박사 학위논문

Ph. D. Dissertation

Linear rank-width의 구조적 특징과 알고리즘 관련 성질에 대하여

On the structural and algorithmic properties of linear rank-width

권 오 정 (權 五 政 Kwon, O-joung)

수리과학과

Department of Mathematical Sciences

KAIST

2015

Linear rank-width의 구조적 특징과 알고리즘 관련 성질에 대하여

On the structural and algorithmic properties of linear rank-width

On the structural and algorithmic properties of linear rank-width

Advisor : Professor Oum, Sang-il

by

Kwon, O-joung
Department of Mathematical Sciences
KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematical Sciences . The study was conducted in accordance with Code of Research Ethics[1].

2015. 5. 13.

Approved by

Professor Oum, Sang-il

[Advisor]

___

[1]Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Linear rank-width의 구조적 특징과 알고리즘 관련 성질에 대하여

권 오 정

위 논문은 한국과학기술원 박사학위논문으로
학위논문심사위원회에서 심사 통과하였음.

2015년 5월 13일

심사위원장	엄 상 일	(인)
심사위원	Otfried Cheong	(인)
심사위원	Andreas Holmsen	(인)
심사위원	진 교 택	(인)
심사위원	김 석 진	(인)

DMAS 20125386

권 오 정. Kwon, O-joung. On the structural and algorithmic properties of linear rank-width. Linear rank-width의 구조적 특징과 알고리즘 관련 성질에 대하여. Department of Mathematical Sciences . 2015. 155p. Advisor Prof. Oum, Sang-il. Text in English.

ABSTRACT

The *linear rank-width* of a graph is the minimum width over all possible linear layouts $(v_1, v_2, \ldots, v_n)$ of the vertex set of the graph, where the width of a linear layout is the maximum rank of the $(0,1)$-adjacency matrices induced by the vertex partitions $(\{v_1, \ldots, v_i\}, \{v_{i+1}, \ldots, v_n\})$. Linear rank-width is the linearized variant of rank-width, just as path-width is the linearized variant of tree-width. Motivated by numerous results on path-width, we investigate several properties of linear rank-width of graphs.

In Chapters 6, 7, and 8, we study structural properties of graphs related to linear rank-width. As a corollary of known theorems by Oum [2008], for each k, there is a finite set of graphs such that a graph G has linear rank-width at most k if and only if no vertex-minor of G is isomorphic to a graph in the set. We show that the number of pairwise locally non-equivalent vertex-minor minimal graphs for the class of graphs of linear rank-width at most k is at least $2^{\Omega(3^k)}$.

For a fixed tree T, we ask whether every graph with sufficiently large linear rank-width contains a vertex-minor isomorphic to T. We show that this question is true if it is true for prime graphs. Prime graphs are graphs with no vertex partition (A, B) with $|A|, |B| \geqslant 2$ such that the set of edges joining A and B induces a complete bipartite graph.

We also investigate a Ramsey type result for prime graphs. We prove that for each n, there exists N such that every prime graph on at least N vertices contains a vertex-minor isomorphic to either a cycle of length n or the line graph of the complete bipartite graph $K_{2,n}$.

In Chapters 9, 10, and 11, we develop graph algorithms related to linear rank-width. We first verify that computing linear rank-width on graphs is NP-hard, using the result on matroid path-width by Kashyap [2008]. We then ask which graph classes admit a polynomial-time algorithm for computing linear rank-width. For distance-hereditary graphs, we show that it is possible to compute the linear rank-width in time $\mathcal{O}(n^2 \log_2 n)$. As a corollary, we can compute the path-width of n-element matroids of branch-width at most 2 in time $\mathcal{O}(n^2 \log_2 n)$, provided that the matroid is given by an independent set oracle.

We also discuss graph modification problems related to linear rank-width. We prove that for a positive integer k and an input graph G with n vertices, we can decide in time $8^k \cdot n^{\mathcal{O}(1)}$ whether G contains a vertex subset S of size at most k such that $G \backslash S$ has linear rank-width at most 1. We also show that this problem admits a polynomial kernel, which means that there exists a polynomial-time algorithm to transform an input graph G and a positive integer k into another instance G' and k' such that (G, k) is a Yes-instance if and only if (G', k') is a Yes-instance, and $|V(G')|$ is bounded by a polynomial function in k. Additionally, for a positive integer k and an input graph G with n vertices, we can decide in time $2^{\mathcal{O}(k \log_2 k)} \cdot n^{\mathcal{O}(1)}$ whether G contains a vertex subset S of size at most k such that $G \backslash S$ has rank-width at most 1.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Linear rank-width is a width parameter of graphs that measures how much the given graph has a path-like structure using the matrix rank function. Linear rank-width is the linearized variant of rank-width [150], and it is similar to path-width which is the linearized variant of tree-width [169, 170]. It is formulated as a graph layout problem, and is known to be equivalent to linear clique-width [62, 89] and linear boolean-width [40, 158].

A linear layout of a graph is an ordering of the vertices of the graph. Graph layout problems are a class of optimization problems where the goal is to find a linear layout of an input graph in such a way that a certain objective function is optimized. Several width parameters are defined in terms of graph layout problems, for instance, band-width [111, 122], cut-width [4, 141], vertex separation number [140, 139]. For motivations and applications of these parameters, we refer to the survey by Diaz [75].

Among these parameters, much less was known for linear rank-width and its equivalent parameters. Together with the vertex-minor relation, we investigate several properties of linear rank-width. Our works are mainly motivated from similar results for path-width [169, 58, 13, 182, 85, 23, 137, 67, 157]. We compare properties of linear rank-width and path-width in Table 1.1 and 1.2. We introduce linear rank-width and vertex-minors in Section 1.1 and 1.2, respectively, and define basic notions in Section 1.3.

Vertex-minor obstruction sets for bounded linear rank-width

Tree-width and path-width have important roles in the Graph Minor Theorem, proved by Robertson and Seymour [169, 171, 175, 176, 177]. They proved that for every infinite sequence $G_1, G_2, \ldots$ of graphs, there exist G_i and G_j with $i < j$ such that G_i is isomorphic to a minor of G_j. In other words, graphs are *well-quasi-ordered* under the minor relation. Surprisingly, this property yields polynomial-time algorithms for many problems; for instance, testing whether an input graph can be embedded on a fixed surface [172, 173], or testing whether an input graph can be embedded in $\mathbb{R}^3$ so that no two cycles are linked [146, 88], or testing, for a fixed k and any minor-closed class $\mathcal{F}$, whether an input graph G contains at most k vertices whose removal makes it belong to $\mathcal{F}$ [88]. For each problem, the set of all YES-instances are closed under taking minors, and the Graph Minor Theorem implies that the number of minor obstructions for each set is finite. Thus, testing whether an instance is a YES-instance can be done in time $\mathcal{O}(n^3)$ using the minor testing algorithm by Robertson and Seymour [176, 166, 167].

Generally, well-quasi-orderings are useful to understand graph classes with forbidden structures. In graphs, the following relations are usually considered as possible quasi-orderings: induced subgraphs (vertex deletions), subgraphs (vertex or edge deletions), minors (vertex or edge deletions, and edge contractions), induced minors (vertex deletions and edge contractions), and topological minors (a graph H is called a *topological minor* of a graph G if a subdivision of H is isomorphic to a subgraph of G). For instance, the class of all graphs with path-width at most k for some fixed k is characterized by a finite list of minor obstructions because it is closed under taking minors and the minor relation is a well-quasi-ordering on this graph class. For linear rank-width and its equivalent parameters, they do not increase when taking induced subgraphs, among the aforementioned five relations. However mostly, a class of graphs of bounded linear rank-width (or rank-width) is not well-quasi-ordered under the induced subgraph relation. For instance, all of cycles have linear rank-width at most 2, but the infinite sequence $C_3, C_4, \ldots$ of cycles contains no pair C_i, C_j with $i < j$ such that C_i is an induced subgraph of C_j.

Local complementation [131] is a useful operation when studying rank-width and linear rank-width. When we apply a local complementation at some vertex in a graph, we swap the adjacency relation between two vertices in the neighborhood of the chosen vertex. Local complementation was first introduced by Kotzig [131] and further studied by Bouchet [28, 29, 30, 32]. Bouchet [32] observed that a local complementation at some vertex preserves the rank of the matrix induced from each vertex partition, and this implies that the rank-width or the linear rank-width of a graph is preserved when applying a local complementation.

A graph G is a *vertex-minor* of a graph H if G can be obtained from H by a sequence of local complementations and vertex deletions. From above observation, the rank-width or the linear rank-width of a graph does not increase when taking vertex-minors. Moreover, Oum [151] showed that every class of graphs with bounded rank-width is well-quasi-ordered under the vertex-minor relation (Theorem 1.11). It implies that every class of graphs with bounded linear rank-width is also well-quasi-ordered under the vertex-minor relation, and therefore for each k, the class of all graphs of linear rank-width at most k can be characterized by a finite list of vertex-minor minimal graphs for the class [151].

Corollary 1.1. *For each positive integer k, there exists a finite set $\mathcal{O}_k$ of graphs such that a graph has linear rank-width at most k if and only if it has no vertex-minor isomorphic to a graph in $\mathcal{O}_k$.*

We can use this result to devise an algorithm to test whether an input graph has linear rank-width at most k, by testing whether it has each graph in the obstruction set as a vertex-minor. Adler, Farley, and Proskurowski [1] proved that the three graphs in Figure 1.1 form a vertex-minor obstruction set for the class of graphs of linear rank-width at most 1. However, the well-quasi-ordered result does not provide any bound on the size of the obstruction set and there were no known upper bound on the size of a vertex-minor obstruction set when $k \geqslant 2$.

In Chapter 6, we prove that for each integer $k \geqslant 2$, there is a set of at least $2^{\Omega(3^k)}$ vertex-minor minimal graphs for the class of graphs of linear rank-width at most k, where no two graphs in the set are equivalent up to local complementations. Two graphs G and H are called *locally equivalent* if G can be obtained from H by applying a sequence of local complementations.

Theorem 6.1. *Let $k \geqslant 2$ be an integer. There exist at least $2^{\Omega(3^k)}$ pairwise locally non-equivalent graphs that are vertex-minor minimal graphs for the class of graphs of linear rank-width at most k.*

There is a technical point in proving that a constructed set of graphs is indeed a minimal set, that is, any two graphs in the set are not locally equivalent to each other. Bouchet [31] showed that no two locally equivalent trees are isomorphic to each other. However, our constructions are not trees. For proving Theorem 6.1, we extend the result on trees [31] into a special type of block graphs in Theorem 5.20.

We note that there is no general way to construct a vertex-minor obstruction set. If we know an upper bound on the maximum number of vertices in a vertex-minor minimal graph for the class of graphs of linear rank-width at most k, then in theory we can enumerate all of the obstructions. We ask an upper bound on the size of vertex-minor minimal graphs for bounded linear rank-width as an open problem.

For path-width of graphs, the following are known.

Theorem 1.2. *Let k be a positive integer.*

- (Takahashi, Ueno, and Kajitani [182]; Ellis, Sudborough, and Turner [85]) *The number of minor obstructions for the class of graphs of path-width at most k is at least $(k!)^2$.*

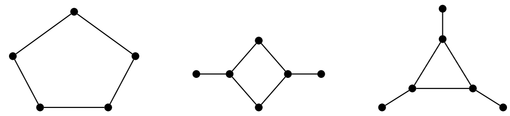


Figure 1.1: A vertex-minor obstruction set for thread graphs.

- (Lagergren [137]) *The number of edges in a minor obstruction for the class of graphs of path-width at most k is at most $2^{\mathcal{O}(k^4)}$. (Since an obstruction is connected, the number of vertices is also bounded.)*

Vertex-minors in graphs of large linear rank-width

In the papers on Graph Minor Theorem, Robertson and Seymour proved that for a fixed r, every graph of sufficiently large tree-width contains an $r \times r$-grid as a minor [171], and for a fixed forest F, every graph of large path-width contains F as a minor [169]. These results not only capture the essential properties of those parameters, but also can be used to devise algorithms for some graph problems such as the Disjoint Paths problem. For a given set of pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$ in a graph, the Disjoint Paths problem asks whether there exist k vertex-disjoint paths $P_1, \ldots, P_k$ where each path P_i links s_i to t_i. Roughly speaking, if an input graph has sufficiently large tree-width, then it is possible to find a large grid in the graph, and identify a vertex whose removal does not affect on whether the question is yes or no [167]. By recursively removing such vertices until an input graph has tree-width bounded by some function in k, we can finally obtain a graph of bounded tree-width where the problem can be solved efficiently on the graph. Based on this argument, Robertson and Seymour [176, 167] showed that the Disjoint Paths problem can be solved in time $f(k) \cdot n^3$ for some function f, where n is the number of vertices in the input graph. Also, this grid theorem has been used to obtain meta-algorithmic results for parameterized problems [71, 72, 81, 73, 93].

Oum [154] conjectured that for a fixed bipartite circle graph H, every graph with sufficiently large rank-width contains a vertex-minor (originally, pivot-minor) isomorphic to H. Bipartite circle graphs naturally appear because every fundamental graph of a planar matroid is bipartite circle [69]. (A planar matroid is the graphic matroid of a planar graph.) This conjecture is still open, and it is true for bipartite graphs [150], circle graphs [154], and line graphs [154].

Similar to this conjecture, for a fixed tree T, we ask whether every graph of sufficiently large linear rank-width contains T as a vertex-minor. We remark that the bipartite version of this question was already asked in [59]. Note that trees have unbounded linear rank-width [97, 133, 2].

Question 1.3. *For any fixed tree T, does every graph of sufficiently large linear rank-width contain a vertex-minor isomorphic to T?*

In Chapter 7, we prove that it is true if it is true for prime graphs with respect to split decompositions [64]. We prove the following.

Theorem 7.1. *Let $p \geqslant 3$ be an integer and let T be a tree. Let G be a graph such that every prime induced subgraph of G has linear rank-width at most p. If G has linear rank-width at least $30(p+4)|V(T)|$, then G contains a vertex-minor isomorphic to T.*

Prime graphs are the graphs having no vertex partition (A, B) with $|A|$, $|B| \geqslant 2$ such that the set of edges joining A and B induces a complete bipartite graph, in other words, the rank of the matrix

$A(G)[X, Y]$ is at most 1 where $A(G)$ is the adjacency matrix of G. Split decompositions and prime graphs were introduced by Cunningham [65, 64]. Prime graphs play an important role in the study of circle graphs (intersection graphs of chords in a circle) and their recognition algorithms [29, 145, 96, 180, 56]. We will discuss basic properties of prime graphs in Chapter 4. We remark that the rank-width of a graph is equal to the maximum rank-width over all its prime induced subgraphs.

To prove this theorem, we essentially prove that for a fixed tree T, every graph admitting a split decomposition whose decomposition tree has sufficiently large path-width contains a vertex-minor isomorphic to T. The vertex-minor relation is indeed necessary because there is a cograph admitting a split decomposition whose decomposition tree has sufficiently large path-width [57, 104]. See Section 5.1 for split decompositions of cographs. However, cographs have no path of length 3 as an induced subgraph [57].

A Ramsey type result for prime graphs

We further study prime graphs. Bouchet [29] proved several theorems for prime graphs from his work on *isotropic systems*, which unify the properties of 4-regular graphs and binary matroids. For instance, he [29] proved that every prime graph with $n > 5$ vertices contains a prime graph with $n - 1$ vertices as a vertex-minor (Theorem 4.1). This is parallel to the Tutte's wheel and whirl theorem [186] for reducing 3-connected matroids. Using this result, Bouchet [29] also proved that every prime graph contains a cycle of length 5 as a vertex-minor (Corollary 4.2), similar to that every 3-connected matroid contains a wheel or a whirl matroid as a minor. Geelen [102, Corollary 5.11] developed a splitter theorem for prime graphs with respect to the vertex-minor relation, which is a variant of the Seymour's splitter theorem [178] for matroids.

Ramsey's theorem [161] states that for a fixed n, every sufficiently large graph contains either a complete graph K_n or the complement of K_n as an induced subgraph. There are several variants of Ramsey's theorem with given some connectivity assumptions; for instance, for a fixed n, every sufficiently large connected graph contains an induced subgraph isomorphic to either a complete graph K_n, or a star graph $K_{1,n}$ or a path of length n [76]. We list similar Ramsey type theorems in the beginning of Chapter 8. In particular, Ding, Oporowski, Oxley, and Vertigan [79, 80] investigated a Ramsey type result for 3-connected matroids, and we are motivated from their result.

We prove the following in Chapter 8. The graph $K_n \boxminus K_n$ is the graph obtained by joining two copies of K_n by a matching of size n. See Figure 1.2. We remark that $K_n \boxminus K_n$ is the line graph of $K_{2,n}$.

Theorem 8.1. *For every n, there exists an integer N such that every prime graph on at least N vertices has a vertex-minor isomorphic to a cycle of length n or $K_n \boxminus K_n$.*

To prove Theorem 8.1, we will use the concept of *blocking sequences* developed by Geelen [102] to construct certain vertex-minors, and further used by Bouchet, Cunningham, and Geelen [36] in the study of delta-matroids. For a graph G, a vertex partition (X, Y) of the vertex set of G is called a *split* if $|X|, |Y| \geqslant 2$, and the rank of the matrix $A(G)[X, Y]$ is at most 1 where $A(G)$ is the adjacency matrix of G. Note that prime graphs have no splits. If a prime graph G has an induced subgraph H which admits a split (X_H, Y_H), then the blocking sequence is a certificate verifying the fact that this vertex partition (X_H, Y_H) cannot be extended to a split of G. For our purpose, we will develop a way to bound the length of blocking sequences using local complementations, in Section 8.2.

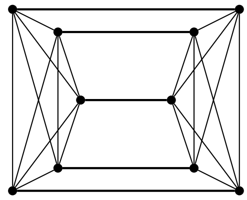


Figure 1.2: The line graph of $K_{2,5}$ ($K_5 \boxminus K_5$).

Algorithms for computing linear rank-width

Tree-decompositions play an important role in many graph algorithms. Arnborg, Proskurowski [7] and Bern, Lawler, Wong [12] and Bodlaender [17] independently developed efficient algorithms to solve NP-complete problems on graphs of bounded tree-width. Later, Courcelle [58] proved a much generalized theorem that every graph property expressible in a monadic second-order logic formula of the second type (MSO_2) can be decided in linear time on graphs of bounded tree-width. This can be applied to many problems such as 3-COLORING, HAMILTONIAN CYCLE, DOMINATING SET problems.

Rank-width and linear rank-width have been studied in the context of generalizing these results into bigger classes. Even for linear rank-width, graphs of bounded linear rank-width may contain dense graphs such as all complete graphs, or all complete bipartite graphs, which cannot be contained in a class of graphs of bounded tree-width. Courcelle, Makowsky, and Rotics [61] showed that every graph property expressible in a monadic second-order logic formula of the first type (MSO_1) can be decided in cubic time on graphs of bounded rank-width. We remark that every MSO_1 formula is an MSO_2 formula, but not vice versa. For instance, a graph property of having a cycle through all vertices can be written as an MSO_2 formula, but it cannot be written as an MSO_1 formula. We will observe the difference of two types of the logic formulas in Section 3.3.

We discuss algorithms for computing linear rank-width in Chapter 9. It is known that computing the path-width of graphs is NP-hard [6]. For various restricted graph classes, computing the exact value of path-width has been studied: forests [85], graphs of bounded tree-width [20, 23], split graphs [16, 109], the complements of chordal graphs [98], permutation graphs [24], cographs [26], and circular-arc graphs [181].

We first prove that computing the linear rank-width of a graph is also NP-hard in Section 9.1 by reducing from matroid path-width. Kashyap [123] proved that computing path-width of representable matroids is NP-hard. Then we ask which graph classes admit a polynomial-time algorithm for computing linear rank-width.

Ellis, Sudborough, and Turner [85] showed that the path-width of forests (graphs of tree-width 1) can be computed in linear time. They used the characterization of path-width on forests, and this allows to have a natural algorithm to compute it based on dynamic programming. Previously, the only known polynomial-time algorithm to compute linear rank-width was for forests [2]. This follows from the fact that the linear rank-width and the path-width of a tree are equal [133, 2].

In Chapter 9, we investigate a new $\mathcal{O}(n^2 \log_2 n)$-time algorithm to compute the linear rank-width of *distance-hereditary graphs*. Distance-hereditary graphs are the graphs G where every connected induced subgraph H of G and two vertices v, w in H, the distance between v and w in H is equal to the distance in G [113, 9], and they include all forests, as well as complete graphs, complete bipartite graphs, threshold graphs [54, 105], and cographs [37] that are not forests. Oum [150] showed that distance-hereditary graphs are exactly the graphs of rank-width at most 1. We will discuss distance-hereditary graphs in

Chapter 5.

Theorem 9.1. *The linear rank-width of an n-vertex distance-hereditary graph can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$.*

Since computing the path-width of distance-hereditary graphs is NP-hard [129], this class is the first class which satisfies that it is NP-hard to compute path-width but linear rank-width can be computed in polynomial time. To prove Theorem 9.1, we present a characterization of linear rank-width on distance-hereditary graphs in Theorem 5.11, and devise a direct dynamic programming algorithm to compute linear rank-width. As a corollary of Theorem 9.1, we also prove the following.

Corollary 9.2. *The path-width of an n-element matroid of branch-width at most 2 with a given independent set oracle can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$.*

Motivated from Theorem 9.1, for $k \geqslant 2$, we ask whether there is a polynomial-time algorithm to compute linear rank-width for the class of graphs of rank-width at most k.

Question 1.4. *For a fixed $k \geqslant 2$, is there a polynomial-time algorithm to compute linear rank-width on graphs of rank-width at most k?*

Bodlaender and Kloks [23] show the following for path-width.

Theorem 1.5 (Bodlaender and Kloks [23])**.** *For fixed k, there is a polynomial-time algorithm to compute path-width on graphs of tree-width at most k.*

We discuss parameterized problems related to linear rank-width. For the definitions related to parameterized algorithms, we refer to the book written by Downey and Fellows [84]. *Parameterized problems* deal with an instance (x, k) where k is a secondary measurement, called as the *parameter*, and the main goal is to find whether a problem admits an algorithm with running time $f(k) \cdot |x|^{\mathcal{O}(1)}$ where f is a function depending on the parameter k alone, and $|x|$ is the input size. As we study parameterized problems where the unparameterized decision versions are NP-complete, the function f is generally super-polynomial. A parameterized problem admitting such an algorithm is said to be *fixed parameter tractable*, or *FPT* in short. For many natural parameterized problems, the function f is overwhelming [95] or even non-explicit [176], especially when the algorithm is indicated by a meta-theorem. Therefore, researchers focus mainly on designing FPT algorithms with affordable super-exponential part in the running time. We are particularly interested in solving parameterized problems in single-exponential time, that is, in time $c^k \cdot |x|^{\mathcal{O}(1)}$ for some constant c.

We consider the problem of determining whether an input graph has linear rank-width at most k for some fixed k. For $k = 1$, we can test whether an input graph G has linear rank-width at most 1 in time $\mathcal{O}(n + m)$ using split decompositions [65, 68, 39, 3], where n and m are the number of the vertices and edges in G, respectively. We will see the characterization of graphs of linear rank-width 1 in terms of split decompositions in Section 5.6. Combined with the algorithm to compute the split decomposition of a graph [65, 68], we can decide whether a graph has linear rank-width at most 1. However, for $k \geqslant 2$, there is no known simple characterization of graphs of linear rank-width at most k, and we need a different approach.

Nagamochi [144] investigated testing algorithms for parameters defined in terms of linear layouts in a general framework. Using his result, we can show that there is an $\mathcal{O}(n^{2k+4})$-time algorithm to determine whether an n-vertex input graph G has linear rank-width at most k. This algorithm is valid for parameters defined by any submodular function, such as cut-width and vertex separation number [144]. However, this algorithm is not a fixed parameter tractable algorithm.

Currently, the only known fixed parameter tractable algorithm for determining whether an input graph has linear rank-width at most k is using the finite list of a vertex-minor obstruction set. Using the obstruction set, we can test whether an input graph G with n vertices has linear rank-width at most k in time $f(k) \cdot n^3$ for some function f, combined with the vertex-minor testing algorithm by Courcelle and Oum [63] and an algorithm to find an approximate rank-decomposition by Oum [153], for instance. In this context, it is interesting to address an upper bound on the size of vertex-minor minimal graphs for the class of graphs of linear rank-width at most k.

Vertex deletion problems related to linear rank-width

We now discuss graph modification problems related to linear rank-width. Generally, for an input graph G and a fixed set O of elementary operations and a class Π of graphs, the objective is to transform G into a graph in Π by applying at most k operations from O. Graph modification problems formulate a number of interesting computational problems arising from both theory and its applications. They have received a significant amount of attention from the perspective of parameterized complexity. We first give some motivation for graph modification problems.

As an application to the real world, we consider a situation that a bank supervisor wants to place automated teller machines (ATMs) in a city so that there is at least one ATM on each street, but also want to place at most k ATMs, as each additional ATM is expensive. For efficiency, we may assume that an ATM is always placed on the intersections. We can model this real world problem as a graph modification problem where the streets of the city are edges of the graph, and each intersection is a vertex of the graph, and ask whether there exists a vertex set S of size at most k such that S meets all edges in G, that is, the remaining graph after removing the vertices in S has no edges. This problem is called the VERTEX COVER problem, and it is known to admit a fixed parameter tractable algorithm when the parameter is the solution size k [83, 42, 8, 147, 51, 49, 148, 50]. In the CLUSTER EDITING problem, we allow k edge deletions and additions to make an input graph G a disjoint union of complete graphs. This problem has been studied in different contexts, such as computational biology [11, 179, 189], and machine learning problems [10]. This problem is also known to admit an FPT algorithm when the parameter is the solution size k [106, 107, 15, 14].

The graph class Π with tree-width at most w is of particular interest as many problems become tractable on graphs of small tree-width (TREE-WIDTH w VERTEX DELETION). When $w = 0$ and $w = 1$, the corresponding graph modification problem with $\mathsf{O} = \{\text{vertex deletion}\}$ coincides with the VERTEX COVER problem and the FEEDBACK VERTEX SET problem, respectively. Since every YES-instance of the TREE-WIDTH w VERTEX DELETION problem has tree-width at most $w + k$, using the Courcelle's meta-theorem [58] on graphs of bounded tree-width, the TREE-WIDTH w VERTEX DELETION problem can be solved in $f(k) \cdot n$ for some function f. Since the function f in the meta-theorem is huge, it is natural to ask whether the exponential function in the running time can be made realistic. Recent endeavor pursuing this question culminated in establishing that for any fixed w, the TREE-WIDTH w VERTEX DELETION problem admits an FPT algorithm that runs in time $c^k \cdot n^{\mathcal{O}(1)}$ for some constant c [92, 127].

We consider the problem to test whether an input graph G has a vertex subset of size at most k whose removal makes G a graph of linear rank-width at most w (Linear rank-width w Vertex Deletion). In particular, we state the problem when $w = 1$. Graphs of linear rank-width 1 are called *thread graphs* [97], and thus, we call this problem as the Thread Vertex Deletion problem for convenience.

Thread Vertex Deletion (Linear rank-width 1 Vertex Deletion)
Input : A graph G, an integer k
Parameter : k
Question : Is there a vertex subset $S \subseteq V(G)$ of size at most k such that $G \backslash S$ is a thread graph, that is, $G \backslash S$ has linear rank-width at most 1?

We will see that the Linear rank-width w Vertex Deletion problem can be formulated as an MSO_1 formula [150, 63] in Section 3.3. We remark that every Yes-instance of Linear rank-width w Vertex Deletion has linear rank-width (or rank-width) at most $k + w$ as linear rank-width can decrease by at most one by removing a vertex. From the meta-theorem for graphs of bounded rank-width by Courcelle, Makowsky and Rotics [61], it is possible to solve the Linear rank-width w Vertex Deletion problem in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function f. However, the involved exponential function is huge and it is not clear whether it can be solved significantly faster. In Chapter 10, we prove that the Thread Vertex Deletion problem can be solved in time $8^k \cdot n^{\mathcal{O}(1)}$.

Theorem 10.1. *For an input graph G with n vertices and a fixed k, we can test whether G has a vertex subset S of size at most k such that $G \backslash S$ has linear rank-width at most 1 in time $8^k \cdot n^{\mathcal{O}(1)}$.*

A powerful technique to handle parameterized problems is the *kernelization algorithm*. A kernelization algorithm takes an instance (x, k) and outputs an instance (x', k') in time polynomial in $|x| + k$ satisfying that (1) (x, k) is a Yes-instance if and only if (x', k') is a Yes-instance, (2) $k' \leqslant k$, and (3) $|x'| \leqslant g(k)$ for some function g. The reduced instance is called a *kernel* and the function g is called the *size* of the kernel. It is folklore that admitting a kernel is equivalent to being fixed-parameter tractable. See [84, Proposition 4.7.1]. Therefore, most kernelization research is focused on finding an algorithm that yields a small-sized kernel, ideally of polynomial size.

We prove that the Thread Vertex Deletion problem admits a polynomial kernel.

Theorem 10.2. *Let k be a fixed integer, and let G be a graph. Then there exists a polynomial-time algorithm to generate a pair (G', k') such that*

1. *G has a vertex subset S of size at most k such that $G \backslash S$ has linear rank-width at most 1 if and only if G' has a vertex subset S' of size at most k' such that $G' \backslash S'$ has linear rank-width at most 1, and*
2. *$k' \leqslant k$ and $|V(G')| \leqslant \mathcal{O}(k^{33})$.*

We remark that a similar deletion problem for graphs of path-width at most 1 has been studied recently, and we call it the Path-width 1 Vertex Deletion problem. Philip, Raman, and Villanger [157] showed that Path-width 1 Vertex Deletion can be solved in time $\mathcal{O}(7^k k \cdot n^2)$, and it admits a kernel of size $\mathcal{O}(k^4)$. Later, Cygan, Pilipczuk, Pilipczuk, and Wojtaszczyk [67] improved the running time by showing that Path-width 1 Vertex Deletion can be solved in time $4.65^k \cdot n^{O(1)}$ and it admits a kernel of size $\mathcal{O}(k^2)$. Several graph classes with a certain path-like structure have been researched for vertex deletion problems. Proper interval graphs [90, 188], unit interval graphs [187], and interval graphs [46] are some such classes.

Additionally, we consider a parameterized deletion problem related to distance-hereditary graphs. The problem is formulated as followss.

DISTANCE-HEREDITARY VERTEX DELETION (RANK-WIDTH 1 DELETION)
Input : A graph G, an integer k
Parameter : k
Question : Is there a vertex subset $S \subseteq V(G)$ of size at most k such that $G \backslash S$ is distance-hereditary, that is, $G \backslash S$ has rank-width at most 1?

Similar to linear rank-width, we ask the following question.

Question 1.6. *Can* DISTANCE-HEREDITARY DELETION *be solved in time $c^k \cdot n^{\mathcal{O}(1)}$ for some constant c?*

We prove that this problem can be solved in time $2^{\mathcal{O}(k \log_2 k)} \cdot n^{\mathcal{O}(1)}$.

Theorem 11.1. *For an input graph G with n vertices and a fixed k, we can test whether G has a vertex subset S of size at most k such that $G \backslash S$ has rank-width at most 1, in time $2^{\mathcal{O}(k \log_2 k)} \cdot n^{\mathcal{O}(1)}$.*

A similar deletion problem for graphs of tree-width at most 1 (forests) is called the FEEDBACK VERTEX SET problem. This problem is one of the most intensively studied problems in parameterized complexity. It was proved in the 1990's that this problem admits an FPT algorithm by Bodlaender [18], and by Downey and Fellows [82]. Then by a series of papers [160, 120, 159, 108, 70, 48, 44, 66], the running time has been subsequently improved, and the current best running time is $3.619^k \cdot n^{\mathcal{O}(1)}$, proved by Kociumaka and Pilipczuk [130]. Thomassè [183] proved that this problem admits a kernel of size $5k^2 + k$.

1.1 Linear rank-width

Let $A(G)$ be the adjacency matrix of a graph G, which is defined on the binary field. For a graph G, we define a function ρ_G: $V(G) \to \mathbb{Z}$ such that $\rho_G(X)$:= $\operatorname{rank} A(G)[X, V(G) \backslash X]$ for $X \subseteq V(G)$. We call it the *cut-rank function* of G.

A *linear layout* σ of a set S is a bijective function from S to $\{1, \ldots, |S|\}$, and for convenience, we denote it as a sequence $(\sigma^{-1}(1), \sigma^{-1}(2), \ldots, \sigma^{-1}(|S|))$. For a linear layout σ of S and $a, b \in S$, we denote $a \leqslant_\sigma b$ or $b \geqslant_\sigma a$ if $\sigma(a) \leqslant \sigma(b)$, and we denote $a <_\sigma b$ or $b >_\sigma a$ if $\sigma(a) < \sigma(b)$.

A linear layout of the vertex set of G is called a linear layout of G. The *width* of a linear layout L of G is defined as the maximum over all values $\rho_G(\{w : w \leqslant_L v\})$ for $v \in V(G)$. We say that the width of L is 0 if $|V(G)| \leqslant 1$. The *linear rank-width* of G, denoted by $\operatorname{lrw}(G)$, is the minimum width of all linear layouts of G.

For some classes of graphs, the exact values of linear rank-width are known. Complete graphs have linear rank-width 1 because for every $\varnothing \subsetneq S \subsetneq V(G)$, $\rho_G(S) = 1$. Complete bipartite graphs $K_{n,m}$ also have linear rank-width 1 because we can put the vertices in one part first, and then put the vertices in the other part in the ordering.

Graphs of linear rank-width at most 1 are characterized by Ganian [97], and Adler, Farley, and Proskurowski [1]. Ganian [97] defined *thread graphs* using a linear layout satisfying a certain condition, and showed that the class of thread graphs is equal to the class of graphs of linear rank-width at most

Path-width	Linear rank-width
The number of minor obstructions for path-width $\leqslant k$ is at least $(k!)^2$ [182, 85].	The number of pairwise locally non-equivalent vertex-minor minimal graphs for linear rank-width $\leqslant k$ is at least $2^{\Omega(3^k)}$. (Theorem 6.1)
The number of edges in a minor obstruction for path-width $\leqslant k$ is at most $2^{\mathcal{O}(k^4)}$ [137].	For $k \geqslant 2$, an upper bound on the number of vertices in a vertex-minor minimal graph for linear rank-width $\leqslant k$ is open.
For a fixed tree T, every graph of sufficiently large path-width contains a minor isomorphic to T [169, 13].	For a fixed tree T and an integer ℓ and a graph G whose prime induced subgraph has linear rank-width $\leqslant \ell$, if G has sufficiently large linear rank-width, then it contains a vertex-minor isomorphic to T. (Theorem 7.1)
Every connected graph with no $K_{1,n}$ minor has path-width $\leqslant n-1$ [13].	Every graph with no $K_{1,n}$ vertex-minor has $\leqslant 2^{4n^2+1}$ vertices. (Theorem 8.4)
Every connected graph with no P_n minor has path-width $\leqslant n-2$ [13].	For $n \leqslant 5$, every graph with no P_n vertex-minor has bounded linear rank-width. (Theorems 7.1 and 8.1) For $n \geqslant 6$, an upper bound on the linear rank-width of a graph with no P_n vertex-minor is open.
Trees have unbounded path-width [182, 85].	Every graph with no K_n vertex-minor has $\leqslant 2^{4(n-1)^2+1}$ vertices. (Theorem 8.4)

Table 1.1: Comparing properties of linear rank-width and path-width.

	Path-width	Linear rank-width
Forests	Linear [85, 143]	Linear [2]
Graphs of tree-width $\leqslant t$	Polynomial [23]	Open
Distance-hereditary graphs	NP-complete [129]	$\mathcal{O}(n^2 \log_2 n)$ (Theorem 9.1)
Graphs of rank-width $\leqslant t$	NP-complete [129]	Open
An FPT algorithm for deciding whether width is $\leqslant k$	A constructive FPT algorithm [23]	Only known FPT algorithm is using the obstruction set [151]
An FPT algorithm for k-vertex deletion to width 1	$4.65^k \cdot n^{\mathcal{O}(1)}$ [67, 157]	$8^k \cdot n^{\mathcal{O}(1)}$ (Theorem 10.1)
A polynomial kernel for k-vertex deletion to width 1	$\mathcal{O}(k^2)$ vertices [67, 157]	$\mathcal{O}(k^{33})$ vertices (Theorem 10.2)

Table 1.2: Algorithms for computing linear rank-width or path-width. If a parameterized problem with an input x and a parameter k admits an algorithm with running time $f(k) \cdot |x|^{\mathcal{O}(1)}$, then the problem is called fixed parameter tractable (shortly, FPT).

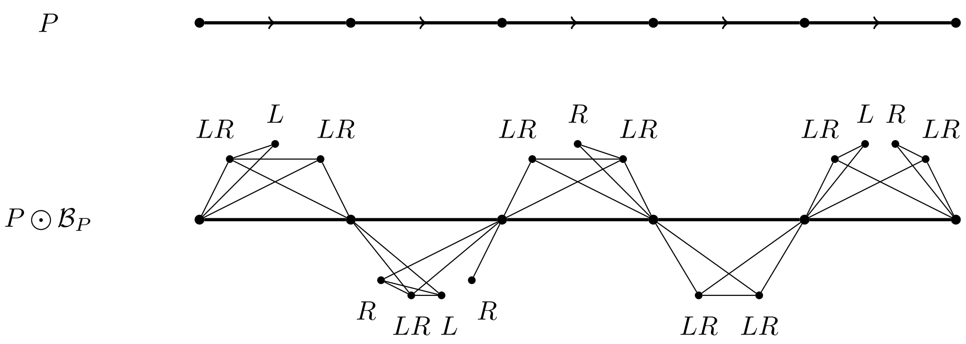


Figure 1.3: A thread graph.

1. Adler, Farley, and Proskurowski [1] proposed a new definition of thread graphs. Here, we propose a convenient form of the definition of thread graphs, motivated by Adler, Farley, and Proskurowski [1].

Thread graphs

A triple $B(x, y) = (G, \sigma, \ell)$, where x and y are two vertices of the graph G, σ is a linear layout of $V(G)$ whose first and last vertices are x and y, respectively, and ℓ is a function from $V(G)$ to $\{\{L\}, \{R\}, \{L, R\}\}$, is a *thread block* if

1. $\ell(x) = \{R\}$ and $\ell(y) = \{L\}$,
2. for $v, w \in V(G)$ with $v <_\sigma w$, $vw \in E(G)$ if and only if $R \in \ell(v)$ and $L \in \ell(w)$,
3. $\ell(\sigma^{-1}(2)) \neq \{L\}$ if $\sigma^{-1}(2) \neq y$.

The aim of the third condition is to guarantee a unique decomposition of thread graphs into thread blocks.

For a digraph $D = (V_D, A_D)$, a set of thread blocks $\{B(x, y) = (G_{xy}, \sigma_{xy}, \ell_{xy}) : xy \in A_D\}$ is said to be *mergeable with* D if for any two arcs x_1y_1, x_2y_2 of A_D, $V(G_{x_1y_1}) \cap V(G_{x_2y_2}) = \{x_1, y_1\} \cap \{x_2, y_2\}$. For a digraph $D = (V_D, A_D)$ and a mergeable set of thread blocks $\mathcal{B}_D = \{B(x, y) = (G_{xy}, \sigma_{xy}, \ell_{xy}) : xy \in A_D\}$, the graph $G = D \odot \mathcal{B}_D$ has the vertex set $V(G) = \bigcup_{xy \in A_D} V(G_{xy})$ and the edge set $E(G) = \bigcup_{xy \in A_D} E(G_{xy})$.

A connected graph G is a *thread graph* if G is either the one vertex graph or $G = P \odot \mathcal{B}_P$ for some directed path P, called the *underlying directed path*, and some set of thread blocks $\mathcal{B}_P$ mergeable with P. A graph is a *thread graph* if each of its connected components is a thread graph. See Figure 1.3 for an example of a thread graph.

We prove the following in Section 5.6.

Theorem 1.7 (Ganian [97]; Adler, Farley, and Proskurowski [1])**.** *A graph has linear rank-width at most* 1 *if and only if it is a thread graph.*

Trees have unbounded linear rank-width [97, 133, 2]. Kwon proved in [133] that the rooted complete binary tree of height n has linear rank-width $\lceil \frac{n}{2} \rceil$, where the height is the length from the root to any leaf. Linear rank-width and path-width are equal on trees [133, 2].

Courcelle, Makowsky, and Rotics [61] showed that every graph property expressible in monadic second order logic (MSO_1) can be decided in cubic time on graphs of bounded rank-width, and thus for

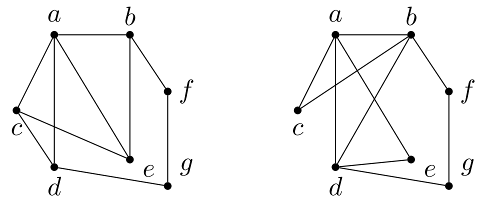


Figure 1.4: Local complementation at a.

the class of graphs of bounded linear rank-width as well. Ganian [97] proved that for some problem, there is a polynomial-time algorithm to solve it for thread graphs even though the problem is NP-hard on other classes, for instance, there exists a polynomial-time algorithm to compute path-width of thread graphs while it is NP-hard to compute the path-width of distance-hereditary graphs [129].

1.2 Vertex-minors

The *local complementation* at a vertex v of a graph is an operation to replace the subgraph induced by the neighborhood of v with its complement. We write $G * v$ to denote the graph obtained from G by applying a local complementation at v. See Figure 1.4 for an example. The graph obtained from G by *pivoting* an edge uv is defined by $G \wedge uv := G * u * v * u$. To see how we obtain the resulting graph by pivoting an edge uv, let $V_1 := N_G(u) \cap N_G(v)$, $V_2 := N_G(u) \backslash N_G(v) \backslash \{v\}$, and $V_3 := N_G(v) \backslash N_G(u) \backslash \{u\}$. One can easily verify that $G \wedge uv$ is identical to the graph obtained from G by complementing the adjacency relations of vertices between distinct sets V_i and V_j, and swapping the vertices u and v [150]. See Figure 1.5 for an example.

A graph H is a *vertex-minor* of G if H can be obtained from G by applying a sequence of vertex deletions and local complementations. A graph H is a *pivot-minor* of G if H can be obtained from G by applying a sequence of vertex deletions and pivotings. A graph H is *locally equivalent* to G if H can be obtained from G by applying a sequence of local complementations. A graph H is *pivot equivalent* to G if H can be obtained from G by applying a sequence of pivotings.

The local complementation was introduced by Kotzig [131] on the study of 4-regular graphs. It has been studied by Bouchet [28, 29, 30, 32, 34], in his papers on isotropic systems. Roughly speaking, *isotropic systems* are linear algebraic objects that capture all graphs equivalent up to local complementations. The graphs associated with an isotropic system are called *fundamental graphs* (parallel with fundamental graphs of matroids), and a certain minor notion of isotropic systems is related to the vertex-minor of their fundamental graphs. Moreover, Bouchet [32] observed that a local complementation at a vertex in a graph preserves the cut-rank function of the graph, and thus it preserves the rank-width and linear rank-width of the graph as well. Recently, the local complementation has been used in quantum information theory [190, 142, 117, 163, 38, 43].

Interestingly, for $n \geqslant 2$, the complete graph K_n is a vertex-minor of a path of length $2n-3$. We will see this in Theorem 8.4. This is not the case for graph minors because every minor of a path is the disjoint union of paths. Since $K_{1,n-1}$ is locally equivalent to K_n, every sufficiently large connected graph has K_n as a vertex-minor (Theorem 8.4).

A vertex-minor or a pivot-minor H of G is *elementary* if $|V(H)| = |V(G)| - 1$, and it is *proper*

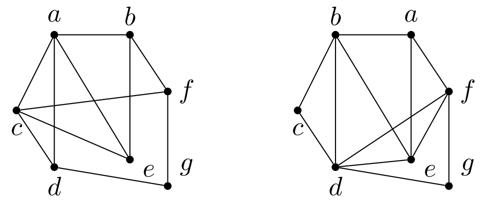


Figure 1.5: Pivoting an edge ab.

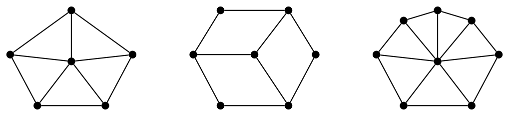


Figure 1.6: A vertex-minor obstruction set for circle graphs.

if $|V(H)| \leqslant |V(G)| - 1$. For a class $\mathcal{C}$ of graphs closed under taking vertex-minors, a graph G is a *vertex-minor minimal graph* for $\mathcal{C}$ if $G \notin \mathcal{C}$ and $H \in \mathcal{C}$ for every elementary vertex-minor H of G. For a class $\mathcal{C}$ of graphs closed under taking pivot-minors, a graph G is a *pivot-minor minimal graph* for $\mathcal{C}$ if $G \notin \mathcal{C}$ and $H \in \mathcal{C}$ for every elementary pivot-minor H of G.

The following lemma by Bouchet provides a key tool to investigate vertex-minors.

Lemma 1.8 (Bouchet [30]; see Geelen and Oum [103])**.** *Let H be a vertex-minor of G and let $v \in V(G)\backslash V(H)$. Then H is a vertex-minor of either $G\backslash v$, $G * v\backslash v$, or $G \wedge vw\backslash v$ for a neighbor w of v.*

The choice of a neighbor w in Lemma 1.8 does not matter, because if x is adjacent to y and z, then $G \wedge xy = (G \wedge xz) \wedge yz$ by the following lemma.

Lemma 1.9 (Oum [150])**.** *Let G be a graph and $x, y, z \in V(G)$ such that $xy, xz \in E(G)$. Then $G \wedge xy = (G \wedge xz) \wedge yz$.*

For a fixed graph H, it is not known if there is a polynomial-time algorithm to decide whether an input graph G has a vertex-minor isomorphic to H. Courcelle and Oum [63] showed that this testing can be done in polynomial time when an input graph has bounded rank-width.

Theorem 1.10 (Courcelle and Oum [63])**.** *Let ℓ be an integer and let H be a fixed graph. For an input graph G with n vertices and rank-width at most ℓ, we can test whether G contains a vertex-minor isomorphic to H in time $\mathcal{O}(n^3)$.*

We remark that it is possible to test in time $\mathcal{O}(n^3)$ [33] whether two given graph G and H with n vertices are locally equivalent without considering isomorphism.

For a class $\mathcal{C}$ of graphs closed under taking vertex-minors, a graph G is called a *vertex-minor minimal graph* for $\mathcal{C}$ if $G \notin \mathcal{C}$ and $H \in \mathcal{C}$ for every elementary vertex-minor H of G. We remark that if a graph G is a vertex-minor minimal graph for a class $\mathcal{C}$ and G' is locally equivalent to G, then G' is also a vertex-minor minimal graph for $\mathcal{C}$. Therefore, we distinguish a minimal set of vertex-minor minimal graphs for $\mathcal{C}$ from the set of all vertex-minor minimal graphs for $\mathcal{C}$. For a vertex-minor closed class $\mathcal{C}$, a set $\mathcal{O}_{\mathcal{C}}$ of vertex-minor minimal graphs for $\mathcal{C}$ is called a *vertex-minor obstruction set* for $\mathcal{C}$ if

1. for every graph G, $G \in \mathcal{C}$ if and only if G has no vertex-minor isomorphic to a graph in $\mathcal{O}_{\mathcal{C}}$, and

2. there are no two graphs G_1, G_2 in $\mathcal{O}_\mathcal{C}$ such that G_1 and G_2 are locally equivalent.

Oum [151] proved that for every infinite sequence $G_1, G_2, \ldots$ of graphs of bounded rank-width, there exist G_i and G_j with $i < j$ such that G_i is isomorphic to a vertex-minor of G_j. This implies that for every vertex-minor closed class $\mathcal{C}$ of graphs with bounded rank-width, a vertex-minor obstruction set for $\mathcal{C}$ is finite.

Theorem 1.11 (Oum [151])**.** *For every vertex-minor closed class $\mathcal{C}$ of graphs that has bounded rank-width, there exists a finite set $\mathcal{O}_\mathcal{C}$ of graphs such that a graph is in $\mathcal{C}$ if and only if it has no vertex-minor isomorphic to a graph in $\mathcal{O}_\mathcal{C}$.*

Kotzig [132] observed that for a vertex v in a graph G, G is a circle graph if and only if $G * v$ is a circle graph. Bouchet [35] characterized circle graphs in terms of three vertex-minor minimal graphs. See Figure 1.6 for the list of obstructions. The pivot-minor obstructions for circle graphs were classified by Geelen and Oum [101].

1.3 Notation

In this thesis, all graphs have no loops and no parallel edges, and all graphs are undirected if not stated. For a set A, we denote the power set of A by 2^A. For a finite set X, we say that a function $f : 2^X \to \mathbb{N}$ is *symmetric* if for any $S \subseteq X$, $f(S) = f(X \backslash S)$, and f is *submodular* if for any $S, T \subseteq X$, $f(S \cup T) + f(S \cap T) \leqslant f(S) + f(T)$.

A binary relation $\precsim$ on a set X is a *quasi-ordering* if it is reflexive and transitive. A quasi-ordering $\precsim$ on a set X is a *well-quasi-ordering* if for any infinite sequence of elements $x_1, x_2, \ldots$ in X, there exists $i < j$ such that $x_i \precsim x_j$.

A graph G is a pair $(V(G), E(G))$ where $V(G)$ is the vertex set of G and $E(G)$ is the edge set of G. For $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced on S and $G \backslash S := G[V(G) \backslash S]$. A graph H is an *induced subgraph* of G if $H = G[S]$ for some $S \subseteq V(G)$. For a vertex x of G, let $G \backslash x := G \backslash \{x\}$. For $F \subseteq E(G)$, let $G \backslash F$ be the graph on the vertex set $V(G)$ with the edge set $E(G \backslash F) = E(G) \backslash F$. For an edge e of G, let $G \backslash e := G \backslash \{e\}$. For an edge e of G, we denote G/e to be the graph obtained from G by contracting e. For a graph G, we denote by $\overline{G}$ the *complement* of G, that is, G and $\overline{G}$ have the same set of vertices and two vertices in G are adjacent if and only if they are not adjacent in $\overline{G}$.

For $v \in V(G)$, we let $N_G(v)$ denote the set of the neighbors of v in G. Let $\deg_G(v) := |N_G(v)|$, and we call it the *degree* of v in G. For $X \subseteq V(G)$, let $\delta_G(X)$ be the set of edges having one end in X and the other end in $V(G) \backslash X$. For two disjoint subsets S, T of $V(G)$, let $G[S, T] = G[S \cup T] \backslash (E(G[S]) \cup E(G[T]))$. Two graphs G and H are *isomorphic* if there exists a bijection $h : V(G) \to V(H)$ such that $xy \in E(G)$ if and only if $h(x)h(y) \in E(H)$. For a set $\mathcal{F}$ of graphs, a graph G is called *$\mathcal{F}$-free* if G has no induced subgraph isomorphic to a graph in $\mathcal{F}$.

A graph G is *connected* if for each pair of vertices $v, w \in V(G)$, there exists a path from v to w in G. A graph G is 2-*connected*, if $|V(G)| \geqslant 3$ and $G \backslash X$ is connected for every vertex set $X \subseteq V(G)$ with $|X| \leqslant 1$. A vertex v of a graph G is a *cut vertex* if the number of components of $G \backslash v$ increases. An edge e of a graph G is a *cut edge* if the number of components of $G \backslash e$ increases. A *block* of a graph G is a maximal connected subgraph of G without a cut vertex.

A vertex v in G is called a *leaf* if $\deg_G(v) = 1$. A vertex v is called a *twin* of another vertex w in a graph if no vertex other than v and w is adjacent to exactly one of v and w. A twin w of a vertex v is called a *true twin* if v and w are adjacent, and called a *false twin* otherwise.

A graph H is a *minor* of G if H can be obtained from G by a sequence of deleting vertices, deleting edges and contracting edges. For a class $\mathcal{C}$ of graphs closed under taking minors, a graph G is a *minor obstruction* for $\mathcal{C}$ if $G \notin \mathcal{C}$ and for every minor H of G with $|V(H)| < |V(G)|$, $H \in \mathcal{C}$.

For an $X \times Y$ matrix A, if $X' \subseteq X$ and $Y' \subseteq Y$, then we write $A[X', Y']$ to denote the submatrix of A obtained by taking rows in X' and columns in Y'. If $X' = Y'$, then we simply denote $A[X'] := A[X', X']$. The *adjacency matrix* of a graph G, which is a $(0, 1)$-matrix over the binary field, will be denoted by $A(G)$.

Graph classes

A *tree* is a connected acyclic graph, and a *forest* is a disjoint union of trees. A tree is a *path* if every vertex has degree at most 2. The *length* of a path is the number of its edges. A tree is a *caterpillar* if it contains a path P such that every vertex of a tree has distance at most 1 to some vertex of P. A *cycle* is a connected graph where every vertex has degree exactly 2. A *hole* is an induced cycle of length at least 5. We write P_n and C_n to denote a graph that is a path and a cycle on n vertices, respectively. A *complete graph* is the graph with all possible edges. We write K_n to denote a complete graph on n vertices. A *star* is a tree with a distinguished vertex, called its *center*, adjacent to all other vertices. A *clique* is a set of pairwise adjacent vertices. A *stable set* is a set of pairwise non-adjacent vertices. A vertex is *simplicial* if the set of its neighbors is a clique.

A graph G is a *bipartite graph* with a bipartition (A, B) if $V(G) = A \cup B$ and $G[A]$ and $G[B]$ has no edges. We write $K_{m,n}$ to denote the complete bipartite graph with a bipartition (A, B) such that $|A| = m$, $|B| = n$. For a graph G, the *line graph* $L(G)$ of G is a graph where $V(L(G)) = E(G)$ and two vertices in $e_1, e_2 \in V(L(G))$ are adjacent in $L(G)$ if and only if the corresponding edges meet at some vertex in G. A graph is a *block graph* if every its block is a complete graph.

A set Φ of chords in a circle are called a *chord diagram.* A graph G is an *intersection graph of chords* in a chord diagram Φ if the vertex set of G is Φ and two vertices are adjacent in G if and only if the chords are intersect in the circle. We say that Φ is a *circle representation* of G, and a graph G is a *circle graph* if it has a circle representation. Circle graphs were independently introduced by several authors in the 1970's [27, 86, 132].

Matroids

We will use matroids. We refer to the book written by Oxley [156] for our matroid notations and basic properties.

A pair $(E(M), \mathcal{I}(M))$ is called a *matroid* M if $E(M)$, called the *ground set of* M, is a finite set and $\mathcal{I}(M)$, called the set of *independent sets of* M, is a nonempty collection of subsets of $E(M)$ satisfying the following conditions:

(I1) if $I \in \mathcal{I}(M)$ and $J \subseteq I$, then $J \in \mathcal{I}(M)$,

(I2) if $I, J \in \mathcal{I}(M)$ and $|I| < |J|$, then $I \cup \{z\} \in \mathcal{I}(M)$ for some $z \in J \backslash I$.

A maximal independent set in M is called a *base of* M. It is known that, if B_1 and B_2 are bases of M, then $|B_1| = |B_2|$.

For a matroid M and a subset X of $E(M)$, we let $(X, \{I \subseteq X : I \in \mathcal{I}(M)\})$ be the matroid denoted by $M|_X$. The size of a base of $M|_X$ is called the *rank* of X in M and the *rank function* of M is the function $r_M : 2^{E(M)} \to \mathbb{N}$ that maps every $X \subseteq E(M)$ to its rank. The rank of $E(M)$ is called the rank of M.

If M is a matroid, then we define λ_M, called the *connectivity function of* M, such that for every subset X of $E(M)$,

$$\lambda_M(X) = r_M(X) + r_M(E(M)\backslash X) - r_M(E(M)) + 1.$$

It is known that the function λ_M is symmetric and submodular.

Let A be a binary matrix and let E be the column labels of A. Let $\mathcal{I}$ be the collection of all those subsets I of E such that the columns of A with index in I are linearly independent. Then $M(A) := (E, \mathcal{I})$ is a matroid. Any matroid isomorphic to $M(A)$ for some matrix A is called a *binary matroid* and A is called a *representation of* M *over the binary field*.

Let G be a graph. Let $\mathcal{I}$ be the collection of all subsets I of $E(G)$ such that $(V(G), I)$ is a forest. Then $M(G) := (E(G), \mathcal{I})$ is a matroid. Any matroid isomorphic to $M(G)$ for some graph G is called a *graphic matroid*.

For $n \geqslant r \geqslant 1$, the *uniform matroid* $U_{r,n}$ is the matroid on a ground set E of size n where the independent sets are all subsets of size at most r in E.

Here, we observe that every matroid of branch-width at most 2 is binary. We will use this fact in Chapter 9. This can be observed from the known minor characterizations for binary matroids and matroids of branch-width at most 2. For the definition of matroid minors, we refer to [156].

Theorem 1.12 (Tutte [184, 185])**.** *A matroid is binary if and only if it has no minor isomorphic to* $U_{2,4}$.

Theorem 1.13 (Dharmatilake [74])**.** *A matroid has branch-width at most* 2 *if and only if it has no minor isomorphic to* $U_{2,4}$ *and* $M(K_4)$.

Corollary 1.14. *Every matroid of branch-width at most* 2 *is binary.*

Proof. This follows from Theorems 1.12 and 1.13. □

Note. Chapter 6 is a joint work with Jisu Jeong and Sang-il Oum, and it is published in [119]. Chapter 8 is a joint work with Sang-il Oum, and it is published in [135]. Chapter 9 is a joint work with Isolde Adler and Mamadou Moustapha Kanté, and its extended abstract appears in [3], and Chapter 7 is based on the work for distance-hereditary graphs established in the same paper. Chapter 10 is a joint work with Mamadou Moustapha Kanté, Eun jung Kim, and Christophe Paul [121]. Chapter 11 is a joint work with Eun jung Kim and Sang-il Oum.

Chapter 2. Cut-rank function

We discuss the cut-rank function of a graph and its properties. Let G be a graph. For two subsets X, Y of $V(G)$, we define the function $\rho^*_G(X,Y) := \operatorname{rank} A(G)[X,Y]$ where $A(G)$ is the adjacency matrix of G. The *cut-rank* function ρ_G of a graph G is defined as

$$\rho_G(X) := \rho^*_G(X, V(G)\backslash X) = \operatorname{rank} A(G)[X, V(G)\backslash X].$$

It was motivated from the matroid rank function [30, 150]. Bouchet [30] observed that the connectivity function of a binary matroid is equal to the cut-rank function of its fundamental graph. We recall this relation in Section 2.2.

The cut-rank function is invariant under taking local complementation, and it satisfies the submodular inequality.

Lemma 2.1 (Bouchet [32]; See Oum [150])**.** *If two graphs G and H are locally equivalent, then $\rho_G(X) = \rho_H(X)$ for all $X \subseteq V(G)$.*

Lemma 2.2 (Oum and Seymour [155])**.** *For a graph G and all $A, B, A', B' \subseteq V(G)$,*

$$\rho^*_G(A,B) + \rho^*_G(A',B') \geqslant \rho^*_G(A \cap A', B \cup B') + \rho^*_G(A \cup A', B \cap B').$$

By Lemma 2.2, we have the submodular inequality:

$$\rho_G(A) + \rho_G(B) \geqslant \rho_G(A \cap B) + \rho_G(A \cup B)$$

for all $A, B \subseteq V(G)$.

We will use the following lemmas.

Lemma 2.3 (Oum [150, Lemma 4.4])**.** *Let G be a graph and $v \in V(G)$. Let (X_1, X_2), (Y_1, Y_2) be vertex partitions of $V(G)\backslash\{v\}$. Then we have*

$$\rho_{G\backslash v}(X_1) + \rho_{G*v\backslash v}(Y_1) \geqslant \rho_G(X_1 \cap Y_1) + \rho_G(X_2 \cap Y_2) - 1.$$

Similarly if w is a neighbor of v, then

$$\rho_{G\backslash v}(X_1) + \rho_{G\wedge vw\backslash v}(Y_1) \geqslant \rho_G(X_1 \cap Y_1) + \rho_G(X_2 \cap Y_2) - 1.$$

Lemma 2.3 is equivalent to the following lemma, which we will use in Chapter 8.

Lemma 2.4. *Let G be a graph and $v \in V(G)$. Let X_1, X_2, Y_1, Y_2 be subsets of $V(G)\backslash\{v\}$ such that $X_1 \cup X_2 = Y_1 \cup Y_2$ and $X_1 \cap X_2 = Y_1 \cap Y_2 = \varnothing$. Then*

$$\rho^*_G(X_1, X_2) + \rho^*_{G*v}(Y_1, Y_2) \geqslant \rho^*_G(X_1 \cap Y_1, X_2 \cup Y_2 \cup \{v\}) + \rho^*_G(X_1 \cup Y_1 \cup \{v\}, X_2 \cap Y_2) - 1.$$

Similarly if $w \in X_1 \cup X_2$ is a neighbor of v, then

$$\rho^*_G(X_1, X_2) + \rho^*_{G\wedge vw}(Y_1, Y_2) \geqslant \rho^*_G(X_1 \cap Y_1, X_2 \cup Y_2 \cup \{v\}) + \rho^*_G(X_1 \cup Y_1 \cup \{v\}, X_2 \cap Y_2) - 1.$$

Proof. Apply Lemma 2.3 with $G' = G[X_1 \cup X_2 \cup \{v\}]$. □

2.1 Blocking sequences

Let A, B be two disjoint subsets of the vertex set of a graph G. By the definition of ρ_G^* and ρ_G, it is clear that

$$\text{if } A \subseteq X \subseteq V(G)\backslash B, \text{ then } \rho_G^*(A, B) \leqslant \rho_G(X).$$

What prevents us to achieve the equality for some X? We present a tool called a blocking sequence, that is a certificate to guarantee that no such X exists. Blocking sequences were introduced by Geelen [102] for studying binary even delta-matroids. Oum [153] used blocking sequences to implement an algorithm for approximating rank-width. For a similar concept for matroids, we refer to [156, Section 13.3]. In Chapter 8, we will widely use this concept to find certain vertex-minors in large prime graphs.

A sequence $v_1, v_2, \ldots, v_m$ ($m \geqslant 1$) is called a *blocking sequence* of a pair (A, B) of disjoint subsets A, B of $V(G)$ if

(a) $\rho_G^*(A, B \cup \{v_1\}) > \rho_G^*(A, B)$,

(b) $\rho_G^*(A \cup \{v_i\}, B \cup \{v_{i+1}\}) > \rho_G^*(A, B)$ for all $i = 1, 2, \ldots, m-1$,

(c) $\rho_G^*(A \cup \{v_m\}, B) > \rho_G^*(A, B)$,

(d) no proper subsequence of $v_1, \ldots, v_m$ satisfies (a), (b), and (c).

The condition (d) is essential for the following standard lemma.

Lemma 2.5. *Let $v_1, v_2, \ldots, v_m$ be a blocking sequence for (A, B) in a graph G. Let X, Y be disjoint subsets of $\{v_1, v_2, \ldots, v_m\}$ such that if $v_i \in X$ and $v_j \in Y$, then $i < j$. Then*

$$\rho_G^*(A \cup X, B \cup Y) = \rho_G^*(A, B)$$

if and only if $v_1 \notin Y$, $v_m \notin X$, and for all $i \in \{1, 2, \ldots, m-1\}$, either $v_i \notin X$ or $v_{i+1} \notin Y$.

Proof. The forward direction is trivial. Let us prove the backward implication. Let $k = \rho_G^*(A, B)$. It is enough to prove $\rho_G^*(A \cup X, B \cup Y) \leqslant k$. Suppose that $v_1 \notin Y$, $v_m \notin X$, and for all $i \in \{1, 2, \ldots, m-1\}$, either $v_i \notin X$ or $v_{i+1} \notin Y$ and yet $\rho_G^*(A \cup X, B \cup Y) > k$. We may assume that $|X| + |Y|$ is chosen to be minimum. If $|X| \geqslant 2$, then we can partition X into two nonempty sets X_1 and X_2. Then by the hypothesis, $\rho_G^*(A \cup X_1, B \cup Y) = \rho_G^*(A \cup X_2, B \cup Y) = k$. By Lemma 2.2, we deduce that $\rho_G^*(A \cup X_1, B \cup Y) + \rho_G^*(A \cup X_2, B \cup Y) \geqslant k + \rho_G^*(A \cup X, B \cup Y)$ and therefore we deduce that $\rho_G^*(A \cup X, B \cup Y) \leqslant k$. So we may assume $|X| \leqslant 1$. By symmetry we may also assume $|Y| \leqslant 1$. Then by the condition (d), this is clear. □

The following proposition states that a blocking sequence is a certificate that $\rho_G(X) > \rho_G^*(A, B)$ for all $A \subseteq X \subseteq V(G)\backslash B$. This appears in almost all applications of blocking sequences. The proof uses the submodular inequality in Lemma 2.2.

Proposition 2.6 (Geelen [102, Lemma 5.1]; see Oum [153])**.** *Let G be a graph and A, B be two disjoint subsets of $V(G)$. Then G has a blocking sequence for (A, B) if and only if $\rho_G(X) > \rho_G^*(A, B)$ for all $A \subseteq X \subseteq V(G)\backslash B$.*

2.2 Matroid rank function

We discuss the direct relation between the rank function of a binary matroid and the cut-rank function of its fundamental graph, observed in [30, 150]. This relation will be used to show that the computation of linear rank-width is NP-hard in Section 9.1. Also, we will use the fundamental graph of a binary matroid in Section 9.3 to provide an algorithm to compute the path-width of a matroid of branch-width 2, provided that the matroid is given by an independent set oracle.

We define a fundamental graph of a binary matroid. Let G be a bipartite graph with a bipartition (S,T). We define $M(G,S,T)$ as the binary matroid represented by the $S \times V(G)$ matrix

$$(I_S \quad A(G)[S,T])$$

where I_S is the $|S| \times |S|$ identity matrix. If $M = M(G,S,T)$, then we call G a *fundamental graph* of M. We remark that $|E(M)| = |V(G)|$.

If a binary matroid M is given with an independent set oracle, then we can compute a fundamental graph of M in time $\mathcal{O}(|E(M)|^2)$ as follows. We first run a greedy algorithm to find a base B of M in time $\mathcal{O}(|E(M)|)$ [156, Section 1.8]. After choosing one base B, for each $e \in B$ and $e' \in E(M)\backslash B$, we test whether $(B\backslash\{e\}) \cup \{e'\}$ is again a base in time $\mathcal{O}(|E(M)|^2)$. We create a bipartite graph G_M on the bipartition $(B, E(M)\backslash B)$ where for each $e \in B$ and $e' \in E(M)\backslash B)$, $ee' \in E(G_M)$ if and only if $(B\backslash\{e\}) \cup \{e'\}$ is again a base. It is not hard to check that

$$(I_B \quad A(G_M)[B, E(M)\backslash B])$$

is a representation of the matroid M, and thus G_M is a fundamental graph of M.

The following relation is observed by Bouchet [30]; see also Oum [150].

Proposition 2.7 (Bouchet [30]; Oum [150])**.** *Let G be a bipartite graph with a bipartition (S,T) and let $M := M(G,S,T)$. For every $X \subseteq V(G)$, $\rho_G(X) = \lambda_M(X) - 1$.*

A binary matroid may have many fundamental graphs, but it is known that two different fundamental graphs of a binary matroid are pivot-equivalent [30, 150]. Roughly speaking, the new fundamental graph that corresponds to a new base obtained from a given base by removing an element e and adding an element e', can be obtained by pivoting the edge ee' in the original fundamental graph.

Chapter 3. Width parameters

We introduce graph width parameters, such as rank-width, tree-width, and path-width.

Rank-width is a graph parameter introduced by Oum and Seymour [155, 150] for efficiently approximating the *clique-width* [62] of a graph. Linear rank-width can be seen as a linearized variant of rank-width. We survey known theorems of rank-width in Section 3.1.

Tree-width and *path-width* are graph parameters introduced by Robertson and Seymour, in their papers on the Graph Minor Theorem [169, 171, 175, 176, 177]. We define these parameters in Section 3.2, and compare basic properties of them with the properties of rank-width and linear rank-width.

Since the *branch-width* and the *path-width* of a matroid are related to rank-width and linear rank-width of its fundamental graph, we introduce those parameters in the last section. This relation will be used to obtain a polynomial-time algorithm to compute the path-width of a matroid of branch-width at most 2 in Section 9.3.

3.1 Rank-width and linear rank-width

For the definition of rank-width, we refer to the papers by Oum [155, 150]. A tree is *subcubic* if it has at least two vertices and every inner vertex has degree 3. A *rank-decomposition* of a graph G is a pair (T, L), where T is a subcubic tree and L is a bijection from the vertices of G to the leaves of T. For an edge e in T, $T\backslash e$ induces a partition (X_e, Y_e) of the leaves of T. The *width* of an edge e is defined as $\rho_G(L^{-1}(X_e))$. The *width* of a rank-decomposition (T, L) is the maximum width over all edges of T. The *rank-width* of G, denoted by $\mathrm{rw}(G)$, is the minimum width over all rank-decompositions of G. If $|V(G)| \leqslant 1$, then G admits no rank-decomposition and $\mathrm{rw}(G) = 0$.

Oum [151] proved that the vertex-minor relation is a well-quasi-ordering on the class of graphs of bounded rank-width.

Theorem 3.1 (Oum [151]). *For a positive integer k and an infinite sequence $G_1, G_2, \ldots$ of graphs of rank-width at most k, there exist G_i and G_j with $i < j$ such that G_i is isomorphic to a vertex-minor of G_j.*

We briefly mention how Theorem 3.1 implies that every vertex-minor closed class of graphs with bounded rank-width can be characterized by a finite set of vertex-minor minimal graphs (Theorem 1.11). Let $\mathcal{C}$ be a vertex-minor closed class of graphs with bounded rank-width, and let $\mathcal{O}_\mathcal{C}$ be a vertex-minor obstruction set for $\mathcal{C}$. If $\mathcal{O}_\mathcal{C}$ is infinite, then Theorem 3.1 implies that there exist two graphs G, G' in $\mathcal{O}_\mathcal{C}$ such that G is a vertex-minor isomorphic to G', contradicting to the fact that there are no two locally equivalent graphs in $\mathcal{O}_\mathcal{C}$ and both G and G' are vertex-minor minimal graphs for $\mathcal{C}$. Therefore, we conclude that $\mathcal{O}_\mathcal{C}$ is finite.

From Theorem 1.11, for a fixed k, the class of graphs with rank-width at most k can be characterized by a finite list of vertex-minor minimal graphs. Oum [150] establishes an upper bound on the number of vertices in a vertex-minor minimal graph for the class of graphs of rank-width at most k.

Theorem 3.2 (Oum [150]). *For $k \geqslant 1$, the number of vertices in a vertex-minor minimal graph for the class of graphs of rank-width at most k is at most $\frac{6^{k+1}-1}{5}$.*

From Theorem 3.2, we can enumerate all vertex-minor minimal graphs from graphs up to the bound on the obstructions.

There are some developments on algorithms for testing whether an input graph has rank-width at most k and compute a rank-decomposition if it has rank-width at most k. Oum and Seymour [155] first provided an algorithm to find a rank-decomposition of width at most $3k+1$ or confirm that the input n-vertex graph has rank-width larger than k in time $O(8^k \cdot n^9 \log n)$. Later, Oum [153] improved the running time into $\mathcal{O}(8^k \cdot n^4)$, and in the same paper, he remarked that it is possible to find a rank-decomposition of width $3k-1$ or confirm that the rank-width is bigger than k in time $\mathcal{O}(f(k) \cdot n^3)$ with some function f.

As a structural result, Oum proved the following.

Theorem 3.3 (Oum [150, 154]). *For a fixed bipartite circle graph H, every bipartite (or line, or circle) graph of sufficiently large rank-width contains a pivot-minor isomorphic to H.*

Robertson and Seymour [171] proved that for a fixed planar graph H, every graph of sufficiently large tree-width contain a minor isomorphic to a planar graph (Thoerem 3.14). Later, Geelen, Gerards, and Whittle [100] generalize this theorem into representable matroids, which states that for a fixed finite field $\mathbb{F}$ and a fixed planar matroid M, every matroid representable over the finite field $\mathbb{F}$ of sufficiently large branch-width must contain a minor isomorphic to M.

Linear rank-width is a variation of rank-width by restricting its tree to a caterpillar. However, we mostly use the following alternative definition of linear rank-width for convenience. We will observe that this definition is equivalent to the definition using rank-decompositions that are caterpillars.

A *linear layout* σ of a set S is a bijective function from S to $\{1, \ldots, |S|\}$, and for convenience, we denote it as a sequence $(\sigma^{-1}(1), \sigma^{-1}(2), \ldots, \sigma^{-1}(|S|))$. For a linear layout σ of S and $a, b \in S$, we denote by $a \leqslant_\sigma b$ or $b \geqslant_\sigma a$ if $\sigma(a) \leqslant \sigma(b)$, and we denote by $a <_\sigma b$ or $b >_\sigma a$ if $\sigma(a) < \sigma(b)$. For two linear layouts $\sigma_1 : S_1 \to \{1, \ldots, |S_1|\}$ and $\sigma_2 : S_2 \to \{1, \ldots, |S_2|\}$, we define the *sum* $\sigma_1 \oplus \sigma_2$ as a bijective mapping from $S_1 \cup S_2$ to $\{1, \ldots, |S_1| + |S_2|\}$ such that $(\sigma_1 \oplus \sigma_2)(x) = \sigma_1(x)$ if $x \in S_1$ and $(\sigma_1 \oplus \sigma_2)(x) = \sigma(x) + |S_1|$ if $x \in S_2$.

A linear layout of the vertex set of G is called a *linear layout of G*. The *width* of a linear layout L of G is defined as

$$\max_{v \in V(G)} (\rho_G(\{w : w \leqslant_L v\}).$$

We say that the width of L is 0 if $|V(G)| \leqslant 1$. The *linear rank-width* of G, denoted by $\mathrm{lrw}(G)$, is the minimum width over all linear layouts of G.

We clarify that this definition is the same as defining with caterpillar subcubic trees.

Lemma 3.4. *For a graph G, the minimum width over all rank-decompositions (T, L) of G where T is a caterpillar tree, is equal to the linear rank-width of G.*

Proof. Let t be the minimum width over all rank-decompositions (T, L) of G where T is a caterpillar tree. If G has no edges, then clearly, $t = \mathrm{lrw}(G) = 0$. We may assume that G has at least one edge.

Suppose G has a linear layout L of G of width k. From the assumption $k \geqslant 1$. Let T be a caterpillar subcubic tree with $|V(G)|$ leaves and let P be the longest path in T. Clearly, $|V(P)| = |V(G)|$, and for each internal vertex p of P, there is one leaf p' in $V(G)\backslash V(P)$ that is adjacent to p. Let $P = p_1 p_2 \cdots p_{|V(G)|}$ and we define the function L_G from $V(G)$ to the leaves of T such that

1. $L_G(L^{-1}(1)) = p_1$, $L_G(L^{-1}(|V(G)|)) = p_{|V(G)|}$, and
2. for each $2 \leqslant j \leqslant |V(G)|$, $L_G(L^{-1}(j)) = p'_j$.

For each edge $p_i p_{i+1}$ in the path P, the width of it is the same as

$$\rho^*_G(\{L^{-1}(j) : 1 \leqslant j \leqslant i\}, \{L^{-1}(j) : i+1 \leqslant j \leqslant |V(G)|\}).$$

Since G has at least one edge, there exists i such that $p_i p_{i+1}$ has width at least one. Since every edge in P incident with a leaf has width at most 1, we conclude that $t \leqslant k$.

The other direction is trivial because we can just take a linear layout from the caterpillar subcubic tree. □

If two graphs G and H are locally equivalent, then $\mathrm{rw}(G) = \mathrm{rw}(H)$ and $\mathrm{lrw}(G) = \mathrm{lrw}(H)$ by Lemma 2.1. It follows that if H is a vertex-minor or a pivot-minor of G, then $\mathrm{rw}(H) \leqslant \mathrm{rw}(G)$ and $\mathrm{lrw}(H) \leqslant \mathrm{lrw}(G)$. Clearly, $\mathrm{rw}(G) \leqslant \mathrm{lrw}(G)$ for any graph G.

We prove an upper bound on the linear rank-width of a graph. The following bound will be used to obtain an $\mathcal{O}(n^2 \log n)$-time algorithm to compute the linear rank-width of n-vertex distance-hereditary graphs in Chapter 9. Bodlaender, Gilbert, Hafsteinsson and Kloks [22] proved a similar relation between tree-width and path-width (Lemma 3.8).

Lemma 3.5. *Let k be a positive integer and let G be a graph of rank-width k such that $|V(G)| \geqslant 2$. Then* $\mathrm{lrw}(G) \leqslant k\lfloor \log_2 |V(G)| \rfloor$.

Proof. Let (T, L) be a rank-decomposition of G having width k. For convenience, we choose an edge e of T and subdivide it with introducing a new vertex x, and regard x as the root of T. For each internal vertex t of T with two subtrees T_1 and T_2 of $T \backslash t$ not containing x, let $\ell(t) := T_1$ and $r(t) := T_2$ if the number of leaves of T in T_1 is at least the number of leaves of T in T_2. Let S be a linear layout of G satisfying that

- for each $v_1, v_2 \in V(G)$ with the first common ancestor w of v_1 and v_2 in T, $L(v_1) <_S L(v_2)$ if $L(v_1) \in V(\ell(w))$.

We can construct such a linear layout inductively.

We show that S has width at most $k\lfloor \log_2 |V(G)| \rfloor$. Let w be a vertex of G that is not the first vertex of S and let $S_w := \{v : v <_S w\}$. Let P_w be the path from $L(w)$ to the root x in T. Note that for each $t \in V(P_w) \backslash \{L(w)\}$ and the subtree T' of $T \backslash t$ not containing x and $L(w)$,

- if $r(t) = T'$, then all leaves of T in T' are not contained in S_w, and
- if $\ell(t) = T'$, then all leaves of T in T' are contained in S_w.

Let Q be the set of all vertices t in P_w except w such that the subtree $\ell(t)$ does not contain x and $L(w)$.

	rank-width	tree-width
tree	1	1
cycle of length $\geqslant 5$	2	2
$n \times n$ grid	$n-1$ (Jelínek [118])	n (Robertson, Seymour [175])
complete graphs K_n	1	$n-1$ (Robertson, Seymour [175])
complete bipartite $K_{m,n}$	1	$\min\{m,n\}$ (Bodlaender, Möhring [26])

Table 3.1: Rank-width and tree-width.

Let $q_1, q_2, \ldots, q_m$ be the sequence of all vertices in Q such that for each $1 \leqslant j \leqslant m-1$, q_j is a descendant of q_{j+1} in T, and let Q_i be the set of all leaves of T contained in $\ell(q_i)$. Clearly, $S_w = Q_1 \cup Q_2 \cup \cdots \cup Q_m$ and $V(G) \backslash S_w \neq \varnothing$. Therefore, we have

$$\begin{aligned} |V(G)| &= |Q_1| + \cdots + |Q_m| + |V(G)\backslash S_w| \\ &\geqslant 1 + 2 + 4 + \cdots 2^{m-1} + 1 \\ &= 2^m. \end{aligned}$$

Thus, $m \leqslant \lfloor \log_2 |V(G)| \rfloor$.

Note that for each $1 \leqslant j \leqslant m$, $\rho^*_G(Q_i, V(G)\backslash S_w) \leqslant k$. Therefore, we have that

$$\rho_G(S_w) = \rho^*_G(Q_1 \cup \cdots \cup Q_m, V(G)\backslash S_w) \leqslant km \leqslant k\lfloor \log_2 |V(G)| \rfloor.$$

Since w was arbitrarily chosen, it implies that $\mathrm{lrw}(G) \leqslant k\lfloor \log_2 |V(G)| \rfloor$. □

3.2 Tree-width and path-width

For the definitions of tree-width and path-width, we refer to the book by Diestel [76]. A *tree-decomposition* of a graph G is a pair (T, B) of a tree T and a family $\mathcal{B} = \{B_t\}_{t \in V(T)}$ of vertex sets $B_t \subseteq V(G)$, called *bags*, satisfying the following three conditions:

(T1) $V(G) = \bigcup_{v \in V(T)} B_t$.

(T2) For every edge uv of G, there exists a vertex t of T such that $u, v \in B_t$.

(T3) For t_1, t_2 and $t_3 \in V(T)$, $B_{t_1} \cap B_{t_3} \subseteq B_{t_2}$ whenever t_2 is on the path from t_1 to t_3.

The *width* of a tree-decomposition $(T, \mathcal{B})$ is $\max\{|B_t| - 1 : t \in V(T)\}$. The *tree-width* of G, denoted by $\mathrm{tw}(G)$, is the minimum width over all tree-decompositions of G. A *path-decomposition* of a graph G is a tree-decomposition $(T, \mathcal{B})$ where T is a path. The *path-width* of G, denoted by $\mathrm{pw}(G)$, is the minimum width over all path-decompositions of G. It is known that tree-width and path-width do not increase when taking minors.

We compare values of the rank-width and the tree-width of some graphs in Table 3.1. As we discussed before, the big difference appear at dense graphs, such as complete graphs and complete bipartite graphs. A *rooted binary tree* is a tree with a root vertex such that the root has degree 2 and all other internal vertices have degree 3. For a positive integer n, the *complete rooted binary tree* of height n is denoted by T_n. We also compare values of the linear rank-width and the path-width of some graphs in Table 3.2. An

	linear rank-width	path-width
path	1	1
complete binary trees T_n	$\lceil \frac{n}{2} \rceil$ (Kwon [133])	$\lceil \frac{n}{2} \rceil$ (Ellis et al. [85]; Takahashi et al. [182])
cycle of length $\geqslant 5$	2	2
$n \times n$ grid	$n-1$ or n (Jelínek [118])	n (Robertson, Seymour [175])
complete graphs K_n	1	$n-1$ (Robertson, Seymour [175])
complete bipartite $K_{m,n}$	1	$\min\{m,n\}$ (Bodlaender, Möhring [26])

Table 3.2: Linear rank-width and path-width.

interesting fact is that path-width and linear rank-width are the same on trees [2, 133]. This is because the exactly same lemmas hold for both parameters.

Lemma 3.6 (Kwon [133])**.** *Let T be a tree and let $k \geqslant 1$. Then T has linear rank-width at most k if and only if for all vertices x in T at most two of the subtrees of $T \backslash x$ have linear rank-width k and all other subtrees have linear rank-width at most $k-1$.*

Lemma 3.7 (Ellis, Sudborough, and Turner [85]; Takahashi, Ueno and Kajitani [182])**.** *Let T be a tree and let $k \geqslant 1$. Then T has path-width at most k if and only if for all vertices x in T at most two of the subtrees of $T \backslash x$ have path-width k and all other subtrees have path-width at most $k-1$.*

We generalize Lemma 3.6 to distance-hereditary graphs in Section 5.3.

Similar to Lemma 3.5 the following relation is known.

Lemma 3.8 (Bodlaender, Gilbert, Hafsteinsson and Kloks [22])**.** *Let k be a positive integer and let G be a graph of tree-width k. Then $\mathrm{pw}(G) \leqslant (k+1)\log_2|V(G)|$.*

Oum [152] proved the following relation between rank-width and tree-width.

Theorem 3.9 (Oum [152])**.** *For a graph G, $\mathrm{rw}(G) \leqslant \mathrm{tw}(G) + 1$.*

Adler and Kanté [2] announced the following relation between linear rank-width and path-width. For completeness, we add a proof for it. It is easy to show this using the fact that the vertex separation number is equal to the path-width of a graph [128]. Let G be a graph and let L be a linear layout of G. For each $v \in V(G)$, we define

$$V_L(v) := \{u \in V(G) : u \leqslant_L v, \text{ and there is a vertex } w \text{ where } u <_L w \text{ and } uw \in E(G)\}.$$

The *vs-width* of the linear layout L is the maximum over all $V_L(v)$ for $v \in V(G)$. The *vertex separation number* is the minimum vs-width over all linear layouts of G, and we denote it by $\mathrm{vs}(G)$.

Theorem 3.10 (Kinnersley [128])**.** *For a graph G, $\mathrm{vs}(G) = \mathrm{pw}(G)$.*

Theorem 3.11 (Adler and Kanté [2])**.** *For a graph G, $\mathrm{lrw}(G) \leqslant \mathrm{pw}(G)$.*

Proof. Let $k = \mathrm{pw}(G)$. By Theorem 3.10, $\mathrm{vs}(G) = k$. Let L be a linear layout of G with vs-width k. It is easy to see that L also has width at most k with respect to linear rank-width. By the definition of vs-width, for each $v \in V(G)$, $\{u \in V(G) : u \leqslant_L v\}$ has at most k vertices that has a neighbor on $\{u \in V(G) : u >_L v\}$. Thus the matrix

$$A(G)[\{u \in V(G) : u \leqslant_L v\}, \{u \in V(G) : u >_L v\}]$$

has rank at most k. Since v is arbitrary, L has width at most k, and thus $\mathrm{lrw}(G) \leqslant k$. □

We cannot hope to bound tree-width or path-width in terms of a function of rank-width or linear rank-width because of complete graphs. However, as a reverse direction, Kwon and Oum [134] proved the following relationship.

Theorem 3.12 (Kwon and Oum [134]). *1. Every graph of rank-width k is a pivot-minor of a graph of tree-width at most $2k$.*

2. Every graph of linear rank-width k is a pivot-minor of a graph of path-width at most $k+1$.

The following two theorems give obstructions for graphs of large tree-width or path-width. For $r \geqslant 1$, the $r \times r$-grid is the graph with the vertex set $\{v_{i,j} : 1 \leqslant i, j \leqslant r\}$ such that $v_{i,j}$ and $v_{i',j'}$ are adjacent if $|i - i'| + |j - j'| = 1$. Note that the $r \times r$-grid has tree-width r [175].

Theorem 3.13 (Robertson and Seymour [169]; Bienstock, Robertson, Seymour and Thomas [13]). *For a fixed forest T, every graph of path-width at least $|V(T)| - 1$ contains a minor isomorphic to T.*

Note that for every forest T, the bound $|V(T)| - 1$ is best possible, because the complete graph $K_{|V(T)|-1}$ has path-width $|V(T)| - 2$ but it does not contain a minor isomorphic to T. We will use Theorem 3.13 in Chapter 7 for finding a tree as a vertex-minor.

Theorem 3.14 (Robertson and Seymour [171]; Robertson, Seymour, Thomas [168]). *For every $r \geqslant 1$, every graph of tree-width at least 20^{2r^5} contains a minor isomorphic to the $r \times r$-grid.*

Diestel, Jensen, Gorbunov, and Thomassen [77] discovered a simpler proof of Theorem 3.14. Also, there have been many attempts to improve this theorem. This bound was improved into $2^{\mathcal{O}(r^2 \log_2 r)}$ by Kawarabayashi and Kobayashi [124], and into $2^{\mathcal{O}(r \log_2 r)}$ by Leaf and Seymour [138]. Chekuri and Chuzhoy [47] proved that the function can be taken to be $\mathcal{O}(r^c)$ for some constant c, which is polynomial in r.

Theorem 3.14 has many applications on graph algorithms. One of the remarkable applications is for the Disjoint Paths problem, which was studied by Robertson and Seymour [176]. Given a graph G and pairs of vertices $(x_1, y_1), \dots, (x_k, y_k)$, the Disjoint Paths problem asks whether there exist pairwise vertex-disjoint paths $P_1, \dots, P_k$ such that for each $1 \leqslant i \leqslant k$, the first and last vertices of P_i are x_i and y_i. In the contexts of VLSI layout design and virtual circuit routing in high-speed internet, the Disjoint Paths problem has been focused as a central problem [94, 174]. Based on Theorem 3.14, Robertson and Seymour [176] showed that the Disjoint Paths problem can be solved in time $\mathcal{O}(n^3)$. Kawarabayashi, Kobayashi, Reed [125] improved this running time into $\mathcal{O}(n^2)$.

There have been several works on developing algorithms that either outputs that the tree-width of an input graph has tree-width larger than k or gave an approximate tree-decomposition [6, 176, 136, 165, 20, 23, 5, 87, 21]. Arnborg, Corneil, Proskurowski [6] first gives an algorithm to compute tree-decomposition of width k in time $\mathcal{O}(n^{k+2})$ if exists. Bodlaender [20] improved this running time into

$\mathcal{O}(k^{\mathcal{O}(k^3)}) \cdot n$. Later, Bodlaender, Drange, Dregi, Fomin, Lokshtanov, Pilipczuk [21] proved that there exists an algorithm that in time $2^{\mathcal{O}(k)} \cdot n$ either outputs that the tree-width of an input graph G is larger than k, or gives a tree-decomposition of G of width at most $5k + 4$.

We will review meta-algorithmic results for graphs of bounded tree-width in the next subsection. For further algorithmic applications, we refer to papers [7, 12, 17, 25].

3.3 Meta-theorems for graphs of bounded width

We describe known meta-algorithmic results for the class of graphs of bounded tree-width and bounded rank-width. For clear description, we define monadic second-order logic formulas. We refer to the book [60] written by Courcelle and Engelfriet for intensive study on logic formulas and graph decompositions.

Let D be a finite set. A function $F : D^m \to \{\mathsf{true}, \mathsf{false}\}$ is a *relation symbol* on D with arity m. A function $F : (2^D)^m \to \{\mathsf{true}, \mathsf{false}\}$ is a *set predicate* on D with arity m. A pair $S = (D, \{F_1, \ldots, F_k\})$ is called a *relational structure* if

1. D is a finite set, and
2. for each i, F_i is either a relation symbol or a set predicate on D.

For instance, we can give an adjacency relation adj on the vertex set of a graph G, where for $v, w \in V(G)$, $\mathsf{adj}(v, w)$ is true if and only if v and w are adjacent in G. Also, we can give an incidency relation inc on $V(G) \cup E(G)$ in G, where for $v, e \in V(G) \cup E(G)$, $\mathsf{inc}(v, e)$ is true if and only if $v \in V(G)$, $e \in E(G)$ and v is incident with e in G.

Now, we define logic formulas. Let $(D, \{F_1, \ldots, F_k\})$ be a relational structure. A variable is a *first-order variable* if it denotes an element of D, and is a *set variable* if it denotes a set of elements of D. A logic formula on the relational structure is called a *monadic second-order logic formula* if it is written by using $\exists, \forall, \wedge, \neg, \vee, \in, \mathsf{true}$ and F_i, with first-order variables and set variables.

Let G be a graph and we denote by adj the adjacency relation on $V(G)$ (or $V(G) \cup E(G)$), and we denote by inc the incidency relation on $V(G) \cup E(G)$. A monadic second-order logic formula on $(V(G), \{\mathsf{adj}\})$ is called a monadic second-order logic formula of the first type (MSO_1), and a monadic second-order logic formula on $(V(G) \cup E(G), \{\mathsf{inc}, \mathsf{adj}\})$ is called a monadic second-order logic formula of the second type (MSO_2).

We give some examples of logic formulas for some properties of graphs.

1. (G has a 3-coloring on the vertices such that adjacent vertices have different colors. (3-coloring))

$$\exists X, Y[X \subseteq V(G) \wedge Y \subseteq V(G) \wedge \forall u, v\{\neg \mathsf{adj}(u, v) \vee$$
$$((\neg(u \in X) \vee \neg(v \in X)) \wedge (\neg(u \in Y) \vee \neg(v \in Y)) \wedge ((u \in X \vee u \in Y) \vee (v \in X \vee v \in Y)))\}]$$

2. (G has a cycle containing all vertices. (Hamiltonian cycle))

$$\begin{aligned}
&\exists X[X \subseteq E(G) \\
&\wedge \forall v\{\neg(v \in V(G)) \vee \exists e_1 \exists e_2(e_1 \in X \wedge e_2 \in X \wedge \neg(e_1 = e_2) \wedge \mathsf{inc}(v, e_1) \wedge \mathsf{inc}(v, e_2)) \wedge \\
&\quad \forall e(\neg(\mathsf{inc}(v, e) \wedge e \in X) \vee (e = e_1 \vee e = e_2))\} \\
&\wedge \forall Y\{\neg(Y \subseteq V(G) \wedge \exists x(x \in Y) \wedge \exists x' \neg(x' \in Y)) \\
&\quad \vee \exists e(e \in X \wedge \exists v(v \in Y \wedge \mathsf{inc}(v, e)) \wedge \exists w(\neg(w \in Y) \wedge \mathsf{inc}(w, e)))\}]
\end{aligned}$$

We remark that the property of 3-colorability is represented by the MSO_1 formula, and the property of having a cycle containing all vertices is represented by the MSO_2 formula. It is known that the property of having a cycle containing all vertices cannot be written in an MSO_1 formula [60]. Note that every MSO_1 formula for a graph property is also an MSO_2 formula, but not vice versa.

Courcelle [58] provided the following algorithmic meta-theorem.

Theorem 3.15 (Courcelle [58])**.** *Every graph property expressible in a monadic second-order logic formula of the second type (*MSO_2*) can be decided in linear time on graphs of bounded tree-width.*

As we can observe from the relation of rank-width and tree-width, graphs of bounded rank-width is bigger than graphs of bounded tree-width. Courcelle, Makowsky and Rotics [61] proved a meta-theorem on graphs of bounded rank-width as well.

Theorem 3.16 (Courcelle, Makowsky and Rotics [61])**.** *Every graph property expressible in a monadic second-order logic formula of the first type (*MSO_1*) can be decided in cubic time on graphs of bounded rank-width.*

3.4 Matroid branch-width and path-width

We define the notion of the *branch-width of a matroid* and the *path-width of a matroid* using the connectivity function of a matroid. Let M be a matroid, and let λ_M be the connectivity function of M. We remind that for every subset X of $E(M)$,

$$\lambda_M(X) = r_M(X) + r_M(E(M) \backslash X) - r_M(E(M)) + 1.$$

A *branch-decomposition of* M is a pair (T, L), where T is a subcubic tree and L is a bijection from $E(M)$ to the leaves of T. For an edge e in T, $T \backslash e$ induces a partition (X_e, Y_e) of the leaves of T. The *width* of an edge e is defined as $\lambda_M(L^{-1}(X_e))$. The *width* of a branch-decomposition (T, L) is the maximum width over all edges of T. The *branch-width* of M, denoted by $\mathrm{bw}(M)$, is the minimum width of all branch-decompositions of M. If $|E(M)| \leqslant 1$, then $\mathrm{bw}(M) = 0$.

An ordering $e_1, \ldots, e_n$ of the ground set $E(M)$ is called a *linear layout* of M. The *width* of a linear layout $e_1, \ldots, e_n$ of M is

$$\max_{1 \leqslant i \leqslant n-1} \{\lambda_M(\{e_1, \ldots, e_i\})\}.$$

The *path-width* of M, denoted by $\mathrm{pw}(M)$, is defined as the minimum width over all linear layouts of M.

The following relation is obtained from Proposition 2.7.

Proposition 3.17 (Oum [150])**.** *Let G be a bipartite graph with a bipartition (A, B) and let $M := M(G, A, B)$. Then* $\mathrm{rw}(G) = \mathrm{bw}(M) - 1$ *and* $\mathrm{lrw}(G) = \mathrm{pw}(M) - 1$.

Similar to the case of linear rank-width, there is no known polynomial-time algorithm to compute the path-width of a matroid of bounded branch-width. Using Proposition 3.17, we prove in Section 9.3 that we can compute in polynomial time the path-width of a matroid of branch-width at most 2, provided that the matroid is given with an independent set oracle.

Chapter 4. Split decompositions

Split decompositions are graph decompositions, introduced by Cunningham [65, 64]. A *split* of a graph G is a vertex partition (X, Y) of G such that $|X|, |Y| \geqslant 2$ and $\rho_G(X) \leqslant 1$. In other words, (X, Y) is a split in G if $|X|, |Y| \geqslant 2$ and there exist $X' \subseteq X$ and $Y' \subseteq Y$ such that $\{xy \in E(G) : x \in X, y \in Y\} = \{xy : x \in X', y \in Y'\}$. Splits are also called as 1-*joins*, or *joins* [96]. A connected graph is called a *prime graph* if it has no split.

Roughly speaking, split decompositions of a graph G can be obtained by successively decomposing a graph along *splits*. Split decompositions have been used to design efficient algorithms to solve graph problems. A usual approach to design an algorithm with a certain decomposition is to recursively decompose an input graph into smaller graphs, until there are no decomposable graphs. Then we solve for these small graphs, and recursively combine the solutions to find a solution for the original graph. In this direction, there are several results which show that a problem can be solved in polynomial-time for each bag of the split decomposition, then the solution for whole graph can be merged in polynomial-time. For instance, such relations were established for Independent Set [64, 162], Clique Number, Dominating Number [162] problems. Using those relations, Rao [162] showed that above problems can be solved in polynomial time on graphs whose prime induced subgraphs have bounded size, and the Independent Set problem can be solved in polynomial time on parity graphs [41, 55].

Circle graphs are deeply related to split decompositions. One circle graph may have different circle representations, but Bouchet [29], Naji [145], and Gabor, Hsu, and Supowit [96] independently showed that every prime circle graph has a unique chord diagram. Based on this, they gave a polynomial-time algorithm to recognize circle graphs. Later Spinrad [180] developed an $\mathcal{O}(n^2)$ time algorithm to recognize circle graphs, and Corneil, Gioan, Paul, and Tedder [56] developed an $\mathcal{O}(n+m)\alpha(n+m)$ time algorithm for the same problem, where α is the inverse Ackermann function, and n, m are the number of vertices and edges in an input graph.

The class of distance-hereditary graphs is equal to the class of graphs totally decomposable with respect to the split decomposition [9, 31, 110]. We approach many problems based on this structure. Bouchet [31] developed the concept of local complementations in split decompositions. We further develop local complementations and vertex-minors in split decompositions, and investigate the characterization of linear rank-width on distance-hereditary graphs in Chapter 5. Bouchet [31] proved that any two locally equivalent trees are isomorphic, and we extend this result in Chapter 5 into a certain type of block graphs. We also use split decompositions of distance-hereditary graphs for Thread Vertex Deletion and Distance-Hereditary Vertex Deletion in Chapter 10 and 11.

4.1 Prime graphs

We recall that a prime graph is a connected graph having no split. Since every connected graph G with at most 3 vertices cannot have vertex partitions (A, B) with $|A|, |B| \geqslant 2$, it is prime. We remark that every connected graph with 4 vertices has a split; see Figure 4.1. Note that every cycle of length at least 5 is prime. If a connected graph G has a cut vertex x such that $G \backslash x$ has components $G_1, G_2, \ldots, G_m$,

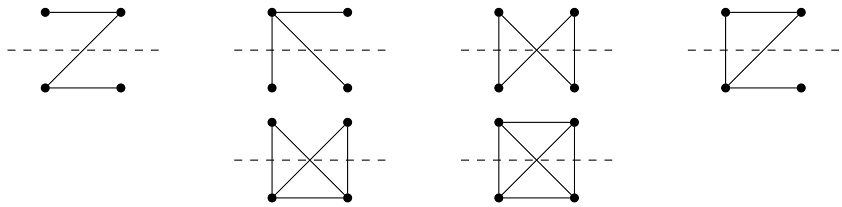


Figure 4.1: Connected 4-vertex graphs. Dashed edges represent splits.

then $(V(G_1) \cup \{x\}, V(G)\backslash(V(G_1) \cup \{x\}))$ is a split. Therefore, every prime graph of at least 5 vertices is 2-connected.

Bouchet showed useful properties of prime graphs. From the definition of prime graphs, every graph locally equivalent to a prime graph is prime.

Theorem 4.1 (Bouchet [29]). *Let G be a prime graph with $|V(G)| \geqslant 6$. Then there exist a vertex v and a neighbor w of v such that one of $G\backslash v$, $G * v\backslash v$ and $G \wedge vw\backslash v$ is prime.*

We remark that every prime graph of 5 vertices is locally equivalent to C_5. As a corollary of Theorem 4.1, we obtain the following.

Corollary 4.2 (Bouchet [29]). *Every prime graph on at least 5 vertices must contain a vertex-minor isomorphic to C_5.*

The following lemma is natural, and will be used in Chapter 8.

Lemma 4.3. *If a prime graph H on at least 5 vertices is a vertex-minor of a graph G, then G has a prime induced subgraph G_0 such that G_0 has a vertex-minor isomorphic to H.*

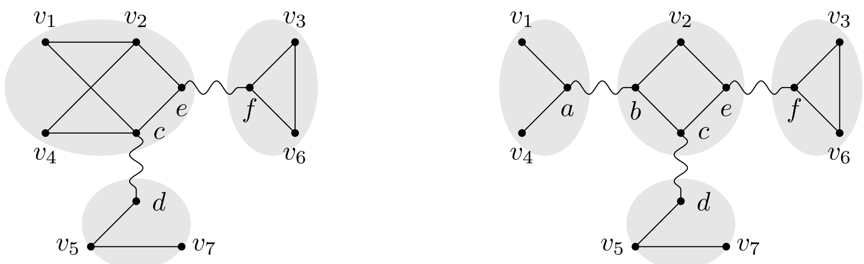
Proof. We may assume that G is connected. It is enough to prove the following claim: if G has a split (A, B), then there exists a vertex v such that H is isomorphic to a vertex-minor of $G\backslash v$. Let G' be a graph locally equivalent to G such that H is an induced subgraph of G'. We have $\rho_H(V(H) \cap A) = \rho^*_{G'}(V(H) \cap A, V(H) \cap B) \leqslant \rho^*_{G'}(A, B) \leqslant 1$ and therefore $|V(H) \cap A| \leqslant 1$ or $|V(H) \cap B| \leqslant 1$ because H is prime. By symmetry, let us assume $|V(H) \cap B| \leqslant 1$. Let us choose $x \in B$ such that x has a neighbor in A and $x \in V(H)$ if $V(H) \cap B$ is nonempty.

Let H' be a vertex-minor of G on $A \cup \{x\}$ such that H is isomorphic to a vertex-minor of H'. Then $H' = G * v_1 * v_2 \cdots * v_n\backslash(B\backslash\{x\})$ for some sequence $v_1, v_2, \ldots, v_n$ of vertices. We may choose H' and n so that n is minimized.

Suppose $n > 0$. Then $v_n \in B\backslash\{x\}$. Let $H_0 = G * v_1 * v_2 \cdots * v_{n-1}\backslash(B\backslash\{x, v_n\})$. Since $(A, \{x, v_n\})$ is a split of H_0, one of the following holds.

1. The two vertices v_n and x have the same set of neighbors in A.
2. The vertex v_n has no neighbors in A.
3. The vertex x has no neighbors in A.

If we have the case (i), then $(H_0\backslash v_n) * x = H'$ and therefore H is isomorphic to a vertex-minor of $H_0\backslash v_n$, contradicting our assumption that H is chosen to minimize n. If we have the case (ii), then $H_0\backslash v_n = H'$, contradicting the assumption too. Finally if we have the case (iii), then x is adjacent to v_n in G because G is connected. Then $H_0 * v_n\backslash v_n$ is isomorphic to $H_0 * v_n\backslash x$. Then $H_0\backslash x$ has a vertex-minor isomorphic to H, contradicting our assumption that n is minimized. $\square$



($\Longrightarrow$) Replacing a bag with its simple decomposition

($\Longleftarrow$) Recomposing along a marked edge ab

Figure 4.2: Two operations on a split decomposition.

4.2 Split decompositions

For the definitions related to split decompositions, we will follow the notations used by Bouchet [31]. A *marked graph* D is a connected graph D with a distinguished set of edges $M(D)$, called *marked edges*, that form a matching such that every edge in $M(D)$ is a cut edge. The ends of the marked edges are called *marked vertices*, and the components of $D\backslash M(D)$ are called *bags* of D. Edges and vertices that are not marked are called *unmarked*. If (X, Y) is a split in G, then we construct a marked graph D with the vertex set $V(G) \cup \{x', y'\}$ for two distinct new vertices $x', y' \notin V(G)$ and the edge set $E(G[X]) \cup E(G[Y]) \cup \{x'y'\} \cup E'$ where we define $x'y'$ as marked and

$$\begin{aligned} E' := \{x'x : x \in X &\text{ and there exists } y \in Y \text{ such that } xy \in E(G)\} \cup \\ \{y'y : y \in Y &\text{ and there exists } x \in X \text{ such that } xy \in E(G)\}. \end{aligned}$$

The marked graph D is called a *simple decomposition of* G. We remark that $(D \wedge x'y')\backslash\{x', y'\} = G$.

A *split decomposition* of a connected graph G is a marked graph D defined inductively to be either G or a marked graph defined from a split decomposition D' of G by replacing a component H of $D'\backslash M(D')$ by a simple decomposition of H. For a marked edge xy in a split decomposition D, the *recomposition of* D *along* xy is the split decomposition $D' := (D \wedge xy)\backslash\{x, y\}$. See Figure 4.2 for an example of decomposing or recomposing a split decomposition. For a split decomposition D, let $\widehat{D}$ denote the connected graph obtained from D by recomposing all marked edges. Note that if D is a split decomposition of G, then $\widehat{D} = G$. Since marked edges of a split decomposition D are cut edges and form a matching, if we contract all the unmarked edges in D, then we obtain a tree and we call it the *decomposition tree* of D and denote it by T_D. Obviously, the vertices of T_D are in bijection with the bags of D. To distinguish the vertices of a split decomposition tree from the vertices of the original graph, we call a vertex of T_D as a *node* of it.

We can observe that a complete graph or a star has many different ways to decompose it because every non-trivial vertex partition of it is a split. Cunningham and Edmonds [65] developed a canonical way to decompose a graph into a split decomposition. A split decomposition D of G is called a *canonical split decomposition* if each bag of D is either a prime graph, a star graph, or a complete graph, and D is not the refinement of a split decomposition with the same property. The following is due to Cunningham and Edmonds [65], and Dahlhaus [68].

Theorem 4.4 (Cunningham and Edmonds [65]; Dahlhaus [68])**.** *Every connected graph G has a unique canonical split decomposition, up to isomorphism, and it can be computed in time $\mathcal{O}(|V(G)| + |E(G)|)$.*

For convenience, a tree T is called a *decomposition tree of* G if it is a split decomposition tree of D where D is the canonical split decomposition of G. Note that all split decomposition trees of G are isomorphic by Theorem 4.4.

Canonical split decompositions can be characterized as follows. Let D be a split decomposition of G with bags that are either primes, or complete graphs or stars (it is not necessarily a canonical split decomposition). The *type of a bag* of D is either P, K, or S depending on whether it is a prime, a complete graph, or a star, respectively. The *type of a marked edge uv* is AB where A and B are the types of the bags containing u and v respectively. If $A = S$ or $B = S$, then we can replace S by S_p or S_c depending on whether the end of the marked edge is a leaf or the center of the star.

Theorem 4.5 (Bouchet [31])**.** *Let D be a split decomposition of a connected graph with bags of types P, K, or S. Then D is a canonical split decomposition if and only if it has no marked edge of type KK or S_pS_c.*

We now relate two vertices in different bags of in a split decomposition D. A vertex v of D *represents* an unmarked vertex x (or is a *representative* of x) if either $v = x$ or there is a path of even length from v to x in D starting with a marked edge such that marked edges and unmarked edges appear alternately in the path. Two unmarked vertices x and y are *linked* in D if there is a path from x to y in D such that marked edges and unmarked edges appear alternatively in the path.

The following lemma characterizes when two unmarked vertices of D are adjacent in the original graph G.

Lemma 4.6. *Let D be a split decomposition of a connected graph G. Let v' and w' be two vertices in a same bag of D, and let v and w be two unmarked vertices of D represented by v' and w', respectively. The following are equivalent.*

1. *v and w are linked in D.*
2. *$vw \in E(G)$.*
3. *$v'w' \in E(D)$.*

Proof. It is not hard to show that v' and w' are adjacent in D if and only if there is an alternating path from v to w in D. Now the proof follows from this and the definition of representativity. □

We sometimes remove vertices from a given split decomposition and obtain several components. The following notations are useful to call marked vertices, and especially, we will use them when characterizing the linear rank-width of distance-hereditary graphs in Chapter 5. For a bag B of a split decomposition D and a component T of $D \backslash V(B)$,

1. let $\zeta_b(D, B, T)$, $\zeta_t(D, B, T)$ be the marked vertices of D such that $\zeta_b(D, B, T) \in V(B)$, $\zeta_t(D, B, T) \in V(T)$ and $\zeta_b(D, B, T)\zeta_b(D, B, T)$ is the marked edge connecting B and T in D.

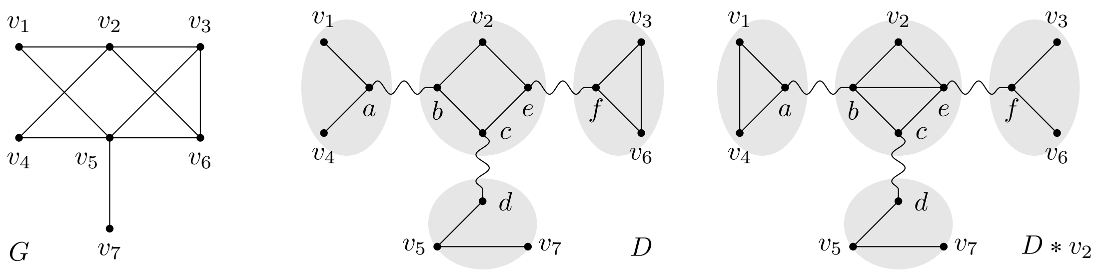


Figure 4.3: From a split decomposition D of a graph G, we obtain a split decomposition $D * v_2$ of $G * v_2$. Note that v_2 is represented by v_2, a, f in D.

The subscript 'b' means the marked vertex that is in the removed bag, and the subscript 't' means the marked vertex that is in a component after removing the bag.

A subgraph of a split decomposition is called a *subdecomposition*. We consider T as a subdecomposition of D. Note that $\zeta_t(D, B, T)$ is not incident with any marked edge in T. When we take a subdecomposition T from D, we regard $\zeta_t(D, B, T)$ as an unmarked vertex of T. It can be viewed as choosing one unmarked vertex which is represented by $\zeta_t(D, B, T)$ in D. If the split decomposition D is clear from the context, then we remove D from the notation $\zeta_b(D, B, T)$ or $\zeta_t(D, B, T)$.

The following is an easy lemma.

Lemma 4.7. *Let D be a split decomposition of a connected graph G. If B is a bag of D, then G has an induced subgraph isomorphic to B.*

Proof. For each vertex in the bag B, we can choose one unmarked vertex represented by it. Then the set of unmarked vertices induces a subgraph isomorphic to B in G. $\square$

4.3 Local complementations in split decompositions

We now discuss how split decompositions change when we apply a local complementation at a vertex in the original graph.

Let D be a split decomposition of a connected graph G. A *local complementation* at an unmarked vertex v in a split decomposition D, denoted by $D*v$, is the operation that replaces each bag B containing a representative w of v with $B * w$. See Figure 4.3 for an example of applying a local complementation in a split decomposition. We observe that if D is a split decomposition of G, then $D * v$ is a split decomposition of $G * v$, and therefore $M(D) = M(D * v)$ [31]. Two split decompositions D and D' are *locally equivalent* if D can be obtained from D' by applying a sequence of local complementations. Moreover, we obtain a relation for canonical split decompositions.

Lemma 4.8 (Bouchet [31])**.** *Let D be the canonical split decomposition of a connected graph G. If v is an unmarked vertex of D, then $D * v$ is the canonical split decomposition of $G * v$.*

From Lemma 4.8 if D is a split decomposition and $D' = D * x$, then $T_{D'}$ and T_D are isomorphic because $M(D) = M(D')$. For every node v of T_D associated with the bag B in D, its corresponding node v' in $T_{D'}$ is associated in D' either with the bag B if x has no representative in B or with the bag $B * w$ where w is the representative of x in B. Hence for each $X \subseteq V(D)$, $D[X]$ induces a bag in D if and only if $D'[X]$ induces a bag in D'. To avoid tracking bags as graphs when we apply local complementations, for each node v of a split decomposition tree T and each canonical split decomposition D with T as a

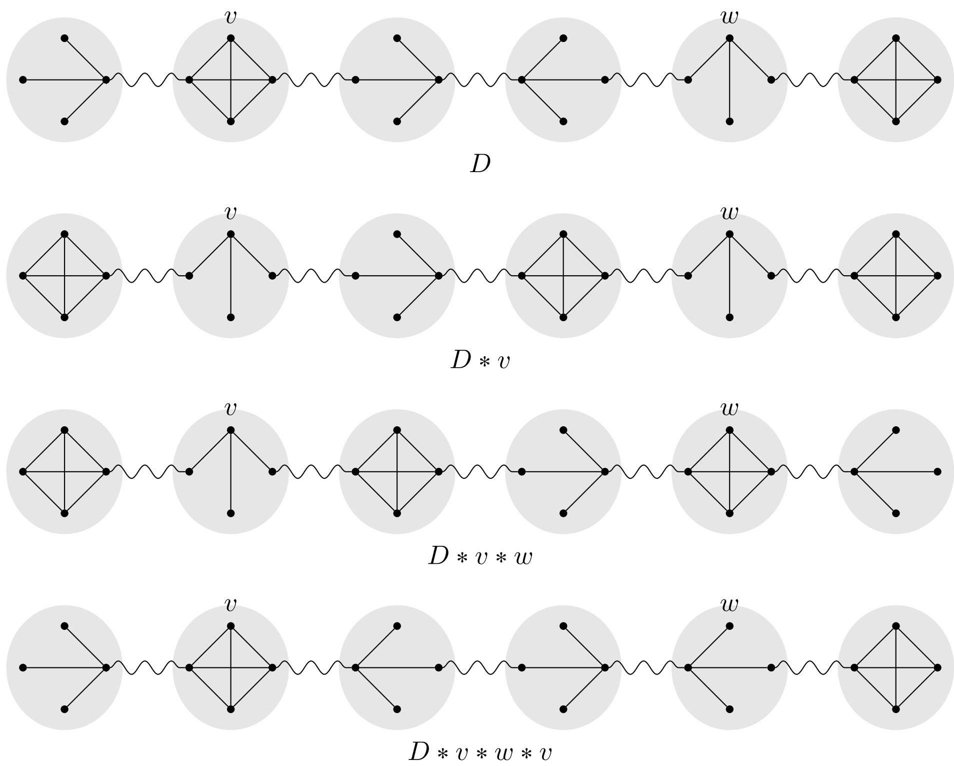


Figure 4.4: The split decomposition $D * v * w * v$, which is the same as $D \wedge vw$.

split decomposition tree we write $\mathsf{bag}_{(D,T)}(v)$ to denote the bag of D with which it is in correspondence. If the split decomposition tree T is clear, then we remove it from the notation.

Let v and w be linked unmarked vertices in a split decomposition D, and let B_v and B_w be the bags containing v and w, respectively. Note that if B is a bag of type S in the path from B_v to B_w in T_D, then the center of B is a representative of either v or w. *Pivoting* vw *of* D, denoted by $D \wedge vw$, is the split decomposition obtained as follows: for each bag B on the path from B_v to B_w in T_D, if $v', w' \in V(B)$ represent v and w in D, respectively, then we replace B with $B \wedge v'w'$. (Note that by Lemma 4.6, we have $v'w' \in E(B)$, hence $B \wedge v'w'$ is well-defined.)

Lemma 4.9. *Let D be a split decomposition of a connected graph G. If $xy \in E(G)$, then $D \wedge xy = D * x * y * x = D * y * x * y$.*

Proof. Since $xy \in E(G)$, by Lemma 4.6, x and y are linked in D. It is easy to see that by the operation $D * x * y * x$, only the bags in the path from x to y are modified, and they are modified according to the definition of $D \wedge xy$. See Figure 4.4. □

As a corollary of Lemmas 4.8 and 4.9, we get the following.

Corollary 4.10. *Let D be the canonical split decomposition of a graph G. If $xy \in E(G)$, then $D \wedge xy$ is the canonical split decomposition of $G \wedge xy$.*

We introduce some useful lemmas on local complementations and their canonical split decomposition versions.

Lemma 4.11. *Let G be a graph and $x, y \in V(G)$ such that $xy \notin E(G)$. Then $G * x * y = G * y * x$.*

Proof. We define vertex sets $W_1 := N_G(x) \cap N_G(y)$, $W_2 := N_G(x) \backslash N_G(y)$, and $W_3 := N_G(y) \backslash N_G(x)$. The graph $G * x * y$ is obtained from G by flipping the adjacency between two vertices in W_2 and in W_3, respectively, and flipping the adjacency between W_1 and $W_2 \cup W_3$. From the symmetry, the resulting graph is the same as $G * y * x$. $\square$

Lemma 4.12. *Let G be a graph and $x, y, z \in V(G)$ such that $xy, xz \notin E(G)$ and $yz \in E(G)$. Then $G * x \wedge yz = G \wedge yz * x$.*

Proof. By the definition of pivoting, $G * x \wedge yz = G * x * y * z * y$. Note that $xy \notin E(G)$, $xz \notin E(G * y)$, and $xy \notin E(G * y * z)$. Therefore, by Lemma 4.11, $G * x * y * z * y = (G * y) * x * z * y = (G * y * z) * x * y = (G * y * z * y) * x = G \wedge yz * x$. $\square$

The followings can be easily verified using the proofs of Lemmas 1.9, 4.11, and 4.12.

Lemma 4.13. *Let D be the canonical split decomposition of a connected graph. The following are satisfied.*

1. *If x, y are unmarked vertices of D that are not linked, then $D * x * y = D * y * x$.*
2. *If x, y, z are unmarked vertices of D such that x is linked to neither y nor z, and y and z are linked, then $D * x \wedge yz = D \wedge yz * x$.*
3. *If x, y, z are unmarked vertices of D such that y is linked to both x and z, then $D \wedge xy \wedge xz = D \wedge yz$.*

Chapter 5. Distance-hereditary graphs

We discuss structures of distance-hereditary graphs. A graph G is *distance-hereditary* if for every connected induced subgraph H of G and $v, w \in V(H)$, the distance between v and w in H is equal to the distance in G. Distance-hereditary graphs were first introduced by Howorka [113]. Bandelt and Mulder [9] provided several characterizations of distance-hereditary graphs. Oum [150] showed that distance-hereditary graphs are exactly the graphs of rank-width at most 1, and thus, these characterizations of distance-hereditary graphs are useful when discussing graphs of rank-width at most 1.

We characterize the linear rank-width of distance-hereditary graphs in Section 5.3. This characterization will be used to devise a polynomial-time algorithm to compute the linear rank-width of distance-hereditary graphs in Chapter 9. We present an extension of Bouchet's theorem about locally equivalent trees in Section 5.5, which will be used to obtain a lower bound on the number of graphs in a vertex-minor obstruction set for the class of graphs of linear rank-width at most k. In the last section, we survey characterizations of thread graphs, which will be used to obtain an FPT algorithm for the Thread Vertex Deletion problem in Chapter 10.

5.1 Characterizations of distance-hereditary graphs

We summarize useful characterizations of distance-hereditary graphs. The induced subgraph obstructions for distance-hereditary graphs are depicted in Figure 5.1.

Theorem 5.1 (Howorka [113]; Bandelt and Mulder [9]; Bouchet [31]; Oum [150]; Kwon and Oum [134])**.** *Let G be a graph. The following are equivalent.*

1. *G is distance-hereditary.*
2. *G has rank-width at most 1.*
3. *G is {house, gem, domino, hole}-free.*
4. *G has no vertex-minor isomorphic to C_5.*
5. *G has no pivot-minor isomorphic to C_5 and C_6.*
6. *Every bag of the canonical split decomposition of each connected component of G is either a complete bag or a star bag.*

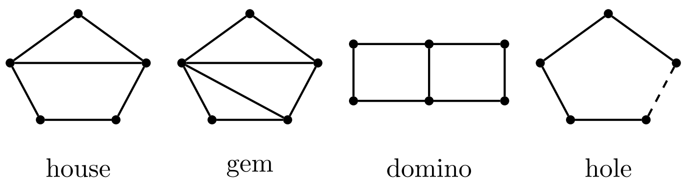


Figure 5.1: The induced subgraph obstructions for distance-hereditary graphs.

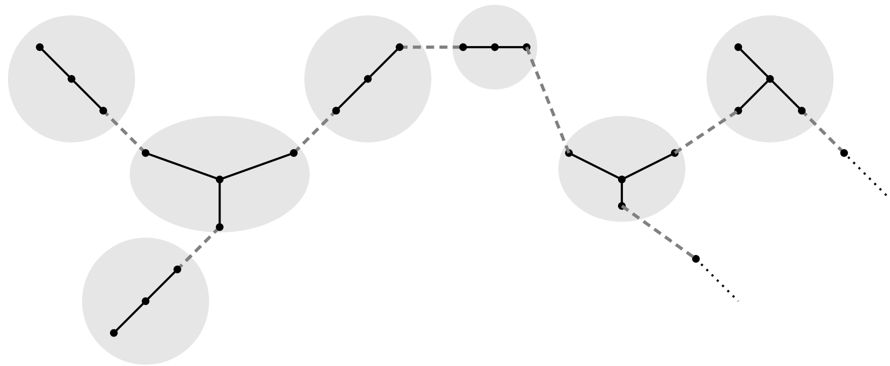


Figure 5.2: A canonical split decomposition of a tree.

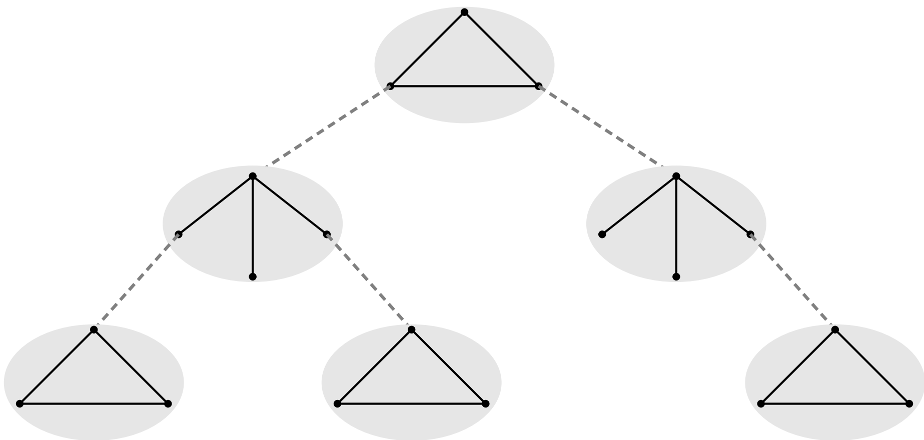


Figure 5.3: A canonical split decomposition of a cograph.

7. *G can be constructed from a vertex by a sequence of adding a false twin, a true twin, or a pendant vertex.*

8. *G is a vertex-minor of a tree.*

The structure of canonical split decompositions of distance-hereditary graphs is widely used in this thesis. To help understanding the structure of distance-hereditary graphs, we provide some examples here. Bouchet [31] provided a characterization of trees in terms of canonical split decompositions.

Theorem 5.2 (Bouchet [31])**.** *A connected graph is a tree if and only if each bag of its canonical split decomposition is a star bag whose center is an unmarked vertex.*

See Figure 5.2 for an example of the canonical split decomposition of a tree. Using this characterization, Bouchet [31] proved that two locally equivalent trees must be isomorphic. We will extend this result in Section 5.5.

Cographs [57, 37] are one of the interesting subclasses of distance-hereditary graphs. Cographs are exactly P_4-free graphs. See Figure 5.3 for an example of the canonical split decomposition of a cograph. Roughly speaking, the star bags of a canonical split decomposition of a connected cograph are directed towards one bag, if we regard the center vertex as the direction of each star bag [57, 104]. We remark that for every vertex v in a distance-hereditary graph, the neighborhood of v induces a cograph because a distance-hereditary graph is {gem}-free.

A graph is a *block graph* [126, 114, 9] if all of its blocks are complete graphs. A *diamond graph* is the graph obtained from K_4 by removing one edge. Bandelt and Mulder [9] showed that a graph is a block graph if and only if it has no induced subgraph isomorphic to a diamond graph or C_k for $k \geqslant 4$. We can regard them as $\{\text{diamond}, C_4\}$-free distance-hereditary graphs. We characterize block graphs from their canonical split decompositions.

Lemma 5.3. *Let D be the canonical split decomposition of a connected graph G. Then G is a block graph if and only if every bag of D is either a star or a complete bag, and the center of each star bag of D is unmarked.*

Proof. We may assume that G is distance-hereditary because otherwise D has a bag that is neither star nor complete, and G is not a block graph.

We first suppose that D has a star bag B having a marked center w. There exists a marked edge ww' joining B with a bag B'. Since D is a canonical split decomposition, B' is either complete or star with the center w'. If B' is complete, then by recomposing ww' we obtain a bag which has an induced subgraph isomorphic to a diamond graph. Thus G has an induced subgraph isomorphic to a diamond graph by Lemma 4.7. Since a diamond graph is not a block graph, we deduce that G is not a block graph. If B' is a star bag with the center w', then by recomposing ww', we obtain a bag which has an induced subgraph isomorphic to C_4. By Lemma 4.7, G should have an induced subgraph isomorphic to C_4, and therefore G is not a block graph.

To prove the converse, we claim a stronger statement: if D is a *split* decomposition of a connected graph G whose bags are star or complete and no center of a star bag in D is marked, then G is a block graph. We proceed by induction on $|V(D)|$. We may assume that D has a star bag B because otherwise G is a complete graph. Let v be the center of B. If B has another unmarked vertex w, then let G' be a graph obtained by recomposing all marked edges in $D \backslash w$. Here G is obtained from G' by adding a pendant vertex w to v. By the induction hypothesis, G' is a block graph and so is G. We may now assume that every vertex in B other than v is marked. Let $B = \{v, v_1, v_2, \ldots, v_n\}$ and let $v_1w_1, v_2w_2, \ldots, v_nw_n$ be the marked edges incident with B. Let D_i be the component of $D \backslash V(B)$ containing w_i. By the induction hypothesis, the graph G_i obtained by recomposing all marked edges in D_i is a block graph. The graph G is obtained from $G_1, G_2, \ldots, G_n$ by identifying $w_1, w_2, \ldots, w_n$ with a new vertex v. Since each block of G is a block of G_i for some i, we deduce that G is a block graph. □

Parity graphs [41, 55] are graphs G where for every $u, v \in V(G)$ and two induced paths from u to v in G, the parity of the length of paths are the same. Cicerone and Stefano [55] show that a graph G is a parity graph if and only if every bag of its canonical split decomposition is bipartite or complete. *Ptolemaic graphs* [126, 115, 9] are exactly $\{\text{gem}, C_4, \text{hole}\}$-free graphs, and we can regard ptolemaic graphs as C_4-free distance-hereditary graphs. We provide a hierarchy of related graph classes in Figure 5.4.

The incremental characterization of distance-hereditary graphs

Lastly, we introduce the incremental characterization of distance-hereditary graphs, given by Gioan and Paul [104]. For a vertex subset S of a graph G and $x \notin V(G)$, we denote by $G + (x, S)$ the graph obtained from G by adding a vertex x and adding edges between x and the vertices in S. Given a canonical split decomposition of a connected distance-hereditary graph G, Gioan and Paul [104] characterize when $G + (x, S)$ is again distance-hereditary, and they describe how to put the new vertex to make a canonical split decomposition of $G + (x, S)$. Since the definition of canonical split decompositions used by Gioan

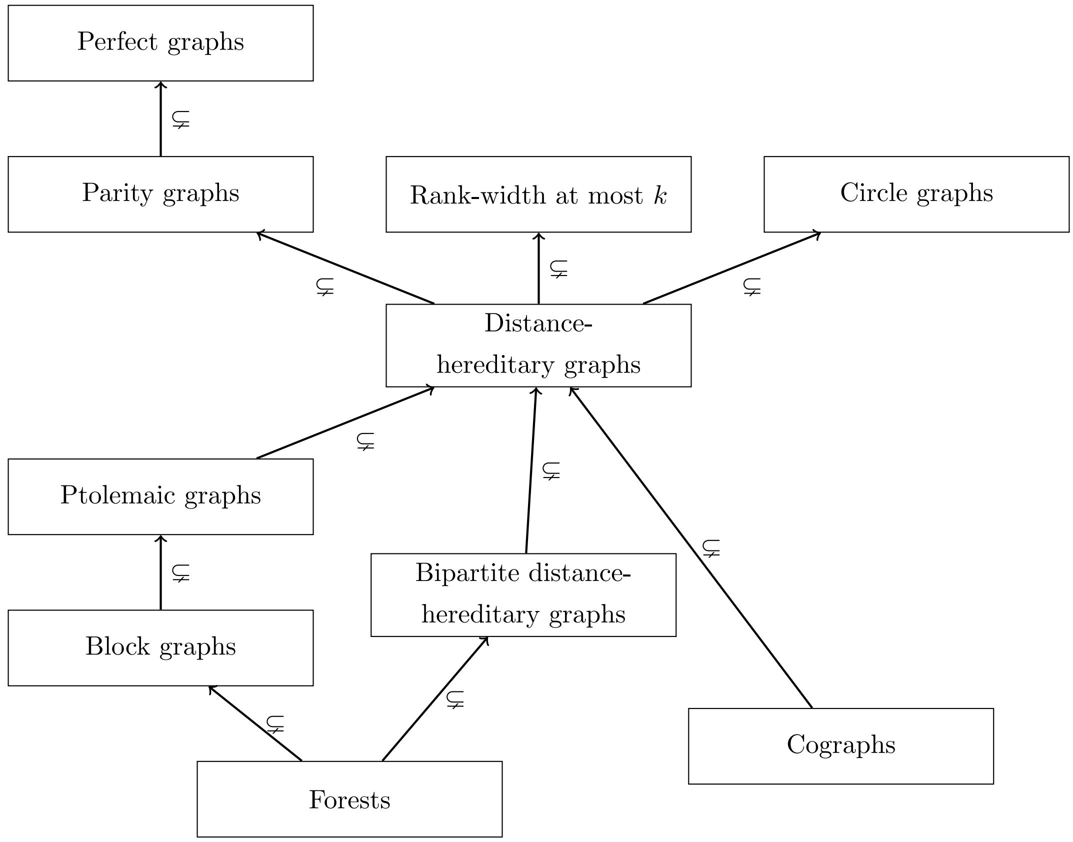


Figure 5.4: A hierarchy of graph classes.

and Paul is slightly different, we rephrase their result into our notation. We will use this incremental characterization in Section 6.4 to generate vertex-minor minimal graphs for the class of graphs of linear rank-width at most k, and in Section 10.1 to analyze necklace graphs which extend thread graphs.

Let D be the canonical split decomposition of a connected distance-hereditary graph G. For $S \subseteq V(G)$ and a vertex v in D with a bag B containing v, v is *accessible* with respect to S if either $v \in S$, or the component of $D \setminus V(B)$ having a neighbor of v contains a vertex in S. For $S \subseteq V(G)$ and a bag B of D,

1. B is *fully accessible* with respect to S if all vertices in B are accessible with respect to S,
2. B is *singly accessible* with respect to S if B is a star bag of D, and exactly two vertices of B including the center of B are accessible with respect to S, and
3. B is *partially accessible* with respect to S if otherwise.

A star bag B of D is *oriented towards* a bag B' (or a marked edge e) in D if the path from the center of B to a vertex of B' (or to end vertices of e) contain the marked edge incident with the center of B. For $S \subseteq V(G)$, we define $D(S)$ as the minimal connected subdecomposition of D such that

1. $D(S)$ is induced by a union of bags of D, and
2. $D(S)$ contains all vertices of S.

Theorem 5.4 (Gioan and Paul [104]). *Let D be the canonical split decomposition of a connected distance-hereditary graph G, and let $S \subseteq V(G)$ with $|S| \geqslant 2$. Then $G + (x, S)$ is distance-hereditary if and only if*

1. *at most one bag of $D(S)$ is partially accessible in D,*
2. *every complete bag of $D(S)$ is either fully accessible or partially accessible in D, and*
3. (a) *if there is a partially accessible bag B in $D(S)$, then each star bag $B' \neq B$ in $D(S)$ is oriented towards B if and only if it is fully accessible,*

   (b) *otherwise, there exists a marked edge e of $D(S)$ such that each star bag B in $D(S)$ is oriented towards e if and only if it is fully accessible.*

Now we describe how we update canonical split decompositions when adding a vertex. Let $S \subseteq V(G)$ with $|S| \geqslant 2$ and $x \notin V(G)$ such that x and S satisfy the condition of Theorem 5.4 and thus, $G + (x, S)$ is distance-hereditary. Let $D(S)$ be the minimal connected subdecomposition of D induced by a union of bags of D that contains all vertices of S. From Condition 3 in Theorem 5.4, the subdecomposition $D(S)$ contains a bag or a marked edge which has a special role. To identify this bag or edge, we define an orientation among bags in $D(S)$.

We define an *orientation* g on $D(S)$ which maps a bag of $D(S)$ to itself or its adjacent bag in D, such that

1. $g(B) = B$ implies $g(B') = B$ for every adjacent bag B' of B, and
2. $g(B) = B'$ implies $g(B'') = B$ for every adjacent bag $B'' \neq B'$ of B.

It is not hard to check that one of the following is satisfied:

1. There exists a unique bag B with $g(B) = B$. We call it the *root bag* with respect to g.
2. There exists a unique marked edge connecting two bags B and B' with $g(B) = B'$ and $g(B') = B$. We call it the *root edge* with respect to g.

We define an orientation g on the bags of $D(S)$ as follows:

1. Let B be a star bag of $D(S)$. If B is partially accessible in D, then $g(B) := B$. If B is singly accessible in D, then let $g(B)$ be the unique adjacent bag B' of B such that a leaf of B is adjacent to B'. If B is fully accessible in D, then let $g(B)$ be the adjacent bag B' of B such that the center of B is adjacent to B'.
2. Let B be a complete bag of $D(S)$. If B is partially accessible in D, then $g(B) := B$. Otherwise, B is fully accessible and its adjacent bags are star bags. If $g(B') = B$ for every adjacent bag B' of B then $g(B) := B$. If $g(B') = B$ for every adjacent bag B' of B but one, say B'', then $g(B) := B''$.

We first preprocess a partially accessible bag with respect to S if exists, and analyze three cases. Note that we take an orientation after preprocessing.

Preprocessing. There is a partially accessible bag B with the set A_B of all accessible vertices in B such that $|A_B| \geqslant 2$, $|V(B) \backslash A_B| \geqslant 2$.

We replace B with a simple decomposition obtained by decomposing B along the split $(A_B, V(B) \backslash A_B)$ of B. Now the new bag containing A_B consists of exactly A_B with one more marked vertex.

Case 1. The root bag B of $D(S)$ with respect to g is partially accessible.

Let B' be its adjacent bag in D that does not belong to $D(S)$. Then we put a new star bag of size 3 on the marked edge linking B and B', whose center is adjacent to the root bag. The unmarked vertex in the new bag is x.

Case 2. The root bag B of $D(S)$ with respect to g is not partially accessible.

By the definition of the orientation g, the bag B is a complete bag, and we add a new unmarked vertex x in B so that the new bag becomes a complete bag.

Case 3. The root edge of $D(S)$ with respect to g is a marked edge linking B and B'.

In this case, we put a new complete bag of size 3 on the marked edge linking B and B'. The unmarked vertex in the new bag is x.

5.2 Limbs in canonical split decompositions

We define the notion of *limb*, which is a key ingredient in the characterization of the linear rank-width of canonical split decompositions of distance-hereditary graphs.

Let D be the canonical split decomposition of a connected distance-hereditary graph G. We recall from Theorem 4.5 that marked edges of types KK or S_pS_c do not occur in D. For an unmarked vertex y in D and a bag B of D containing a marked vertex that represents y, let T be the component of $D \backslash V(B)$ containing y, and let w be the marked vertex of B adjacent to a vertex of T, and let v be the neighbor of w in T. We define the *limb* $\mathcal{L} := \mathcal{L}_D[B, y]$ with respect to B and y as follows:

1. if B is of type K, then $\mathcal{L} := T * v \backslash v$,
2. if B is of type S and w is a leaf, then $\mathcal{L} := T \backslash v$,
3. if B is of type S and w is the center, then $\mathcal{L} := T \wedge vy \backslash v$.

Since v becomes an unmarked vertex in T, the limb is well-defined and it is a split decomposition. While T is a canonical split decomposition, $\mathcal{L}$ may not be a canonical split decomposition at all, because deleting v may create a bag of size 2. We analyze the cases when such a bag appears, and describe how to transform it into a canonical split decomposition.

Suppose that a bag B' of size 2 appears in $\mathcal{L}$ by deleting v. If B' has no adjacent bags in $\mathcal{L}$, then B' itself is a canonical split decomposition. Otherwise we have two cases.

1. (B' has one adjacent bag B_1)
   If $v_1 \in V(B_1)$ is the marked vertex adjacent to a vertex of B' and r is the unmarked vertex of B' in $\mathcal{L}$, then we can transform the limb into a canonical split decomposition by removing the bag B' and replacing v_1 with r.
2. (B' has two adjacent bags B_1, B_2)
   If $v_1 \in V(B_1)$ and $v_2 \in V(B_2)$ are the two marked vertices that are adjacent to the two marked vertices of B' respectively, then we can first transform the limb into another split decomposition by removing B' and adding a marked edge v_1v_2. If the new marked edge v_1v_2 is of type KK or S_pS_c, then by recomposing along v_1v_2, we finally transform the limb into a canonical split decomposition.

Let $\widetilde{\mathcal{L}} := \widetilde{\mathcal{L}}_D[B, y]$ be the canonical split decomposition obtained from $\mathcal{L}_D[B, y]$ and we call it the *canonical limb*. Let $\widehat{\mathcal{L}} := \widehat{\mathcal{L}}_D[B, y]$ be the graph obtained from $\mathcal{L}_D[B, y]$ by recomposing all marked edges. See Figure 5.5 for an example of a canonical limb. If the original canonical split decomposition D is clear from the context, then we remove the subscript D in the notations $\mathcal{L}_D[B, y]$, $\widetilde{\mathcal{L}}_D[B, y]$ and $\widehat{\mathcal{L}}_D[B, y]$.

Lemma 5.5. *Let B be a bag of D. If an unmarked vertex y of D is represented by a marked vertex of B, then $\mathcal{L}[B, y]$ is connected.*

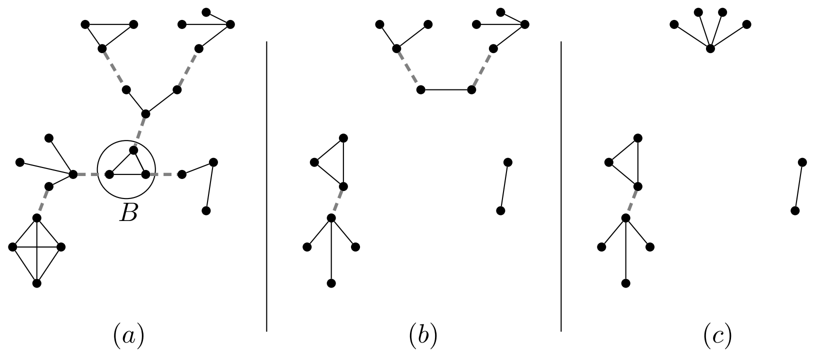


Figure 5.5: In (a), we have a canonical split decomposition D of a distance-hereditary graph and a bag B of D. The dashed edges are marked edges of D. In (b), we have limbs $\mathcal{L}$ associated with the components of $D\backslash V(B)$. The canonical limbs $\widetilde{\mathcal{L}}$ associated with limbs $\mathcal{L}$ are shown in (c).

Proof. Let T be the component of $D\backslash V(B)$ containing y, and $v := \zeta_t(B,T)$, and B' be the bag of D containing v. Since local complementations maintain connectedness, it suffices to verify that $V(B')\backslash v$ induces a connected subgraph in $\mathcal{L}[B,y]$. This is not hard to see for each of the three cases. □

Lemma 5.6. *Let B be a bag of D. If two unmarked vertices x and y are represented by a marked vertex $w \in V(B)$, then $\mathcal{L}[B,x]$ is locally equivalent to $\mathcal{L}[B,y]$.*

Proof. Since x and y are represented by the same vertex w of B in D, they are contained in the same component of $D\backslash V(B)$, say T. Let $v := \zeta_t(B,T)$.

If B is a complete bag or a star bag having w as a leaf, then by the definition of limbs, $\mathcal{L}[B,x] = \mathcal{L}[B,y]$. So, we may assume that w is the center of the star bag B. Since v is linked to both x and y in T, by Lemma 4.13, $T \wedge vx \wedge xy = T \wedge vy$. So, we obtain that $(T \wedge vx\backslash v) \wedge xy = T \wedge vx \wedge xy\backslash v = T \wedge vy\backslash v$. Therefore $\mathcal{L}[B,x]$ is locally equivalent to $\mathcal{L}[B,y]$. □

For a bag B of D and a component T of $D\backslash V(B)$, we define $f_D(B,T)$ as the linear rank-width of $\widehat{\mathcal{L}}_D[B,y]$ for some unmarked vertex $y \in V(T)$. In fact, by Lemma 5.6, $f_D(B,T)$ does not depend on the choice of y. As in the notation $\mathcal{L}_D[B,x]$, if the canonical split decomposition D is clear from the context, then we remove the subscript D in the notation $f_D(B,T)$.

Proposition 5.7. *Let B be a bag of D and y be an unmarked vertex represented in D by $w \in V(B)$. Let $x \in V(\widehat{D})$. If an unmarked vertex y' is represented by w in $D * x$, then $\widehat{\mathcal{L}}_D[B,y]$ is locally equivalent to $\widehat{\mathcal{L}}_{D*x}[(D*x)[V(B)],y']$. Therefore, $f_D(B,T) = f_{D*x}((D*x)[V(B)],T_x)$ where T and T_x are the components of $D\backslash V(B)$ and $(D*x)\backslash V(B)$ containing y, respectively.*

Before proving it, let us recall the following by Geelen and Oum.

Lemma 5.8 (Geelen and Oum [103])**.** *Let G be a graph and x,y be two distinct vertices in G. Let $xw \in E(G*y)$ and $xz \in E(G)$.*

1. *If $xy \notin E(G)$, then $(G*y)\backslash x$, $(G*y*x)\backslash x$, and $(G*y) \wedge xw\backslash x$ are locally equivalent to $G\backslash x$, $G*x\backslash x$, and $G \wedge xz\backslash x$, respectively.*

2. *If $xy \in E(G)$, then $(G*y)\backslash x$, $(G*y*x)\backslash x$, and $(G*y) \wedge xw\backslash x$ are locally equivalent to $G\backslash x$, $G \wedge xz\backslash x$, and $(G*x)\backslash x$, respectively.*

Proof of Proposition 5.7. By Lemma 5.6, it is enough to show the first statement because a local complementation preserves the linear rank-width of a graph. Let $v := \zeta_t(D, B, T)$ and $B' := (D * x)[V(B)]$. Let T and T_x be the components of $D \backslash V(B)$ and $(D * x) \backslash V(B)$ containing y, respectively. Note that $V(T) = V(T_x)$.

We claim that $\widehat{\mathcal{L}}_D[B, y]$ is locally equivalent to $\widehat{\mathcal{L}}_{D*x}[B', y']$ for some unmarked vertex y' represented by w in $D * x$. We divide into cases depending on the type of the bag B and whether $x \in V(T)$.

Case 1. *$x \in V(T)$ and x is not linked to v in T.*

Since x is not linked to v in T, $B' = B$ and v is still linked to y in $T * x$. In this case, let $y' := y$.

Case 1.1. B is of type S and w is a leaf of B.

Since v is not linked to x in T, by Lemma 5.8, $\widehat{T} \backslash v$ is locally equivalent to $\widehat{T} * x \backslash v$.

Case 1.2. B is of type S and w is the center of B.

Since v is linked to y in $T * x$, by Lemma 5.8, $\widehat{T} \wedge vy \backslash v$ is locally equivalent to $\widehat{T} * x \wedge vy \backslash v$.

Case 1.3. B is of type K.

Since v is not linked to x in T, by Lemma 5.8, $\widehat{T} * v \backslash v$ is locally equivalent to $\widehat{T} * x * v \backslash v$.

Case 2. *$x \in V(T)$ and x is linked to v in T.*

Note that x is linked to v in $T * x$. Let $y' := x$ for this case.

Case 2.1. B is of type S and w is a leaf of B.

Applying local complementation at x does not change the type of the bag B. Since v is linked to x in T, by Lemma 5.8, $\widehat{T} \backslash v$ is locally equivalent to $\widehat{T} * x \backslash v$.

Case 2.2. B is of type S and w is the center of B.

Applying local complementation at x changes the bag B into a bag of type K, and the component T into $T * x$. Since v is linked to x in T, by Lemma 5.8, $\widehat{T} \wedge vy \backslash v$ is locally equivalent to $\widehat{T} * x * v \backslash v$.

Case 2.3. B is of type K.

Applying local complementation at x changes the bag B into a bag of type S such that the center of B is w. Since v is linked to x in T, by Lemma 5.8, $\widehat{T} * v \backslash v$ is locally equivalent to $\widehat{T} * x \wedge vx \backslash v$.

Case 3. *$x \notin V(T)$.*

If x has no representative in the bag B, then applying local complementation at x does not change the bag B and the component T. Therefore, we may assume that x is represented by some vertex in B, necessarily adjacent to w. In this case, v is still a representative of y in $D * x$, and we let $y' := y$.

Case 3.1. B is of type S and w is a leaf of B.

Applying local complementation at x changes B into a bag of type K, and T into $T * v$. We have $\mathcal{L}_{D*x}[B', y'] = (T * v) * v \backslash v = T \backslash v = \mathcal{L}_D[B, y]$.

Case 3.2. B is of type S and w is the center of B.

Since w is the center of B, x is represented by a leaf of the bag B. Applying local complementation at x does not change the bag B, but it changes T into $T * v$. We have $\mathcal{L}_{D*x}[B', y'] = (T * v) \wedge vy \backslash v$. Since $((T * v) \wedge vy \backslash v) * y = T * y * v * y \backslash v = T \wedge vy \backslash v$, $\mathcal{L}_D[B, y]$ and $\mathcal{L}_{D*x}[B', y']$ are locally equivalent.

Case 3.3. B is of type K.

After applying local complementation at x in D, B becomes a star such that w is a leaf of B, and T becomes $T * v$. Therefore, we have $\mathcal{L}_{D*x}[B', y'] = T * v \backslash v = \mathcal{L}_D[B, y]$. □

The following lemma will be used widely to reduce cases in several proofs.

Lemma 5.9. *Let B_1 and B_2 be two distinct bags of D and for each $i \in \{1,2\}$, let T_i be the components of $D \backslash V(B_i)$ such that T_1 contains the bag B_2 and T_2 contains the bag B_1. Then there exists a canonical split decomposition D' locally equivalent to D such that for each $i \in \{1,2\}$, $D'[V(B_i)]$ is a star whose leaf is adjacent to a vertex in T_i.*

Proof. Let $v_i := \zeta_b(D, B_i, T_i)$ for $i = 1, 2$. It is easy to make B_1 into a star bag having v_1 as a leaf by applying local complementations. We can assume without loss of generality that v_1 is a leaf of B_1 in D. If v_2 is a leaf of B_2, then we are done. If B_2 is a complete bag, then choose an unmarked vertex w_2 of D that is represented by a vertex of B_2 other than v_2. Then applying a local complementation at w_2 makes B_2 into a star bag having v_2 as a leaf without changing B_1. Therefore, we may assume that v_2 is the center of the star bag B_2. If B_1 and B_2 are adjacent bags in D, then the marked edge connecting B_1 and B_2 is of type S_pS_c, contradicting to the assumption that D is a canonical split decomposition. Thus, B_1 and B_2 are not adjacent bags in D.

Let $T := D[V(T_1) \cap V(T_2)]$ and $w_2 := \zeta_t(D, B_2, T_2)$. By the definition of a canonical split decomposition, w_2 is not a leaf of a star bag in D. Therefore, there exists an unmarked vertex $y \in V(T)$ of D such that y is linked to w_2 in T. Choose an unmarked vertex y' of D represented by w_2 in D. Since y is linked to y' and the alternating path from y to y' in D pass through B_2 but not B_1, pivoting yy' in D makes B_2 into a star bag having v_2 as a leaf without changing B_1. Thus, each v_i is a leaf of B_i in $D \wedge yy'$, as required. □

We conclude the section with the following.

Proposition 5.10. *Let B_1 and B_2 be two distinct bags of D and T_1 be a component of $D \backslash V(B_1)$ such that T_1 does not contain the bag B_2, and T_2 be the component of $D \backslash V(B_2)$ such that T_2 contains the bag B_1. If y_1 and y_2 are two unmarked vertices in T_1 and T_2 that are represented by some vertices in B_1 and B_2, respectively, then $\widehat{\mathcal{L}}_D[B_1, y_1]$ is a vertex-minor of $\widehat{\mathcal{L}}_D[B_2, y_2]$. Therefore $f(B_1, T_1) \leqslant f(B_2, T_2)$.*

Proof. Let $u_2 := \zeta_t(B_2, T_2)$ and $v_2 := \zeta_b(B_2, T_2)$. By Lemma 5.9, there exists a canonical split decomposition D' locally equivalent to D such that B_2 is a star bag in D' and v_2 is a leaf of B_2. For each $i \in \{1,2\}$, let $T_i' := D'[V(T_i)]$, $B_i' := D'[V(B_i)]$ and let y_i' be an unmarked vertex in T_i' that is represented by some vertex in B_i'.

Since v_2 is a leaf of B_2' in D', we have $\mathcal{L}_{D'}[B_2', y_2'] = T_2' \backslash v_2$. Because T_1' is a subgraph of $T_2' \backslash v_2$, we can easily observe that $\widehat{\mathcal{L}}_{D'}[B_1', y_1']$ is a vertex-minor of $\widehat{\mathcal{L}}_{D'}[B_2', y_2']$. Since for each i, $\mathcal{L}_D[B_i, y_i]$ is locally equivalent to $\mathcal{L}_{D'}[B_i', y_i']$, $\widehat{\mathcal{L}}_D[B_1, y_1]$ is a vertex-minor of $\widehat{\mathcal{L}}_D[B_2, y_2]$. We conclude that $f(B_1, T_1) \leqslant f(B_2, T_2)$. □

5.3 Linear rank-width of distance-hereditary graphs

Now, we present the main result of this chapter characterizing the linear rank-width of distance-hereditary graphs using limbs.

Theorem 5.11. *Let k be a positive integer and let D be the canonical split decomposition of a connected distance-hereditary graph G. Then $\operatorname{lrw}(G) \leqslant k$ if and only if for each bag B of D, at most two components T of $D \backslash V(B)$ induce limbs L where $\widehat{L}$ has linear rank-width exactly k, and all other component T' of $D \backslash V(B)$ induce limbs L' where $\widehat{L'}$ has linear rank-width at most $k-1$.*

Let k be a positive integer and let D be the canonical split decomposition of a connected distance-hereditary graph G. We first prove the forward direction.

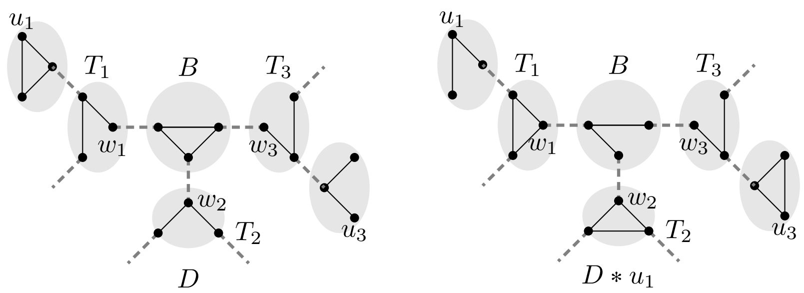


Figure 5.6: We realize a limb without removing the bag in Theorem 5.11. Since B is a complete bag, the limb $\mathcal{L}_D[B, u_2] = (D * u_1)[V(T_2)\backslash w_2]$.

Proof of the forward direction of Theorem 5.11. Suppose that there exists a bag B of D such that $D\backslash V(B)$ has at least three components T which induce limbs L where $\widehat{L}$ has linear rank-width k.

We claim that $\text{lrw}(G) \geqslant k + 1$. We may assume that $D\backslash V(B)$ has exactly three components T_1, T_2 and T_3, where each component T_i satisfies $f_D(B, T_i) = k$. For each $1 \leqslant i \leqslant 3$, let $w_i := \zeta_t(B, T_i)$, and N_i be the set of the unmarked vertices in T_i linked to w_i. Choose a vertex u_i in N_i and let $D_i := \mathcal{L}_D[B, u_i]$. We remark that N_i is exactly the set of the vertices in $V(\widehat{D_i})$ that have a neighbor in $V(\widehat{D})\backslash V(\widehat{D_i})$.

Since removing a vertex from a graph does not increase the linear rank-width, we assume that B consists of exactly three marked vertices that are adjacent to one of T_1, T_2 and T_3. Now, every unmarked vertex of D is contained in one of T_1, T_2 and T_3.

Note that by Proposition 5.7 and Lemmas 2.1 and 4.8, for any canonical split decomposition D' locally equivalent to D, we have $\text{lrw}(\widehat{D}) = \text{lrw}(\widehat{D'})$ and $f_D(B, T_i)$ does not change. So, we may assume that B is a complete bag of D.

We first claim that $D_2 = (D * u_1)[V(T_2)\backslash w_2]$. Since the bag B is complete, by definition, $D_2 = T_2 * w_2 \backslash w_2$. Since u_1 is linked to w_1 in T_1 and there is an alternating path from w_1 to w_2 in D, by concatenating alternating paths it is easy to see that $(D * u_1)[V(T_2)\backslash w_2] = T_2 * w_2 \backslash w_2 = D_2$, as claimed. See Figure 5.6.

Towards a contradiction, suppose that $\widehat{D}$ has a linear layout L of width k. Let a and b be the first vertex and the last vertex of L, respectively. Since B has no unmarked vertices, without loss of generality, we may assume that $a, b \in V(\widehat{D_1}) \cup V(\widehat{D_3})$. With this assumption, we will prove that $\widehat{D_2}$ has linear rank-width at most $k - 1$.

Let $v \in V(\widehat{D_2})$ and $S_v := \{x \in V(\widehat{D}) : x \leqslant_L v\}$ and $T_v := V(\widehat{D})\backslash S_v$. Since v is arbitrary, it is sufficient to show that $\rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) \leqslant k - 1$.

We divide into three cases. We first check two cases that are either ($N_1 \cap S_v \neq \varnothing$ and $N_3 \cap T_v \neq \varnothing$) or ($N_1 \cap T_v \neq \varnothing$ and $N_3 \cap S_v \neq \varnothing$). If both of them are not satisfied, then we can easily deduce that $N_1 \cup N_3 \subseteq S_v$ or $N_1 \cup N_3 \subseteq T_v$.

Case 1. *$N_1 \cap S_v \neq \varnothing$ and $N_3 \cap T_v \neq \varnothing$.*

Let $x_1 \in N_1 \cap S_v$ and $x_3 \in N_3 \cap T_v$. We claim that

$$\rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) = \rho_{\widehat{D}[V(\widehat{D_2}) \cup \{x_1, x_3\}]}((S_v \cap V(\widehat{D_2})) \cup \{x_1\}) - 1.$$

Because $\rho_{\widehat{D}[V(\widehat{D_2}) \cup \{x_1, x_3\}]}((S_v \cap V(\widehat{D_2})) \cup \{x_1\}) \leqslant \rho_{\widehat{D}}(S_v) \leqslant k$, the claim implies that $\rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) \leqslant k - 1$.

Note that x_1 and x_3 have the same neighbors in $\widehat{D}[V(\widehat{D_2}) \cup \{x_1, x_3\}]$ because they are in N_1 and N_3, respectively. Since x_1 is adjacent to x_3 in $\widehat{D}[V(\widehat{D_2}) \cup \{x_1, x_3\}]$, x_3 becomes a leaf in $\widehat{D}[V(\widehat{D_2}) \cup$

$\{x_1, x_3\}] * x_1$ having exactly one neighbor, x_1. Since $(D * x_1)[V(T_2) \backslash w_2] = D_2$, we have

$$\widehat{D}[V(\widehat{D_2}) \cup \{x_1, x_3\}] * x_1 \backslash x_1 \backslash x_3 = (\widehat{D} * x_1)[V(\widehat{D_2})] = \widehat{D_2}.$$

Therefore,

$$\begin{aligned}
&\rho_{\widehat{D}[V(\widehat{D_2}) \cup \{x_1,x_3\}]}((S_v \cap V(\widehat{D_2})) \cup \{x_1\}) \\
&= \rho_{\widehat{D}[V(\widehat{D_2}) \cup \{x_1,x_3\}] * x_1}((S_v \cap V(\widehat{D_2})) \cup \{x_1\}) \\
&= \operatorname{rank}\left(\begin{array}{cc} & \begin{array}{cc} x_3 & T_v \cap V(\widehat{D_2}) \end{array} \\ \begin{array}{c} x_1 \\ S_v \cap V(\widehat{D_2}) \end{array} & \left(\begin{array}{c|c} 1 & * \\ \hline 0 & * \end{array}\right) \end{array} \right) \\
&= \operatorname{rank}\left(\begin{array}{cc} & \begin{array}{cc} x_3 & T_v \cap V(\widehat{D_2}) \end{array} \\ \begin{array}{c} x_1 \\ S_v \cap V(\widehat{D_2}) \end{array} & \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & * \end{array}\right) \end{array} \right) \\
&= \rho_{\widehat{D}[V(\widehat{D_2}) \cup \{x_1,x_3\}] * x_1 \backslash x_1 \backslash x_3}(S_v \cap V(\widehat{D_2})) + 1 \\
&= \rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) + 1,
\end{aligned}$$

as claimed.

Case 2. *$N_1 \cap T_v \neq \varnothing$ and $N_3 \cap S_v \neq \varnothing$.*

We can prove $\rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) \leqslant k-1$ in the same way as for **Case 1**.

Case 3. *$N_1 \cup N_3 \subseteq S_v$ or $N_1 \cup N_3 \subseteq T_v$.*

We can assume without loss of generality that $N_1 \cup N_3 \subseteq S_v$ because $N_1 \cup N_3 \subseteq T_v$ is similar. Since $a, b \in V(\widehat{D_1}) \cup V(\widehat{D_3})$ and the graph $\widehat{D}[V(\widehat{D_1}) \cup V(\widehat{D_3})]$ is connected, there exists a vertex $t \in T_v \cap (V(\widehat{D_1}) \cup V(\widehat{D_3}))$ such that t is adjacent to a vertex of $N_1 \cup N_3$. Let $x \in N_{\widehat{D}}(t) \cap (N_1 \cup N_3)$. Since t cannot have a neighbor in N_2, we have

$$\begin{aligned}
\rho_{\widehat{D}}(S_v) &\geqslant \operatorname{rank}\left(\begin{array}{cc} & \begin{array}{cc} t & T_v \cap V(\widehat{D_2}) \end{array} \\ \begin{array}{c} x \\ S_v \cap V(\widehat{D_2}) \end{array} & \left(\begin{array}{c|c} 1 & * \\ \hline 0 & * \end{array}\right) \end{array} \right) \\
&= \rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) + 1.
\end{aligned}$$

Therefore, we conclude $\rho_{\widehat{D_2}}(S_v \cap V(\widehat{D_2})) \leqslant k-1$.

Thus, $\widehat{D_2}$ has linear rank-width at most $k-1$, which yields a contradiction. □

To prove the converse direction, we use the following lemmas.

Lemma 5.12. *Let B be a bag of D of type S having two unmarked vertices x and y such that x is the center and y is a leaf of B. If $f(B,T) \leqslant k-1$ for every component T of $D \backslash V(B)$, then the graph $\widehat{D}$ has a linear layout of width at most k whose first and last vertices are x and y, respectively.*

Proof. Let $T_1, T_2, \ldots, T_\ell$ be the components of $D \backslash V(B)$ and for each $1 \leqslant i \leqslant \ell$, let $w_i := \zeta_t(B, T_i)$ and let y_i be a vertex in T_i represented by a vertex of B. Since each w_i is adjacent to a leaf of B, $T_i \backslash w_i$ is the limb of D with respect to B and y_i.

Suppose that $f(B,T) \leqslant k-1$ for every component T of $D\backslash V(B)$. We may assume without loss of generality that B has only two unmarked vertices x and y. For each $1 \leqslant i \leqslant \ell$, let L_i be a linear layout of $\widehat{T_i\backslash w_i}$ of width at most $k-1$. We claim that

$$L := (x) \oplus L_1 \oplus L_2 \oplus \cdots \oplus L_\ell \oplus (y)$$

is a linear layout of $\widehat{D}$ of width at most k. It is sufficient to prove that for every $w \in V(\widehat{D})\backslash\{x,y\}$, $\rho_{\widehat{D}}(\{v : v \leqslant_L w\}) \leqslant k$.

Let $w \in V(\widehat{D})\backslash\{x,y\}$, and let $S_w := \{v : v \leqslant_L w\}$ and $T_w := V(\widehat{D})\backslash S_w$. Then $w \in L_j$ for some $1 \leqslant j \leqslant \ell$ and

$$\begin{aligned}
\rho_{\widehat{D}}(S_w) &= \operatorname{rank}\left(\begin{array}{cc} & \begin{array}{ccc} y & T_w \cap V(\widehat{T_j}) & T_w\backslash\{y\}\backslash V(\widehat{T_j}) \end{array} \\ \begin{array}{c} x \\ S_w \cap V(\widehat{T_j}) \\ S_w\backslash\{x\}\backslash V(\widehat{T_j}) \end{array} & \begin{pmatrix} 1 & * & * \\ 0 & * & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}\right) \\
&= \operatorname{rank}\left(\begin{array}{cc} & \begin{array}{ccc} y & T_w \cap V(\widehat{T_j}) & T_w\backslash\{y\}\backslash V(\widehat{T_j}) \end{array} \\ \begin{array}{c} x \\ S_w \cap V(\widehat{T_j}) \\ S_w\backslash\{x\}\backslash V(\widehat{T_j}) \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}\right) \\
&= \rho_{\widehat{T_j\backslash w_j}}(S_w \cap V(\widehat{T_j})) + 1 \leqslant (k-1)+1 = k.
\end{aligned}$$

Therefore, L is a linear layout of $\widehat{D}$ of width k whose first and last vertices are x and y, respectively. $\square$

Proposition 5.13. *Let B be a bag of D with two unmarked vertices x and y. If $f(B,T) \leqslant k-1$ for every component T of $D\backslash V(B)$, then the graph $\widehat{D}$ has a linear layout of width at most k whose first and last vertice are x and y, respectively.*

Proof. Suppose that $f(B,T) \leqslant k-1$ for every component T of $D\backslash V(B)$. If B is a complete graph, then let $D' := D * x$, and if B is a star such that x is the center, then let $D' := D$, and if x is not the center then let $D' := D \wedge xy$ if y is the center, otherwise let $D' := D \wedge xz$ where z is an unmarked vertex represented by the center of B. It is clear that $D[V(B)]$ is a star with x the center. By Proposition 5.7, for each component T of $D\backslash V(B)$, $f_D(B,T) = f_{D'}(D'[V(B)], D'[V(T)])$. Since $\widehat{D'}$ is locally equivalent to $\widehat{D}$, by Lemma 5.12, we conclude that $\widehat{D}$ has a linear layout of width at most k whose first and last vertice are x and y, respectively. $\square$

Lemma 5.14. *If*

1. *for each bag B of D, there are at most two components T of $D\backslash V(B)$ satisfying $f(B,T) = k$, and*

2. *for every other component T' of $D\backslash V(B)$, $f(B,T') \leqslant k-1$, and*

3. *P is the set of nodes v in T_D such that exactly two components T of $D\backslash V(\mathsf{bag}_D(v))$ satisfy $f(\mathsf{bag}_D(v), T) = k$,*

then either $P = \varnothing$ or $T_D[P]$ is a path.

Proof. Suppose that $P \neq \varnothing$. If v_1 and v_2 are in P, then there exists a component T_1 of $D\backslash V(\mathsf{bag}_D(v_1))$ not containing $V(\mathsf{bag}_D(v_2))$ such that $f(\mathsf{bag}_D(v_1), T_1) = k$, and there exists a component T_2 of $D\backslash V(\mathsf{bag}_D(v_2))$

not containing $V(\mathsf{bag}_D(v_1))$ such that $f(\mathsf{bag}_D(v_2), T_2) = k$. By Proposition 5.10, for every node v on the path from v_1 to v_2 in T_D, v must be contained in P. Thus P forms a connected subtree in T_D.

Suppose now that P contains a node v having three neighbors v_1, v_2, and v_3 in P. Then, again by Proposition 5.10, D must have three components T of $D \backslash V(\mathsf{bag}_D(v))$ such that $f(\mathsf{bag}_D(v), T) = k$, which contradicts the assumption. Therefore, P forms a path in T_D. □

Lemma 5.15. *If*

1. *for each bag B of D, there are at most two components T of $D \backslash V(B)$ satisfying $f(B,T) = k$, and*
2. *$f(B,T') \leqslant k-1$ for all the other components T' of $D \backslash V(B)$,*

then T_D has a path P such that for each node v in P and each component T of $D \backslash V(\mathsf{bag}_D(v))$ not containing a bag $\mathsf{bag}_D(w)$ with $w \in V(P)$, $f(B,T) \leqslant k-1$.

Proof. Let P' be the set of nodes v in T_D such that exactly two components T of $D \backslash V(\mathsf{bag}_D(v))$ satisfy $f(\mathsf{bag}_D(v), T) = k$. By Lemma 5.14, either $P' = \varnothing$ or $T_D[P']$ is a path.

We first assume that $P' \neq \varnothing$. Let $T_D[P'] = v_1 v_2 \cdots v_n$, and let $B_i := \mathsf{bag}_D(v_i)$. By the definition, there exists a component T_1 of $D \backslash V(B_1)$ such that T_1 does not contain a bag of P' and $f(B_1, T_1) = k$. Let v_0 be the node of T_D such that $\mathsf{bag}_D(v_0)$ is the bag of T_1 that is the adjacent bag of B_1 in D. Similarly, there exists a component T_n of $D \backslash V(B_n)$ such that T_n does not contain a bag of P' and $f(B_n, T_n) = k$. Let v_{n+1} be the node of T_D such that $\mathsf{bag}_D(v_{n+1})$ is the bag of T_n that is the adjacent bag of B_n in D. Then $P = v_0 v_1 v_2 \cdots v_n v_{n+1}$ is the required path.

Now we assume that $P' = \varnothing$. We choose a node v_0 in T_D and let $B_0 := \mathsf{bag}_D(v_0)$. If D has no component T of $D \backslash V(B_0)$ such that $f(B_0, T) = k$, then $P := v_0$ satisfies the condition. If not, we take a maximal path $P := v_0 v_1 \cdots v_{n+1}$ in T_D such that (with $B_i := \mathsf{bag}_D(v_i)$)

- for each $0 \leqslant i \leqslant n$, $D \backslash V(B_i)$ has one component T_i such that $f(B_i, T_i) = k$, and B_{i+1} is the bag of T_i that is the adjacent bag of B_i in D.

By the maximality of P, P is a path in T_D such that for each bag B in P and a component T of $D \backslash V(B)$ not containing a bag of P, $f(B,T) \leqslant k-1$. □

We are now ready to prove the converse direction of the proof of Theorem 5.11.

Proof of the backward direction of Theorem 5.11. Suppose that for each bag B of D, at most two components T of $D \backslash V(B)$ induce limbs L where $\widehat{L}$ has linear rank-width exactly k, and all other component T' of $D \backslash V(B)$ induce limbs L' where $\widehat{L'}$ has linear rank-width at most $k-1$. We claim that $\mathrm{lrw}(G) \leqslant k$.

Let $P = v_0 v_1 \cdots v_n v_{n+1}$ be a path in T_D such that for each node v in P and a component T of $D \backslash V(\mathsf{bag}_D(v))$ not containing a bag $\mathsf{bag}_D(w)$ with $w \in V(P)$, $f(\mathsf{bag}_D(v), T) \leqslant k-1$ (such a path exists by Lemma 5.15). For each $0 \leqslant i \leqslant n+1$, let $B_i := \mathsf{bag}_D(v_i)$. If P consists of one vertex, then by Proposition 5.13, $\mathrm{lrw}(G) = \mathrm{lrw}(\widehat{D}) \leqslant k$. Thus, we may assume that $n \geqslant 0$.

By adding unmarked vertices on B_0 and B_{n+1} if necessary, we assume that B_0 and B_{n+1} have unmarked vertices a_0 and b_{n+1} in D, respectively.

For each $0 \leqslant i \leqslant n$, let b_i be a marked vertex of B_i and let a_{i+1} be a marked vertex B_{i+1} such that $b_i a_{i+1}$ is the marked edge connecting B_i and B_{i+1}.

Let D_0 be the component of $D \backslash V(B_1)$ containing the bag B_0. Let D_{n+1} be the component of $D \backslash V(B_n)$ containing the bag B_{n+1}. For each $1 \leqslant i \leqslant n$, let D_i be the component of $D \backslash (V(B_{i-1}) \cup V(B_{i+1}))$ containing the bag B_i. Notice that the vertices a_i and b_i are unmarked vertices in D_i.

Since every component T of $D_i \backslash V(B_i)$ satisfies that $f_{D_i}(B_i, T) \leqslant k-1$, by Proposition 5.13, $\widehat{D_i}$ has a linear layout L_i' of width k whose first and last vertices are a_i and b_i, respectively. For each $1 \leqslant i \leqslant n$, let L_i be the linear layout obtained from L_i' by removing a_i and b_i. Let L_0 and L_{n+1} be obtained from L_0' and L_{n+1}' by removing b_0 and a_{n+1}, respectively, and also a_0 and b_{n+1} if they were added. Then we can easily check that $L = L_0 \oplus L_1 \oplus \cdots \oplus L_{n+1}$ is a linear layout of $\widehat{D}$ having width at most k. Therefore $\mathrm{lrw}(G) = \mathrm{lrw}(\widehat{D}) \leqslant k$. □

5.4 Canonical limbs

We now prove two statements on canonical limbs which will be useful to design the algorithm for computing the linear rank-width of distance-hereditary graphs. Let D be the canonical split decomposition of a connected distance-hereditary graph G.

Proposition 5.16. *Let B_1 and B_2 be two distinct bags of D and T_1, T_2 be the component of $D \backslash V(B_1)$, $D \backslash V(B_2)$ respectively, such that T_1 contains the bag B_2 and T_2 contains the bag B_1, and $V(T_1) \cap V(T_2)$ has at least two unmarked vertices of D. For each $i = 1, 2$, let $w_i := \zeta_b(D, B_i, T_i)$ and y_i be an unmarked vertex in D represented by w_i. We define that*

1. *$B_1' := \widetilde{\mathcal{L}}_D[B_2, y_2][V(B_1)]$,*
2. *$B_2' := \widetilde{\mathcal{L}}_D[B_1, y_1][V(B_2)]$,*
3. *y_1' is an unmarked vertex in $\widetilde{\mathcal{L}}_D[B_2, y_2]$ represented by w_1, and*
4. *y_2' is an unmarked vertex in $\widetilde{\mathcal{L}}_D[B_1, y_1]$ represented by w_2.*

Then $\widetilde{\mathcal{L}}[\widetilde{\mathcal{L}}_D[B_1, y_1], B_2', y_2']$ is locally equivalent to $\widetilde{\mathcal{L}}[\widetilde{\mathcal{L}}_D[B_2, y_2], B_1', y_1']$.

Proof. For each $i = 1, 2$, let $v_i := \zeta_t(D, B_i, T_i)$. By Lemma 5.9, there exists a canonical split decomposition D' locally equivalent to D such that for each $i \in \{1, 2\}$, w_i is a leaf of $D'[V(B_i)]$ in D'.

For each $i = 1, 2$, let $P_i := D'[V(B_i)]$, $T_i' := D'[V(T_i)]$, and z_i be an unmarked vertex represented in D' by w_i. Let $T' := D'[V(T_1') \cap V(T_2')]$, and we define

1. $P_1' := \widetilde{\mathcal{L}}_{D'}[P_2, z_2][V(P_1)]$,
2. $P_2' := \widetilde{\mathcal{L}}_{D'}[P_1, z_1][V(P_2)]$,
3. let z_1' be an unmarked vertex in $\widetilde{\mathcal{L}}_{D'}[P_2, z_2]$ represented by w_1,
4. let z_2' be an unmarked vertex in $\widetilde{\mathcal{L}}_{D'}[P_1, z_1]$ represented by w_2.

Since D is locally equivalent to D', by Proposition 5.7, $\widetilde{\mathcal{L}}_D[B_1, y_1]$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[P_1, z_1]$. Again, since $\widetilde{\mathcal{L}}_D[B_1, y_1]$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[P_1, z_1]$, by Proposition 5.7,

$$\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_1, y_1]}[B_2', y_2'] \text{ is locally equivalent to } \widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_1, z_1]}[P_2', z_2'].$$

Similarly, we obtain that

$$\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_2, y_2]}[B_1', y_1'] \text{ is locally equivalent to } \widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_2, z_2]}[P_1', z_1'].$$

Since each v_i is a leaf of P_i in D',

$$\mathcal{L}_{\mathcal{L}_{D'}[P_1, z_1]}[P_2', z_2'] = T' \backslash v_1 \backslash v_2 = \mathcal{L}_{\mathcal{L}_{D'}[P_2, z_2]}[P_1', z_1'],$$

and it implies that

$$\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_1,z_1]}[P_2', z_2'] = \widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_2,z_2]}[P_1', z_1'].$$

Therefore, $\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_1,y_1]}[B_2', y_2']$ is locally equivalent to $\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_2,y_2]}[B_1', y_1']$. □

Proposition 5.17. *Let B_1 and B_2 be two distinct bags of D. Let T_1 be a component of $D \backslash V(B_1)$ that does not contain B_2 and T_2 be the component of $D \backslash V(B_2)$ containing the bag B_1. For $i = 1, 2$, let $w_i := \zeta_b(D, B_i, T_i)$, and y_i be an unmarked vertex in D represented by w_i. If $V(B_1)$ induces a bag B_1' of $\widetilde{\mathcal{L}}_D[B_2, y_2]$, then $\widetilde{\mathcal{L}}_D[B_1, y_1]$ is locally equivalent to $\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_2,y_2]}[B_1', y_1']$, where y_1' is an unmarked vertex in $\widetilde{\mathcal{L}}_D[B_2, y_2]$ represented by w_1.*

Proof. Suppose $V(B_1)$ induces a bag B_1' of $\widetilde{\mathcal{L}}_D[B_2, y_2]$ and y_2' is an unmarked vertex in $\widetilde{\mathcal{L}}_D[B_2, y_2]$ represented by w_1. By Lemma 5.9, there exists a canonical split decomposition D' locally equivalent to D such that w_2 is a leaf of a star bag $P_2 = D'[V(B_2)]$ in D'. We define

- $P_1 := D'[V(B_1)]$,
- z_i is an unmarked vertex in D' represented by w_i,
- $P_1' := \widetilde{\mathcal{L}}[D', P_2, z_2][V(B_1)]$, and
- z_1' is an unmarked vertex in $\widetilde{\mathcal{L}}[D', P_2, z_2]$ represented by w_1.

Since D is locally equivalent to D', by Proposition 5.7, $\widetilde{\mathcal{L}}_D[B_1, y_1]$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[P_1, z_1]$. Similarly, we obtain that $\widetilde{\mathcal{L}}_D[B_2, y_2]$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[P_2, z_2]$. Since $\widetilde{\mathcal{L}}_D[B_2, y_2]$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[P_2, z_2]$, by Proposition 5.7,

$$\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_2,y_2]}[B_1', y_1'] \text{ is locally equivalent to } \widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_2,z_2]}[P_1', z_1'].$$

Since w_2 is a leaf of P_2 in D', $\widetilde{\mathcal{L}}_{D'}[P_1, z_1] = \widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_{D'}[P_2,z_2]}[P_1', z_1']$, and therefore, $\widetilde{\mathcal{L}}_D[B_1, y_1]$ is locally equivalent to $\widetilde{\mathcal{L}}_{\widetilde{\mathcal{L}}_D[B_2,y_2]}[B_1', y_1']$, as required. □

We conclude this section with the following which more or less says that when taking limbs successively, the chosen order is not matter since they are all locally equivalent.

Proposition 5.18. *Let B be a bag of D, let T be a component of $D \backslash V(B)$ and let y be an unmarked vertex in $V(T)$ represented by a marked vertex of B. A canonical split decomposition $\mathcal{L}'$ is locally equivalent to $\widetilde{\mathcal{L}}_D[B, y]$ if and only if there exists D' locally equivalent to D and $y' \in V(T)$ represented by a marked vertex in $V(B)$ such that $\mathcal{L}' = \widetilde{\mathcal{L}}_{D'}[D'[V(B)], y']$.*

Proof. If $\mathcal{L}' = \widetilde{\mathcal{L}}_{D'}[D'[V(B)], y']$ for some canonical split decomposition D' locally equivalent to D with $y' \in V(T)$ represented by some vertex in $V(B)$, then by Lemma 4.8 and Proposition 5.7 we can conclude that $\mathcal{L}'$ is locally equivalent to $\widetilde{\mathcal{L}}_D[B, y]$.

Let us now prove the other direction. It is enough to show it when $\mathcal{L}' := \widetilde{\mathcal{L}}_D[B, y] * x$ for some unmarked vertex x of $\widetilde{\mathcal{L}}_D[B, y]$. Observe that x is necessarily in $V(T)$. From the definition of canonical limb, $\widetilde{\mathcal{L}}_D[B, y]$ is obtained from $D_1 := \mathcal{L}_D[B, y]$, and then $\mathcal{L}'$ is obtained in the same way from $D_2 := D_1 * x$. So, it is sufficient to prove that $D_2 = \mathcal{L}_{D'}[D'[V(B)], y']$ where D' is locally equivalent to D and $y' \in V(T)$ is represented in D' by some marked vertex of $V(B)$. Let $v := \zeta_t(D, B, T)$ and $w := \zeta_b(D, B, T)$. We divide into cases depending on the type of B.

Case 1. *B is of type S and w is a leaf of B.*

In this case, $D_1 = T\backslash v$ and $D_2 = (T\backslash v) * x$. Now, $\mathcal{L}_{D*x}[B, y] = (T * x)\backslash v = (T\backslash v) * x = D_2$.

Case 2. *B is of type K.*

Note that $D_1 = (T * v)\backslash v$ and $D_2 = (T * v)\backslash v * x$.

Case 2.1. x is linked to v in T.

Now $\mathcal{L}_{D*x}[(D * x)[V(B)], x] = (T * x) * x * v * x\backslash v = (T * v)\backslash v * x = D_2$.

Case 2.2. x is not linked to v in T.

Since x is not linked to v in T, by Lemma 4.11, $T * v * x = T * x * v$. So, we have $\mathcal{L}_{D*x}[(D * x)[V(B)], y] = (T * x) * v\backslash v = (T * v)\backslash v * x = D_2$.

Case 3. *B is of type S and w is the center of $D[B]$.*

In this case, $D_1 = (T \wedge vy)\backslash v$ and $D_2 = (T \wedge vy)\backslash v * x$. Let v' be an unmarked vertex represented by v in D. Note that $v' \notin V(T)$.

Case 3.1. x is linked to neither v nor y.

Since x is linked to neither v nor y, by Lemma 4.12, $T * x \wedge vy = T \wedge vy * x$. Thus, we have $\mathcal{L}_{D*x}[(D * x)[V(B)], y] = (T * x) \wedge vy\backslash v = (T \wedge vy\backslash v) * x = D_2$.

Case 3.2. x is not linked to y, but linked to v.

The vertex set $V(B)$ induces a complete bag in $D * v' * y * x$, and x and v are not linked in $D * v' * y$. Thus by Lemma 4.11, $\mathcal{L}_{D*v'*y*x}[(D * v' * y * x)[V(B)], y] = (T * v * y * x) * v\backslash v = (T \wedge vy\backslash v) * x = D_2$.

Case 3.3. x is not linked to v, but linked to y.

The vertex set $V(B)$ induces a star with a leaf w in $D * y * v' * y * x$. Thus, $\mathcal{L}_{D*y*v'*y*x}[(D * y * v' * y * x)[V(B)], y] = (T * y * v * y * x)\backslash v = (T \wedge vy\backslash v) * x = D_2$.

Case 3.4. x is linked to both v and y.

The vertex set $V(B)$ induces a star having the center at w in $D * v' * y * x$. Thus, $\mathcal{L}_{D*v'*y*x}[(D * v' * y * x)[V(B)], x] = (T * v * y * x) * x * v * x\backslash v = (T \wedge vy\backslash v) * x = D_2$. □

5.5 Locally equivalent block graphs

In this section, we extend the following theorem.

Theorem 5.19 (Bouchet [31]). *If two trees are locally equivalent, then they are isomorphic.*

We recall that a vertex of a graph is a simplicial vertex if its neighborhood induces a clique in the graph. We prove the following.

Theorem 5.20. *If two block graphs without simplicial vertices of degree at least 2 are locally equivalent, then they are isomorphic.*

To prove Theorem 5.20, we use the characterization of block graphs in terms of their canonical split decompositions in Lemma 5.3.

Proposition 5.21. *Let G be a connected block graph with at least 3 vertices, and let D be the canonical split decomposition of G. Then G has a simplicial vertex of degree at least 2 if and only if D has a complete bag B containing at least one unmarked vertex.*

Proof. Suppose that $v \in V(G)$ is a simplicial vertex of degree at least 2 in G. Clearly v is not a center of a star bag of D by Lemma 4.7. Because the center of a star bag is unmarked by Lemma 5.3 and v has degree at least 2, v cannot belong to a star bag. So v is in a complete bag of D.

Conversely suppose that D has a complete bag B having at least 1 unmarked vertex v. Since $|V(G)| \geqslant 3$, the bag B contains at least 3 vertices. By Lemma 4.7, the degree of v is at least 2. Since all adjacent bags of B are star bags whose centers are unmarked by Lemma 5.3, v is a simplicial vertex of G. □

Now we are ready to prove Theorem 5.20. This theorem is best possible for block graphs, because if v is a simplicial vertex of a block graph G, then $G * v$ is also a block graph.

Proof of Theorem 5.20. Let G and H be two locally equivalent graphs. Suppose that G and H have no simplicial vertices of degree at least 2. Let D_G and D_H be the canonical split decompositions of G and H, respectively. We may assume that $|V(G)| = |V(H)| \geqslant 3$ and therefore each bag of D_G or D_H has at least 3 vertices.

Since G and H are locally equivalent, by Lemma 4.8 we assume that D_H is obtained from D_G by a sequence of local complementations. Note that applying a local complementation in a split decomposition does not change the number of marked vertices and unmarked vertices in each bag.

Suppose that a bag B of D_G corresponds to a bag $B' = D_H[V(B)]$ of D_H. If B is a complete bag in D_G, then by Proposition 5.21, B has no unmarked vertex in D_G and therefore B' has no unmarked vertex in D_H. Since every star bag of D_H should have at least one unmarked vertex by Lemma 5.3, B' is a complete bag in D_H. Similarly, if B' is a complete bag in D_H, then B is a complete bag in D_G.

Thus B is a star bag of D_G if and only if B' is a star bag of D_H. By Lemma 5.3, the center of a star bag in D_G or D_H is an unmarked vertex. Since a bag B in D_G and B' in D_H have the same number of adjacent bags and unmarked vertices in each canonical split decomposition, the unmarked vertices of B in D_G must be mapped to the unmarked vertices of B' in D_H. Therefore, D_G is isomorphic to D_H and so G is isomorphic to H. □

5.6 Thread graphs

We survey known characterizations of thread graphs, and prove Theorem 1.7. A characterization of thread graphs using canonical split decompositions was first announced by Bui-Xuan, Kanté and Limouzy [39]. Adler, Farley and Proskurowski [1] characterize the complete set of induced subgraph, vertex-minor, pivot-minor obstructions for thread graphs. The induced subgraph obstructions consist of the induced subgraph obstructions for distance-hereditary graphs [9] in Figure 5.1, and 14 additional induced subgraph obstructions for thread graphs that are distance-hereditary, depicted in Figure 5.7.

Based on the characterization of linear rank-width on distance-hereditary graphs, we obtain an alternative proof of the characterization in terms of canonical split decompositions. We add the proof of it.

Theorem 5.22 (Bui-Xuan, Kanté and Limouzy [39]; Adler, Farley and Proskurowski [1]; Kwon and Oum [134])**.** *Let G be a connected graph and let D be the canonical split decomposition of G. The following are equivalent.*

1. *G has linear rank-width at most 1.*
2. *G is distance-hereditary and T_D is a path.*
3. *G is distance-hereditary and G is $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4\}$-free.*
4. *G has no pivot-minor isomorphic to C_5, C_6, α_1, α_3, α_4, α_6, γ_1, and γ_3.*

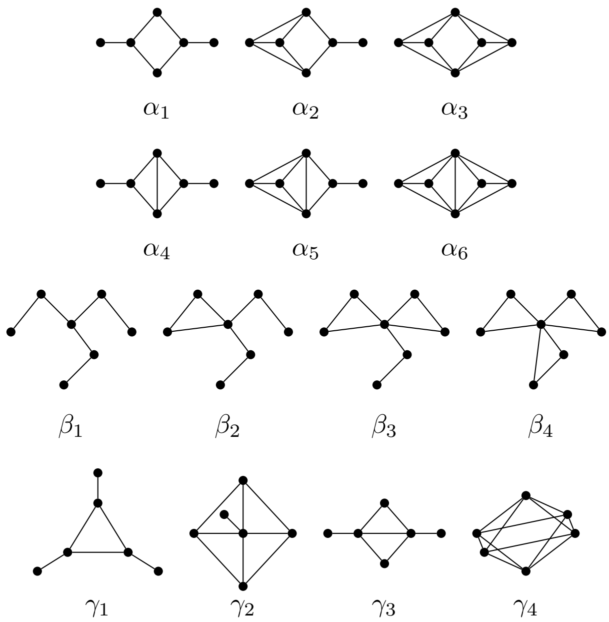


Figure 5.7: The induced subgraph obstructions for thread graphs that are distance-hereditary.

5. G has no vertex-minor isomorphic to C_5*,* α_1 *and* γ_1*.*

6. G is a vertex-minor of a path.

7. G is locally equivalent to a caterpillar tree.

Proof. The proofs of ((1) ↔ (6)) and ((1) ↔ (7)) are given by Kwon and Oum [134], and Bui-Xuan, Kanté and Limouzy [39], respectively. By Lemma 2.1, ((1) → (5)) is clear as C_5, α_1 and γ_1 have linear rank-width 2. ((5) → (4) → (3)) is proved by Adler, Farley and Proskurowski [1].

We add proofs for the remaining parts.

((3) → (2)) We may assume that G is distance-hereditary. Suppose T_D is not a path. Then there exists a bag B of D such that $D \backslash V(B)$ has at least three components T_1, T_2, T_3. For each $i \in \{1, 2, 3\}$, let $v_i := \zeta_b(B, T_i)$ and $w_i := \zeta_t(B, T_i)$. We have three cases; B is a complete bag, or B is a star bag with the center at one of v_1, v_2, v_3, or B is a star bag with the center at a vertex of $V(B) \backslash \{v_1, v_2, v_3\}$.

If B is a complete bag, then G has an induced subgraph isomorphic to one of $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ depending on the types of the marked edges $v_i w_i$. If B is a star bag with the center at one of v_1, v_2, v_3, then G has an induced subgraph isomorphic to one of $\beta_1, \beta_2, \ldots, \beta_6$. Finally, if B is a star bag with the center at a vertex of $V(B) \backslash \{v_1, v_2, v_3\}$, then G has an induced subgraph isomorphic to one of $\gamma_1, \gamma_2, \gamma_3, \gamma_4$. We summarize all cases in Table 5.1.

((2) → (1)) Suppose G has linear rank-width at least 2, and we assume that G is distance-hereditary. By Theorem 5.11, there exists a bag B of D such that B has at least three adjacent bags in D. Thus, T_D is not a path. □

Bui-Xuan, Kanté, and Limouzy [39] announced that we can test whether an input graph has linear rank-width at most 1 in polynomial time using the characterization in terms of split decompositions. For completeness, we add its proof.

Theorem 5.23 (Bui-Xuan, Kanté, and Limouzy [39])**.** *For a given graph G, we can test whether G has linear rank-width at most* 1 *in time* $\mathcal{O}(|V(G)| + |E(G)|)$.

type of B	type of v_1w_1	type of v_2w_2	type of v_3w_3	induced subgraph
A complete bag	KS_p	KS_p	KS_p	α_1
	KS_c	KS_p	KS_p	α_2
	KS_c	KS_c	KS_p	α_3
	KS_c	KS_c	KS_c	α_4
A star bag	S_cS_c	S_pS_p	S_pS_p	β_1
with center at v_1	S_cS_c	S_pS_p	S_pK	β_2
	S_cS_c	S_pK	S_pK	β_3
	S_cK	S_pS_p	S_pS_p	β_4
	S_cK	S_pS_p	S_pK	β_5
	S_cK	S_pK	S_pK	β_6
A star bag	S_pS_p	S_pS_p	S_pS_p	γ_1
with center at	S_pK	S_pS_p	S_pS_p	γ_2
a vertex	S_pK	S_pK	S_pS_p	γ_3
other than v_i	S_pK	S_pK	S_pK	γ_4

Table 5.1: Summary of cases in Corollary 5.22.

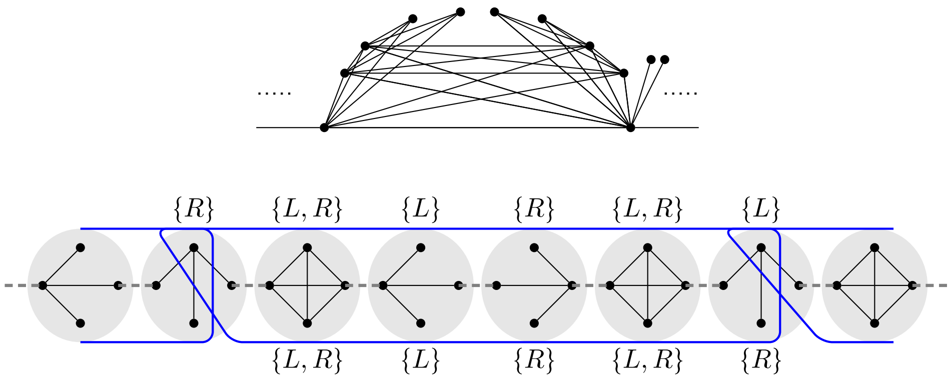


Figure 5.8: The canonical split decomposition of a connected thread graph. The unmarked vertices in each box form a thread block.

Proof. We first compute the canonical split decomposition D of each connected component of G using the algorithm from Theorem 4.4 in time $\mathcal{O}(|V(G)| + |E(G)|)$. This also gives the information that each bag is either prime or a complete bag or a star bag. Then we check whether T_D is a path, and whether each bag is star or complete. By Theorem 5.22, if T_D is a path and each bag is star or complete, then we conclude that G has linear rank-width at most 1, and otherwise, G has linear rank-width at least 2. Because the total number of bags in all split decompositions is $\mathcal{O}(|V(G)|)$, it takes $\mathcal{O}(|V(G)|)$ time. $\square$

Now we discuss a proof of Theorem 1.7. We recall the definition of thread graphs and Theorem 1.7.

A triple $B(x, y) = (G, \sigma, \ell)$, where x and y are two vertices of the graph G, σ is a linear layout of $V(G)$ whose first and last vertices are x and y respectively, and ℓ is a function from $V(G)$ to $\{\{L\}, \{R\}, \{L, R\}\}$, is a *thread block* if

1. $\ell(x) = \{R\}$ and $\ell(y) = \{L\}$,

2. for $v, w \in V(G)$ with $v <_\sigma w$, $vw \in E(G)$ if and only if $R \in \ell(v)$ and $L \in \ell(w)$,

3. $\ell(\sigma^{-1}(2)) \neq \{L\}$ if $\sigma^{-1}(2) \neq y$.

For a digraph $D = (V_D, A_D)$, a set of thread blocks $\{B(x, y) = (G_{xy}, \sigma_{xy}, \ell_{xy}) : xy \in A_D\}$ is said to be *mergeable with* D if for any two arcs x_1y_1, x_2y_2 of A_D, $V(G_{x_1y_1}) \cap V(G_{x_2y_2}) = \{x_1, y_1\} \cap \{x_2, y_2\}$. For a digraph $D = (V_D, A_D)$ and a mergeable set of thread blocks $\mathcal{B}_D = \{B(x, y) = (G_{xy}, \sigma_{xy}, \ell_{xy}) : xy \in A_D\}$, the graph $G = D \odot \mathcal{B}_D$ has the vertex set $V(G) = \bigcup_{xy\in A_D} V(G_{xy})$ and the edge set $E(G) = \bigcup_{xy\in A_D} E(G_{xy})$.

A connected graph G is a *thread graph* if G is either an one vertex graph or $G = P \odot \mathcal{B}_P$ for some directed path P, called the *underlying path*, and some set of thread blocks $\mathcal{B}_P$ mergeable with P. A graph is a *thread graph* if each of its connected components is a thread graph.

Theorem 1.7 (Ganian [97]; Adler, Farley, Proskurowski [1])**.** *A graph has linear rank-width at most* 1 *if and only if it is a thread graph.*

We first prove it for connected graphs. If this is true, then Theorem 1.7 is true because a graph is a thread graph if each component is a thread graph, and a graph has linear rank-width at most 1 if each component has linear rank-width at most 1.

Proof of Theorem 1.7. We first show the case when G is connected. We use the characterization of graphs of linear rank-width 1 in terms of canonical split decompositions in Theorem 5.22.

Suppose G is a connected graph of linear rank-width at most 1, and let D be the canonical split decomposition of G. Since every graph of linear rank-width 1 is distance-hereditary, by Theorem 5.1, every bag of D is either a complete bag or a star bag. We may assume that $|V(G)| \geqslant 2$. From Theorem 5.22, the split decomposition tree of D is a path. Let $B_1B_2 \cdots B_m$ be the sequence of bags representing D. Note that for each bag B_i of D, for $1 \leqslant i \leqslant m$, exactly one of the following is satisfied.

1. (Type 1) B_i is a complete bag.

2. (Type 2) B_i is a star bag whose center is an unmarked vertex.

3. (Type 3) B_i is a star bag whose center is a marked vertex adjacent to a vertex of B_{i-1}.

4. (Type 4) B_i is a star bag whose center is a marked vertex adjacent to a vertex of B_{i+1}.

The center of every bag of Type 2 is a cut vertex in G, and thus, two vertices in different parts of $D\backslash V(B_i)$ are not adjacent. It will be the point where we divide a thread graph into thread blocks.

Let $B_{i_1}, B_{i_2}, \ldots, B_{i_t}$ be the set of all star bags whose centers are unmarked vertices such that $1 \leqslant i_1 < i_2 < \cdots < i_t \leqslant m$. For each $1 \leqslant j \leqslant t$, let v_j be the center of the bag B_{i_j}, and let $v_0 \neq v_{i_1}$ and $v_m \neq v_{i_t}$ be respectively unmarked vertices in B_1 and B_m. Let R_0 be the set of unmarked vertices in the bags on the path from B_1 to B_{i_1}, and for each $1 \leqslant j \leqslant t$ let R_j be the set of unmarked vertices in the bags on the path from B_{i_j+1} to $B_{i_{j+1}}$.

We claim that G is a thread graph whose underlying directed path is $v_0v_1v_2 \ldots v_tv_m$. Suppose that $0 \leqslant i < j \leqslant t$. Then the vertex v_{i+1} is a cut vertex of G and it separates R_i from R_j, and it is easy to observe that there are no edges between the vertex sets $R_i\backslash\{v_i, v_{i+1}\}$ and $R_j\backslash\{v_j, v_{j+1}\}$. Thus, to prove the claim, it is enough to show that for each $0 \leqslant i \leqslant t$, $G[R_i \cup \{v_i\}]$ is a thread block whose linear layout starts at v_i and ends at v_{i+1}.

Let σ_i be a linear layout of $R_i \cup \{v_i\}$ satisfying that for two vertices $x, y \in R_i \cup \{v_i\}$, $x <_{\sigma_i} y$ if the bag containing x appears before the bag containing y in the decomposition path, and for all vertices in the same bag, we give any linear layout among them. We define a labelling ℓ_i of the vertices in $R_i \cup \{v_i\}$ such that for $v \in R_i \cup \{v_i\}$ in a bag B,

1. $\ell_i(v) := \{L, R\}$ if B is of Type 1,
2. $\ell_i(v) := \{L\}$ if B is of Type 3,
3. $\ell_i(v) := \{R\}$ if B is of Type 4,
4. $\ell_i(v_i) =: \{R\}$, $\ell_i(v_{i+1}) := \{L\}$.

Note that v_i and the second vertex of σ_i are contained in different bags, except when $i_1 = 1$, and since D is a canonical split decomposition, the bag containing the second vertex cannot be of Type 3. Thus, $\ell_i(\sigma_i^{-1}(2)) \neq \{L\}$ unless $\sigma_i^{-1}(2) = v_{i+1}$. Thus, it is sufficient to check that for $v, w \in V(G[R_i \cup \{v_i\}])$ with $v <_{\sigma_i} w$, $vw \in E(G[R_i \cup \{v_i\}])$ if and only if $R \in \ell_i(v)$ and $L \in \ell_i(w)$.

When v, w are contained in the same bag, they are adjacent if and only if both have labels $\{L, R\}$ or $w = v_{i+1}$ and v is a pendant vertex in the bag containing v_{i+1}. Thus, it satisfies the condition because v, w should have same labels, or v is labeled $\{R\}$ and w is labeled $\{L\}$. Suppose $v \in B_v$ and $w \in B_w$ with $B_v \neq B_w$. If $vw \in E(G[R_i \cup \{v_i\}])$, then by Lemma 4.6, v and w are linked. Therefore, $R \in \ell_i(v)$ and $L \in \ell_i(w)$. Conversely, if $R \in \ell_i(v)$ and $L \in \ell_i(w)$, then v and w are linked because there are no bags of Type 2 between the bags B_v and B_w in D. Therefore, $vw \in E(G[R_i \cup \{v_i\}])$. It proves the claim.

For the converse direction, suppose that $G = P \odot \mathcal{B}_P$ for some directed path $P = p_1 p_2 \cdots p_m$ from p_1 to p_m, and some set of thread blocks $\mathcal{B}_P$, and for each $1 \leqslant i \leqslant m-1$, let $B(p_i, p_{i+1}) := (G_i, \sigma_i, \ell_i)$. For each $1 \leqslant i \leqslant m-1$, we take a canonical split decomposition D_i of G_i. Since each p_i is a cut-vertex of the graph, it is not hard to see that

$$\sigma_1 \oplus \sigma_2 \cdots \oplus \sigma_{m-1}$$

is a linear layout of width at most 1, which implies that G has linear rank-width at most 1.

Now we consider the case when G is disconnected. From the definition of thread graphs, we have the following.

G has linear rank-width at most 1.
$\Leftrightarrow$ Each component of G has linear rank-width at most 1.
$\Leftrightarrow$ Each component of G is a thread graph.
$\Leftrightarrow$ G is a thread graph. $\square$

Chapter 6. Vertex-minor obstruction sets for graphs of bounded linear rank-width

We provide a lower bound on the number of graphs in a vertex-minor obstruction set for the class of graphs of linear rank-width at most k. From Corollary 1.1, for each k, there exists a finite set $\mathcal{O}_k$ of graphs such that a graph has linear rank-width at most k if and only if it has no vetex-minor isomorphic to a graph in $\mathcal{O}_k$. Adler, Farley, Proskurowski [1] prove that three graphs in Figure 1.1 form a vertex-minor obstruction set for the class of graphs of linear rank-width at most 1.

We show that for each $k \geqslant 2$, there are at least $2^{\Omega(3^k)}$ pairwise locally non-equivalent vertex-minor minimal graphs for the class of graphs of linear rank-width at most k, thus proving that every vertex-minor obstruction set for the class of graphs of linear rank-width at most k has at least $2^{\Omega(3^k)}$ vertex-minor minimal graphs.

Theorem 6.1. *Let $k \geqslant 2$ be an integer. There exist at least $2^{\Omega(3^k)}$ pairwise locally non-equivalent graphs that are vertex-minor minimal with the property that they have linear rank-width larger than k. In other words, if $\mathcal{O}_k$ is a vertex-minor obstruction set for the class of graphs of linear rank-width at most k, then $|\mathcal{O}_k| \geqslant 2^{\Omega(3^k)}$.*

To prove Theorem 6.1, we construct a set Δ_k of graphs that are vertex-minor minimal with the property that the linear rank-width is larger than k.

Constructions of graphs in Δ_k

A *delta composition* G of graphs G_1, G_2, and G_3 is a graph obtained from the disjoint union of G_1, G_2, and G_3 by adding a triangle $v_1v_2v_3$ where $v_i \in V(G_i)$ for $i = 1, 2, 3$. We call $v_1v_2v_3$ the *central triangle* of G. For a non-negative integer k, we define Δ_k as follows:

1. $\Delta_0 = \{(\{x, y\}, \{xy\})\}$. (It is isomorphic to K_2.)
2. For $i \geqslant 1$, Δ_i is the set of all delta compositions of three graphs in Δ_{i-1}.

All non-isomorphic graphs in Δ_2 are depicted in Figure 6.1.

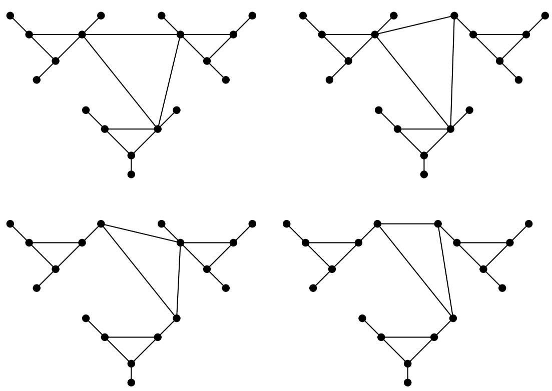


Figure 6.1: All non-isomorphic graphs in Δ_2.

We use Theorem 5.20 to prove that if two graphs in Δ_k are locally equivalent, then they are isomorphic. Generally, by Theorem 5.20, if two block graphs without simplicial vertices of degree at least 2 are locally equivalent, then they are isomorphic. We verify that for an integer k, every graph in Δ_k is a block graph and has no simplicial vertices of degree at least two.

Lemma 6.2. *Every graph in Δ_k is a block graph without simplicial vertices of degree at least 2.*

Proof. Let G be a graph in Δ_k. From the construction of Δ_k, every vertex of G has odd degree and each block of G is isomorphic to K_2 or K_3. Therefore G is a block graph and has no simplicial vertex of degree at least 2. □

In other words, two non-isomorphic graphs in Δ_k cannot be equivalent up to locally equivalence, and it is sufficient to count the number of graphs in Δ_k for providing a lower bound on the number of graphs in a vertex-minor obstruction set for the class of graphs of linear rank-width at most k. We count the number of graphs in Δ_k in Section 6.2.

We note that it is not clear whether every vertex-minor minimal graph for the class of graphs of linear rank-width at most k that is distance-hereditary, is locally equivalent to one of the graph in Δ_k. In Section 6.4, we generalize the constructions of Δ_k and generate a set of canonical split decompositions Ψ_k^+ that has the following property.

- Every distance-hereditary graph of linear rank-width at least $k+1$ contains a vertex-minor isomorphic to a graph whose canonical split decomposition is isomorphic to a split decomposition in Ψ_k^+ (Theorem 6.20).

Using Theorem 6.20, we can generate all vertex-minor minimal distance-hereditary graphs for the class of graphs of linear rank-width at most k.

6.1 Linear rank-width of a graph in Δ_k and its vertex-minor

In this section, we prove the following.

Proposition 6.3. *Let k be a non-negative integer. Every graph in Δ_k is a vertex-minor minimal graph for the class of graphs of linear rank-width at most k.*

First, we prove that every graph in Δ_k has linear rank-width $k+1$. We remark that using the canonical split decompositions of graphs in Δ_k presented in Section 6.3 and Theorem 5.11, we can obtain the following lemma as a corollary. However, we give a direct proof without using canonical split decompositions.

Lemma 6.4. *The linear rank-width of a graph in Δ_k is at least $k+1$.*

Proof. We use induction on k. We may assume that $k \geqslant 1$. Since $G \in \Delta_k$, G is a delta composition of $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$.

Suppose that G has linear rank-width at most k. By the induction hypothesis, G_1 has linear rank-width at least k and therefore G has linear rank-width exactly k. Let L be a linear layout of G having width k. For $v \in V(G)$, we define $S_v = \{x \in V(G) : x \leqslant_L v\}$ and $T_v = V(G) \backslash S_v$. Let a and b be the first and the last vertices in L such that $\rho_G(S_a) = \rho_G(S_b) = k$. Without loss of generality, we may assume that $\{a, b\} \subseteq V(G_2) \cup V(G_3)$. Let L_1 be the subsequence of L whose elements are the vertices of G_1.

For contradiction, we claim that L_1 is a linear layout of G_1 having width at most $k-1$. Let $v \in V(G_1)$. It is sufficient to show that $\rho_{G_1}(S_v \cap V(G_1)) \leqslant k-1$. Note that $v \neq a$ and $v \neq b$. If $v \leqslant_L a$ or $v \geqslant_L b$, then

$$\rho_{G_1}(S_v \cap V(G_1)) \leqslant \rho_G(S_v) \leqslant k-1.$$

So we may assume that $a \leqslant_L v \leqslant_L b$. Note that one of $S_v \cap V(G_1)$ and $T_v \cap V(G_1)$ does not have a neighbor in $G[V(G)\backslash V(G_1)]$ because v_1 is the unique vertex in G_1 which has a neighbor in $G[V(G)\backslash V(G_1)]$. And since $G[V(G)\backslash V(G_1)]$ is connected and $a \in S_v\backslash V(G_1)$ and $b \in T_v\backslash V(G_1)$, there is an edge u_1u_2 in $G[V(G)\backslash V(G_1)]$ such that $u_1 \in S_v\backslash V(G_1)$ and $u_2 \in T_v\backslash V(G_1)$. So

$$A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]$$

is a non-zero matrix. Depending on whether $v_1 \in S_v \cap V(G_1)$ or $v_1 \in T_v \cap V(G_1)$,

$$\begin{aligned}\rho_G(S_v) &= \operatorname{rank}\begin{pmatrix} A(G)[S_v \cap V(G_1), T_v \cap V(G_1)] & 0 \\ A(G)[S_v\backslash V(G_1), T_v \cap V(G_1)] & A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]\end{pmatrix} \\ &\geqslant \operatorname{rank}\left(A(G)[S_v \cap V(G_1), T_v \cap V(G_1)]\right) + \operatorname{rank}\left(A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]\right),\end{aligned}$$

or

$$\begin{aligned}\rho_G(S_v) &= \operatorname{rank}\begin{pmatrix} A(G)[S_v \cap V(G_1), T_v \cap V(G_1)] & A(G)[S_v \cap V(G_1), T_v\backslash V(G_1)] \\ 0 & A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]\end{pmatrix} \\ &\geqslant \operatorname{rank}\left(A(G)[S_v \cap V(G_1), T_v \cap V(G_1)]\right) + \operatorname{rank}\left(A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]\right),\end{aligned}$$

respectively. Thus, we have

$$\begin{aligned}\rho_{G_1}(S_v \cap V(G_1)) &= \operatorname{rank}\left(A(G)[S_v \cap V(G_1), T_v \cap V(G_1)]\right) \\ &\leqslant \rho_G(S_v) - \operatorname{rank}\left(A(G)[S_v\backslash V(G_1), T_v\backslash V(G_1)]\right) \\ &\leqslant \rho_G(S_v) - 1 \leqslant k-1.\end{aligned}$$

So L_1 is a linear layout of G_1 having width at most $k-1$, which is contradiction. Hence, $\operatorname{lrw}(G) \geqslant k+1$. □

If w is a twin of v in a graph G and $G\backslash w$ has linear rank-width $k+1$ with a linear layout of width $k+1$ starting with v, then clearly G also admits a linear layout of width $k+1$ starting with v because we can easily put w in the second place. But the following lemma claims that we can place w at the end if $G\backslash w \in \Delta_k$. This lemma implies that every graph in Δ_k has linear rank-width k. Moreover, it will be mainly used to show that every elementary vertex-minor of a graph in Δ_k has linear rank-width k.

Lemma 6.5. *Let v be a vertex of a graph G and let w be a twin of v. If $G\backslash w \in \Delta_k$, then G has a linear layout L of width $k+1$ such that the first vertex of L is v and the last vertex of L is w.*

Before proving the lemma, we first show that Lemma 6.5 implies the following proposition determining the exact linear rank-width of a graph in Δ_k.

Proposition 6.6. *Every graph in Δ_k has linear rank-width $k+1$. Moreover, for every vertex v of $G \in \Delta_k$, there exists a linear layout of G having width $k+1$ whose first vertex is v.*

Proof. By Lemma 6.4, the linear rank-width of a graph G in Δ_k is at least $k+1$. Let $v \in V(G)$ and let G' be a graph obtained by adding a twin w of v to G. Then Lemma 6.5 implies that G' has a linear layout L of width $k+1$ starting at v and ending at w. We discard w from L to obtain a linear layout of G starting with v having width $k+1$. □

Proof of Lemma 6.5. We prove by induction on k. If $k = 0$, then G is a connected graph on three vertices and therefore every linear layout of G has width 1. Thus we may assume that $k \geqslant 1$. Let $G\backslash w$ be a delta composition of $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. We may assume that $v \in V(G_2)$.

We first claim that G_1 has a linear layout L_1 of width k ending at v_1, and G_3 has a linear layout L_3 of width k starting at v_3. For $i \in \{1, 3\}$, let G'_i is a graph obtained from G_i by adding a twin w_i of v_i. Since $G'_i\backslash w_i \in \Delta_{k-1}$, by the induction hypothesis, G'_i has a linear layout L'_i of width k starting at w_i and ending at v_i, and by discarding w_i from each L'_i, we obtain a linear layout L''_i of G_i ending at v_i. So, $L_1 = L''_1$ and the reverse layout L_3 of L''_3 are the linear layouts of G_1 and G_3 having width k, respectively, such that the last vertex of L_1 is v_1 and the first vertex of L_3 is v_3.

Let

$$H = \begin{cases} G\backslash(V(G_1) \cup V(G_3)) & \text{if } v \neq v_2, \\ G\backslash(V(G_1) \cup V(G_3))\backslash vw & \text{if } v = v_2, \text{ and } v,\ w \text{ are adjacent in } G, \\ G\backslash(V(G_1) \cup V(G_3)) + vw & \text{otherwise.} \end{cases}$$

By the induction hypothesis, H has a linear layout $(v) \oplus L_H \oplus (w)$ of width k.

(1) Clearly, $\rho_G(V(G_1) \cup \{v\}) \leqslant 2 \leqslant k + 1$ and $\rho_G(V(G_3) \cup \{w\}) \leqslant 2 \leqslant k + 1$.

(2) We claim that for $X \subseteq V(G_1)\backslash\{v_1\}$, if $\rho_{G_1}(X) \leqslant k$, then $\rho_G(X \cup \{v\}) \leqslant k + 1$. This is because no vertex in X has a neighbor in $V(G)\backslash V(G_1)$ and therefore $\rho_{G_1}(X) = \rho_G(X) \geqslant \rho_G(X \cup \{v\}) - 1$ by the submodular inequality.

(3) Similar to (2), we deduce that for $X \subseteq V(G_3)\backslash\{v_3\}$, if $\rho_{G_3}(X) \leqslant k$, then $\rho_G(X \cup \{v\}) \leqslant k + 1$.

(4) We claim that if $v \neq v_2$, $X \subseteq V(H)$, and $\rho_H(X) \leqslant k$, then $\rho_G(V(G_1) \cup X) \leqslant k + 1$. By symmetry between G_1 and G_3, we may assume that $v_2 \notin X$. By the submodular inequality, $\rho_G(V(G_1) \cup X) \leqslant \rho_G(X) + \rho_G(V(G_1)) = \rho_H(X) + 1 \leqslant k + 1$.

(5) We claim that if $v = v_2$, $v \in X \subseteq V(H)$, $w \notin X$, and $\rho_H(X) \leqslant k$, then $\rho_G(V(G_1) \cup X) \leqslant k + 1$. By adding the row of v_1 to that of v_2 in $A(G)[X \cup V(G_1), (V(H)\backslash X) \cup V(G_3)]$, we see that $\rho_G(X \cup V(G_1)) \leqslant \rho_H(X) + 1 \leqslant k + 1$.

By combining (1), (2), (3), (4), and (5), we conclude that $(v) \oplus L_1 \oplus L_H \oplus L_3 \oplus (w)$ is a linear layout of G having width at most $k + 1$. Clearly it has width $k + 1$ because $G\backslash w$ has linear rank-width $k + 1$ by Lemma 6.4. □

The following two lemmas will help us to prove that elementary vertex-minors of graphs in Δ_k have linear rank-width at most k.

Lemma 6.7. *Let k be a positive integer and let $G_1, G_2 \in \Delta_{k-1}$. Let G be a graph obtained from the disjoint union of G_1 and G_2 by adding an edge w_1w_2 for fixed $w_1 \in V(G_1)$ and $w_2 \in V(G_2)$. Then G has linear rank-width k.*

Proof. It is trivial that the linear rank-width of G is at least k because an induced subgraph G_1 of G has linear rank-width k by Proposition 6.6. By Proposition 6.6, there is a linear layout L_1 of G_1 having width k such that the last vertex of L_1 is w_1, and there is a linear layout L_2 of G_2 having width k such that the first vertex of L_2 is w_2. Then obviously $L_1 \oplus L_2$ is a linear layout of G having width at most k. □

Lemma 6.8. *Let k be a positive integer. Let G_1, $G_2 \in \Delta_{k-1}$, and let G_3 be a graph having linear rank-width at most $k-1$. Then every delta composition of G_1, G_2 and G_3 has linear rank-width k.*

Proof. Let G be a delta composition of G_1, G_2 and G_3 with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. Clearly the linear rank-width of G is at least k because an induced subgraph G_1 of G has linear rank-width k by Proposition 6.6.

Since G_1, $G_2 \in \Delta_{k-1}$, by Proposition 6.6, there is a linear layout L_1 of G_1 having width k such that the last vertex of L_1 is v_1, and there is a linear layout L_2 of G_2 having width k such that the first vertex of L_2 is v_2. Let L_3 be a linear layout of G_3 having width at most $k-1$.

We claim that $L = L_1 \oplus L_3 \oplus L_2$ is a linear layout of G having width at most k. Let $v \in V(G)$, $S_v = \{x : x \leqslant_L v\}$, and $T_v = V(G) \backslash S_v$. We need to show that $\rho_G(S_v) \leqslant k$ for all $v \in V(G)$. This is clearly true if $v \in V(G_1) \cup V(G_2)$. So let us assume that $v \in V(G_3)$. By symmetry we may assume $v_3 \notin S_v$, because we can swap G_1 and G_2. Then no vertex of G_2 has a neighbor in $S_v \cap V(G_3)$ and therefore

$$\begin{aligned}\rho_G(S_v) &\leqslant \operatorname{rank}(A(G)[V(G_1), T_v]) + \operatorname{rank}(A(G)[S_v \cap V(G_3), T_v]) \\ &= 1 + \rho_{G_3}(S_v \cap V(G_3)) \leqslant k.\end{aligned}$$

Therefore, G has linear rank-width at most k. □

We now prove that every elementary vertex-minor of G in Δ_k has linear rank-width at most k. To prove it, we will use the following lemmas. By Lemma 1.8, it is sufficient to prove that $G\backslash v$, $G * v\backslash v$, and $G \wedge vw\backslash v$ has linear rank-width one less than the linear rank-width of G.

Lemma 6.9. *Let k be a non-negative integer and $G \in \Delta_k$. Then $G\backslash v$ has linear rank-width at most k for each vertex v.*

Proof. We use induction on k. We may assume $k \geqslant 1$. So G is a delta composition of three graphs in Δ_{k-1}, say G_1, G_2 and G_3 with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. We may assume that $v \in V(G_1)$. By the induction hypothesis, $G_1\backslash v$ has linear rank-width at most $k-1$.

If $v = v_1$, then $G\backslash v$ is obtained from the disjoint union of three graphs $G_1\backslash v$, G_2, G_3 by adding an edge v_2v_3 and so $G\backslash v$ has linear rank-width k by Lemma 6.7.

If $v \neq v_1$, then $G\backslash v$ is a delta composition of two graphs in Δ_{k-1} and one graph having linear rank-width at most $k-1$. Thus by Lemma 6.8, $\operatorname{lrw}(G\backslash v) = k$. □

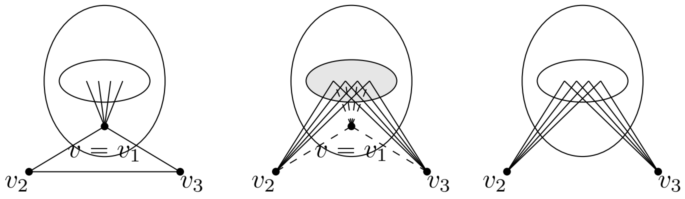
Lemma 6.10. *Let k be a non-negative integer and $G \in \Delta_k$. Then $G * v\backslash v$ has linear rank-width at most k for each vertex v.*

Proof. We use induction on k. We may assume $k \geqslant 1$. Let G be a delta composition of $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. We may assume that $v \in V(G_1)$.

If $v \neq v_1$, then $G * v\backslash v$ is a delta composition of $G_1 * v\backslash v$, G_2 and G_3 where $G_1 * v\backslash v$ has linear rank-width at most $k-1$ by the induction hypothesis. Thus by Lemma 6.8, $G * v\backslash v$ has linear rank-width k.

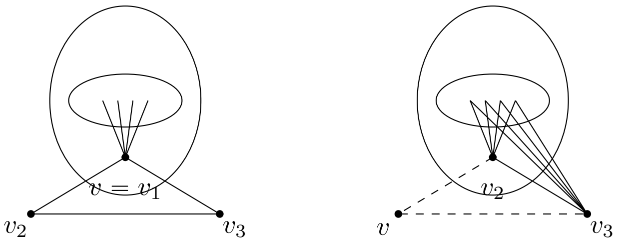
So we may assume $v = v_1$. let $G_1' = (G * v\backslash v)[V(G_1) \cup \{v_2, v_3\}]$. Since v_3 is a twin of v_2 in G_1' and v_3 is not adjacent to v_2 in $G_1' * v_2$ and $G_1' * v_2\backslash v_3$ is isomorphic to G_1 (see Figure 6.2), by Lemma 6.5, G_1' has a linear layout $(v_2) \oplus L_1 \oplus (v_3)$ of width k.

By Proposition 6.6, G_2 has a linear layout L_2 of width k whose last vertex is v_2, and G_3 has a linear layout L_3 of width k whose first vertex is v_3.



$G[V(G_1) \cup \{v_2, v_3\}]$ $\qquad$ G_1' $\qquad$ $G_1' * v_2$

Figure 6.2: The case $G * v \backslash v$ where $v = v_1$ in the proof of Lemma 6.10.



$G[V(G_1) \cup \{v_2, v_3\}]$ $\qquad$ $G[V(G_1) \cup \{v_2, v_3\}] \wedge v_1 v_2 \backslash v_1$

Figure 6.3: The case $G \wedge v_1 v_2 \backslash v$ in the proof of Lemma 6.11.

It follows easily that $L = L_2 \oplus L_1 \oplus L_3$ is a linear layout of $G * v \backslash v$ having width k because $(G * v \backslash v)[V(G_2)] = G_2$, $(G * v \backslash v)[V(G_3)] = G_3$, and $(G * v \backslash v)[V(G_1) \cup \{v_2, v_3\}] = G_1'$. $\square$

Lemma 6.11. *Let k be a non-negative integer and $G \in \Delta_k$. Then $G \wedge vw \backslash v$ has linear rank-width at most k for each edge vw.*

Proof. For each vertex v, it is enough to prove it for one neighbor w of v by Lemma 2.1 and Lemma 1.9.

We use induction on k. We may assume $k \geqslant 1$. Let G be a delta composition of $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1 v_2 v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. We may assume that $v \in V(G_1)$.

If v has only one neighbor w, then $G \wedge vw \backslash v$ is isomorphic to $G \backslash w$ and by Lemma 6.9 we know that $G \backslash w$ has linear rank-width at most k. So we may assume that v has at least two neighbors.

If $v \neq v_1$, then we choose a neighbor w of v such that $w \neq v_1$. It is easy to observe that $G \wedge vw \backslash v$ is a delta composition of $G_1 \wedge vw \backslash v$, G_2, G_3 where $G_1 \wedge vw \backslash v$ has linear rank-width at most $k - 1$ by the induction hypothesis. Hence, by Lemma 6.8, $G \wedge vw \backslash v$ has linear rank-width k.

Thus we may assume $v = v_1$. Since $G[V(G_1) \cup \{v_2, v_3\}] \wedge vv_2 \backslash v$ is isomorphic to a graph obtained from G_1 by adding a twin of v (see Figure 6.3), by Lemma 6.5, $G[V(G_1) \cup \{v_2, v_3\}] \wedge vv_2 \backslash v$ has a linear layout $(v_2) \oplus L_1 \oplus (v_3)$ of width k.

Let w be a neighbor of v in G_1 and let $G_1' = G[V(G_1) \cup \{v_2, v_3\}] \wedge vw \backslash v$. By Lemma 1.9, $G_1' \wedge v_2 w = G[V(G_1) \cup \{v_2, v_3\}] \wedge vw \wedge v_2 w \backslash v = G[V(G_1) \cup \{v_2, v_3\}] \wedge vv_2 \backslash v$ and therefore $(v_2) \oplus L_1 \oplus (v_3)$ is also a linear layout of G_1' having width k.

By Proposition 6.6, G_2 has a linear layout L_2 of width k whose last vertex is v_2, and G_3 has a linear layout L_3 of width k whose first vertex is v_3.

It is now easy to see that $L = L_2 \oplus L_1 \oplus L_3$ is a linear layout of $G \wedge vw \backslash v$ having width at most k because $(G \wedge vw \backslash v)[V(G_2)] = G_2$, $(G \wedge vw \backslash v)[V(G_3)] = G_3$, and $(G \wedge vw \backslash v)[V(G_1) \cup \{v_2, v_3\}] = G_1'$. $\square$

Finally we are ready to prove the main theorem of this section.

Proof of Proposition 6.3. Let $G \in \Delta_k$. By Proposition 6.6, G has linear rank-width $k+1$. And by lemmas 6.9, 6.10, and 6.11, every elementary vertex-minor of G has linear rank-width at most k. Therefore, G is a vertex-minor obstruction for the class of graphs of linear rank-width at most k. □

6.2 The number of graphs in Δ_k

We now prove that Δ_k has at least $2^{\Omega(3^k)}$ pairwise non-isomorphic graphs. A *rooted graph* is a pair of a graph and a specified vertex called a *root*. Two rooted graphs (G, v) and (G', v') are *isomorphic* if there exists a graph isomorphism ϕ from G to G' that maps v to v'. Let us write $\mathrm{Aut}(G)$ to denote the automorphism group of a graph G. For a rooted graph (G, v), we write $\mathrm{Aut}(G, v)$ to denote the automorphism group of (G, v). In other words, $\mathrm{Aut}(G, v) = \{\phi \in \mathrm{Aut}(G) : \phi(v) = v\}$.

First we show that each graph in Δ_k has a unique central triangle.

Lemma 6.12. *Let $k \geqslant 1$ and $G \in \Delta_k$. Then G has a unique cycle $v_1v_2v_3$ of length 3 such that $G\backslash v_1v_2\backslash v_2v_3\backslash v_3v_1$ has exactly three components G_1, G_2, G_3, each of which is in Δ_{k-1}.*

Proof. Clearly there is at least one such cycle because of the construction. Suppose there are two such cycles $T = v_1v_2v_3$ and $T' = v_1'v_2'v_3'$. Let H be a component of $G\backslash v_1v_2\backslash v_2v_3\backslash v_3v_1$ having no vertex of T'. By the condition, $H \in \Delta_{k-1}$ and so H has exactly $2 \cdot 3^{k-1}$ vertices. We may assume $v_1 \in V(H)$. The component J of $G\backslash v_1'v_2'\backslash v_2'v_3'\backslash v_3'v_1'$ intersecting $V(H)$ should be equal to H because T' does not intersect H and $|V(J)| = |V(H)|$. Thus $v_2, v_3 \in T'$ and so v_2 and v_3 have a common neighbor other than v_1. However, this contradicts our assumption that $G\backslash v_1v_2\backslash v_2v_3\backslash v_3v_1$ has exactly three components. □

Let $k \geqslant 2$ and let G be a graph in Δ_k. By the construction, G is a delta composition of three graphs $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for $i = 1, 2, 3$. We call $G \in \Delta_k$

- *Type-A* if (G_1, v_1), (G_2, v_2), and (G_3, v_3) are pairwise isomorphic,
- *Type-B* if exactly two of (G_1, v_1), (G_2, v_2), (G_3, v_3) are isomorphic,
- *Type-C* otherwise.

Lemma 6.13. *Let $k \geqslant 1$ and G be a delta composition of three graphs $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for all $i = 1, 2, 3$. Then,*

1. $\mathrm{Aut}(G) \simeq S_3 \times \mathrm{Aut}(G_1, v_1) \times \mathrm{Aut}(G_2, v_2) \times \mathrm{Aut}(G_3, v_3)$ *if G is Type-A.*
2. $\mathrm{Aut}(G) \simeq S_2 \times \mathrm{Aut}(G_1, v_1) \times \mathrm{Aut}(G_2, v_2) \times \mathrm{Aut}(G_3, v_3)$ *if G is Type-B.*
3. $\mathrm{Aut}(G) \simeq \mathrm{Aut}(G_1, v_1) \times \mathrm{Aut}(G_2, v_2) \times \mathrm{Aut}(G_3, v_3)$ *if G is Type-C.*

Proof. Let $g \in \mathrm{Aut}(G)$. By Lemma 6.12, $g(\{v_1, v_2, v_3\}) = \{v_1, v_2, v_3\}$ and therefore

$$g(V(G_1)), g(V(G_2)), g(V(G_3)) \in \{V(G_1), V(G_2), V(G_3)\}.$$

So $\mathrm{Aut}(G)$ induces a subgroup Γ of S_3 on $\{v_1, v_2, v_3\}$ based on the type of G. It is clear that $\mathrm{Aut}(G)/\Gamma$ is a composition of automorphism groups of three rooted graphs (G_1, v_1), (G_2, v_2) and (G_3, v_3). □

For a graph G and $x \in V(G)$, we define the *orbit* of x in G as the set

$$\{w \in V(G) : w = f(x) \text{ for some automorphism } f \text{ of } G\},$$

and we denote $\#\mathrm{Orb}(G)$ as the number of all distinct orbits of G. For a rooted graph (G, v) and $x \in V(G)$, we define the *orbit* of x in (G, v) as the set

$$\{w \in V(G) : w = f(x) \text{ for some automorphism } f \text{ of } (G, v)\},$$

and we denote $\#\mathrm{Orb}(G, v)$ as the number of all distinct orbits of (G, v).

Lemma 6.14. *Let $k \geqslant 1$ and G be a delta composition of three graphs $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for all $i = 1, 2, 3$. If $v \in V(G_1)$, then*

$$\#\mathrm{Orb}(G, v) \geqslant \#\mathrm{Orb}(G_1, v_1) + \#\mathrm{Orb}(G_2, v_2).$$

Proof. By Lemma 6.12, no vertex in G_1 can be mapped to a vertex in G_2 or G_3 by an automorphism of G fixing v. Thus orbits of (G, v) intersecting $V(G_1)$ cannot contain a vertex in G_2 or G_3. The number of orbits of (G, v) intersecting $V(G_1)$ is equal to the number of distinct subsets of $V(G_1)$ that can be represented as

$$\{f(x) \in V(G_1) \colon f \text{ is an automorphism of } G_1 \text{ such that } f(v) = v,\ f(v_1) = v_1\}$$

for some $x \in V(G_1)$ and this number is at least $\#\mathrm{Orb}(G_1, v_1)$. The number of orbits of (G, v) not intersecting $V(G_1)$ is at least $\#\mathrm{Orb}(G_2, v_2)$ by Lemma 6.13. Thus, we obtain the desired inequality. $\square$

Lemma 6.15. *Let k be a non-negative integer and $G \in \Delta_k$ and $v \in V(G)$. Then (G, v) has at least 2^{k+1} orbits.*

Proof. Trivial if $k = 0$. It follows easily by induction from Lemma 6.14. $\square$

Lemma 6.16. *Let k be a positive integer and $G \in \Delta_k$.*

1. *If G is Type-A, then G has at least 2^k orbits.*
2. *If G is Type-B, then G has at least $2 \cdot 2^k$ orbits.*
3. *If G is Type-C, then G has at least $3 \cdot 2^k$ orbits.*

Proof. Let G be a delta composition of $G_1, G_2, G_3 \in \Delta_{k-1}$ with the central triangle $v_1v_2v_3$ such that $v_i \in V(G_i)$ for all $i = 1, 2, 3$. By Lemma 6.13,

1. $\#\mathrm{Orb}(G) = \#\mathrm{Orb}(G_1, v_1)$ if G is Type-A,
2. $\#\mathrm{Orb}(G) = \#\mathrm{Orb}(G_1, v_1) + \#\mathrm{Orb}(G_2, v_2)$ if G is Type-B and (G_1, v_1) is isomorphic to (G_3, v_3),
3. $\#\mathrm{Orb}(G) = \#\mathrm{Orb}(G_1, v_1) + \#\mathrm{Orb}(G_2, v_2) + \#\mathrm{Orb}(G_3, v_3)$ if G is Type-C.

By Lemma 6.15, we deduce the lemma. $\square$

Let p_k be the number of non-isomorphic rooted graphs (G, v) with $G \in \Delta_k$. Then $p_0 = 1$, $p_1 = 2$, and $p_2 = 24$ (see Figure 6.1). We can easily verify that Δ_k has

- exactly p_{k-1} non-isomorphic Type-A graphs,
- exactly $p_{k-1}(p_{k-1} - 1)$ non-isomorphic Type-B graphs,
- exactly $\binom{p_{k-1}}{3}$ non-isomorphic Type-C graphs.

We are now ready to provide a lower bound on the number of non-isomorphic graphs in Δ_k.

Proposition 6.17. *Let $k \geqslant 2$ be an integer. Then Δ_k has at least $2^{\Omega(3^k)}$ non-isomorphic graphs.*

Proof. Let a_k, b_k, c_k be the number of non-isomorphic graphs in Δ_k that is Type-A, Type-B, and Type-C respectively. By Lemma 6.16,

$$p_k \geqslant 2^k a_k + 2 \cdot 2^k b_k + 3 \cdot 2^k c_k.$$

Since $a_k = p_{k-1}$, $b_k = p_{k-1}(p_{k-1} - 1)$ and $c_k = \binom{p_{k-1}}{3}$, we obtain the following recurrence relation;

$$a_{k+1} = p_k \geqslant 2^k a_k + 2 \cdot 2^k b_k + 3 \cdot 2^k c_k = 2^{k-1} a_k^2 (a_k + 1) \geqslant 2^{k-1} a_k^3$$

and $a_2 = 2$. We deduce that $a_k \geqslant 2^{(1-2k)/4+7\cdot 3^k/36} = 2^{\Omega(3^k)}$. □

Now we can combine all to prove our main theorem.

Proof of Theorem 6.1. By Proposition 6.3, every graph in Δ_k is a vertex-minor minimal graph for the class of graphs of linear rank-width at most k. Proposition 6.17 states that Δ_k has at least $2^{\Omega(3^k)}$ non-isomorphic graphs. Lemma 6.2 and Theorem 5.20 show that two non-isomorphic graphs in Δ_k cannot be locally equivalent. Therefore, there are at least $2^{\Omega(3^k)}$ pairwise locally non-equivalent vertex-minor minimal graphs for the class of graphs of linear rank-width at most k. In other words, if $\mathcal{O}_k$ is a vertex-minor obstruction set for the class of graphs of linear rank-width at most k, thn $|\mathcal{O}_k| \geqslant 2^{\Omega(3^k)}$. □

6.3 Canonical split decompositions of graphs in Δ_k

We now aim to describe the canonical split decomposition S_G of each graph G in Δ_k for $k \geqslant 1$ explicitly. Using this structure, we can show that two locally equivalent graphs in Δ_k are isomorphic, even though we showed it using a general theorem. In the next subsection, we generalize this construction to obain all vertex-minor minimal distance-hereditary graphs for the class of graphs of linear rank-width at most k.

Let us call the edges of the graph in Δ_0 *thick*. In graphs in Δ_k, the edges originated from Δ_0 are *thick* and all other edges introduced by a delta composition are *thin*. Observe the set of thick edges of $G \in \Delta_k$ is a perfect matching and therefore we deduce the following.

Lemma 6.18. *For graphs in Δ_k, each leaf is incident only with a thick edge and no two leaves have a common neighbor.*

For $G \in \Delta_k$, let $\mathcal{C}(G)$ be the set of triangles in G. First let us describe the set $\Theta(G)$ of marked vertices of H_G. For each thick edge uv joining two non-leaf vertices, we have two new vertices $m(u, v)$ and $m(v, u)$ in $\Theta(G)$ and for each pair of a vertex v and a triangle C containing v, we have two new vertices $m(C, v)$ and $m(v, C)$. We will construct S_G so that $V(S_G)$ is the disjoint union of $V(G)$ and $\Theta(G)$. For convenience, if w is a leaf incident with an (thick) edge vw, then $m(v, w) := w$.

Now we describe all bags of S_G. For each vertex v in G of degree $n > 1$, if w is the unique neighbor of v joined by a thick edge, then let $B(v)$ be the graph isomorphic to $K_{1,(n-1)/2+1}$ on the vertex set

$$\{v, m(v, w)\} \cup \{m(v, C) : C \in \mathcal{C}(G), v \in V(C)\}$$

with the center v. For each triangle C of G, let $B(C)$ be the graph isomorphic to K_3 on the vertex set $\{m(C, v) : v \in V(C)\}$.

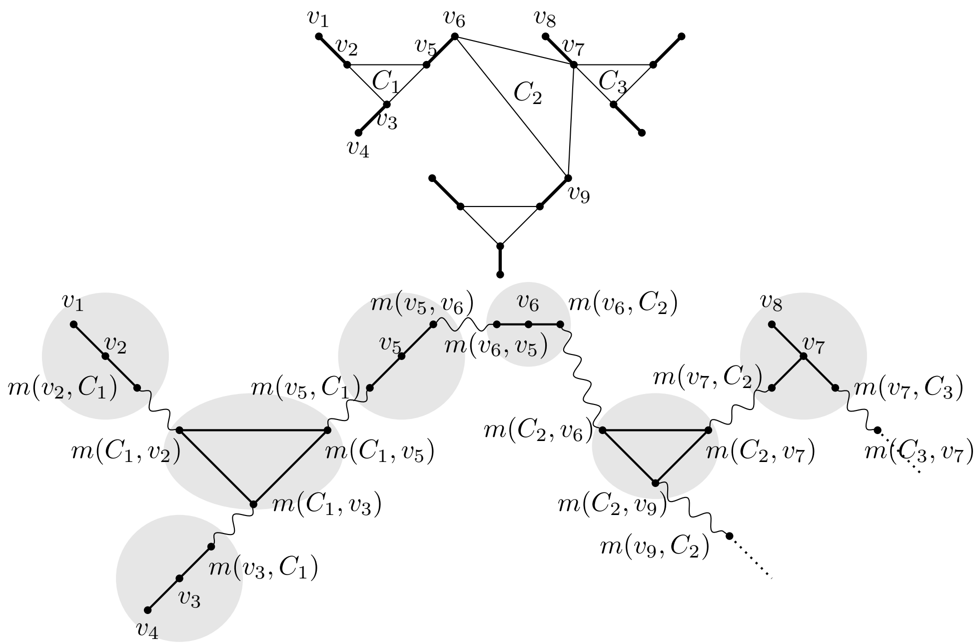


Figure 6.4: A graph $G \in \Delta_2$ with thick edges, and a part of S_G.

Let S_G be the marked graph on the vertex set $\Theta(G) \cup V(G)$ such that all bags of S_G are

$$\{B(v) : v \text{ is a non-leaf vertex in } G\} \cup \{B(C) : C \in \mathcal{C}(G)\}$$

and the set $M(S_G)$ of all marked edges is exactly

$$\begin{aligned}&\{m(v, C)m(C, v) : C \in \mathcal{C}(G), v \in V(C)\}\\ \cup&\{m(v, w)m(w, v) : vw \text{ is the thick edge joining two non-leaf vertices}\}\end{aligned}$$

For a graph G in Δ_2, the marked graph S_G is depicted in Figure 6.4.

We now show that if $G \in \Delta_k$, then S_G is the canonical split decomposition of G.

Proposition 6.19. *For each graph $G \in \Delta_k$ with $k \geqslant 1$, the marked graph S_G is the canonical split decomposition of G.*

Proof. We first prove that S_G is a split decomposition of G. We use induction on k. We may assume that $k \geqslant 2$ and let C be the central triangle $v_1v_2v_3$ of G. For each $1 \leqslant i \leqslant 3$, let G_i be the component of $G \backslash v_1v_2 \backslash v_2v_3 \backslash v_3v_1$ such that $v_i \in V(G_i)$, and let S_i be the component of

$$S_G \backslash \{m(C, v_1), m(C, v_2), m(C, v_3)\}$$

such that $v_i \in V(S_i)$. Let w_i be the neighbor of v_i such that v_iw_i is thick.

If v_i is not a leaf in G_i, then by construction, $S_i \backslash m(v_i, C) = S_{G_i}$. If v_i is a leaf of G_i, then S_{G_i} is obtained from $S_i \backslash m(v_i, C)$ by recomposing a marked edge joining $m(v_i, w_i)$ and $m(w_i, v_i)$. By the induction hypothesis, S_{G_i} is a split decomposition of G_i and therefore in both cases, $S_i \backslash m(v_i, C)$ is a split decomposition of G_i because we obtain G_i from $S_i \backslash m(v_i, C)$ by recomposing all marked edges of $S_i \backslash m(v_i, C)$.

Let G'_i be the graph obtained from S_i by recomposing all marked edges of S_i. Then $m(v_i, C)$ is a leaf of G'_i and $G'_i \backslash m(v_i, C) = G_i$.

If we recompose all marked edges of S_G except three marked edges associated with C, then we obtain a marked graph obtained from the disjoint union of G'_1, G'_2, G'_3, and $B(C)$ by adding three marked edges in $\{m(v_i, C), m(C, v_i)\}_{1 \leqslant i \leqslant 3}$. It is then clear that G is obtained from this graph by recomposing three marked edges in $\{m(v_i, C), m(C, v_i)\}_{1 \leqslant i \leqslant 3}$ from this graph. This proves that S_G is a split decomposition of G.

It remains to check whether S_G is a canonical split decomposition. From the construction, every bag of S_G is a complete bag or a star bag, and every star bag has marked vertices only on its leaves and no two complete bags are adjacent bags. This proves the lemma. □

6.4 General constructions of distance-hereditary obstructions

We generalize the constructions of Δ_k to generate all vertex-minor minimal graphs that are distance-hereditary graphs. We use the incremental characterization of distance-hereditary graphs, described in Section 5.1.

We say that for a distance-hereditary graph G, a graph G' is an *one-vertex DH-extension* of G if $G = G' \backslash v$ for some vertex $v \in V(G')$ and G' is distance-hereditary. For convenience, if G' is an *one-vertex DH-extension* of G and D, D' are canonical split decompositions of G, G', respectively, then D' is also called an *one-vertex DH-extension* of D.

For a set $\mathcal{D}$ of canonical split decompositions, we define

$$\mathcal{D}^+ := \mathcal{D} \cup \{D' : D' \text{ is an one vertex DH-extension of } D \in \mathcal{D}\}.$$

From Theorem 5.4, we can generate the set $\mathcal{D}^+$ from $\mathcal{D}$. For a set $\mathcal{D}$ of canonical split decompositions, we define a new set $\Delta(\mathcal{D})$ of canonical split decompositions D as follows:

- Choose three split decompositions D_1, D_2, D_3 in $\mathcal{D}$ and for each $1 \leqslant i \leqslant 3$, take an one-vertex extension D'_i of D_i with a new vertex w_i. We introduce a new bag B of type K or S having three vertices v_1, v_2, v_3 and

  1. if v_i is in a complete bag, then we define $D''_i := D'_i * w_i$,
  2. if v_i is the center of a star bag, then we define $D''_i := D'_i \wedge w_i z_i$ for some z_i linked to w_i in D'_i,
  3. if v_i is a leaf of a star bag, then we define $D''_i := D'_i$.

  Let D be the canonical split decomposition obtained by the disjoint union of D''_1, D''_2, D''_3 and B by adding the marked edges v_1w_1, v_2w_2, v_3w_3.

For each non-negative integer k, we recursively construct the sets Ψ_k and Φ_k of canonical split decompositions as follows.

1. $\Psi_0 = \Phi_0 := \{K_2\}$ (K_2 is the canonical split decomposition of itself.)
2. For $k \geqslant 0$, let $\Psi_{k+1} := \Delta(\Psi_k^+)$.
3. For $k \geqslant 0$, let $\Phi_{k+1} := \Delta(\Phi_k)$

We prove the following.

Theorem 6.20. *Let $k \geqslant 0$. Every distance-hereditary graph of linear rank-width at least $k+1$ contains a vertex-minor isomorphic to a graph whose canonical split decomposition is isomorphic to a split decomposition in Ψ_k^+.*

We remark that for each non-negative integer k, starting with the set Ψ_k^+, we can construct a set of vertex-minor minimal graphs $\mathcal{O}$ such that

1. every distance-hereditary graph of linear rank-width at least $k+1$ contains a vertex-minor isomorphic to a graph in $\mathcal{O}$, and
2. for $G \in \mathcal{O}$, $\mathrm{lrw}(G) = k+1$ and every its proper vertex-minor has linear rank-width at most k.

For computing the linear rank-width values of distance-hereditary graphs, we can use the result in Theorem 9.1.

Let D be the canonical split decomposition of a connected distance-hereditary graph G.

Lemma 6.21. *Let B_1 and B_2 be two distinct bags of D, and for each $i \in \{1,2\}$, let T_i be the components of $D \backslash V(B_i)$ such that T_1 contains the bag B_2 and T_2 contains the bag B_1. If*

- *$y_1 := \zeta_b(D, B_1, T_1)$ is not a center of a star bag, and*
- *B_2 is a star bag and $y_2 := \zeta_b(D, B_2, T_2)$ is a leaf of B_2,*

then there exists a canonical split decomposition D' such that

1. *$\widehat{D}$ has $\widehat{D'}$ as a vertex-minor,*
2. *$D[V(T_2)\backslash V(T_1)] = D'[V(T_2)\backslash V(T_1)]$,*
3. *$D[V(T_1)\backslash V(T_2)] = D'[V(T_1)\backslash V(T_2)]$, and*
4. *either D' has no bags between B_1 and B_2, or D' has only one bag B between B_1 and B_2 such that $|V(B)| = 3$, B is star, the center of B is an unmarked vertex, and the two leaves are adjacent to y_1 and y_2 in D'.*

Proof. If B_1 and B_2 are adjacent bags in D, then we are done. We assume that there exists at least one bag between B_1 and B_2 in D. Let $P = p_1p_2 \ldots p_\ell$ be the shortest path from $y_1 = p_1$ to $y_2 = p_\ell$ in D. Note that $\ell \geqslant 4$.

Let C be a bag in D that contains exactly two vertices p_i, p_{i+1} of P. Then we remove C and all components of $D\backslash V(C)$ which does not contains a vertex of B_1 or B_2, and add a marked edge $p_{i-1}p_{i+2}$. Since this operation does not change the parts $D[V(T_2)\backslash V(T_1)]$ and $D[V(T_1)\backslash V(T_2)]$, applying this operation consecutively, we may assume that except B_1 and B_2, all bags of D having a vertex of P contain three vertices of P. Those bags should be star bags where the middle vertices of them are the centers.

If there exist two adjacent bags C_1 and C_2 in D such that $p_i, p_{i+1}, p_{i+2} \in V(C_1)$ and $p_{i+3}, p_{i+4}, p_{i+5} \in V(C_2)$. Take two unmarked vertices x_{i+1} and x_{i+4} of D that are represented by p_{i+1}, p_{i+4}, respectively. By pivoting $x_{i+1}x_{i+4}$ in D, we can modify two bags C_1 and C_2 so that $p_ip_{i+2}p_{i+3}p_{i+5}$ become a path. By Lemma 4.9, this pivoting does not affect on the parts $D[V(T_2)\backslash V(T_1)]$ and $D[V(T_1)\backslash V(T_2)]$. We remove C_1 and C_2 from D (with all components of $D\backslash V(C_i)$ which does not contain a vertex of B_1 or B_2), and add a marked edge $p_{i-1}p_{i+6}$. Because by the assumption that y_1 is not the center of B_1 we know that by removing C_1 and C_2 the bag B_1 will not be merged with the bag just after C_2 in the path between y_1 and y_2, we obtain a canonical split decomposition satisfying the condition (1), (2), (3), and the number of bags containing P is decreased by two. By recursively doing this procedure, at the end, we have either no bags between B_1 and B_2, or only one star bag whose two leaves are adjacent to y_1 and y_2. □

The next proposition says how we can replace limbs having linear rank-width $\geqslant k = 1$ into a split decomposition in Ψ^+_{k-1} using Lemma 6.21.

Proposition 6.22. *Let B be a star bag of D and v be a leaf of B. Let T be a component of $D \backslash V(B)$ such that $\zeta_b(D, B, T) = v$, and w be an unmarked vertex of D represented by v. Let A be the canonical split decomposition of a distance-hereditary graph. If $\widehat{\mathcal{L}}_D[B, w]$ has a vertex-minor that is either $\widehat{A}$ or an one-vertex extension of $\widehat{A}$, then there exists a canonical split decomposition D' on a subset of $V(D)$ such that*

1. *either $D' \backslash V(T) = D \backslash V(T)$ or $D' \backslash V(T) = (D \backslash V(T)) * v$, and*
2. *$\widetilde{\mathcal{L}}_{D'}[B, w']$ is either A or an one-vertex DH-extension of A for some unmarked vertex w' of D' represented by v.*

Proof. Suppose that there exists a sequence $x_1, x_2, \ldots, x_m$ of vertices of $\widehat{\mathcal{L}}_D[B, w]$ and $S \subseteq V(\widehat{\mathcal{L}}_D[B, w])$ such that $(\widehat{\mathcal{L}}_D[B, w] * x_1 * x_2 * \ldots * x_m) \backslash S$ is either $\widehat{A}$ or an one-vertex DH-extension of $\widehat{A}$. Note that $(\mathcal{L}_D[B, w] * x_1 * x_2 * \ldots * x_m) \backslash S$ is not necessary a split decomposition, because it could have some marked vertices that does not represent any unmarked vertices. However, by removing such vertices successively, we make it a split decomposition of either $\widehat{A}$ or an one-vertex DH-extension of $\widehat{A}$. Let $Q \subseteq V(D)$ such that $(\mathcal{L}_D[B, w] * x_1 * x_2 * \ldots * x_m)[Q]$ is a split decomposition of either $\widehat{A}$ or an one-vertex DH-extension of $\widehat{A}$. Since $\mathcal{L}_D[B, w]$ is an induced subgraph of D, we have

$$(\mathcal{L}_D[B, w] * x_1 * x_2 * \ldots * x_m)[Q] = (D * x_1 * x_2 * \ldots * x_m)[Q].$$

For convenience, let $D^* = D * x_1 * x_2 * \ldots * x_m$. Note that $D[V(B)] = D^*[V(B)]$.

We choose a bag B' in D^* such that

1. B' has a vertex of Q, and
2. the distance from B' to B in T_{D^*} is minimum.

Here, we want to shrink all the bags between B' and B using Lemma 6.21. Let T_1 be the component of $D^* \backslash V(B')$ containing the bag B and let T_2 be the component of $D^* \backslash V(B)$ containing the bag B'. Let $y := \zeta_b(D, B', T_1)$. From the choice of B', $y \notin Q$. (If $y \in Q$, then there exists an unmarked vertex represented by y, and all vertices on the path from y to it should be contained in Q.) Since $D^*[Q]$ is connected and B' has at least two vertices of Q, y is not the center of a star bag.

Applying Lemma 6.21, there exists a canonical split decomposition D_1 such that

1. $\widehat{D^*}$ has $\widehat{D_1}$ as a vertex-minor,
2. $D^*[V(T_2) \backslash V(T_1)] = D_1[V(T_2) \backslash V(T_1)]$,
3. $D^*[V(T_1) \backslash V(T_2)] = D_1[V(T_1) \backslash V(T_2)]$,
4. either D_1 has no bags between B and B', or D_1 has exactly one bag B_s between B and B' such that $|V(B_s)| = 3$, B_s is star whose center is an unmarked vertex, and the two leaves of B_s are adjacent to y and v in D_1.

We first remove the vertices of $V(T_2) \backslash V(T_1)$ that are not contained in $Q \cup \{y\}$. Let $D_2 := D_1 \backslash ((V(T_2) \backslash V(T_1)) \backslash (Q \cup \{y\}))$. From the choice of Q, we know that every marked vertex of D_2 represents at least one unmarked vertex, and therefore D_2 is a split decomposition. We consider $\widetilde{D_2}$. Because

v is a leaf of B, and $D_2[B']$ is either a complete graph or a star with y a leaf, B' and B are still bags in $\widetilde{D_2}$. Moreover, if B_s exists in D_1, then B_s is still a bag of $\widetilde{D_2}$.

Let B_2 be the bag of $\widetilde{D_2}$ containing y. Clearly, $D_2[Q] = D_1[Q] = D^*[Q]$, and therefore $\widetilde{D_2[Q]}$ is either A or an one vertex DH-extension of A. We divide into cases.

Case 1. *$\widetilde{D_2}$ has no bags between B and B_2.*

In this case, $\widetilde{D_2}$ is a required decomposition. Choose an unmarked vertex z in $\widetilde{D_2}$ that is represented by v. Then $\widetilde{\mathcal{L}}_{\widetilde{D_2}}[B,z] = \widetilde{D_2[Q]}$ because v is a leaf of the bag B.

Case 2. *$\widetilde{D_2}$ has one bag B_s between B and B_2 where $|V(B_s)| = 3$, B_s is star whose center c is an unmarked vertex, and two leaves c_1, c_2 of B_s are adjacent to y and v, respectively.*

Choose an unmarked vertex z in $\widetilde{D_2}$ that is represented by c_1. From the construction, we can easily observe that $\widetilde{\mathcal{L}}_{\widetilde{D_2}}[B_s,z] = \widetilde{D_2[Q]}$.

If $\widetilde{D_2[Q]} = A$, then we can regard $\widetilde{\mathcal{L}}_{\widetilde{D_2}}[B,c]$ as an one-vertex DH-extension of A with the new vertex c. Therefore, we may assume that $\widetilde{D_2[Q]}$ is an one-vertex DH-extension of A with a newly added vertex a for some unmarked vertex a of $\widetilde{D_2[Q]}$. Note that since y is not the center of a star bag, either y is a leaf of a star bag or B_2 is a complete bag.

If B_2 is a star whose center is an unmarked vertex in $\widetilde{D_2}$, then we obtain a new decomposition D_3 by applying local complementation at c and removing c and recomposing the two marked edges incident with B_s. Note that D_3 is exactly the decomposition obtained from the disjoint of the two components of $\widetilde{D_2}\backslash V(B_s)$ by adding a marked edge yv, and it is canonical. Also, z is represented by v in D_3, and therefore $\widetilde{\mathcal{L}}_{D_3}[B,z] = \widetilde{D_2[Q]}$. Thus, D_3 is a required decomposition.

If B_2 is a complete bag and its vertex y' is an unmarked vertex in $\widetilde{D_2}$, then we obtain a new decomposition D_3 by pivoting $y'c$ on $\widetilde{D_2}$ and removing c and recomposing the two marked edges incident with B_s. Here, we also have a split decomposition obtained from the disjoint of the two components of $\widetilde{D_2}\backslash V(B_s)$ by adding a marked edge yv, and therefore D_3 is a required decomposition.

Now we may assume that at least two unmarked vertices of $\widetilde{D_2}$ are represented by c_1. So, c is linked to at least two vertices of $\widehat{A}$ in $\widetilde{D_2}$. Since $\widehat{A}$ is an one vertex DH-extension of a connected distance-hereditary graph, $\widehat{A}\backslash a$ is connected. So, if we define $D_3 := \widetilde{D_2\backslash a}$, then D_3 is connected and $\widetilde{\mathcal{L}}_{D_3}[B,c]$ can be regarded as an one vertex DH-extension of A. Therefore, D_3 is a required decomposition. $\square$

Proof of Theorem 6.20. We prove it by induction on k. If $k = 0$, then $\mathrm{lrw}(G) \geqslant 1$ and G has an edge. Therefore, we may assume that $k \geqslant 1$.

Let D be the canonical split decomposition of G. Since G has linear rank-width at least $k+1$, by Theorem 5.11, there exists a bag B in D with three components T_1, T_2, T_3 of $D\backslash V(B)$ such that $f_D(B, T_i) \geqslant k$ for each $1 \leqslant i \leqslant 3$. For each $1 \leqslant i \leqslant 3$, let $v_i := \zeta_b(D, B, T_i)$ and $w_i := \zeta_t(D, B, T_i)$, and z_i be an unmarked vertex of D that is represented by v_i in D.

By Proposition 5.7, we may assume that B is a star with the center v_3. We also assume that B has exactly three vertices, by removing all components of $D\backslash V(B)$ other than T_1, T_2, T_3. Since v_1 and v_2 are leaves of B, for each $i \in \{1,2\}$, $\mathcal{L}_D[B, z_i] = T_i\backslash w_i$ and $\widehat{\mathcal{L}}_D[B, z_i]$ has linear rank-width at least k. So, by the induction hypothesis, there exists a canonical split decomposition D_i in Ψ^+_{k-1} such that $\widehat{\mathcal{L}}_D[B, z_i]$ has a vertex-minor isomorphic to a graph $\widehat{D_i}$. Note that D_i is a split decomposition in Ψ_{k-1} or an one vertex DH-extension of a split decomposition in Ψ_{k-1}. Then by applying Proposition 6.22 twice, we can obtain a canonical split decomposition D' satisfying that

1. $D'[V(B)] = D[V(B)]$,

2. either $D'[V(T_3)] = T_3$ or $D'[V(T_3)] = T_3 * w_3$, and

3. for each $i \in \{1, 2\}$, $\widetilde{\mathcal{L}}_{D'}[B, z'_i]$ is isomorphic to a split decomposition in Ψ^+_{k-1} for some unmarked vertex z'_i of D' represented by v_i.

For each $i \in \{1, 2, 3\}$, let T'_i be the component of $D' \backslash V(B)$ containing z'_i, and $w'_i := \zeta_t(D', D'[V(B)], T'_i)$. Note that $T'_1 \backslash w'_1$ and $T'_2 \backslash w'_2$ are contained in Ψ^+_{k-1}. We choose an unmarked vertex z'_3 that is represented by v'_3 in D'. If we apply local complementation at z'_3 and z'_2 subsequently in D', then

1. B is changed to a star with the center v_2,

2. T'_1 is the same as before,

3. T'_2 is changed to $T'_2 * w'_2 * z'_2$,

4. T'_3 is changed to $T'_3 * z'_3 * w'_3$.

Now, we again apply Proposition 6.22 to $D' * z'_3 * z'_2$, and obtain a canonical split decomposition D'' satisfying that

1. $D''[V(B)] = (D' * z'_3 * z'_2)[V(B)]$ and $D''[V(T'_1)] = (D' * z'_3 * z'_2)[V(T'_1)]$,

2. either $D''[V(T'_2)] = (D' * z'_3 * z'_2)[V(T'_2)]$ or $(D' * z'_3 * z'_2)[V(T'_2)] * w'_2$, and

3. $\widetilde{\mathcal{L}}_{D''}[B, z''_3]$ is isomorphic to a split decomposition in Ψ^+_{k-1} for some unmarked vertex z''_3 of D'' represented by v_3.

Let T''_3 be the component of $D'' \backslash V(B)$ containing z''_3, and $w''_3 := \zeta_t(D'', D''[V(B)], T''_3)$. Note that $T''_3 \backslash w''_3 \in \Psi^+_{k-1}$ and for $i \in \{1, 2\}$, z'_i is still represented by v_i in D''.

Now we claim that $D'' \in \Psi_k$ or $D'' * z'_2 \in \Psi_k$. We observe two cases depending on whether $D''[V(T'_2)] = (D' * z'_3 * z'_2)[V(T'_2)]$ or $(D' * z'_3 * z'_2)[V(T'_2)] * w'_2$.

Case 1. $D''[V(T'_2)] = (D' * z'_3 * z'_2)[V(T'_2)]$.

We observe that B is a star with the center v_2 in D'', and the three components of $D'' \backslash V(B)$ are T'_1, $T'_2 * w'_2 * z'_2$, and T''_3. In this case, $D'' * z'_2 \in \Psi_k$ because

1. B is a complete bag in $D'' * z'_2$, and

2. the three components of $D'' \backslash V(B)$ are $T'_1 * w'_1$, $T'_2 * w'_2$, and $T''_3 * w''_3$,

and each limb of $D'' * z'_2$ with respect to B are $T'_1 \backslash w'_1$, $T'_2 \backslash w'_2$, $T''_3 \backslash w''_3$, which are contained in Ψ'_{k-1}.

Case 2. $D''[V(T'_2)] = (D' * z'_3 * z'_2)[V(T'_2)] * w'_2$.

We observe that B is a star with the center v_2 in D'', and the three components of $D'' \backslash V(B)$ are T'_1, $T'_2 * w'_2 * z'_2 * w'_2$, and T''_3. We can see that $D'' \in \Psi_k$ because each limb with respect to B are $T'_1 \backslash w'_1$, $T'_2 \backslash w'_2$, $T''_3 \backslash w''_3$, which are contained in Ψ'_{k-1}.

We conclude that G has a vertex-minor isomorphic to $\widehat{D''}$ where $D'' \in \Psi_k \subseteq \Psi'_k$, as required. □

In order to prove that Ψ_k is the set of canonical split decompositions of distance-hereditary vertex-minor obstructions for linear rank-width at most k, we need to prove that for every $D \in \Psi_k$, $\widehat{D}$ has linear rank-width $k + 1$ and its proper vertex-minors have linear rank-width at most k. However, while by Theorem 5.11 we know that $\mathrm{lrw}(\widehat{D}) \geqslant k + 1$, it is not true that for all $D \in \Psi_k$, all proper vertex-minors of $\widehat{D}$ have linear rank-width k. For instance, the canonical split decomposition in Figure 6.5 is in Ψ_1,

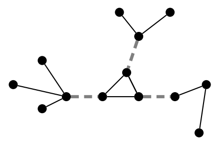


Figure 6.5: A canonical split decomposition in Ψ_1.

has linear rank-width 2 but not all proper vertex-minors have linear rank-width 1. Instead we show the property for canonical split decompositions in Φ_k.

Proposition 6.23. *Let $k \geqslant 0$ and let $D \in \Phi_k$. Then $\mathrm{lrw}(\widehat{D}) = k + 1$ and every proper vertex-minor of $\widehat{D}$ has linear rank-width at most k.*

To prove Proposition 6.23, we need some lemmas.

Lemma 6.24. *Let $D \in \Phi_k$ and let v be an unmarked vertex in D. Then $D * v \in \Phi_k$.*

Proof. We proceed by induction on k. We may assume that $k \geqslant 1$. By the construction, there exists a bag B of D such that the three limbs D_1, D_2, D_3 in D corresponding to the bag B are contained in Φ_{k-1}.

Let D_1', D_2', D_3' be the three limbs of $D * v$ corresponding to the bag B such that D_i' and D_i came from the same component of $D \backslash V(B)$. Then by Proposition 5.7, D_i' is locally equivalent to D_i. So by the induction hypothesis, $D_i' \in \Phi_{k-1}$. And $D * v$ is the canonical split decomposition obtained from D_i' following the construction of Φ_k. Therefore, $D * v \in \Phi_k$. □

Proof of Proposition 6.23. By Lemma 6.24 and Lemma 1.8, it is sufficient to show that if $D \in \Phi_k$ and v is an unmarked vertex of D, then $\widehat{D} \backslash v$ has linear rank-width at most k. We use induction on k to prove it. We may assume that $k \geqslant 1$. Let B be the bag of D such that $D \backslash V(B)$ has exactly three limbs whose underlying graphs are contained in Φ_{k-1}. Clearly there is no other bag having the same property. Since B has no unmarked vertices, v is contained in one of the limbs D', and by induction hypothesis, $\widehat{D'} \backslash v$ has linear rank-width at most $k - 1$. Therefore, by Theorem 5.11, $\widehat{D} \backslash v$ has linear rank-width at most k. □

One can observe that for two graphs other than C_5 in Figure 1.1 [1], their canonical split decompositions are contained in Φ_1. Also, all of the canonical split decompositions of graphs in Δ_k are contained in Φ_k for each $k \geqslant 1$.

We leave an open question to identify a set $\Phi_k \subset \Theta_k \subset \Psi_k$ that forms the set of canonical split decompositions of distance-hereditary vertex-minor minimal graphs for the class of graphs of linear rank-width k.

Chapter 7. Finding a tree as a vertex-minor

For the path-width of graphs, it is known that for a fixed forest T, every graph of sufficiently large path-width contains T as a minor [169, 13]. For the tree-width of graphs, it is known that for fixed n, every graph of sufficiently large tree-width contains an $n \times n$-grid graph as a minor [171].

Oum [150, 154] proved that for a fixed bipartite circle graph H, every bipartite (or line, or circle) graph of sufficiently large rank-width contains a vertex-minor isomorphic to H. We ask a similar question for linear rank-width.

Question 1.3. *For a fixed tree T, does every graph of sufficiently large linear rank-width contain a vertex-minor isomorphic to T?*

The author [133] showed that if we replace the containment relation with pivot-minor, then this question becomes false. We remark that there exists a class of trees of unbounded linear rank-width which do not have a pivot-minor isomorphic to a graph in Figure 7.1 [133]. However, we still have no counterexamples for Question 1.3.

We show that Question 1.3 is true if it is true for prime graphs. To support this statement, we show the following.

Theorem 7.1. *Let $p \geqslant 3$ be an integer and T be a tree. Let G be a graph such that every prime induced subgraph of G has linear rank-width at most p. If $\operatorname{lrw}(G) \geqslant 30(p+4)|V(T)|$, then G contains a vertex-minor isomorphic to T.*

We prove that for fixed p and a graph G whose prime induced subgraphs have linear rank-width at most p, if G has large linear rank-width, then its split decomposition tree has large path-width. We first establish a relation between the linear rank-width of a distance-hereditary graph and the path-width of its split decomposition tree. Next we observe a relation for graphs whose prime induced subgraphs have bounded linear rank-width. At the last moment, we show that for a fixed tree T, if a graph G admits a split decomposition tree of sufficiently large path-width, then G contains a vertex-minor isomorphic to T.

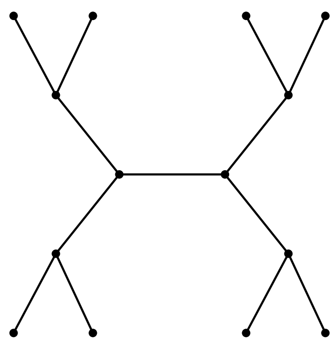


Figure 7.1: This tree is not a pivot-minor of a tree obtained from a tree by replacing each edge with a path of length 2 [133].

7.1 Path-width of split decomposition trees

We will observe a relation between the linear rank-width of a distance-hereditary graph and the path-width of its split decomposition tree. This analysis is natural because path-width on trees and linear rank-width on distance-hereditary graphs admit similar characterizations.

We prove the following.

Theorem 7.2. *Let D be the canonical split decomposition of a connected distance-hereditary graph G. Let T_D be the split decomposition tree of D. Then $\frac{1}{2}\,\mathrm{pw}(T_D) \leqslant \mathrm{lrw}(G) \leqslant \mathrm{pw}(T_D) + 1$.*

For example, every complete graph G with at least two vertices has linear rank-width 1 and the path-width of its split decomposition tree has path-width 0. For the set Δ_1 defined in Chapter 6 and $G \in \Delta_1$, G has linear rank-width 2 and the path-width of its split decomposition tree is 1. We recall the following lemmas on trees from [85].

Lemma 3.7 (Ellis, Sudborough and Turner [85]; Takahashi, Ueno and Kajitani [182])**.** *Let k be a positive integer and let T be a tree. Then $\mathrm{pw}(T) \leqslant k$ if and only if for each vertex v of T, at most two subtrees of $T\backslash v$ have path-width k and all the other subtrees of $T\backslash v$ have path-width at most $k-1$.*

Lemma 7.3 (Ellis, Sudborough and Turner [85])**.** *Let T be a tree. Let P be a path in T such that for each vertex v of P and each component T' of $T\backslash v$ having no vertices of P, $\mathrm{pw}(T') \leqslant k-1$. Then $\mathrm{pw}(T) \leqslant k$.*

Lemma 7.4 (Ellis, Sudborough and Turner [85])**.** *Let T be a tree such that for each $v \in V(T)$, at most two subtrees of $T\backslash v$ have path-width k and all the other subtrees of $T\backslash v$ have path-width at most $k-1$. Then T has a path P such that for each vertex v of P and a component T' of $T\backslash v$ having no vertices of P, $\mathrm{pw}(T') \leqslant k-1$.*

We first show the lower bound in Theorem 7.2.

Lemma 7.5. *Let G be a graph and let $uv \in E(G)$. Then $\mathrm{pw}(G) \leqslant \mathrm{pw}(G/uv) + 1$.*

Proof. Let w be the contracted vertex from the edge uv in G/uv, and let $(P, \mathcal{B})$ be a path-decomposition of G/uv having the minimum width. It is not hard to check that a new path-decomposition obtained by replacing w with u and v in each bag containing w is a path-decomposition of G. We conclude that $\mathrm{pw}(G) \leqslant \mathrm{pw}(G/uv) + 1$. □

Lemma 7.6. *Let G be a graph. Let u be a vertex of degree 2 in G such that v_1, v_2 are the neighbors of u in G and $v_1v_2 \notin E(G)$. Then $\mathrm{pw}(G) \leqslant \mathrm{pw}(G/uv_1/uv_2) + 1$.*

Proof. Let w be the contracted vertex from the two edges uv_1, uv_2 in $G/uv_1/uv_2$, and let $(P, \mathcal{B})$ be a path-decomposition of $G/uv_1/uv_2$ having the minimum width $\mathrm{pw}(G/uv_1/uv_2)$. Let $t := \mathrm{pw}(G/uv_1/uv_2)$. We may assume that every two consecutive bags are not equal.

We first obtain a path-decomposition $(P, \mathcal{B}')$ from $(P, \mathcal{B})$ by replacing w with v_1 and v_2 in all bags containing w. Since every consecutive two bags in $(P, \mathcal{B})$ are not equal, every consecutive two bags in $(P, \mathcal{B}')$ are not equal.

We first assume that there are two adjacent bags B_1 and B_2 in $(P, \mathcal{B}')$ containing both v_1 and v_2, respectively. We obtain a path-decomposition $(P', \mathcal{B}'')$ from $(P, \mathcal{B}')$ by subdividing between two bags B_1 and B_2 with adding a new bag $B' = (B_1 \cap B_2) \cup \{u\}$. Since B_1 and B_2 are not same, $|B_1 \cap B_2| \leqslant t + 1$

and therefore, $|B'| \leqslant t+2$. Thus, $(P', \mathcal{B}'')$ is a path-decomposition of G of width at most $t+1$, and $\mathrm{pw}(G) \leqslant \mathrm{pw}(G/uv_1/uv_2) + 1$.

Now we assume that there are only one bag B in $(P, \mathcal{B}')$ containing both v_1 and v_2. In this case, since $v_1v_2 \notin E(G)$, we can obtain a path decomposition of G by replacing this bag B with a sequence of two bags B_1 and B_2, where $B_1 = B\backslash\{v_2\} \cup \{u\}$ and $B_2 = B\backslash\{v_1\} \cup \{u\}$. This implies that $\mathrm{pw}(G) \leqslant \mathrm{pw}(G/uv_1/uv_2) + 1$. □

Let D be the canonical split decomposition of a connected distance-hereditary graph G.

Lemma 7.7. $\mathrm{pw}(T_D) \leqslant 2\,\mathrm{lrw}(G)$.

Proof. We prove by induction on $k := \mathrm{lrw}(G)$. If $k = 0$, then G is an one vertex graph, and $\mathrm{pw}(T_D) = 0$. If $k = 1$, then by Theorem 5.22, T_D is a path. Therefore, $\mathrm{pw}(T_D) = 0$ or 1, and we have $\mathrm{pw}(T_D) \leqslant 2k$. Thus, we may assume that $k \geqslant 2$.

Since $\mathrm{lrw}(G) = k \geqslant 2$, by Theorem 5.11 and Lemma 5.15, there exists a path $P = v_0v_1 \cdots v_nv_{n+1}$ in T_D such that for each node v in P and a component C of $D\backslash V(\mathsf{bag}_D(v))$ not containing a bag of P, $f(B, C) \leqslant k-1$. By induction hypothesis, for each corresponding limb L_C of linear rank-width at moat $k-1$, the split decomposition tree T_{L_C} of it has path-width at most $2k-2$. We compare the path-width of T_C (the split decomposition tree of C) and the path-width of T_{L_C}.

We claim that $\mathrm{pw}(T_C) \leqslant 2k-1$. As described in Section 5.2, when we take a canonical split decomposition from a limb, if there is a bag having 2 vertices, then we apply one of the following operations:

1. Removing a bag having exactly one adjacent bag,
2. Removing a bag having exactly two adjacent bags and linking the adjacent bags, or
3. Removing a bag having exactly two adjacent bags and merging the adjacent bags into one bag.

First two cases correspond to contracting at most one edge in view of subtrees of T_D. So, $\mathrm{pw}(T_C) \leqslant \mathrm{pw}(T_{L_C}) + 1 \leqslant (2k-2)+1 = 2k-1$ by Lemma 7.5. The last case corresponds to contracting two incident edges where the middle node has degree 2 and its neighbors are not adjacent. By Lemma 7.6, $\mathrm{pw}(T_C) \leqslant \mathrm{pw}(T_{L_C}) + 1 \leqslant 2k-1$.

Therefore, By Lemma 7.3, T_D has path-width at most $2k$, as required. □

Now, we prove the upper bound part.

Lemma 7.8. $\mathrm{lrw}(G) \leqslant \mathrm{pw}(T_D) + 1$.

Proof. We prove by induction on $k := \mathrm{pw}(T_D)$. If $k = 0$, then T_D consists of one node, $\mathrm{lrw}(G) = 0$ or 1. So, we have $\mathrm{lrw}(G) \leqslant \mathrm{pw}(T_D) + 1$. We assume that $k \geqslant 1$.

Since $\mathrm{pw}(T_D) = k$, by Lemma 3.7, for each vertex v of T_D, at most two subtrees of $T\backslash v$ have path-width k and all the other subtrees of $T\backslash v$ have path-width at most $k-1$. Also by Lemma 7.4, there exists a path $P = v_0v_1 \cdots v_nv_{n+1}$ in T_D such that for each node v in P and a component T of $T_D\backslash v$ not containing a node of P, $\mathrm{pw}(T) \leqslant k-1$. By induction hypothesis, the graph obtained from a corresponding canonical split decomposition has linear rank-width at most $(k-1)+1 = k$. From the definition of limbs, we have that for each node v in P and a component T of $D\backslash V(\mathsf{bag}_D(v))$ not containing a bag of P, $f(B, T) \leqslant k$. By Theorem 5.11, we conclude that $\mathrm{lrw}(G) \leqslant k+1$. □

Proof of Theorem 7.2. From Lemmas 7.7 and 7.8, we conclude that $\frac{1}{2}\operatorname{pw}(T_D) \leqslant \operatorname{lrw}(G) \leqslant \operatorname{pw}(T_D) + 1$. $\square$

We could not confirm that the lower bound of $\operatorname{lrw}(G)$ is tight. We remain the following as an open question.

Question 7.9. *Let D be the canonical split decomposition of a connected distance-hereditary graph G. Is it true that* $\operatorname{pw}(T_D) \leqslant \operatorname{lrw}(G)$*?*

Now we consider graphs whose prime induced subgraphs have bounded linear rank-width. Note that every prime induced subgraph of a distance-hereditary graph has at most 3 vertices [31]. Thus, the following can be regarded as a general version of Lemma 7.8.

Lemma 7.10. *Let $p \geqslant 3$ be an integer. Let G be a graph such that every prime induced subgraph of G has linear rank-width at most p, and let D be the canonical split decomposition of G. Then* $\operatorname{lrw}(G) \leqslant 2(p+4)(\operatorname{pw}(T_D)+1)$.

From the condition, it is easy to observe that every prime bag of D has bounded linear rank-width. A basic strategy to bound linear rank-width is to recursively replace an optimal linear layout of some bag with linear layouts of subdecompositions branched from this bag.

Proof. We prove by induction on $k := \operatorname{pw}(T_D)$. If $k = 0$, then T_D consists of one vertex, and by the assumption, $\operatorname{lrw}(G) \leqslant p \leqslant 2(p+4)$. We assume that $k \geqslant 1$.

We follow similar steps in the proof of Theorem 5.11. Since $\operatorname{pw}(T_D) = k$, by Lemma 3.7 and Lemma 7.4, there exists a path $P = v_0 v_1 \cdots v_n v_{n+1}$ in T_D such that for each node v in P and a component T of $T_D \backslash v$ not containing a node of P, $\operatorname{pw}(T) \leqslant k-1$. For each $0 \leqslant j \leqslant n+1$, let $B_i := \mathsf{bag}_D(v_i)$. For each $0 \leqslant i \leqslant n$, let b_i be a marked vertex of B_i and let a_{i+1} be a marked vertex B_{i+1} such that $b_i a_{i+1}$ is the marked edge connecting B_i and B_{i+1}. If necessary, by adding unmarked vertices on B_0 and B_{n+1} which are twins of one of the first or last vertex in the optimal linear layout, we may assume that B_0 and B_{n+1} have unmarked vertices a_0 and b_{n+1} in D, respectively, and the linear rank-width of B_0 and B_{n+1} are at most p.

In case when $n = -1$, let $D_0 := D$. If $n \geqslant 0$, then we define the following subdecompositions.

1. Let D_0 be the component of $D \backslash V(B_1)$ containing the bag B_0.
2. Let D_{n+1} be the component of $D \backslash V(B_n)$ containing the bag B_{n+1}.
3. For each $1 \leqslant i \leqslant n$, let D_i be the component of $D \backslash (V(B_{i-1}) \cup V(B_{i+1}))$ containing the bag B_i.

Notice that the vertices a_i and b_i are unmarked vertices in D_i

We claim that for each $i \in \{0, 1, \ldots, n+1\}$, $\widehat{D_i}$ has a linear layout of width at most $2(p+4)(k+1)$ whose first and last vertices are a_i and b_i, respectively. Let $i \in \{0, 1, \ldots, n+1\}$ and let D' be a component of $D_i \backslash V(B_i)$. Since $\operatorname{pw}(T_{D'}) \leqslant k-1$, by induction hypothesis, $\widehat{D'}$ has linear rank-width at most $2(p+4)k$, and in particular, the graph $\widehat{D'} \backslash \zeta_t(D, B_i, D')$ has linear rank-width at most $2(p+4)k$.

By the assumption the bag B_i has linear rank-width at most p. Since the rank of any matrix can be increased by at most 2 when we move one element in the column indices (or the row indices) to the row indices (or the column indices, respectively), B_i admits a linear layout of width at most $p+4$ whose first and last vertices are a_i and b_i, respectively. Let

$$L_{B_i} := (w_1, w_2, \ldots, w_m)$$

be a linear layout of B_i of width at most $p+4$ whose first and last vertices are a_i and b_i, respectively. For each $1 \leqslant j \leqslant m$,

1. if w_j is an unmarked vertex, then let $L(w_j) := (w_j)$, and
2. if w_j is a marked vertex adjacent to a vertex of a component of $D_i \backslash V(B_i)$, say D_{w_j}, then let $L(w_j)$ be the optimal linear layout of $\widehat{D_{w_j}} \backslash \zeta_t(D, B_i, D_{w_j})$.

Since each T_{w_j} has path-width at most $k-1$, the width of $L(w_j)$ is at most $2(p+4)k$. We define that

$$L_i := L(w_1) \oplus L(w_2) \cdots \oplus L(w_m).$$

We claim that L_i has width at most $2(p+4)(k+1)$. It is sufficient to prove that for every $w \in V(\widehat{D_i}) \backslash \{a_i, b_i\}$, $\rho_{\widehat{D_i}}(\{v : v \leqslant_{L_i} w\}) \leqslant 2(p+4)(k+1)$. For each $w \in V(\widehat{D_i}) \backslash \{a_i, b_i\}$, let $S_w := \{v : v \leqslant_{L_i} w\}$ and $T_w := V(\widehat{D_i}) \backslash S_w$. We fix $2 \leqslant j \leqslant m-1$. If w_j is an unmarked vertex of the bag B_i, then clearly,

$$\rho_{\widehat{D_i}}(S_{w_j}) = \rho_{B_i}(\{v : v \leqslant_{L_{B_i}} w_j\}) \leqslant p+4$$

by the assumption. Thus, we may assume that $w_j \notin V(B_i)$.

From the assumption we have the following.

1. $\rho^*_{\widehat{D_i}}(S_{w_j}, T_{w_j} \backslash V(\widehat{D_{w_j}})) \leqslant \min\{\rho_{B_i}(\{v : v \leqslant_{L_{B_i}} w_j\}), \rho_{B_i}(\{v : v \leqslant_{L_{B_i}} w_{j-1}\})\} \leqslant p+4$,
2. $\rho^*_{\widehat{D_i}}(S_{w_j} \backslash V(\widehat{D_{w_j}}), T_{w_j}) \leqslant \min\{\rho_{B_i}(\{v : v \leqslant_{L_{B_i}} w_j\}), \rho_{B_i}(\{v : v \leqslant_{L_{B_i}} w_{j-1}\})\} \leqslant p+4$,
3. $\rho^*_{\widehat{D_i}}(S_{w_j} \cap V(\widehat{D_{w_j}}), T_{w_j} \cap V(\widehat{D_{w_j}})) \leqslant 2(p+4)k$.

Therefore,

$$\begin{aligned}
&\rho_{\widehat{D_i}}(S_{w_j}) \\
&\leqslant \rho^*_{\widehat{D_i}}(S_{w_j}, T_{w_j} \backslash V(\widehat{D_{w_j}})) + \rho^*_{\widehat{D_i}}(S_{w_j} \backslash V(\widehat{D_{w_j}}), T_{w_j}) + \rho^*_{\widehat{D_i}}(S_{w_j} \cap V(\widehat{D_{w_j}}), T_{w_j} \cap V(\widehat{D_{w_j}})) \\
&\leqslant 2(p+4) + 2(p+4)k \leqslant 2(p+4)(k+1).
\end{aligned}$$

We show that $\widehat{D_i}$ has a linear layout L_i of width $2(p+4)(k+1)$ whose first and last vertices are a_i and b_i, respectively. For each i, let L'_i be the linear layout obtained from L_i by removing a_i and b_i. Then it is not hard to check that

$$(a_0) \oplus L'_0 \oplus \cdots \oplus L'_{n+1} \oplus (b_{n+1})$$

is a linear layout of G having width at most $2(p+4)(k+1)$. We conclude that $\operatorname{lrw}(G) \leqslant 2(p+4)(\operatorname{pw}(T_D)+1)$. □

7.2 Containing a tree as a vertex-minor

Now we prove Theorem 7.1 using Lemma 7.10. Let G be a graph where every prime induced subgraph has linear rank-width at most p. We first prove that if G has sufficiently large linear rank-width, then the split decomposition tree of the canonical split decomposition of G has large path-width, by applying Lemma 7.10. Then this tree must contain a subdivision of a complete binary tree as a subgraph by Theorem 3.13. We take a corresponding decomposition from the canonical split decomposition of G. The next step is to replace each prime bag with some canonical split decomposition that consists of only complete bags or star bags by removing some vertices and applying local complementations such that

- the new split decomposition tree is still a subdivision of the same complete binary tree.

In the last step, we modify to obtain a canonical split decomposition of trees. For this, we will use the Bouchet's characterization of trees in terms of canonical split decompositions in Theorem 5.2.

We first prove some lemmas to reduce general trees into subcubic trees. For a tree T, we denote by $\phi(T)$ the sum of the degrees of vertices of T whose degree is at least 4. Note that for every subcubic tree T, $\phi(T) = 0$.

Lemma 7.11. *Let k be a positive integer and let T be a tree with $\phi(T) = k$. Then T is a vertex-minor of a tree T' with $\phi(T') = k - 1$ and $|V(T')| = |V(T)| + 2$.*

Proof. Since $\phi(T) \geqslant 1$, T has a vertex of degree at least 4. Let $v \in V(T)$ be a vertex of degree at least 4, and $v_1, v_2, \ldots, v_m$ be the neighbors of v. We obtain T' from T by replacing v with a path vp_1p_2 and adding edges between v and $v_3, v_4, \ldots, v_m$, and between p_2 and v_1, v_2. It is easy to verify that $T' \wedge p_1p_2 \backslash p_1 \backslash p_2 = T$. Because p_1 and p_2 are vertices of degree at most 3 in T', and the degree of v in T' is one less than the degree of v in T, we have $\phi(T) = k - 1$. □

Lemma 7.12. *Let T be a tree. Then T is a vertex-minor of a subcubic tree T' with $|V(T')| \leqslant 5|V(T)|$.*

Proof. By Lemma 7.11, T is a vertex-minor of a subcubic tree T' with $|V(T')| \leqslant |V(T)| + 2\phi(T)$. Since $\phi(T) \leqslant 2|E(T)| \leqslant 2|V(T)|$, we conclude that $|V(T')| \leqslant |V(T)| + 2\phi(T) \leqslant 5|V(T)|$. □

For a tree T, let $\eta(T)$ be the tree obtained from T by replacing each edge with a path of length 3. We recall a characterization of trees in Theorem 5.2 that a connected graph is a tree if and only if each bag of its canonical split decomposition is a star bag whose center is an unmarked vertex.

We now prove two lemmas which tell how to modify the canonical split decompositions.

Lemma 7.13. *Let D be a split decomposition obtained from a path of bags $B_1B_2B_3B_4$ by attaching two bags B_5 and B_6 on B_4 such that each bag B_i consists of exactly three vertices, and B_1, B_2, B_3 are star bags whose centers are unmarked vertices. Let v_4, w_4 be the two marked vertices in B_4 adjacent to a vertex of B_5 and B_6, respectively. Then $\widehat{D}$ has a vertex-minor whose canonical split decomposition is D' where*

1. *$T_{D'}$ is a star whose center is B_4, and the leaves are B_1, B_5, B_6,*
2. *for $i \in \{1, 5, 6\}$, $B_i = D'[V(B_i)]$, and*
3. *$|V(B_4)| = 4$ and B_4 is a star bag whose center is an unmarked vertex other than v_4 and w_4.*

Proof. By applying local complementations at some vertices in B_5 or B_6, we may assume that B_4 is a star bag, and without loss of generality, we assume that v_4 is the center of B_4.

Let v_2 and v_3 be the unmarked vertices of B_2 and B_3, respectively. Consider $D \wedge v_2v_3 \backslash v_3$. The bag B_3 can be shrunk by recomposing in $D \wedge v_2v_3 \backslash v_3$ and the marked edge connecting B_2 and B_4 becomes a marked edge of type S_cS_p. By Theorem 4.5, it is not a canonical split decomposition, and by recomposing this marked edge of type S_cS_p, we obtain a canonical split decomposition where B_4 contains v_2 as a leaf which is unmarked. By pivoting v_2 with an unmarked vertex represented by v_4, v_2 becomes the center of B_4 as required. □

Lemma 7.14. *Let k be a positive integer and T be a subcubic tree. Let D be the canonical split decomposition of a connected distance-hereditary graph whose split decomposition tree T_D is isomorphic to a subdivision of $\eta(T)$ and each bag of D consists of exactly 3 vertices. Then D contains a vertex-minor D' where $T_{D'}$ is isomorphic to a subdivision of T and for each bag B of D',*

1. *B is a star bag,*
2. *if B is a leaf bag or a bag having 2 adjacent bags, then $|V(B)| = 3$,*
3. *if B is a bag having 3 adjacent bags, then $|V(B)| = 4$. (that is, the center of B is an unmarked vertex, and the other vertices of B are marked vertices.)*

Proof. We choose a leaf bag R in D and call it the root. By applying a local complementation if necessary, we may assume that R is a star bag whose center is an unmarked vertex. From the root to bottom, we do the following two procedures recursively to obtain the required decomposition D'. Let B be a bag that is not chosen before. Note that B has at most 3 adjacent bags. We may assume that the parent bag of B is a star bag whose center is an unmarked vertex.

Suppose that B is a bag having 2 adjacent bags. Let c be the unmarked vertex of B and let y be a vertex represented by a vertex of B that belongs to a descendant bag of B. Since the parent bag of B is already a star bag and whose center is an unmarked vertex, B is either a complete bag or a star bag where c is not linked to a vertex of the parent bag. If it is a complete bag, then we modify it into a star bag by applying local complementation at c. If it is a star bag, we pivot cy to turn this bag into a star bag whose center is c.

Now suppose that B is a bag having 3 adjacent bags. Since T_D is isomorphic to a subdivision of $\eta(T)$, there are at least 2 ancestor bags above B, which are already processed. Therefore, using Lemma 7.13, we can modify it into a star bag of size 4 by shrinking the two ancestor bags, where its center is an unmarked vertex.

If we do this procedure recursively, we finally obtain the canonical split decomposition satisfying the condition. □

Now we introduce lemmas which tells how to replace each prime bag with some canonical split decomposition that consists of complete bags or star bags.

Lemma 7.15. *Let D be the canonical split decomposition of a connected distance-hereditary graph and let B be a prime bag of D such that $D\backslash V(B)$ has two components T_1, T_2 and for each $i \in \{1,2\}$, the bag containing $\zeta_t(D,B,T_i)$ is either a star bag whose leaf is $\zeta_t(D,B,T_i)$ or a prime bag. Then by applying local complementations at vertices and deleting vertices in B, we can transform D into a canonical split decomposition D' such that*

1. *B is transformed to a complete bag of size 3 in D',*
2. *for every bag B' in D other than B, $D'[V(B)]$ is a bag of D', and*

3. *the types of adjacent bags of B in D' are the same as in D.*

Proof. For each $i \in \{1,2\}$, let $v_i := \zeta_b(D,B,T_i)$, and let B_i be the adjacent bag of B containing $\zeta_t(D,B,T_i)$. Since B is prime, there exists a path from v_1 to v_2 except a possible edge v_1v_2. We take a shortest one among those paths, and say P. Then by applying local complementations at internal vertices of P, we can shrink it into a path of length exactly 2, without changing the types of adjacent bags. Finally, by applying local complementation at the middle vertex of the path of length 2, we can create the edge v_1v_2 if not exists. Therefore, by removing all other vertices in B except the vertices of the path of length 2, we can turn B into a star bag of size 3 whose center is an unmarked vertex. Since each marked edge connecting B and B_i is of type KS_p or KP, it is not recomposable. ☐

Lemma 7.16. *Let D be the canonical split decomposition of a connected distance-hereditary graph and let B be a prime bag of D such that $D\backslash V(B)$ has one components T_1 and bag containing $\zeta_t(D,B,T_1)$ is either a star bag whose leaf is $\zeta_t(D,B,T_1)$ or a prime bag. Then by applying local complementations at vertices and deleting vertices in B, we can transform D into a canonical split decomposition D' such that*

1. *B is transformed to a complete bag of size 3 in D',*
2. *for every bag B' in D other than B, $D'[V(B)]$ is a bag of D', and*
3. *the types of adjacent bags of B in D' are the same as in D.*

Proof. We choose any vertex v_2 other than the vertex $\zeta_b(D,B,T_1)$. Then by the same argument in Lemma 7.15, we can turn B into a star bag of size 3 whose center is an unmarked vertex. ☐

Lemma 7.17. *Let D be the canonical split decomposition of a connected distance-hereditary graph and let B be a prime bag of D such that $D\backslash V(B)$ has three components T_1, T_2, T_3, and for each $i \in \{1,2,3\}$, the bag containing $\zeta_t(D,B,T_i)$ is either a star bag whose leaf is $\zeta_t(D,B,T_i)$ or a prime bag. Then by applying local complementations at vertices and deleting vertices in B, we can transform D into a canonical split decomposition D' such that*

1. *for every bag B' in D other than B, $D'[V(B)]$ is a bag of D', and*
2. *the types of adjacent bags of B in D' are the same as in D, and*
3. *B is transformed into a split decomposition D_B whose split decomposition tree is a star with at most 3 leaves where its center corresponds to a complete bag of size 3 and its leaves correspond to star bags of size 3 whose centers are unmarked vertices, and a leaf of a star bag or a vertex of a complete bag is adjacent to a vertex of one of T_i.*

Proof. For each $i \in \{1,2,3\}$, let $v_i := \zeta_b(D,B,T_i)$ and let B_i be the adjacent bag of B containing $\zeta_t(D,B,T_i)$. We take a minimal induced subgraph B' of B containing v_1, v_2 and v_3. It is not hard to observe that B' is one of the following:

1. There exist a vertex c in B' and three internally vertex-disjoint paths P_i from c to each v_i such that $V(B') = \bigcup_{1\leqslant i\leqslant 3} V(P_i)$ and $E(B') = \bigcup_{1\leqslant i\leqslant 3} E(P_i)$.
2. There exist a triangle $c_1c_2c_3$ in B' and three vertex-disjoint paths P_i from c_i to each v_i such that $V(B') = \bigcup_{1\leqslant i\leqslant 3} V(P_i)$ and $E(B') = (\bigcup_{1\leqslant i\leqslant 3} E(P_i)) \cup \{c_1c_2, c_2c_3, c_3c_1\}$.

In both cases, we may assume that each P_i has length at most 1, by applying local complementation at internal vertices of P_i and taking a shorter path. Note that we did not yet remove vertices and the modified bag is still prime. We analyze each case, separately.

Case 1. *There exist a vertex c in B' and three internally vertex-disjoint paths P_i from c to each v_i.*

If $c \notin \{v_1, v_2, v_3\}$, then B' is exactly isomorphic to $K_{1,3}$, and it is enough to remain this subgraph by removing all other vertices. Without loss of generality, we may assume that $c = v_1$. Note that v_2, v_3 are the neighbors of v_1 in B and $v_2v_3 \notin E(B)$. Since B is prime, it is 2-connected and there exists a path P' from v_2 to v_3 in $B \backslash v_1$. We assume that P' is a shortest path among those paths. By applying local complementations at internal vertices of P', we can create an edge between v_2 and v_3 so that $\{v_1, v_2, v_3\}$ forms a triangle. Then by removing all vertices except v_1, v_2, v_3, we can turn B into a complete bag. Since each marked edge connecting B and B_i is of type KS_p or KP, it is not recomposable.

Case 2. *There exist a triangle $c_1c_2c_3$ in B' and three vertex-disjoint paths P_i from c_i to each v_i.*

In this case, we just replace B with the canonical split decomposition of B'. Note that a split decomposition tree of B' is a star where the bag corresponding to the center of the star is a complete bag of size 3, say $q_1q_2q_3$, and for each $1 \leqslant i \leqslant 3$, the canonical split decomposition of B' possibly has at most one neighbor star bag Q_i of size 3 where t_i is adjacent to a leaf of Q_i, and

1. if Q_i exists, then v_i is the other leaf of Q_i,
2. if Q_i does not exist, then $t_i = v_i$,

So, the canonical split decomposition obtained by replacing B with the canonical split decomposition of B' is a required decomposition. □

Now we prove the main result.

Proof of Theorem 7.1. Let $t := |V(T)|$ and suppose that $\mathrm{lrw}(G) \geqslant 30(p+4)t$. By Lemma 7.12, there exists a subcubic tree T' such that T is a vertex-minor of T' and $|V(T')| \leqslant 5t$. Note that $|V(\eta(T'))| \leqslant 15t$.

Since $\mathrm{lrw}(G) \geqslant 30(p+4)t$, by Lemma 7.10, $\mathrm{pw}(T_D) \geqslant 15t - 1$. Since $|V(\eta(T'))| \leqslant 15t$, from Theorem 3.13, T_D contains a minor isomorphic to $\eta(T')$. Since the maximum degree of $\eta(T')$ is 3, T_D contains a subgraph isomorphic to a subdivision of $\eta(T')$.

Let D' be the subdecomposition of D whose split decomposition tree is isomorphic to the subdivision of $\eta(T')$. We first describe how to take a vertex-minor D'' of D' whose split decomposition tree is also isomorphic to a subdivision of $\eta(T')$, and $\widehat{D''}$ is distance-hereditary.

Suppose that B is a prime bag of D'. Note that B contains 1, 2 or 3 marked vertices because the split decomposition tree of D' is isomorphic to a subdivision of $\eta(T')$. Let B' be a adjacent bag of B, and let v be the marked vertex in B that is adjacent to a vertex of B'. If B' is a star bag, then by pivoting with some unmarked vertex in B and an unmarked vertex linked to it on the side of B', we can turn B' into a star bag whose leaf is adjacent to v. If B' is a complete bag, then by applying local complementation at the outside of B, we can turn B' into a star bag whose leaf is adjacent to v. This local complementation may affect some other adjacent bags of B, but for every adjacent bag that is already a star bag whose leaf is adjacent to B, is not changed by this local complementation. Therefore, by doing these procedures, we may assume that each adjacent bag of B is either a prime bag, or a star bag whose leaf is adjacent to a vertex of B.

Let T_1, T_2, T_3 be three components of $D' \backslash V(B)$. Since for each $i \in \{1,2,3\}$, the bag containing $\zeta_t(D', B, T_i)$ is either a star bag whose leaf is $\zeta_t(D', B, T_i)$ or a prime bag, by Lemmas 7.15, 7.16 and 7.17, we can transform D' into a canonical split decomposition D'' by applying local complementations at vertices and deleting vertices in B such that

1. for every bag B' in D' other than B, $D''[V(B)]$ is a bag of D'', and
2. the types of adjacent bags of B in D'' are the same as in D, and
3. B is transformed into a split decomposition D_B whose split decomposition tree is a star with at most 3 leaves where its center corresponds to a complete bag of size 3 and its leaves correspond to star bags of size 3 whose centers are unmarked vertices, and a leaf of a star bag or a vertex of a complete bag is adjacent to a vertex of one of T_i.

So, the new canonical split decomposition has a split decomposition tree isomorphic to a subdivision of $\eta(T')$, as required.

Now we prove that we can obtain a tree T from the canonical split decomposition D'' of a distance-hereditary graph whose split decomposition tree is isomorphic to a subdivision of $\eta(T')$. If a leaf bag has at least three unmarked vertices, by removing unmarked vertices, we may assume that every leaf bag contains exactly 3 vertices. If a bag having 2 adjacent bags contains at least two unmarked vertices, by removing unmarked vertices, we may assume that it contains exactly 3 vertices. If a bag having 3 adjacent bags contains a center x of a star that is an unmarked vertex, we apply local complementation at x and remove it. And, if a bag having 3 adjacent bags still has at least two unmarked vertices, by removing unmarked vertices, we may assume that it contains exactly 3 vertices. So, we may assume that every bag of D'' consists of exactly 3 vertices. Note that $T_{D''}$ is not changed.

Since $T_{D''}$ is isomorphic to a subdivision of $\eta(T')$ and each bag of D'' consists of exactly 3 vertices, by Lemma 7.14, D'' contains a vertex-minor D''' where $T_{D'''}$ is isomorphic to a subdivision of T' and for each bag B of D''',

1. B is a star bag,
2. if B is a leaf bag or a bag having 2 adjacent bags, then $|V(B)| = 3$,
3. if B is a bag having 3 adjacent bags, then $|V(B)| = 4$.

By Theorem 5.2, D''' is a split decomposition of a tree, and in fact, it is not hard to observe that $\widehat{D'''}$ has an induced subgraph isomorphic to T' by removing some unmarked vertices in leaf bags. Since T is a vertex-minor of T', we conclude that G contains a vertex-minor isomorphic to T. $\square$

Chapter 8. Unavoidable vertex-minors in large prime graphs

In this chapter, we investigate a Ramsey type theorem on prime graphs. We recall that prime graphs G have no vertex partitions (A, B) with $|A|, |B| \geqslant 2$ and $\rho^*_G(A) \leqslant 1$. The main theorem of this chapter is the following.

Theorem 8.1. *For every n, there is N such that every prime graph on at least N vertices has a vertex-minor isomorphic to C_n or $K_n \boxminus K_n$.*

Ramsey's theorem [161] states that for every n, there exists N such that every graph on at least N vertices contains an induced subgraph isomorphic to K_n or $\overline{K_n}$. There are several variants of Ramsey's theorem with some given connectivity conditions. For instance, if a given graph is connected, then we may expect structures more than $\overline{K_n}$, because $\overline{K_n}$ is not connected. It is known that every large connected graph contains an induced subgraph isomorphic to either K_n, $K_{1,n}$ or P_n [76]. We point out some variants of Ramsey's theorem in this direction.

- (folklore; see Diestel's book [76, Proposition 9.4.1])

  For every n, there exists N such that every *connected* graph on at least N vertices contains an induced subgraph isomorphic to K_n, $K_{1,n}$, or P_n.

- (folklore; see Diestel's book [76, Proposition 9.4.2])

  For every n, there exists N such that every 2-*connected* graph on at least N vertices contains a topological minor isomorphic to C_n or $K_{2,n}$.

- (Oporowski, Oxley, and Thomas [149])

  For every n, there exists N such that every 3-*connected* graph on at least N vertices contains a minor isomorphic to the wheel graph W_n on n vertices or $K_{3,n}$.

- (Ding, Chen [78])

  For every integer n, there exists N such that every *connected and co-connected* graph on at least N vertices contains an induced subgraph isomorphic to P_n, $K^s_{1,n}$ (the graph obtained from $K_{1,n}$ by subdividing one edge once), $K_{2,n}\backslash e$, or $K_{2,n}/e\backslash f\backslash g$ where $\{f, g\}$ is a matching in $K_{2,n}/e$, or one of their complements. A graph is *co-connected* if its complement graph is connected.

- (Chun, Ding, Oporowski, and Vertigan [53])

  For every integer $n \geqslant 5$, there exists N such that every internally 4-connected graph on at least N vertices contains a *parallel minor* isomorphic to K_n, $K'_{4,n}$ ($K_{4,n}$ with a complete graph on the vertices of degree n), TF_n (the n-partition triple fan with a complete graph on the vertices of degree n), D_n (the n-spoke double wheel), D'_n (the n-spoke double wheel with axle), M_n (the $(2n + 1)$-rung Mobius zigzag ladder), or Z_n (the $(2n)$-rung zigzag ladder).

These theorems commonly state that every sufficiently large graph having certain connectivity contains at least one graph in the list of *unavoidable* graphs by certain graph containment relation. Moreover in each theorem, the list of unavoidable graphs is *optimal* in the sense that each unavoidable graph in the list has the required connectivity, can be made arbitrary large, and does not contain other unavoidable graphs in the list.

Theorem 8.1 is a first non-trivial variant of Ramsey's theorem on the vertex-minor relation. The proof of Theorem 8.1 consists of the following steps.

1. We first prove that for each n, there exists N such that every prime graph having an induced path of length N contains a vertex-minor isomorphic to C_n. (In fact, we prove that $N = \lceil 6.75n^7 \rceil$.)

2. Secondly, we prove that for each n, there exists N such that every prime graph on at least N vertices contains a vertex-minor isomorphic to P_n or $K_n \boxminus K_n$.

To prove (1), we actually prove first that every sufficiently large generalized ladder, a certain type of outerplanar graphs, contains C_n as a vertex-minor. This will be shown in Section 8.3. Then, we use the technique of blocking sequences developed by Geelen [102] to construct a large generalized ladder in a prime graph having a sufficiently long induced path, shown in Section 8.4. Blocking sequences will be discussed and developed in Section 2.1. The second part (2) is discussed in Section 8.5, where we iteratively use Ramsey's theorem to find a bigger configuration called a broom inside a graph.

We write $R(n_1, n_2, \ldots, n_k)$ to denote the minimum number N such that in every k coloring of the edges of K_N, there exist i and a clique of size n_i whose edges are all colored with the i-th color. Such a number exists by Ramsey's theorem [161].

We introduce three classes of graphs which are frequently used in this chapter.

Constructions of graphs

For two graphs G and H on the same set of n vertices, we would like to introduce operations to construct graphs on $2n$ vertices by making the disjoint union of them and adding some edges between two graphs. Roughly speaking, $G \boxminus H$ will add a perfect matching, $G \boxtimes H$ will add the complement of a perfect matching, and $G \boxslash H$ will add a bipartite chain graph. Formally, for two graphs G and H on $\{v_1, v_2, \ldots, v_n\}$, let $G \boxminus H$, $G \boxtimes H$, $G \boxslash H$ be graphs on $\{v_1^1, v_2^1, \ldots, v_n^1, v_1^2, v_2^2, \ldots, v_n^2\}$ such that for all $i, j \in \{1, 2, \ldots, n\}$,

1. $v_i^1 v_j^1 \in E(G \boxminus H)$ if and only if $v_i v_j \in E(G)$,
2. $v_i^2 v_j^2 \in E(G \boxminus H)$ if and only if $v_i v_j \in E(H)$,
3. $v_i^1 v_j^2 \in E(G \boxminus H)$ if and only if $i = j$,
4. $v_i^1 v_j^1 \in E(G \boxtimes H)$ if and only if $v_i v_j \in E(G)$,
5. $v_i^2 v_j^2 \in E(G \boxtimes H)$ if and only if $v_i v_j \in E(H)$,
6. $v_i^1 v_j^2 \in E(G \boxtimes H)$ if and only if $i \neq j$,
7. $v_i^1 v_j^1 \in E(G \boxslash H)$ if and only if $v_i v_j \in E(G)$,
8. $v_i^2 v_j^2 \in E(G \boxslash H)$ if and only if $v_i v_j \in E(H)$,
9. $v_i^1 v_j^2 \in E(G \boxslash H)$ if and only if $i \geqslant j$.

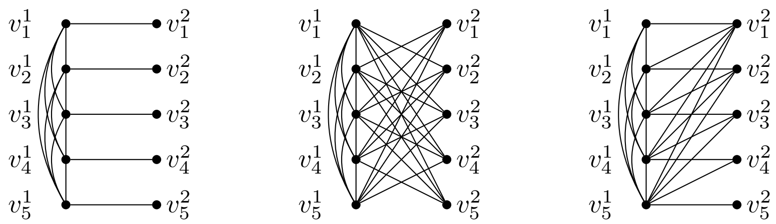


Figure 8.1: $K_5 \boxminus \overline{K_5}$, $K_5 \boxtimes \overline{K_5}$, and $K_5 \boxslash \overline{K_5}$.

See Figure 8.1 for $K_5 \boxminus \overline{K_5}$, $K_5 \boxtimes \overline{K_5}$, and $K_5 \boxslash \overline{K_5}$.

We will use the following lemmas.

Lemma 8.2. *Let $n \geqslant 3$ be an integer.*

1. *$K_n \boxtimes \overline{K_n}$ has a vertex-minor isomorphic to $K_{n-1} \boxminus K_{n-1}$.*

2. *$\overline{K_n} \boxtimes \overline{K_n}$ has a vertex-minor isomorphic to $K_{n-2} \boxminus K_{n-2}$.*

Proof. (1) Let $V(K_n) = V(\overline{K_n}) = \{v_i : 1 \leqslant i \leqslant n\}$. The graph $(K_n \boxtimes \overline{K_n}) * v_1^1 * v_1^2 \backslash v_1^1 \backslash v_1^2$ is isomorphic to $K_{n-1} \boxminus K_{n-1}$.

(2) Let $V(\overline{K_n}) = \{v_1, v_2, \ldots, v_n\}$. The graph $(\overline{K_n} \boxtimes \overline{K_n}) * v_1^1 \backslash v_1^1 \backslash v_1^2$ is isomorphic to $\overline{K_{n-1}} \boxminus K_{n-1}$. By (1), $\overline{K_n} \boxtimes \overline{K_n}$ has a vertex-minor isomorphic to $K_{n-2} \boxminus K_{n-2}$. $\square$

Lemma 8.3. *Let n be a positive integer.*

1. *The graph $\overline{K_n} \boxslash \overline{K_n}$ is pivot-equivalent to P_{2n}.*

2. *The graph $K_n \boxslash \overline{K_n}$ is locally equivalent to P_{2n}.*

Proof. (1) Let $P = p_1 p_2 \ldots p_{2n}$. We can check that $\overline{K_n} \boxslash \overline{K_n}$ can be obtained from P by pivoting $p_i p_{i+1}$ for all $i = 1, 3, \ldots, 2n-1$.

(2) Let $V(K_n) = V(\overline{K_n}) = \{v_1, v_2, \ldots, v_n\}$. Since $(K_n \boxslash \overline{K_n}) * v_1^2$ is isomorphic to $\overline{K_n} \boxslash \overline{K_n}$, the result follows from (1). $\square$

Before going to prove Theorem 8.1, we first observe similar theorems of this type on vertex-minors with respect to less restrictive connectivity requirements in Section 8.1.

8.1 Ramsey type theorems on vertex-minors with less connectivity

We present three simple statements on unavoidable vertex-minors.

Theorem 8.4.
1. *For every n, there exists N such that every graph on at least N vertices has a vertex-minor isomorphic to $\overline{K_n}$.*

2. *For every n, there exists N such that every connected graph having at least N vertices has a vertex-minor isomorphic to K_n.*

3. *For every n, there exists N such that every graph having at least N edges has a vertex-minor isomorphic to K_n or $\overline{K_n} \boxminus \overline{K_n}$.*

Proof. (1) If a graph has no $\overline{K_n}$ as a vertex-minor, then it has no vertex-minor isomorphic to K_{n+1}. So we can take $N = R(n, n+1)$.

(2) Let us assume that G has no vertex-minor isomorphic to K_n. Then the maximum degree of G is less than $\Delta = R(n-1, n-1)$ by Ramsey theorem. If $|V(G)|$ is big enough, then it contains an induced path P of length $2n-3$ because the maximum degree is bounded. By Lemma 8.3, P_{2n-2} has a vertex-minor isomorphic to $K_{1,n-1}$, that is locally equivalent to K_n.

(3) Let G be a graph having no vertex-minor isomorphic to K_n or $\overline{K_n} \boxminus \overline{K_n}$. Each component of G has bounded number of vertices, say M, by (2). Since $\overline{K_n} \boxminus \overline{K_n}$ is not a vertex-minor of G, G has less than n non-trivial components. (A component is trivial if it has no edges.) So G has at most $\binom{M}{2}(n-1)$ edges. $\square$

These theorems together with Theorem 8.1 can be restated with the concept of vertex-minor ideals. A set I of graphs is called a *vertex-minor ideal* if for all $G \in I$, all graphs isomorphic to a vertex-minor of G are also contained in I. This formulation allows us to appreciate why these theorems are optimal.

Corollary 8.5. *Let I be a vertex-minor ideal.*

(1) (Theorem 8.4) Graphs in I have bounded number of vertices if and only if $\{\overline{K_n} : n \geqslant 3\} \nsubseteq I$.

(2) (Theorem 8.4) Connected graphs in I have bounded number of vertices if and only if $\{K_n : n \geqslant 3\} \nsubseteq I$.

(3) (Theorem 8.4) Graphs in I have bounded number of edges if and only if $\{K_n : n \geqslant 3\} \nsubseteq I$ and $\{\overline{K_n} \boxminus \overline{K_n} : n \geqslant 1\} \nsubseteq I$.

(4) (Theorem 8.1) Prime graphs in I have bounded number of vertices if and only if $\{C_n : n \geqslant 3\} \nsubseteq I$ and $\{K_n \boxminus K_n : n \geqslant 3\} \nsubseteq I$.

8.2 Short blocking sequences

We will use blocking sequences to investigate vertex-minors. We first recall the definition of blocking sequences, introduced by Geelen [102].

A sequence $v_1, v_2, \ldots, v_m$ ($m \geqslant 1$) is called a *blocking sequence* of a pair (A, B) of disjoint subsets A, B of $V(G)$ if

(a) $\rho^*_G(A, B \cup \{v_1\}) > \rho^*_G(A, B)$,

(b) $\rho^*_G(A \cup \{v_i\}, B \cup \{v_{i+1}\}) > \rho^*_G(A, B)$ for all $i = 1, 2, \ldots, m-1$,

(c) $\rho^*_G(A \cup \{v_m\}, B) > \rho^*_G(A, B)$,

(d) no proper subsequence of $v_1, \ldots, v_m$ satisfies (a), (b), and (c).

The following proposition allows us to change the graph to reduce the length of a blocking sequence. This was pointed out by Geelen [private communication with Oum, 2005]. A special case of the following proposition is presented in [153].

Proposition 8.6. *Let G be a graph and A, B be disjoint subsets of $V(G)$. Let $v_1, v_2, \ldots, v_m$ be a blocking sequence for (A, B) in G. Let $1 \leqslant i \leqslant m$.*

- *If $m > 1$, then $\rho^*_{G*v_i}(A, B) = \rho^*_G(A, B)$ and a sequence*

$$v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_m$$

obtained by removing v_i from the blocking sequence is a blocking sequence for (A, B) in $G * v_i$.

- *If $m = 1$, then $\rho^*_{G*v_i}(A, B) = \rho^*_G(A, B) + 1$.*

Proof. Let $k = \rho^*_G(A, B)$ and $H = G * v_i$.

If $m = 1$, then by Lemma 2.4,

$$\rho^*_H(A, B) + \rho^*_G(A, B) \geqslant \rho^*_G(A \cup \{v_1\}, B) + \rho^*_G(A, B \cup \{v_1\}) - 1 \geqslant 2k + 1$$

and therefore $\rho^*_H(A, B) \geqslant k + 1$. Since $\rho^*_H(A, B) \leqslant \rho^*_H(A, B \cup \{v_1\}) = \rho^*_G(A, B \cup \{v_1\}) \leqslant k + 1$, we deduce that $\rho^*_H(A, B) = k + 1$ if $m = 1$.

Now we assume that $m \neq 1$. First it is easy to observe that $\rho^*_H(X, Y) \leqslant \rho^*_G(X, Y \cup \{v_i\})$ and $\rho^*_H(X, Y) \leqslant \rho^*_G(X \cup \{v_i\}, Y)$ whenever X, Y are disjoint subsets of $V(G)\backslash\{v_i\}$, because the local complementation does not change the cut-rank function of $G[X \cup Y \cup \{v_i\}]$. This with Lemma 2.5 implies that

- $\rho^*_H(A, B) \leqslant k$,
- $\rho^*_H(A \cup \{v_j\}, B) \leqslant k$ for all $j \in \{1, 2, \ldots, m\}\backslash\{i - 1, m\}$,
- $\rho^*_H(A \cup \{v_{i-1}\}, B) \leqslant k$ if $i \neq 1, m$.
- $\rho^*_H(A, B \cup \{v_j\}) \leqslant k$ for all $j \in \{1, 2, \ldots, m\}\backslash\{1, i + 1\}$.
- $\rho^*_H(A, B \cup \{v_{i+1}\}) \leqslant k$ if $i \neq 1, m$.
- $\rho^*_H(A \cup \{v_j\}, B \cup \{v_\ell\}) \leqslant k$ for all $j, \ell \in \{1, 2, \ldots, m\}\backslash\{i\}$ with $\ell - j > 1$, unless $j + 1 = i = \ell - 1$.

Let $B' = B \cup \{v_{i+1}\}$ if $i < m$ and $B' = B$ otherwise. Then $\rho^*_G(A \cup \{v_i\}, B') = k + 1$ and $\rho^*_G(A, B') = k$.

(1) We claim that if $i > 1$, then $\rho^*_H(A, B \cup \{v_1\}) > k$. By Lemma 2.4,

$$\rho^*_H(A, B' \cup \{v_1\}) + \rho^*_G(A, B') \geqslant \rho^*_G(A, B' \cup \{v_1, v_i\}) + \rho^*_G(A \cup \{v_i\}, B') - 1,$$

and therefore we deduce that $\rho^*_H(A, B' \cup \{v_1\}) \geqslant \rho^*_G(A, B' \cup \{v_1, v_i\}) > k$. By Lemma 2.2, $\rho^*_H(A, B' \cup \{v_i\}) + \rho^*_H(A, B \cup \{v_1\}) \geqslant \rho^*_H(A, B' \cup \{v_1, v_i\}) + \rho^*_H(A, B) > 2k$. We deduce that $\rho^*_H(A, B \cup \{v_1\}) > k$ because $\rho^*_H(A, B' \cup \{v_i\}) = \rho^*_G(A, B' \cup \{v_i\}) = k$ by Lemma 2.5.

(2) By (1) and symmetry between A and B, if $i < m$, then $\rho^*_H(A \cup \{v_m\}, B) > k$.

Then we deduce that $\rho^*_H(A, B) \geqslant k$ and therefore $\rho^*_H(A, B) = k$.

(3) We claim that if $j < i - 1$, then $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) > k$. By Lemma 2.4,

$$\begin{aligned}&\rho^*_H(A \cup \{v_j\}, B' \cup \{v_{j+1}\}) + \rho^*_G(A \cup \{v_j\}, B')\\ \geqslant&\rho^*_G(A \cup \{v_j\}, B' \cup \{v_{j+1}, v_i\}) + \rho^*_G(A \cup \{v_j, v_i\}, B') - 1 > 2k,\end{aligned}$$

and therefore $\rho^*_H(A \cup \{v_j\}, B' \cup \{v_{j+1}\}) > k$. By Lemma 2.2, $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) + \rho^*_H(A \cup \{v_j\}, B') \geqslant \rho^*_H(A \cup \{v_j\}, B' \cup \{v_{j+1}\}) + \rho^*_H(A \cup \{v_j\}, B) > 2k$. Note that $\rho^*_H(A \cup \{v_j\}, B) \geqslant \rho^*_H(A, B) = k$. Since $\rho^*_H(A \cup \{v_j\}, B') \leqslant \rho^*_H(A \cup \{v_j\}, B' \cup \{v_i\}) = \rho^*_G(A \cup \{v_j\}, B' \cup \{v_i\}) \leqslant k$, we deduce that $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) > k$.

(4) By symmetry, we deduce from (3) that if $i < j < m$, then $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) > k$.

(5) We claim that $\rho^*_H(A \cup \{v_{i-1}\}, B') > k$. By Lemma 2.4,

$$\begin{aligned}&\rho^*_H(A \cup \{v_{i-1}\}, B') + \rho^*_G(A \cup \{v_{i-1}\}, B')\\ &\geqslant \rho^*_G(A \cup \{v_{i-1}\}, B' \cup \{v_i\}) + \rho^*_G(A \cup \{v_{i-1}, v_i\}, B') - 1 > 2k.\end{aligned}$$

Since $\rho^*_G(A \cup \{v_{i-1}\}, B') = k$, we have $\rho^*_H(A \cup \{v_{i-1}\}, B') > k$.

This completes the proof of the lemma that $v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_m$ is a blocking sequence of (A, B) in $G * v_i$. □

Corollary 8.7. *Let G be a graph and A, B be disjoint subsets of $V(G)$. Let $v_1, v_2, \ldots, v_m$ be a blocking sequence for (A, B) in G. Let $1 \leqslant i \leqslant m$. Suppose that v_i has a neighbor w in $A \cup B$.*

- *If $m > 1$, then $\rho^*_{G \wedge v_i w}(A, B) = \rho^*_G(A, B)$ and the sequence $v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_m$ obtained by removing v_i from the blocking sequence is a blocking sequence for (A, B) in $G \wedge v_i w$.*
- *If $m = 1$, then $\rho^*_{G \wedge v_i w}(A, B) = \rho^*_G(A, B) + 1$.*

Proof. It follows easily from the facts that $G \wedge v_i w = G * w * v_i * w$ and $\rho^*_G(X, Y) = \rho^*_{G*w}(X, Y)$ for all graphs G with $w \in X \cup Y$. □

Corollary 8.8. *Let G be a graph and A, B be disjoint subsets of $V(G)$. Let $v_1, v_2, \ldots, v_m$ be a blocking sequence for (A, B) in G. Let $1 \leqslant i \leqslant m$. Suppose that v_i and $v_{i'}$ are adjacent and $i < i'$.*

- *If $m > 2$, then $\rho^*_{G \wedge v_i v_{i'}}(A, B) = \rho^*_G(A, B)$ and the sequence $v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i'-1}, v_{i'+1}, \ldots, v_m$ obtained by removing v_i and $v_{i'}$ from the blocking sequence is a blocking sequence for (A, B) in $G \wedge v_i v_{i'}$.*
- *If $m = 2$, then $\rho^*_{G \wedge v_i v_{i'}}(A, B) = \rho^*_G(A, B) + 1$.*

Proof. If v_i has a neighbor w in $A \cup B$, then $G \wedge v_i v_{i'} = G \wedge v_i w \wedge w v_{i'}$ and this corollary follows from Corollary 8.7. So we may assume that v_i has no neighbors in $A \cup B$ and similarly $v_{i'}$ has no neighbors in $A \cup B$. Thus $i, i' \notin \{1, m\}$ and $m \geqslant 4$.

Since v_i and $v_{i'}$ are adjacent, we may assume that $i' = i + 1$. Let $H = G \wedge v_i v_{i+1}$ and $k = \rho^*_G(A, B)$. Since v_i and v_{i+1} have no neighbors in $A \cup B$, $\rho^*_H(A, B) = k$.

Then $v_1, v_2, \ldots, v_i$ is a blocking sequence for $(A, B \cup \{v_{i+1}\})$ in G by Lemma 2.5. Similarly $v_{i+1}, v_{i+2}, \ldots, v_m$ is a blocking sequence for $(A \cup \{v_i\}, B)$ in G.

By Corollary 8.7, $v_1, v_2, \ldots, v_{i-1}$ is a blocking sequence for $(A, B \cup \{v_{i+1}\})$ in H. Then $\rho^*_H(A, B \cup \{v_1\}) = \rho^*_H(A, B \cup \{v_1, v_{i+1}\}) > k$, because v_{i+1} has no neighbors of H in A.

For $1 \leqslant j < i - 1$, $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) + \rho^*_H(A \cup \{v_j\}, B \cup \{v_{i+1}\}) \geqslant \rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}, v_{i+1}\}) + \rho^*_H(A \cup \{v_j\}, B) > 2k$ and therefore

$$\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) > k$$

because $\rho^*_H(A \cup \{v_j\}, B) \leqslant \rho^*_H(A \cup \{v_j\}, B \cup \{v_{i+1}\}) \leqslant k$.

Similarly $v_{i+2}, v_{i+3}, \ldots, v_m$ is a blocking sequence for $(A \cup \{v_i\}, B)$ in H. By symmetry, we deduce that $\rho^*_H(A \cup \{v_m\}, B) > k$ and $\rho^*_H(A \cup \{v_j\}, B \cup \{v_{j+1}\}) > k$ for all $i + 1 < j < m$.

We now claim that $\rho^*_H(A \cup \{v_{i-1}\}, B \cup \{v_{i+2}\}) > k$. By Lemma 2.2,

$$\begin{aligned}&\rho^*_H(A \cup \{v_{i-1}\}, B \cup \{v_{i+2}\}) + \rho^*_H(A \cup \{v_{i+1}\}, B \cup \{v_{i+2}\})\\ &\qquad \geqslant \rho^*_H(A \cup \{v_{i-1}, v_{i+1}\}, B \cup \{v_{i+2}\}) + \rho^*_H(A, B \cup \{v_{i+2}\}).\end{aligned}$$

Since v_{i+1} has no neighbors in $A \cup B$, we have $\rho^*_H(A \cup \{v_{i+1}\}, B \cup \{v_{i+2}\}) = \rho^*_G(A \cup \{v_i\}, B \cup \{v_{i+2}\}) = k$ and $\rho^*_H(A, B \cup \{v_{i+2}\}) = \rho^*_G(A, B \cup \{v_{i+2}\}) = k$. Therefore

$$\rho^*_H(A \cup \{v_{i-1}\}, B \cup \{v_{i+2}\}) \geqslant \rho^*_H(A \cup \{v_{i-1}, v_{i+1}\}, B \cup \{v_{i+2}\}).$$

By Lemma 2.4,

$$\begin{aligned}&\rho^*_H(A \cup \{v_{i-1}, v_{i+1}\}, B \cup \{v_{i+2}\}) + \rho^*_G(A \cup \{v_{i-1}\}, B \cup \{v_{i+1}, v_{i+2}\})\\&\geqslant \rho^*_G(A \cup \{v_{i-1}\}, B \cup \{v_i, v_{i+1}, v_{i+2}\}) + \rho^*_G(A \cup \{v_{i-1}, v_i, v_{i+1}\}, B \cup \{v_{i+2}\}) - 1.\end{aligned}$$

By Lemma 2.5, $\rho^*_G(A \cup \{v_{i-1}, v_i, v_{i+1}\}, B \cup \{v_{i+2}\}) > k$ and $\rho^*_G(A \cup \{v_{i-1}\}, B \cup \{v_{i+1}, v_{i+2}\}) = k$. Therefore $\rho^*_H(A \cup \{v_{i-1}\}, B \cup \{v_{i+2}\}) \geqslant \rho^*_H(A \cup \{v_{i-1}, v_{i+1}\}, B \cup \{v_{i+2}\}) \geqslant \rho^*_G(A \cup \{v_{i-1}\}, B \cup \{v_i, v_{i+1}, v_{i+2}\}) > k$. This proves the claim.

So far we have shown that the sequence $v_1, v_2, \ldots, v_{i-1}, v_{i+2}, \ldots, v_m$ satisfies (a), (b), (c) of the definition of blocking sequences. It remains to show (d). For $j \in \{2, 3, \ldots, m\} \backslash \{i, i+1\}$, $\rho^*_H(A, B \cup \{v_j\}) = \rho^*_G(A, B \cup \{v_j\}) = k$ because v_i and v_{i+1} have no neighbors in $A \cup B$. Similarly $\rho^*_H(A \cup \{v_j\}, B) = \rho^*_G(A \cup \{v_j\}, B) = k$ for $j \in \{1, 2, \ldots, m-1\} \backslash \{i, i+1\}$. For $j, \ell \in \{1, 2, \ldots, m\} \backslash \{i, i+1\}$ with $\ell - j > 1$, either $\rho^*_G(A \cup \{v_j\}, B \cup \{v_\ell, v_i, v_{i+1}\}) = k$ or $\rho^*_G(A \cup \{v_j, v_i, v_{i+1}\}, B \cup \{v_\ell\}) = k$ and therefore $\rho^*_H(A \cup \{v_j\}, B \cup \{v_\ell\}) \leqslant k$, unless $j = i-1$ and $\ell = i+2$. This completes the proof. □

We will now prove that without loss of generality, a blocking sequence for (A, B) is short by applying local complementation while keeping the subgraph induced on $A \cup B$.

Proposition 8.9. *Let G be a prime graph and let A, B be disjoint subsets of $V(G)$ with $|A|, |B| \geqslant 2$. Suppose that there exist two nonempty sets $A_0 \subseteq A$ and $B_0 \subseteq B$ such that the set of all edges between A and B is $\{xy : x \in A_0, y \in B_0\}$. Let*

$$\ell_0 = \begin{cases} 3 & \text{if } |A_0| = |B_0| = 1,\\ 4 & \text{if } |A_0| = 1 \text{ or } |B_0| = 1,\\ 6 & \text{otherwise.}\end{cases}$$

Then there exists a graph G' locally equivalent to G satisfying the following.

1. *$G[A \cup B] = G'[A \cup B]$.*
2. *G' has a blocking sequence $b_1, b_2, \ldots, b_\ell$ of length at most ℓ_0 for (A, B).*

Proof. Since G is prime, G has a blocking sequence for (A, B) by Proposition 2.6. Let $\mathcal{G}$ be the set of all graphs G' locally equivalent to G such that $G'[A \cup B] = G[A \cup B]$. We assume that G is chosen in $\mathcal{G}$ so that the length ℓ of a blocking sequence $b_1, b_2, \ldots, b_\ell$ for (A, B) is minimized.

For $1 \leqslant i < \ell$, $N_G(b_i) \cap B = B_0$ or $\varnothing$ because $\rho_G(A \cup \{b_i\}, B) = 1$. For $1 < i \leqslant \ell$, $N_G(b_i) \cap A = A_0$ or $\varnothing$ because $\rho_G(A, B \cup \{b_i\}) = 1$.

Suppose that $N_G(b_i) \cap (A \cup B) = N_G(b_j) \cap (A \cup B)$ for some $1 < i < j < \ell$. If b_i and b_j are adjacent, then $G' = G \wedge b_i b_j \in \mathcal{G}$. If b_i and b_j are non-adjacent, then $G' = G * b_i * b_j \in \mathcal{G}$. In both cases, we found a graph in $\mathcal{G}$ having a shorter blocking sequence by Proposition 8.6 or Corollary 8.8, contradicting our assumption.

If $|B_0| = 1$, then for all $1 < i < \ell$, $N_G(b_i) \cap A = A_0$ because otherwise $G * b_i \in \mathcal{G}$ has a shorter blocking sequence by Proposition 8.6, contradicting our assumption. Similarly if $|A_0| = 1$, then $N_G(b_i) \cap B = B_0$ for all $1 < i < \ell$.

By the pigeonhole principle, we deduce that $\ell \leqslant \ell_0$. □

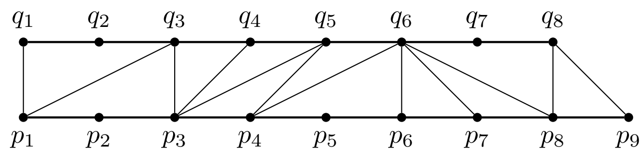


Figure 8.2: An example of a generalized ladder.

8.3 Obtaining a long cycle from a huge generalized ladder

A *generalized ladder* is a graph G with two vertex-disjoint paths $P = p_1p_2 \ldots p_a$, $Q = q_1q_2 \ldots q_b$ $(a, b \geqslant 1)$ with additional edges, called *chords*, each joining a vertex of P with a vertex of Q such that $V(P) \cup V(Q) = V(G)$, p_1 is adjacent to q_1, p_a is adjacent to q_b, and no two chords cross. Two chords p_iq_j and $p_{i'}q_{j'}$ $(i < i')$ *cross* if and only if $j > j'$. We remark that a generalized ladder is an outerplanar graph whose weak dual is a path. We call p_1q_1 the *first chord* and p_aq_b the *last chord* of G. Since no two chords cross, p_1 or q_1 has degree at most 2. Similarly, p_a or q_b has degree at most 2. See Figure 8.2 for an example.

We will prove the following proposition.

Proposition 8.10. *Let $n \geqslant 2$. Every generalized ladder with at least $4608n^5$ vertices has a cycle of length $4n + 3$ as a vertex-minor.*

8.3.1 Lemmas on a fan

Let F_n be a graph on n vertices with a specified vertex c, called the center, such that $F_n \backslash c$ is a path on $n - 1$ vertices and c is adjacent to all other vertices. We call F_n a *fan* on n vertices. Note that F_5 is the gem graph.

Lemma 8.11. *A fan F_{3n} has a vertex-minor isomorphic to a cycle of length $2n + 1$.*

Proof. Let c be the center of F_{3n}. Let $v_1, v_2, \ldots, v_{3n-1}$ be the non-center vertices in F_{3n} forming a path. Let $G = F_{3n} * v_3 * v_6 * v_9 \cdots * v_{3n-3}$. Clearly c is adjacent to v_i in G if and only if $i \in \{1, 3n-1\}$ or $i \equiv 0 \pmod 3$ and furthermore v_{3i-1} is adjacent to v_{3i+1} in G for all i. Let $H = G \backslash \{v_3, v_6, \ldots, v_{3n-3}\}$. Then H is a cycle of length $3n - (n - 1)$. $\square$

Lemma 8.12. *Let $n \geqslant 2$. Let G be a graph with a vertex c such that $G \backslash c$ is isomorphic to an induced path P whose both ends are adjacent to c. If $|V(G)| \geqslant 6(n-1)^2 - 3$, then G has a vertex-minor isomorphic to a cycle of length $2n + 1$.*

Proof. We may assume that $n \geqslant 3$. Let $P = v_1v_2 \ldots v_k$ with $k \geqslant 6$. We may assume that v_2 is adjacent to c because otherwise we replace G with $G * v_1$. Similarly we may assume that v_{k-1} is adjacent to c. We may also assume v_3 is adjacent to c because otherwise we replace G with $G \wedge v_1v_2$. Similarly we may assume that v_{k-2} is adjacent to c.

If c is adjacent to at least $3n - 1$ vertices on P, then G has a vertex-minor isomorphic to F_{3n}. So by Lemma 8.11, G has a vertex-minor isomorphic to a cycle of length $2n + 1$. Thus we may assume that the number of neighbors of c is at most $3n - 2$. The neighbors of c gives a partition of P into at most $3n - 3$ subpaths. We already have 4 subpaths at both ends having length 1. Since

$$|E(P)| \geqslant 6(n-1)^2 - 3 - 2 > (2n-2)((3n-3) - 4) + 4,$$

there exists a subpath P' of P having length at least $2n-1$ such that no internal vertex of P' is adjacent to c and the ends of P' are adjacent to c. This together with c gives an induced cycle of length at least $2n+1$. □

8.3.2 Generalized ladders of maximum degree at most 3

Lemma 8.13. *Let G be a generalized ladder of maximum degree 3. If G has at least $6n$ vertices of degree 3, then G has a cycle of length $4n+3$ as a vertex-minor.*

Proof. We proceed by induction on $|V(G)|$. Let P, Q be two defining paths of G. We may assume that all internal vertices of P or Q has degree 3, because if P or Q has an internal vertex v of degree 2, then we apply the induction hypothesis to $G * v\backslash v$. Since p_1 or q_1 has degree 2, we may assume that p_1 has degree 2 by symmetry. We may assume that q_1 has degree 3 because otherwise we can apply the induction hypothesis to $G * q_1\backslash q_1$. Consequently q_1 is adjacent to p_2 and thus for each internal vertex q_i of Q, q_i is adjacent to p_{i+1} and each internal vertex p_{i+1} of P is adjacent to q_i. Thus either $a = b$ and p_a has degree 3 or $a = b+1$ and p_a has degree 2. But if $a = b+1$ and p_a has degree 2, then we can apply the induction hypothesis to $G * p_a\backslash p_a$. Thus we may assume that $a = b$ and p_a has degree 3. Since G has at least $6n$ vertices of degree 3, $a > 3n$ and $b > 3n$. If $a = b > 3n+1$, then we can apply the induction hypothesis to $G\backslash q_b$. Thus we may assume that $a = b = 3n+1$ and p_a has degree 3 and q_b has degree 2. Note that p_i is adjacent to q_{i-1} for all $i = 2, \ldots, 3n+1$. Then $G * p_1 \wedge p_4q_3 \wedge p_7q_6 \cdots \wedge p_{3n+1}q_{3n}\backslash\{p_4, p_7, \ldots, p_{3n-2}, q_3, q_6, \ldots, q_{3n-3}, q_{3n+1}\}$ is isomorphic to a cycle of length $4n+3$. □

Lemma 8.14. *Let G be a generalized ladder of maximum degree 3. If $|V(G)| \geqslant 12n^2$, then G has a cycle of length $4n+3$ as a vertex-minor.*

Proof. Let P, Q be the two defining paths of G. We may assume $a > 1$ and $b > 1$ because otherwise G has an induced cycle of length at least $6n^2+1 \geqslant 4n+3$.

Let p_xq_y be the unique chord other than p_1q_1 with minimum $x+y$. We claim that we may assume $(x-1)+(y-1) \leqslant 2$. Suppose not. Then p_xq_y, p_1q_1 and subpaths of P and Q form a cycle of length $x+y \geqslant 5$ and $p_1, p_2, \ldots, p_{x-1}, q_1, q_2, \ldots, q_{y-1}$ have degree 2. By moving the first few vertices of P to Q or Q to P, we may assume that $x \geqslant 3$ and $y \geqslant 2$. Then we may replace G with $G * p_1$. This proves the claim.

Thus the induced cycle containing p_1q_1 has at most 2 edges from $E(P) \cup E(Q)$. Similarly we may assume that the induced cycle containing p_aq_b has at most 2 edges from $E(P) \cup E(Q)$.

If G has at least $6n$ vertices of degree 3, then by Lemma 8.13, we obtain a desired vertex-minor. So we may assume that G has at most $6n-1$ vertices of degree 3. Thus G has at most $3n-1$ chords other than p_1q_1 and p_aq_b. These chords give at most $3n$ induced cycles of G where each edge in $E(P) \cup E(Q)$ appears in exactly one of them. If every such induced cycle has length at most $4n+2$, then

$$|E(P) \cup E(Q)| \leqslant (3n-2)(4n) + 4 = 12n^2 - 8n + 4 < 12n^2 - 2.$$

Since $|V(G)| \geqslant 12n^2$, we have $|E(P) \cup E(Q)| \geqslant 12n^2 - 2$. This leads to a contradiction. □

8.3.3 Generalized ladders of maximum degree 4

Lemma 8.15. *Let G be a generalized ladder of maximum degree at most 4. Let α be the number of vertices of G having degree 3 or 4. Then G has a vertex-minor H that is a generalized ladder of* *maximum degree at most* 3 *such that* $|V(H)| \geqslant \alpha/4$.

Proof. Let $P = p_1p_2\dots p_a$, $Q = q_1q_2\dots q_b$ be the paths defining a generalized ladder G. Let $X_{i,j} = \{p_1, p_2, \dots, p_i, q_1, q_2, \dots, q_j\}$. We may assume $\alpha > 8$.

If $a = 1$, then p_1 has at least $\alpha - 1$ neighbors but the maximum degree is 4 and therefore $\alpha \leqslant 5$, contradicting our assumption. Thus $a > 1$. Similarly $b > 1$.

We may also assume that no internal vertex of P or Q has degree 2, because otherwise we can apply local complementation and remove it.

Let $\alpha_{i,j}(G)$ be the number of vertices in $V(G)\backslash X_{i,j}$ having degree 3 or 4. We will prove the following.

Claim 1. *Suppose that there exist* $1 \leqslant i < a$ *and* $1 \leqslant j < b$ *such that* $\delta_G(X_{i,j})$ *has exactly two edges and every vertex in* $X_{i,j}$ *has degree* 2 *or* 3 *in* G*. Then* G *has a vertex-minor* H *that is a generalized ladder of maximum degree at most* 3 *such that* $|V(H)| \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$.

Before proving Claim 1, let us see why this claim implies our lemma. First we would like to see why there exist i and j such that $\delta_G(X_{i,j})$ has exactly two edges. If p_1 has degree bigger than 2, then p_1 is adjacent to q_2 and so $G * q_1 = G\backslash p_1q_2$. Thus we may assume that both p_1 and q_1 have degree 2. Keep in mind that the number of vertices of degree 3 or 4 in $X_{1,1}$ may be decreased by 1 by replacing G with $G * q_1$ and so $\alpha_{1,1}(G) \geqslant \alpha - 2$.

By applying Claim 1 with $i = j = 1$, we obtain a generalized ladder H of maximum degree at most 3 as a vertex-minor such that $|V(H)| \geqslant 2 + (\alpha - 2)/4 \geqslant \alpha/4$. This completes the proof of the lemma, assuming Claim 1.

We now prove Claim 1 by induction on $|V(G)| - |X_{i,j}(G)|$. We may assume that every vertex in $V(G)\backslash(X_{i,j} \cup \{p_a, q_b\})$ has degree 3 or 4 because otherwise we can apply local complementation and delete it while keeping $\alpha_{i,j}$. Then p_{i+1} is obviously adjacent to q_{j+1}.

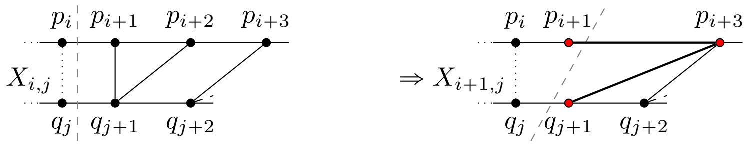
We may assume that $i < a - 1$ because otherwise G is a generalized ladder of maximum degree 3 if p_a has degree 3 and $G\backslash q_b$ is a generalized ladder of maximum degree 3 otherwise. Similarly we may assume $j < b - 1$. Either p_{i+1} or q_{j+1} has degree 4, because otherwise $\delta_G(X_{i+1,j+1})$ has exactly two edges. By symmetry, we may assume that p_{i+1} has degree 3 and q_{j+1} has degree 4 and therefore q_{j+1} is adjacent to p_{i+2}.

If $\alpha_{i,j}(G) \leqslant 12$, then $H = G[X_{i+2,j+1}]$ is a generalized ladder of maximum degree at most 3. Thus we may assume that $\alpha_{i,j}(G) > 12$. If $b - j \leqslant 4$, then $a - i \leqslant 8$ because each vertex in $q_{j+1}, q_{j+2}, \dots, q_b$ has degree at most 4 and each vertex in $p_{i+1}, p_{i+2}, \dots, p_{a-1}$ has degree at least 3. This contradicts our assumption that $\alpha_{i,j}(G) > 12$. So we may assume that $b - j \geqslant 5$ and similarly $a - i \geqslant 5$.

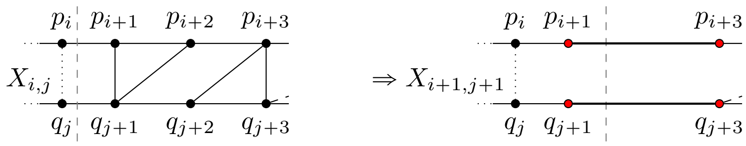
Let R be the component of $G\backslash(E(P) \cup E(Q))$ containing p_{i+1}. Because of the degree condition, R is a path. We now consider six cases, see Figure 8.3.

1. If R has length 2 and p_{i+3} has degree 3 in G, then $G' = G * p_{i+2}\backslash p_{i+2} = (G\backslash p_{i+2} + p_{i+1}p_{i+3} + q_{j+1}p_{i+3})\backslash p_{i+1}q_{j+1}$ is a generalized ladder of maximum degree at most 4. Every vertex in G' not in $X_{i,j}$ has degree at most 4. Furthermore p_{i+1} has degree 2 in G'. Thus, $\delta_{G'}(X_{i+1,j})$ has exactly 2 edges. Then $|X_{i+1,j}| + \alpha_{i+1,j}(G')/4 \geqslant (|X_{i,j}| + 1) + (\alpha_{i,j}(G) - 2)/4 \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$. By the induction hypothesis, we find a desired vertex-minor H in G'.

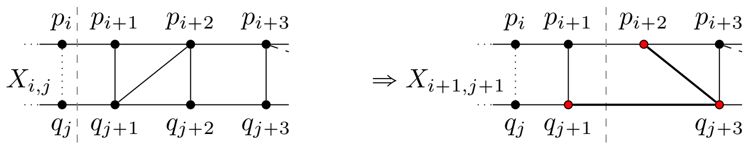
2. If R has length 2 and p_{i+3} has degree 4 in G, then the vertex q_{j+2} has degree 3. Then $G' = G * p_{i+2} * q_{j+2}\backslash p_{i+2}\backslash q_{j+2}$ is a generalized ladder of maximum degree at most 4. Then $\delta_{G'}(X_{i+1,j+1})$ has exactly two edges and $\alpha_{i+1,j+1}(G') \geqslant \alpha_{i,j}(G) - 6$. Again, $|X_{i+1,j+1}| + \alpha_{i+1,j+1}(G')/4 \geqslant |X_{i,j}| + 2 + (\alpha_{i,j}(G) - 6)/4 \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$ and therefore we are done.



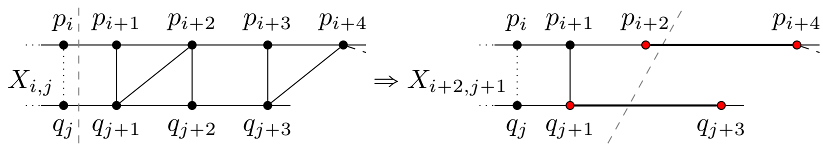
(a) Apply $G * p_{i+2} \backslash p_{i+2}$



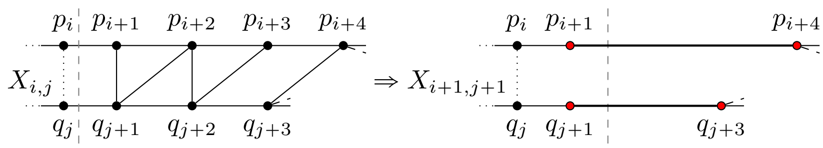
(b) Apply $G * p_{i+2} * q_{j+2} \backslash p_{i+2} \backslash q_{j+2}$



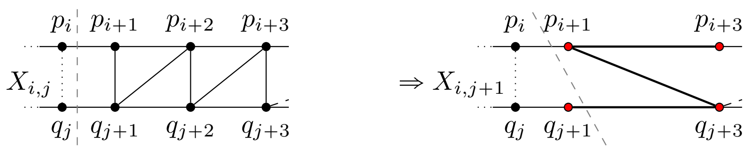
(c) Apply $G * q_{j+2} \backslash q_{j+2}$



(d) Apply $G * q_{j+2} * p_{i+3} \backslash q_{j+2} \backslash p_{i+3}$



(e) Apply $G \wedge p_{i+2}q_{j+2} * p_{i+3} \backslash p_{i+2} \backslash q_{j+2} \backslash p_{i+3}$



(f) Apply $G \wedge p_{i+2}q_{j+2} \backslash p_{i+2} \backslash q_{j+2}$

Figure 8.3: Cases in the proof of Lemma 8.15.

3. If R has length 3 and q_{j+3} has degree 3 in G, then $G' = G * q_{j+2} \backslash q_{j+2}$ is a generalized ladder of maximum degree at most 4. Then $\delta_{G'}(X_{i+1,j+1})$ has exactly two edges and $\alpha_{i+1,j+1}(G') \geqslant \alpha_{i,j}(G) - 3$. We deduce that $|X_{i+1,j+1}| + \alpha_{i+1,j+1}(G')/4 \geqslant |X_{i,j}| + 2 + (\alpha_{i,j}(G) - 3)/4 \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$.

4. If R has length 3 and q_{j+3} has degree 4 in G, then p_{i+3} has degree 3 and $G' = G * q_{j+2} * p_{i+3} \backslash q_{j+2} \backslash p_{i+3}$ is a generalized ladder of maximum degree at most 4. Then $\delta_{G'}(X_{i+2,j+1})$ has exactly two edges and $\alpha_{i+2,j+1}(G') \geqslant \alpha_{i,j}(G) - 7$. We deduce that $|X_{i+2,j+1}| + \alpha_{i+2,j+1}(G')/4 \geqslant |X_{i,j}| + 3 + (\alpha_{i,j}(G) - 7)/4 \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$. By the induction hypothesis, G' has a desired vertex-minor and so does G.

5. If R has length 4, then $G' = G \wedge p_{i+2}q_{j+2} * p_{i+3} \backslash p_{i+2} \backslash p_{i+3} \backslash q_{j+2}$ is a generalized ladder of maximum degree at most 4. Then $\delta_{G'}(X_{i+1,j+1})$ has exactly two edges and $\alpha_{i+1,j+1}(G') \geqslant \alpha_{i,j}(G) - 7$ and therefore $|X_{i+1,j+1}| + \alpha_{i+1,j+1}(G')/4 \geqslant |X_{i,j}| + 2 + (\alpha_{i,j}(G) - 7)/4 \geqslant |X_{i,j}| + \alpha_{i,j}(G)/4$. Our induction hypothesis implies that G' has a desired vertex-minor.

6. If R has length at least 5, then $G' = G \wedge p_{i+2}q_{j+2} \backslash p_{i+2} \backslash q_{j+2}$ is a generalized ladder of maximum degree at most 4. Then $\delta_{G'}(X_{i,j+1})$ has exactly two edges and $\alpha_{i,j+1}(G') \geqslant \alpha_{i,j}(G) - 4$ and therefore $|X_{i,j+1}| + \alpha_{i,j+1}(G')/4 \geqslant |X_{i,j}| + 1 + (\alpha_{i,j}(G) - 4)/4 = |X_{i,j}| + \alpha_{i,j}(G)/4$. Our induction hypothesis implies that G' has a desired vertex-minor.

In all cases, we find the desired vertex-minor H. This completes the proof of Claim 1. $\square$

Lemma 8.16. *Let G be a generalized ladder of maximum degree at most 4. If $|V(G)| \geqslant 192n^3$, then G has a cycle of length $4n+3$ as a vertex-minor.*

Proof. Let P, Q be the two defining paths of G. We may assume $a > 1$ and $b > 1$ because $(192n^3 - 2)/3 + 2 \geqslant 4n + 3$.

Let $p_x q_y$ be the unique chord other than $p_1 q_1$ with minimum $x + y$. We claim that we may assume $(x-1) + (y-1) \leqslant 2$. Suppose not. Then $p_x q_y$, $p_1 q_1$ and subpaths of P and Q form a cycle of length $x + y \geqslant 5$ and $p_1, p_2, \ldots, p_{x-1}, q_1, q_2, \ldots, q_{y-1}$ have degree 2. By moving the first few vertices of P to Q or Q to P, we may assume that $x \geqslant 3$ and $y \geqslant 2$. Then we may replace G with $G * p_1$. This proves the claim.

Thus the induced cycle containing $p_1 q_1$ has at most 2 edges from $E(P) \cup E(Q)$. Similarly we may assume that the induced cycle containing $p_a q_b$ has at most 2 edges from $E(P) \cup E(Q)$.

If G has at least $48n^2$ vertices of degree 3 or 4, then by Lemma 8.15, G has a generalized ladder H as a vertex-minor such that $|V(H)| \geqslant 12n^2$ and H has maximum degree at most 3. By Lemma 8.14, H has a cycle of length $4n+3$ as a vertex-minor.

Thus we may assume that G has less than $48n^2$ vertices of degree 3 or 4. We may assume that G has at least one vertex of degree at least 3. The cycle formed by edges in $E(P) \cup E(Q) \cup \{p_1 q_1, p_a q_b\}$ is partitioned into less than $48n^2$ paths whose internal vertices have degree 2 in G. One of the paths has length greater than $192n^3/(48n^2) = 4n$. Then there is an induced cycle C of G containing this path. Since C does not contain $p_1 q_1$ or $p_a q_b$, C must contain two edges not in $E(P) \cup E(Q) \cup \{p_1 q_1, p_a q_b\}$. Thus the length of C is at least $4n+3$. $\square$

8.3.4 Treating all generalized ladders

Lemma 8.17. *Let G be a generalized ladder. If G has n vertices of degree at least 4, then G has a vertex-minor H that is a generalized ladder such that the maximum degree of H is at most 4 and H has* *at least n vertices.*

Proof. Let P, Q be the two defining paths of G. Let S be the set of vertices having degree at least 4. For each vertex v in S, let P_v be the minimal subpath of Q containing all neighbors of v in Q if $v \in V(P)$ and let P_v be the minimal subpath of P containing all neighbors of v in P if $v \in V(Q)$.

Then each internal vertex of P_v has degree 2 or 3 and has degree 3 if and only if it is adjacent to v. We apply local complementation to each internal vertex and delete all internal vertices of P_v. It is easy to see that the resulting graph H is a generalized ladder and moreover $S \subseteq V(H)$ and every vertex in S has degree at most 4 in H. □

We are now ready to prove the main proposition of this section.

Proof of Proposition 8.10. Let G be such a graph. If G has at least $192n^3$ vertices of degree at least 4, then by Lemma 8.17, G has a vertex-minor H having at least $192n^3$ vertices such that H is a generalized ladder of maximum degree at most 4. By Lemma 8.16, H has a cycle of length $4n+3$ as a vertex-minor.

Thus we may assume that G has less than $192n^3$ vertices of degree at least 4. For a vertex v in P having degree at least 5, let q_i, q_j be two neighbors of v in Q such that if q_k is a neighbor of v in Q, then $i \leqslant k \leqslant j$. By Lemma 8.12, if $j-i+2 \geqslant 24n^2-3$, then G contains a cycle of length $4n+3$ as a vertex-minor. Thus we may assume $j-i \leqslant 24n^2-6$. The subpath of Q from q_i to q_j contains $j-i-1 \leqslant 24n^2-7$ internal vertices. Similarly the same bound holds for a vertex v in Q having degree at least 5. As in the proof of Lemma 8.17, we apply local complementation and delete all internal vertices of the minimal path spanning the neighbors of each vertex of degree at least 5 to obtain H. Then each vertex of degree at least 5 in G will have degree at most 4 in H. Since we remove at most $(192n^3-1)(24n^2-7)$ vertices,

$$|V(H)| \geqslant |V(G)| - (192n^3-1)(24n^2-7) > 192n^3.$$

By Lemma 8.16, H has a cycle of length $4n+3$ as a vertex-minor. □

8.4 Obtaining a long cycle from a huge induced path

In this section we aim to prove the following theorem.

Theorem 8.18. *If a prime graph has an induced path of length $\lceil 6.75n^7 \rceil$, then it has a cycle of length n as a vertex-minor.*

The main idea is to find a big generalized ladder, defined in Section 8.3 as a vertex-minor by using blocking sequences in Section 8.2.

8.4.1 Patching a path

For $1 \leqslant k \leqslant n-2$, a *$k$-patch* of an induced path $P = v_0v_1\cdots v_n$ of a graph G is a sequence $Q = w_1, w_2, \ldots, w_k$ of distinct vertices not on P such that for each $i \in \{1, 2, \ldots, k\}$,

1. v_{i+2} is the only vertex adjacent to w_i among v_{i+1}, v_{i+2}, $\ldots$, v_n,

2. $\varnothing \neq N_G(w_i) \cap \{v_0, \ldots, v_i, w_1, \ldots, w_{i-1}\} \neq \{v_i, w_{i-1}\}$ if $i > 1$,

3. $N_G(w_1) \cap \{v_0, v_1\} = \{v_0\}$.

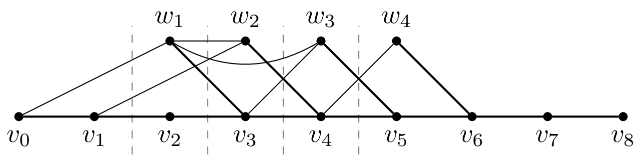


Figure 8.4: An example of a 4-patched path of length 8.

An induced path is called *k-patched* if it has a k-patch. An induced path of length n is called *fully patched* if it is equipped with a $(n-2)$-patch. See Figure 8.4 for an example.

Our goal is to find a fully patched long induced path in a vertex-minor of a prime graph having a very long induced path.

Lemma 8.19. *Let $P = v_0v_1 \dots v_m$ be an induced path from $s = v_0$ to $t = v_m$ in a graph G and let H be a connected induced subgraph of $G \backslash V(P)$. Let v be a vertex in $V(G) \backslash (V(H) \cup V(P))$. Suppose that $N_G(V(H)) \cap V(P) = \{s\}$, $|E(P)| \geqslant 6(n-1)^2 - 5$, and v has neighbors in both $V(P) \backslash \{s\}$ and $V(H)$.*

If G has no cycle of length $2n+1$ as a vertex-minor, then there exist a graph G' locally equivalent to G and an induced path P' from s to t of G' disjoint from $V(H)$ satisfying the following.

1. *$G[V(H) \cup \{s\}] = G'[V(H) \cup \{s\}]$,*
2. *$N_G(v) \cap V(H) = N_{G'}(v) \cap V(H)$,*
3. *$P' = v_0v_iv_{i+1}v_{i+2} \cdots v_m$ for some i,*
4. *v_i is the only vertex on $V(P')$ adjacent to v in G',*
5. *$|E(P')| \geqslant |E(P)| - 6(n-1)^2 + 6$.*

Proof. Since G has a cycle using H with s and P, G is not a forest and therefore $n \geqslant 2$. Let $v_0 = s, v_1, v_2, \dots, v_m = t$ be vertices in P. Let v_k be the neighbor of v with maximum k. Then G has a fan having at least $k+3$ vertices because H is connected and v has a neighbor in H. If $k \geqslant 6(n-1)^2 - 6$, then G has a fan having at least $6(n-1)^2 - 3$ vertices and by Lemma 8.12, G contains a cycle of length $2n+1$ as a vertex-minor. This contradicts to our assumption that G has no such vertex-minor. Thus, $k \leqslant 6(n-1)^2 - 7$.

Let $G_0 = G * v_1 * v_2 * v_3 \cdots * v_{k-2}$ and let $P_0 = v_0v_{k-1}v_kv_{k+1} \cdots v_m$. (If $k \leqslant 2$, then let $G_0 = G$ and $P_0 = P$.) Then clearly P_0 is an induced path of G_0 and $v_k \in N_{G_0}(v) \cap V(P_0) \subseteq \{v_0, v_{k-1}, v_k\}$.

If $N_{G_0}(v) \cap V(P_0) = \{v_k\}$, then we are done by taking $G' = G_0 * v_{k-1}$ and $P' = v_0v_kv_{k+1} \cdots v_m$.

If $N_{G_0}(v) \cap V(P_0) = \{v_{k-1}, v_k\}$, then we can take $G' = G_0 * v_k * v_{k-1}$ and $P' = v_0v_{k+1}v_{k+2} \cdots v_m$.

If $N_{G_0}(v) \cap V(P_0) = \{v_0, v_k\}$, then we can take $G' = G_0 * v_{k-1} * v_k$ and $P' = v_0v_{k+1}v_{k+2} \cdots v_m$.

Finally, if $N_{G_0}(v) \cap V(P_0) = \{v_0, v_{k-1}, v_k\}$, then we can take $G' = G_0 * v_k * v_{k-1} * v_{k+1}$ and $P' = v_0v_{k+2}v_{k+3} \cdots v_m$.

In all cases, $|E(P')| \geqslant |E(P)| - (k+1) \geqslant |E(P)| - 6(n-1)^2 + 6$. $\square$

Lemma 8.20. *Let $n \geqslant 2$. Let G be a prime graph having an induced path of length t. If $t \geqslant 6(n-1)^2 - 3$, then there exists a graph G' locally equivalent to G having a 1-patched induced path of length $t - 6(n-1)^2 + 6$, unless G has a cycle of length $2n+1$ as a vertex-minor.*

Proof. We may choose G so that the length t of an induced path P is maximized among all graphs locally equivalent to G. Let $v_0, v_1, \dots, v_m$ be vertices of P in this order. Since G is prime, v_0 has a neighbor v

other than v_1. We may assume that v is non-adjacent to v_1 because otherwise we can replace G with $G * v_0$.

Since P is a longest induced path, v must have some neighbors in $V(P)\backslash\{v_0, v_1\}$. We now apply Lemma 8.19 with $H = G[\{v_0, v_1\}]$, deducing that there exists a graph G' locally equivalent to G having a 1-patched induced path of length $t - 6(n-1)^2 + 6$, unless G has a cycle of length $2n+1$ as a vertex-minor. □

Lemma 8.21. *Let $n \geqslant 2$. Let G be a prime graph and let P be a k-patched induced path $v_0v_1 \cdots v_t$. If $t \geqslant 6(n-1)^2 + k$, then there exists a graph G' locally equivalent to G having a $(k+1)$-patched induced path $v_0v_1 \cdots v_{k+2}v_iv_{i+1} \cdots v_t$ of length at least $t - 6(n-1)^2 + 3$ with some $i > k+2$, unless G has a cycle of length $2n+1$ as a vertex-minor.*

Proof. Let $P = v_0v_1 \ldots v_t$ be an induced path of length t in G and $Q = w_1, w_2, \ldots, w_k$ be its k-patch. Suppose that G has no vertex-minor isomorphic to a cycle of length $2n+1$.

Let $A = \{v_0, v_1, \ldots, v_{k+1}\} \cup Q$. By Proposition 8.9, we may assume that G has a blocking sequence $b_1, b_2, \ldots, b_\ell$ of length at most 4 for $(A, V(P)\backslash A)$ because v_{k+2} is the only vertex in $V(P)\backslash A$ having neighbors in A.

Notice that $P\backslash A$ is an induced path of G. We say that a blocking sequence $b_1, b_2, \ldots, b_\ell$ for $(A, V(P)\backslash A)$ is *nice* if b_ℓ has a unique neighbor in $V(P)\backslash A$, that is also a unique neighbor of v_{k+2} in $V(P)\backslash A$.

We know that b_ℓ has neighbors in $\{v_{k+3}, \ldots, v_t\}$ by the definition of a blocking sequence. We take $H = G[A \cup Q \cup \{b_1, b_2, \ldots, b_{\ell-1}\}]$. By Lemma 8.19, there exist a graph G_ℓ locally equivalent to G and an induced path $P_\ell = v_0v_1 \cdots v_{k+2}v_iv_{i+1} \cdots v_t$ of G_ℓ for some i with a k-patch Q such that $G_\ell[A \cup \{v_{k+2}\}] = G[A \cup \{v_{k+2}\}]$, a sequence $b_1, b_2, \ldots, b_\ell$ is a nice blocking sequence for $(A, V(P_\ell)\backslash A)$ in G_ℓ, and $|E(P_\ell)| \geqslant t - 6(n-1)^2 + 6$.

Let $r \geqslant 1$ be minimum such that there exist a graph G' locally equivalent to G and an induced path $P' = v_0v_1 \cdots v_{k+2}v_iv_{i+1} \cdots v_m$ for some i with a k-patch Q in G' such that $G'[A \cup \{v_{k+2}\}] = G[A \cup \{v_{k+2}\}]$, a sequence $b_1, b_2, \ldots, b_r$ is a nice blocking sequence for $(A, V(P')\backslash A)$ in G', and $|E(P')| \geqslant t - 6(n-1)^2 + 6 + r - \ell$. Such r exists because G_ℓ and P_ℓ satisfy the condition when $r = \ell$.

We claim that $r = 1$. Suppose $r > 1$.

Suppose that b_r is non-adjacent to v_{k+1} in G'. Then v_i is the only neighbor of b_r in $V(P')$ in G' and b_r is adjacent to b_{r-1} in G'. If b_{r-1} is non-adjacent to v_{k+2}, then take $G'' = G' * b_r$ and $P'' = P'$; in G'', a sequence $b_1, b_2, \ldots, b_{r-1}$ is a nice blocking sequence for $(A, V(P')\backslash A)$ and the length of P' is at least $t - 6(n-1)^2 + 6 + r - \ell$. This leads a contradiction to the assumption that r is minimized. Therefore b_{r-1} is adjacent to v_{k+2}. Then take $G'' = G' * b_r * v_i$ with $P'' = v_0v_1 \cdots v_{k+2}v_{i+1} \cdots v_m$. Then $b_1, b_2, \ldots, b_{r-1}$ is a nice blocking sequence for $(A, V(P'')\backslash A)$ in G'' and the length of P'' is at least $t - 6(n-1)^2 + 6 + r - \ell - 1$. This contradicts to the assumption that r is chosen to be minimum.

Therefore b_r is adjacent to v_{k+1} in G'. Since b_r is the last vertex in the blocking sequence, b_r is also adjacent to w_k in G'. If b_{r-1} is non-adjacent to v_{k+2}, then take $G'' = G' * v_{k+2} * b_r$ and $P'' = P'$; in G'', a sequence $b_1, b_2, \ldots, b_{r-1}$ is a nice blocking sequence for $(A, V(P'')\backslash A)$ and the length of P'' is at least $t - 6(n-1)^2 + 6 + r - \ell$, contradicting our assumption on r. So b_{r-1} is adjacent to v_{k+2}. Then we take $G'' = G' * v_{k+2} * b_r * v_i$ with $P'' = v_0v_1 \cdots v_{k+2}v_{i+1} \cdots v_m$. Then $b_1, b_2, \ldots, b_{r-1}$ is a nice blocking sequence for $(A, V(P'')\backslash A)$ in G'' and the length of P'' is at least $t - 6(n-1)^2 + 6 + r - \ell - 1$. This again contradicts to the assumption on r. This proves that $r = 1$.

Since b_1 is a nice blocking sequence for $(A, V(P')\backslash A)$ in G', b_1 has a neighbor in A in G' and $N_{G'}(b_1) \cap A \neq \{v_{k+1}, w_k\}$. In addition, v_i is the only neighbor of b_1 among $V(P')\backslash A$ in G'. Now it is easy to see that $w_1, w_2, w_3, \ldots, w_k, b_1$ is a $(k+1)$-patch of P' in G'. And, since $\ell \leqslant 4$, we have $|E(P')| \geqslant t - 6(n-1)^2 + 3$. □

Proposition 8.22. *Let $N \geqslant 4$ be an integer. If a prime graph G on at least 5 vertices has an induced path of length $L = (6(n-1)^2 - 2)(N-2) - 1$, then there exists a graph G' locally equivalent to G having a fully patched induced path of length N, unless G has a cycle of length $2n+1$ as a vertex-minor.*

Proof. Suppose that G has no cycle of length $2n+1$ as a vertex-minor. Then $n \geqslant 3$ by Theorem 4.2. By Lemma 8.20, we may assume that G has a 1-patched path of length $L - 6(n-1)^2 + 6$. By Lemma 8.21, we may assume that G has an $(N-2)$-patched path of length

$$L - 6(n-1)^2 + 6 - (N-3)(6(n-1)^2 - 3) = N$$

Thus G has a fully patched induced path of length N. □

8.4.2 Finding a cycle from a fully patched path

We aim to find a cycle as a vertex-minor in a sufficiently long fully patched path.

Let $P = v_0 v_1 \cdots v_n$ be an induced path of a graph G with a $(n-2)$-patch $Q = w_1 w_2 w_3, \ldots w_{n-2}$. Let $A_1 = \{v_0, v_1\}$ and for $i = 2, \ldots, n-2$, let $A_i = \{v_0, v_1, \ldots, v_i, w_1, w_2, \ldots, w_{i-1}\}$ and $B_i = V(P)\backslash A_i$ for all $i \in \{1, 2, \ldots, n-2\}$.

For $i \geqslant 1$, let $L(w_i)$ be the minimum $j \geqslant 0$ such that

$$\rho^*_G(A_{j+1}, B_{j+1} \cup \{w_i\}) > 1.$$

Since w_i is a blocking sequence for (A_i, B_i), $L(w_i)$ is well defined and $L(w_i) < i$.

We classify vertices in Q as follows.

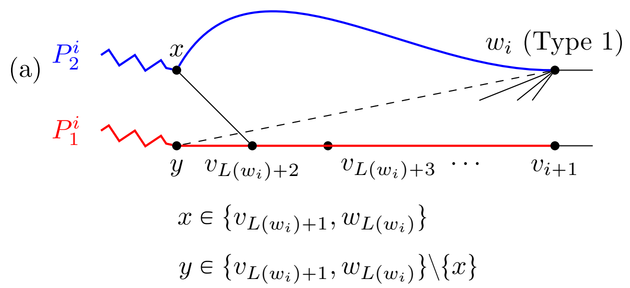
- A vertex w_i has *Type 0* if $L(w_i) = 0$ and w_i is adjacent to v_0.
- A vertex w_i has *Type 1* if $L(w_i) \geqslant 1$ and w_i has no neighbor in $A_{L(w_i)}$ and w_i is adjacent to exactly one of $v_{L(w_i)+1}$ and $w_{L(w_i)}$.
- A vertex w_i has *Type 2* if $L(w_i) = 1$ and w_i is adjacent to v_1, non-adjacent to v_0.
- A vertex w_i has *Type 3* if $L(w_i) \geqslant 2$ and w_i has no neighbor in $A_{L(w_i)-1}$ and w_i is adjacent to both $v_{L(w_i)}$ and $w_{L(w_i)-1}$.

By the definition of fully patched paths, we can deduce the following lemma easily.

Lemma 8.23. *Each vertex in Q has Type 0, 1, 2, or 3.*

Proof. If w_i is adjacent to v_0, then $\rho^*_G(A_1, B_1 \cup \{w_i\}) > 1$ and therefore $L(w_i) = 0$, implying that w_i has Type 0. We may now assume that w_i is non-adjacent to v_0 and so $L(w_i) > 0$.

If w_i has no neighbors in $A_{L(w_i)}$, then $\rho^*_G(A_{L(w_i)+1}, B_{L(w_i)+1} \cup \{w_i\}) = \rho^*_G(A_{L(w_i)+1}\backslash A_{L(w_i)}, B_{L(w_i)+1} \cup \{w_i\}) > 1$. Thus $v_{L(w_i)+2}$ and w_i cannot have the same set of neighbors in $A_{L(w_i)+1}\backslash A_{L(w_i)} = \{v_{L(w_i)+1}, w_{L(w_i)}\}$. By the definition of fully patched paths, $v_{L(w_i)+2}$ is adjacent to both $v_{L(w_i)+1}$ and $w_{L(w_i)}$. It follows that w_i is adjacent to exactly one of $v_{L(w_i)+1}$ and $w_{L(w_i)}$. So w_i has Type 1.



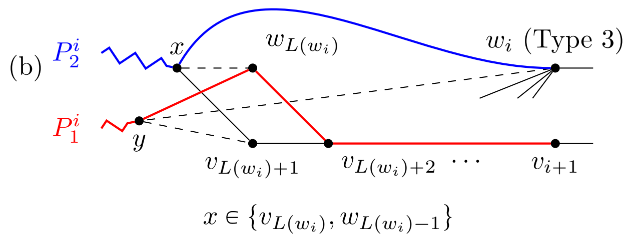


Figure 8.5: Constructing a generalized ladder in a fully patched path. The vertex w_i has Type 1 in (a) and has Type 3 in (b).

Now we may assume that w_i has some neighbors in $A_{L(w_i)}$. By definition,

$$\rho_G^*(A_{L(w_i)}, B_{L(w_i)} \cup \{w_i\}) \leqslant 1$$

and therefore w_i and $v_{L(w_i)+1}$ have the same set of neighbors in $A_{L(w_i)}$. Therefore, if $L(w_i) = 1$, then w_i is adjacent to v_1, implying that w_i has Type 2. If $L(w_i) > 1$, then w_i is adjacent to both $v_{L(w_i)}$ and $w_{L(w_i)-1}$, and so w_i has Type 3. $\square$

We say that a pair of paths P_1^i and P_2^i from $\{v_0, v_1\}$ to $\{v_{i+1}, w_i\}$ is *good* if

1. P_1^i and P_2^i are vertex-disjoint induced paths on A_{i+1},
2. for each $j \in \{1, 2, \ldots, i-1\}$, $w_j \in V(P_1^i) \cup V(P_2^i)$ or $v_{j+1} \in V(P_1^i) \cup V(P_2^i)$,
3. $G[V(P_1^i) \cup V(P_2^i)] + v_{i+1}w_i$ is a generalized ladder with two defining paths P_1^i and P_2^i.

Lemma 8.24. *For all $i \in \{1, 2, \ldots, n-2\}$, G has a good pair of paths P_1^i and P_2^i from $\{v_0, v_1\}$ to $\{v_{i+1}, w_i\}$.*

Proof. We proceed by induction on i. If w_i has Type 0, then let $P_1^i = v_1 v_2 \cdots v_{i+1}$ and $P_2^i = v_0 w_i$. Since v_0 has no neighbors in $\{v_2, v_3, \ldots, v_{i+1}\}$, $G[V(P_1^i) \cup V(P_2^i)] + v_{i+1}w_i$ is a generalized ladder with two defining paths P_1^i and P_2^i. Also, $V(P_1^i) \cup V(P_2^i) \subseteq A_{i+1}$ and for all $j \in \{1, 2, \ldots, i-1\}$, $v_{j+1} \in V(P_1^i)$. Thus, the pair (P_1^i, P_2^i) is good.

If w_i has Type 2, then let $P_1^i = v_0 w_1 v_3 v_4 \cdots v_{i+1}$ and $P_2^i = v_1 w_i$. By the definition of a patched path, v_1 is not adjacent to w_1. So, v_1 has no neighbors in $\{w_1, v_3, v_4, \ldots, v_{i+1}\}$, and therefore $G[V(P_1^i) \cup V(P_2^i)]+v_{i+1}w_i$ is a generalized ladder with two defining paths P_1^i and P_2^i. Clearly, $V(P_1^i) \cup V(P_2^i) \subseteq A_{i+1}$. Moreover, $w_1 \in V(P_1^i)$ and for each $j \in \{2, \ldots, i-1\}$, $v_{j+1} \in V(P_1^i)$. Therefore, the pair (P_1^i, P_2^i) is good.

Now, we may assume that w_i has Type 1 or Type 3. Since $L(w_i) \geqslant 1$, by the induction hypothesis, G has a good pair of paths $P_1^{L(w_i)}$, $P_2^{L(w_i)}$ from $\{v_0, v_1\}$ to $\{v_{L(w_i)+1}, w_{L(w_i)}\}$.

Suppose w_i has Type 1 and therefore w_i is adjacent to exactly one of $v_{L(w_i)+1}$ and $w_{L(w_i)}$. Let $\{x, y\} = \{v_{L(w_i)+1}, w_{L(w_i)}\}$ such that x is adjacent to w_i. We may assume that the paths $P_1^{L(w_i)}$ and $P_2^{L(w_i)}$ end at y and x, respectively. Let P_1^i be a path

$$P_1^{L(w_i)} + yv_{L(w_i)+2}v_{L(w_i)+3} \cdots v_{i+1}$$

and let P_2^i be a path $P_2^{L(w_i)} + xw_i$. See Figure 8.5. By the induction hypothesis, $V(P_1^{L(w_i)}) \cup V(P_2^{L(w_i)}) \subseteq A_{L(w_i)+1} \subseteq A_{i+1}$, and for each $j \in \{1, 2, \ldots, L(w_i) - 1\}$, $V(P_1^{L(w_i)}) \cup V(P_2^{L(w_i)})$ contains w_j or v_{j+1}. Thus it follows that $V(P_1^i) \cup V(P_2^i) \subseteq A_{i+1}$ and for each $j \in \{1, 2, \ldots, i-1\}$, $V(P_1^i) \cup V(P_2^i)$ contains w_j or v_{j+1}.

We claim that $G[V(P_1^i) \cup V(P_2^i)] + v_{i+1}w_i$ is a generalized ladder with the defining paths P_1^i and P_2^i. By the induction hypothesis, it is enough to show that there are no two crossing chords xa and w_ib for some $a, b \in V(P_1^i)$. Since w_i has no neighbor in $A_{L(w_i)}$ and w_i and y are non-adjacent, $b \in X = \{v_k : k \in \{L(w_i) + 2, L(w_i) + 3, \ldots, i + 1\}\}$. Since x has no neighbor in $X \backslash \{v_{L(w_i)+2}\}$, we deduce that xa and w_ib cannot cross and therefore $G[V(P_1^i) \cup V(P_2^i)] + v_{i+1}w_i$ is a generalized ladder. This proves that if w_i has Type 1, then (P_1^i, P_2^i) is a good pair.

Finally, suppose that w_i has Type 3 and so w_i is adjacent to both $v_{L(w_i)}$ and $w_{L(w_i)-1}$. By symmetry, we may assume that $P_2^{L(w_i)}$ ends at $v_{L(w_i)+1}$. Let x be the predecessor of $v_{L(w_i)+1}$ in $P_2^{L(w_i)}$. Since $P_2^{L(w_i)}$ is on $A_{L(w_i)+1}$ and $v_{L(w_i)+1}$ has only two neighbors $v_{L(w_i)}, w_{L(w_i)-1}$ in $A_{L(w_i)+1}$, either $x = v_{L(w_i)}$ or $x = w_{L(w_i)-1}$. Let y be the predecessor of $w_{L(w_i)}$ in $P_1^{L(w_i)}$. Let P_1^i be a path

$$P_1^{L(w_i)} + w_{L(w_i)}v_{L(w_i)+2}v_{L(w_i)+3} \cdots v_{i+1}$$

and let P_2^i be a path obtained from $P_2^{L(w_i)}$ by removing $v_{L(w_i)+1}$ and adding xw_i. See Figure 8.5(b). It follows from our construction and the induction hypothesis that $V(P_1^i) \cup V(P_2^i) \subseteq A_{i+1}$ and $V(P_1^i) \cup V(P_2^i)$ contains w_j or v_{j+1} for each $j \in \{1, 2, \ldots, i-1\}$.

We claim that $G[V(P_1^i) \cup V(P_2^i)] + v_{i+1}w_i$ is a generalized ladder with the defining paths P_1^i and P_2^i. By the induction hypothesis, it is enough to prove that there are no two chords xa and w_ib such that $a, b \in V(P_1^i)$ and b precedes a in P_1^i. Suppose not. Since w_i has no neighbor in $A_{L(w_i)-1}$, neighbors of w_i in P_1^i are in $\{y, w_{L(w_i)}\} \cup \{v_k : k \in \{L(w_i) + 2, L(w_i) + 3, \ldots, i + 1\}\}$. Since x has no neighbor in $\{v_k : k \in \{L(w_i) + 2, L(w_i) + 3, \ldots, i + 1\}\}$, we deduce that $a = w_{L(w_i)}$ and $b = y$. Since w_i has no neighbor in $A_{L(w_i)-1}$, b is one of $v_{L(w_i)}$ and $w_{L(w_i)-1}$ other than x. Thus $w_{L(w_i)}$ is adjacent to both $v_{L(w_i)}$ and $w_{L(w_i)-1}$. This contradicts (iii) because $v_{L(w_i)+1}$ is also adjacent to both $v_{L(w_i)}$ and $w_{L(w_i)-1}$ and so $G[V(P_1^{L(w_i)}) \cup V(P_2^{L(w_i)})] + v_{L(w_i)+1}w_{L(w_i)}$ is not a generalized ladder. $\square$

Lemma 8.25. *If a graph has a fully patched induced path of length n, then it has a generalized ladder having at least $n + 2$ vertices as an induced subgraph.*

Proof. Let $P = v_0v_1 \cdots v_n$ be the induced path of length n with an $(n-2)$-patch $Q = w_1w_2 \cdots w_{n-2}$. Lemma 8.24 provides a good pair of paths P_1^{n-2} and P_2^{n-2} from $\{v_0, v_1\}$ to $\{v_{n-1}, w_{n-2}\}$ such that $G[V(P_1^{n-2}) \cup V(P_2^{n-2})] + v_{n-1}w_{n-2}$ is a generalized ladder and $V(P_1^{n-2}) \cup V(P_2^{n-2})$ contains w_j or v_{j+1} for each $j \in \{1, 2, \ldots, n-3\}$. Since v_n is only adjacent to v_{n-1} and w_{n-2} in G, $G' = G[V(P_1^{n-2}) \cup V(P_2^{n-2}) \cup \{v_n\}]$ is a generalized ladder. Since $v_0, v_1, v_n, v_{n-1}, w_{n-2} \in V(G')$, G' has at least $(n-3)+5 = n + 2$ vertices. $\square$

Now we are ready to prove the main theorem of this section.

Lemma 8.26. *Let $n \geqslant 1$. If a prime graph has an induced path of length $110592n^7$, then it has a cycle of length $4n + 3$ as a vertex-minor.*

Proof. Let G be a prime graph having an induced path of length $110592n^7$. Suppose that G has no cycle of length $4n+3$ as a vertex-minor. Let $N = 4608n^5$. Then

$$(6(2n)^2 - 2)(N-2) - 1 < 110592n^7.$$

Thus by Proposition 8.22, there exists a graph G' locally equivalent to G having a fully patched induced path of length N. By Lemma 8.25, G' must have a generalized ladder having at least $N+2$ vertices as an induced subgraph. By Proposition 8.10, we deduce that G' has a cycle of length $4n+3$ as a vertex-minor. □

Proof of Theorem 8.18. Let $k = \lfloor n/4 \rfloor$. Let G be a prime graph having a path of length at least $6.75n^7$. Then G has a path of length $6.75(4k)^7 = 110592k^7$, and by Lemma 8.26, G has a cycle of length $4k+3 \geqslant n$ as a vertex-minor. □

8.5 Obtaining C_n or $K_n \boxminus K_n$ from a large prime graphs

In this section, we prove the main result of this chapter.

Theorem 8.1. *For every n, there is N such that every prime graph on at least N vertices has a vertex-minor isomorphic to C_n or $K_n \boxminus K_n$.*

By Theorem 8.18, it is enough to prove the following proposition.

Proposition 8.27. *For every c, there exists N such that every prime graph on at least N vertices has a vertex-minor isomorphic to either P_c or $K_c \boxminus K_c$.*

Here is the proof of Theorem 8.1 assuming Proposition 8.27.

Proof of Theorem 8.1. We take $c = \lceil 6.75n^7 \rceil$ and apply Proposition 8.27 and Theorem 8.18. □

For integers $h, w, \ell \geqslant 1$, a (h, w, ℓ)*-broom* of a graph G is a connected induced subgraph H of G such that

1. H contains a vertex v, called the *center*,
2. one component of $H \backslash v$ is an induced path P of length $h-1$, and
3. $H \backslash (V(P) \cup \{v\})$ has w components, each having exactly ℓ vertices.

The path P is called a *handle* of H and each component of $H \backslash V(P)$ is called a *fiber* of H. If $H = G$, then we say that G is a (h, w, ℓ)-broom. We call h, w, ℓ the *height*, *width*, *length*, respectively, of a (h, w, ℓ)-broom. See Figure 8.6. Observe that v has one or more neighbors in each fiber.

Here is the rough sketch of the proof. If a prime graph G has no vertex-minor isomorphic to P_c or $K_c \boxminus K_c$ and G has a broom having huge width as a vertex-minor, then it has a vertex-minor isomorphic to a broom with larger length and sufficiently big width. So, we increase the length of a broom while keeping its width big. If we obtain a broom of big length by repeatedly applying this process, then we will obtain a broom of larger height. By growing the height, we will eventually obtain a long induced path.

To start the process, we need an initial broom with sufficiently big width. For that purpose, we use the following Ramsey-type theorem.

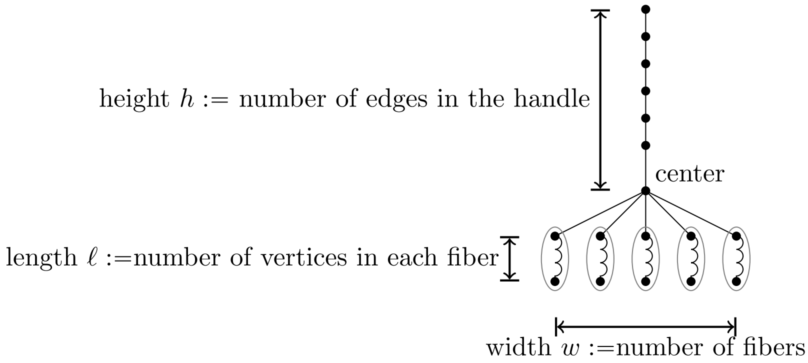


Figure 8.6: A (h, w, ℓ)-broom.

Theorem 8.28 (folklore; see Diestel [76]). *For positive integers c and t, there exists $N = g_0(c, t)$ such that every connected graph on at least N vertices must contain K_{t+1}, $K_{1,t}$, or P_c as an induced subgraph.*

By Theorem 8.28, if G is prime and $|V(G)| \geqslant g_0(c, t+1)$, then either G has an induced subgraph isomorphic to P_c or G has a vertex-minor isomorphic to $K_{1,t+1}$. Since a $(1, t, 1)$-broom is isomorphic to $K_{1,t+1}$, we conclude that every sufficiently large prime graph has a vertex-minor isomorphic to a $(1, t, 1)$-broom, unless it has an induced subgraph isomorphic to P_c.

8.5.1 Increasing the length of a broom

We now show that if a prime graph has a broom having sufficiently large width, we can find a broom having larger length after applying local complementation and shrinking the width.

In the following proposition, we want to find a wide broom of length 2 when we are given a sufficiently wide broom of length 1, when the graph has no P_c or $K_c \boxminus K_c$ as a vertex-minor.

Proposition 8.29. *For all integers $c \geqslant 3$ and $t \geqslant 1$, there exists $N = g_1(c, t)$ such that for each $h \geqslant 1$, every prime graph having a $(h, N, 1)$-broom has a vertex-minor isomorphic to a $(h, t, 2)$-broom, $K_c \boxminus K_c$, or P_c.*

We will use the following theorem.

Theorem 8.30 (Ding, Oporowski, Oxley, Vertigan [79]). *For every positive integer n, there exists $N = f(n)$ such that for every bipartite graph G with a bipartition (S, T), if no two vertices in S have the same set of neighbors and $|S| \geqslant N$, then S and T have n-element subsets S' and T', respectively, such that $G[S', T']$ is isomorphic to $\overline{K_n} \boxminus \overline{K_n}$, $\overline{K_n} \boxslash \overline{K_n}$, or $\overline{K_n} \boxtimes \overline{K_n}$.*

Proof of Proposition 8.29. Let $N = f(R(w, w))$ where f is the function in Theorem 8.30, and $w = \max(t + (c-1)(c-3), 2c-1)$. Suppose that G has a $(h, g_1(c, t), 1)$-broom H. Note that every fiber of H is a single vertex.

Let S be the union of the vertex sets of all fibers of H, and x be the center of H. Let $N_G(S) \backslash \{x\} = T$. Since G is prime, no two vertices in G have the same set of neighbors, and so two distinct vertices in S have different sets of neighbors in T. Since $|S| = N = f(R(w, w))$, by Theorem 8.30, there exist $S_0 \subseteq S$, $T_0 \subseteq T$ such that $G[S_0, T_0]$ is isomorphic to $\overline{K_{R(w,w)}} \boxminus \overline{K_{R(w,w)}}$, $\overline{K_{R(w,w)}} \boxslash \overline{K_{R(w,w)}}$ or $\overline{K_{R(w,w)}} \boxtimes \overline{K_{R(w,w)}}$. Since $|T_0| \geqslant R(w, w)$, by Ramsey's theorem, there exist $S' \subseteq S_0$ and $T' \subseteq T_0$ such that $G[S', T']$ is isomorphic to $\overline{K_w} \boxminus \overline{K_w}$, $\overline{K_w} \boxslash \overline{K_w}$, or $\overline{K_w} \boxtimes \overline{K_w}$, and T' is a clique or a stable set in G. If $G[S', T']$ is isomorphic to $\overline{K_w} \boxslash \overline{K_w}$ or $\overline{K_w} \boxtimes \overline{K_w}$, then by Lemmas 8.2 and 8.3, G has a vertex-minor isomorphic to

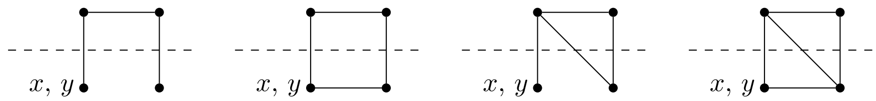


Figure 8.7: Dealing with 4-vertex graphs in Lemma 8.31.

either P_{2w} or $K_{w-2} \boxminus K_{w-2}$. Since $w \geqslant 2c-1$ and $c \geqslant 3$, we have P_c or $K_c \boxminus K_c$. Thus we may assume that $G[S', T']$ is isomorphic to $\overline{K_w} \boxminus \overline{K_w}$.

If T' is a clique in G, then we can remove the edges connecting T' with x by applying local complementation at some vertices in S'. Thus, we can obtain a vertex-minor isomorphic to $K_w \boxminus K_w$ from $G[S' \cup T' \cup \{x\}]$ by applying local complementation at x and deleting x. Therefore we may assume that T' is a stable set in G.

We claim that each vertex $y \neq x$ in the handle of H is adjacent to at most c vertices in T', or G has $K_c \boxminus K_c$ as a vertex-minor. Suppose not. If y is a neighbor of x, then by pivoting an edge of $G[S', T']$, we can delete the edge xy. From there, we obtain a vertex-minor isomorphic to $K_c \boxminus K_c$ by applying local complementation at x and y. If y is not adjacent to x, then we obtain a vertex-minor isomorphic to $K_c \boxminus K_c$ by deleting all vertices in the handle other than x and y, and applying local complementation at x and y. This proves the claim.

By deleting at most $(c-1)h$ vertices in T' and their pairs in S', we can assume that no vertex other than x in the handle has a neighbor in T' and this broom has width at least $w-(c-1)h$. If $h+2 \geqslant c$, then we have P_c as an induced subgraph. Thus we may assume that $h \leqslant c-3$. Since $w-(c-1)h \geqslant w-(c-1)(c-3) \geqslant t$, we obtain a vertex-minor isomorphic to a $(h, t, 2)$-broom. $\square$

We now aim to increase the length of a broom when the broom has length at least 2. For a fiber F of a broom H, we say that a vertex $v \in V(G) \backslash V(H)$ *blocks* F if

$$\rho^*_G(V(F), (V(H) \backslash V(F)) \cup \{v\}) > 1.$$

If G is prime and F has at least two vertices, then G has a blocking sequence for $(V(F), V(H) \backslash V(F))$ by Proposition 2.6 and therefore there exists a vertex v that blocks F because we can take the first vertex in the blocking sequence.

Lemma 8.31. *Let G be a graph and let x, y be two vertices such that $\rho_G(\{x, y\}) = 2$ and $G \backslash x \backslash y$ is connected. Then there exists some sequence $v_1, v_2, \ldots, v_n \in V(G) \backslash \{x, y\}$ of (not necessarily distinct) vertices such that $G * v_1 * v_2 \cdots * v_n$ has an induced path of length 3 from x to y.*

Proof. We proceed by induction on $|V(G)|$. If $|V(G)| = 4$, then it is easy to check all cases to obtain a path of length 3. To do so, first observe that up to symmetry, there are 2 cases in $G[\{x, y\}, V(G) \backslash \{x, y\}]$; either it is a matching of size 2 or a path of length 3. In both cases, one can find a desired sequence of vertices to apply local complementation, see Figure 8.7 for all possible graphs on 4-vertices up to isomorphism.

Now we may assume that G has at least 5 vertices. Let $A_1 = N_G(x) \backslash (N_G(y) \cup \{y\})$, $A_2 = N_G(x) \cap N_G(y)$, and $A_3 = N_G(y) \backslash (N_G(x) \cup \{x\})$. Clearly $\rho_G(\{x, y\}) = 2$ is equivalent to say that at least two of A_1, A_2, A_3 are nonempty.

We say a vertex t in $G \backslash x \backslash y$ *deletable* if $G \backslash x \backslash y \backslash t$ is connected. If there is a deletable vertex not in $A_1 \cup A_2 \cup A_3$, then $\rho_{G \backslash t}(\{x, y\}) = 2$ and we apply the induction hypothesis to find an induced path. Thus we may assume that all deletable vertices are in $A_1 \cup A_2 \cup A_3$.

If $|A_i| > 1$ and A_i has a deletable vertex t for some $i = 1, 2, 3$, then $\rho_{G\backslash t}(\{x, y\}) = 2$ and so we obtain a sequence by applying the induction hypothesis. So we may assume that if A_i has a deletable vertex, then $|A_i| = 1$.

If there are three deletable vertices t_1, t_2, t_3 in $G\backslash x\backslash y$, then we may assume $A_i = \{t_i\}$. However, $\rho_{G\backslash t_1}(\{x, y\}) = 2$ because A_2, A_3 are nonempty and therefore we obtain an induced path from x to y by the induction hypothesis.

Thus we may assume that $G\backslash x\backslash y$ has at most 2 deletable vertices. So $G\backslash x\backslash y$ has maximum degree at most 2 because otherwise we can choose leaves of a spanning tree of $G\backslash x\backslash y$ using all edges incident with a vertex of the maximum degree. If $G\backslash x\backslash y$ is a cycle, then every vertex is deletable and so $G\backslash x\backslash y$ is a path. Let w be a degree-2 vertex in $G\backslash x\backslash y$. Then $G * w$ has at least 3 deletable vertices and therefore we find a desired sequence $v_1, v_2, \ldots, v_n$ such that $G * w * v_1 * v_2 \cdots * v_n$ has an induced path of length 3 from x to y. □

Lemma 8.32. *Let G be a graph and let x, y be two vertices in G, and let F_1, $F_2, \ldots, F_c$ be the components of $G\backslash x\backslash y$. If $\rho^*_G(\{x, y\}, F_i) = 2$ for all $1 \leqslant i \leqslant c$, then G has a vertex-minor isomorphic to $K_c \boxminus K_c$.*

Proof. We proceed by induction on $|V(G)| + |E(G)|$.

Suppose that $G[V(F_i) \cup \{x, y\}]$ is not an induced path of length 3 from x to y. By Lemma 8.31, there exists a sequence $v_1, v_2, \ldots, v_n \in V(F_i)$ such that $G[V(F_i) \cup \{x, y\}] * v_1 * v_2 \cdots * v_n$ has an induced path of length 3 from x to y. If $|V(F_i)| \geqslant 3$, then we delete all vertices in F_i not on this path and apply the induction hypothesis. If $|V(F_i)| = 2$, then $|E(G[V(F_i) \cup \{x, y\}])| > |E(G[V(F_i) \cup \{x, y\}] * v_1 * v_2 * \cdots * v_n)|$ because two vertices in F_i are connected, $G[\{x, y\}, V(F_i)]$ has at least two edges, and $G[V(F_i) \cup \{x, y\}]$ is not an induced path of length 3 from x to y. So we apply the induction hypothesis to $G * v_1 * v_2 * \cdots * v_n$ to obtain a vertex-minor isomorphic to $K_c \boxminus K_c$.

Therefore we may assume that $G[V(F_i) \cup \{x, y\}]$ is an induced path of length 3 from x to y for all i. Thus $G * x * y\backslash x\backslash y$ is indeed isomorphic to $K_c \boxminus K_c$. □

Lemma 8.33. *Let t be a positive integer, and G be a bipartite graph with a bipartition (S, T) such that every vertex in T has degree at least 1. Then either S has a vertex of degree at least $t + 1$ or G has an induced matching of size at least $|T|/t$.*

Proof. We claim that if every vertex in S has degree at most t, then G has an induced matching of size at least $|T|/t$. We proceed by induction on $|T|$. This is trivial if $|T| = 0$. If $0 < |T| \leqslant t$, then we can simply pick an edge to form an induced matching of size 1. So we may assume that $|T| > t$.

We may assume that T has a vertex w of degree 1, because otherwise we can delete a vertex in S and apply the induction hypothesis. Let v be the unique neighbor of w. By the induction hypothesis, $G\backslash v\backslash N_G(v)$ has an induced matching M' of size at least $(|T| - t)/t$. Now $M' \cup \{vw\}$ is a desired induced matching. □

Lemma 8.34. *Let H be a broom in a graph G having n fibers $F_1, F_2, \ldots, F_n$ given with n vertices $v_1, v_2, \ldots, v_n$ in $V(G)\backslash V(H)$ such that*

1. *v_i blocks F_j if and only if $i = j$,*
2. *v_i has a neighbor in F_j if and only if $i \leqslant j$.*

If $n \geqslant R(c + 1, c + 1)$, then G has a vertex-minor isomorphic to P_c.

Proof. If $j > i$, then v_i has a neighbor in F_j, but v_i does not block F_j. Therefore, v_i is adjacent to every vertex in $V(F_j) \cap N_H(x)$ for $j > i$. Since $n \geqslant R(c+1, c+1)$, there exist $1 \leqslant t_1 < t_2 \cdots < t_{c+1} \leqslant n$ such that $\{v_{t_1}, v_{t_2}, \ldots, v_{t_{c+1}}\}$ is a clique or a stable set of G. For $1 \leqslant i \leqslant c+1$, let w_i be a vertex in $V(F_{t_i}) \cap N_H(x)$. Clearly,

$$G[\{v_{t_1}, v_{t_3}, \ldots, v_{t_{2\lceil c/2\rceil-1}}\}, \{w_2, w_4, \ldots, w_{2\lceil c/2\rceil}\}]$$

is isomorphic to $\overline{K_{\lceil c/2\rceil}} \boxslash \overline{K_{\lceil c/2\rceil}}$.

By Lemma 8.3, $\overline{K_{\lceil c/2\rceil}} \boxslash \overline{K_{\lceil c/2\rceil}}$ or $\overline{K_{\lceil c/2\rceil}} \boxslash K_{\lceil c/2\rceil}$ has a vertex-minor isomorphic to P_c. $\square$

Lemma 8.35. *Let H be a broom in a graph G having n fibers $F_1, F_2, \ldots, F_n$. Let $v_1, v_2, \ldots, v_n$ be vertices in $V(G)\backslash V(H)$ such that*

1. *v_i blocks F_j if and only if $i = j$,*
2. *v_i has a neighbor in F_j for all i and j.*

If $n \geqslant R(c+2, c+2)$, then G has a vertex-minor isomorphic to $K_c \boxminus K_c$.

Proof. If $i \neq j$, then v_j does not block F_i and therefore $N_G(v_j) \cap V(F_i) = N_G(x) \cap V(F_i)$. Since $n \geqslant R(c+2, c+2)$, there exist $1 \leqslant t_1 < t_2 \cdots < t_{c+2} \leqslant n$ such that $\{v_{t_1}, v_{t_2}, \ldots, v_{t_{c+2}}\}$ is a clique or a stable set of G.

We claim that for each $1 \leqslant i \leqslant c+2$, there exist a sequence $w_1^{(i)}, w_2^{(i)}, \ldots, w_{k_i}^{(i)}$ of $k_i \geqslant 0$ vertices in $V(F_{t_i})\backslash(N_G(x) \cup N_G(v_{t_i}))$ and $z_i \in V(F_{t_i})$ such that z_i is not adjacent to v_{t_i} in $G * w_1^{(i)} * w_2^{(i)} * \cdots * w_{k_i}^{(i)}$ but z_i is adjacent to v_{t_j} in $G * w_1^{(i)} * w_2^{(i)} * \cdots * w_{k_i}^{(i)}$ for all $j \neq i$.

Let $A_1^{(i)} = (N_G(v_{t_i})\backslash N_G(x)) \cap V(F_{t_i})$, $A_2^{(i)} = (N_G(v_{t_i}) \cap N_G(x)) \cap V(F_{t_i})$ and $A_3^{(i)} = (N_G(x)\backslash N_G(v_{t_i})) \cap V(F_{t_i})$.

If $A_3^{(i)} \neq \varnothing$, then a vertex z_i in $A_3^{(i)}$ satisfies the claim. So we may assume $A_3^{(i)}$ is empty. Then $A_1^{(i)} \neq \varnothing$ and $A_2^{(i)} \neq \varnothing$, otherwise $\rho_G^*(\{v_{t_i}, v_{t_j}\}, V(F_{t_i})) \leqslant 1$ for all $j \neq i$ because $N_G(v_{t_j}) \cap V(F_{t_i}) = N_G(x) \cap V(F_{t_i})$. We choose $a_1^{(i)} \in A_1^{(i)}$ and $a_2^{(i)} \in A_2^{(i)}$ so that the distance from $a_1^{(i)}$ to $a_2^{(i)}$ in F_i is minimum.

Let P_i be a shortest path from $a_1^{(i)}$ to $a_2^{(i)}$ in F_{t_i}. Note that each internal vertex of P_i is not contained in $A_1^{(i)} \cup A_2^{(i)}$. After applying local complementation at all internal vertices of P_i, $a_1^{(i)}$ is adjacent to $a_2^{(i)}$ and v_{t_i}, and non-adjacent to v_{t_j} for all $j \neq i$. So by applying one more local complementation at $a_1^{(i)}$ if necessary, we can delete the edges between $a_2^{(i)}$ and v_{t_j} for all $j \neq i$. And then, $z_i = a_2^{(i)}$ satisfies the claim.

Now, take $G' = G * w_1^{(1)} * \cdots * w_{k_1}^{(1)} * w_1^{(2)} * \cdots * w_{k_2}^{(2)} \cdots * w_1^{(c+2)} * \cdots * w_{k_{c+2}}^{(c+2)}$. Since each $w_k^{(i)}$ has no neighbors in $\{v_{t_1}, v_{t_2}, \ldots, v_{t_{c+2}}\}$ in G, applying local complementation at $w_k^{(i)}$ does not change the adjacency between any two vertices in $\{v_{t_1}, v_{t_2}, \ldots, v_{t_{c+2}}\}$. Thus the induced subgraph of G' on $\{z_1, z_2, \ldots, z_{c+2}\} \cup \{v_{t_1}, v_{t_2}, \ldots, v_{t_{c+2}}\}$ is isomorphic to $\overline{K_{c+2}} \boxtimes \overline{K_{c+2}}$ or $\overline{K_{c+2}} \boxtimes K_{c+2}$, and by Lemma 8.2, G has a vertex-minor isomorphic to $K_c \boxminus K_c$. $\square$

Lemma 8.36. *Let H be a (h, n, ℓ)-broom in a graph G having n fibers $F_1, F_2, \ldots, F_n$ given with n vertices $v_1, v_2, \ldots, v_n$ in $V(G)\backslash V(H)$ such that*

1. *v_i blocks F_j if and only if $i = j$,*
2. *if $i \neq j$, then v_i has no neighbor in F_j.*

If $n \geqslant R(t+(c-1)(c-3), c)$, then G has a vertex-minor isomorphic to P_c, $K_c \boxminus K_c$, or a $(h, t, \ell+1)$-broom.

Proof. Since $n \geqslant R(t+(c-1)(c-3), c)$, there exist $1 \leqslant t_1 < t_2 \cdots < t_k \leqslant n$ such that either

1. $k = c$ and $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$ is a clique in G, or

2. $k = t + (c-1)(c-3)$ and $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$ is a stable set in G.

First, we assume that $k = c$ and $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$ is a clique. For each t_i, since $\rho^*_G(\{x, v_{t_i}\}, V(F_{t_i})) \geqslant 2$, by Lemma 8.31, there exists some sequence $w_1, w_2, \ldots, w_n \in V(F_{t_i})$ of (not necessarily distinct) vertices such that $G[V(F_{t_i}) \cup \{x, v_{t_i}\}] * w_1 * w_2 \cdots * w_n$ has an induced path of length 2 from v_{t_i} to x. By applying local complementation at x, we have a vertex-minor isomorphic to $K_c \boxminus K_c$.

Now, suppose that $k = t + (c-1)(c-3)$ and $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$ is a stable set in G. Let P be the handle of H. If $h + 2 \geqslant c$, then we have P_c as an induced subgraph. Thus we may assume that $h \leqslant c - 3$. We assume that a vertex $y \in V(P)\backslash\{x\}$ adjacent to c vertices in $\{v_1, v_2, \ldots, v_k\}$. Then since $\rho^*_G(\{x, y\}), V(F_i) \cup \{v_{t_i}\}) = 2$ for each i, by Lemma 8.32, we have a vertex-minor isomorphic to $K_c \boxminus K_c$. Thus, every vertex in the handle other than x cannot have more than $c-1$ neighbors in $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$. By deleting at most $(c-1)h$ vertices in $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$, we can remove all edges from $V(P)\backslash\{x\}$ to $\{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$. Since

$$k - (c-1)h \geqslant k - (c-1)(c-3) \geqslant t,$$

we have a vertex-minor isomorphic to a $(h, t, \ell + 1)$-broom. □

Proposition 8.37. *For positive integers c and t, there exists $N = g_2(c, t)$ such that for all integers $\ell \geqslant 2$ and $h \geqslant 1$, every prime graph having a (h, N, ℓ)-broom has a vertex-minor isomorphic to a $(h, t, \ell + 1)$-broom, P_c, or $K_c \boxminus K_c$.*

Proof. Let $N = g_2(c, t) = (c-1)m$, where $m = R(m_1, m_2, m_2, m_2)$, $m_1 = R(t + (c-1)(c-3), c)$, and $m_2 = R(c+2, c+2)$. Let H be a (h, N, ℓ)-broom of G. If a vertex w in $V(G)\backslash V(H)$ blocks c fibers of H, then for each fiber F of them, $\rho^*_G(\{w, x\}, V(F)) = 2$. So by Lemma 8.32, G has a vertex-minor isomorphic to $K_c \boxminus K_c$. Thus, a vertex in $V(G)\backslash V(H)$ can block at most $c - 1$ fibers of H.

For each fiber F of H, there is a vertex $v \in V(G)\backslash V(H)$ that blocks F because G is prime. Thus, by Lemma 8.33, there are $g_2(c, t)/(c-1) = m$ vertices $v_1, v_2, \ldots, v_m$ in $V(G)\backslash V(H)$ and fibers $F_1, F_2, \ldots, F_m$ of H such that for $1 \leqslant i, j \leqslant m$, v_i blocks F_j if and only if $i = j$. For $i \neq j$, either v_i has no neighbor in F_j or v_i has a neighbor in F_j but $\rho^*_G(\{v_i, x\}, V(F_j)) = 1$.

We assume that $V(K_m) = \{1, 2, \ldots, m\}$. We color the edges of K_m such that an edge $\{i, j\}$ is

- green if $N_G(v_i) \cap V(F_j) \neq \varnothing$ and $N_G(v_j) \cap V(F_i) \neq \varnothing$,
- red if $N_G(v_i) \cap V(F_j) \neq \varnothing$ and $N_G(v_j) \cap V(F_i) = \varnothing$,
- yellow if $N_G(v_i) \cap V(F_j) = \varnothing$ and $N_G(v_j) \cap V(F_i) \neq \varnothing$,
- blue if $N_G(v_i) \cap V(F_j) = N_G(v_j) \cap V(F_i) = \varnothing$.

Since $|V(K_m)| = m = R(m_1, m_2, m_2, m_2)$, by Ramsey's theorem, either K_m has a green clique of size m_1, or K_m has a monochromatic clique of size m_2 which is red, yellow, or blue.

If K_m has a red clique C of size m_2, then for $i, j \in C$, v_i has a neighbor in F_j if and only if $i \leqslant j$. Since $m_2 \geqslant R(c+1, c+1)$, by Lemma 8.34, G has a vertex-minor isomorphic to P_c.

Similarly, if K_m has a yellow clique C of size m_2, by Lemma 8.34, G has a vertex-minor isomorphic to P_c.

If K_m has a blue clique C of size m_2, then for distinct $i, j \in C$, v_i has a neighbor in F_j. Since $m_2 = R(c+2, c+2)$, by Lemma 8.35, G has a vertex-minor isomorphic to $K_c \boxminus K_c$.

If K_m has a green clique C of size m_1, then for distinct $i, j \in C$, v_i has no neighbor in F_j. Since $m_1 = R(t + (c-1)(c-3), c)$, by Lemma 8.36, G has a vertex-minor isomorphic to P_c, $K_c \boxminus K_c$, or a $(h, t, \ell+1)$-broom. $\square$

8.5.2 Increasing the height of a broom

Proposition 8.38. *For positive integers c, t, there exists $N = g_3(c,t)$ such that for $h \geqslant 1$, every prime graph having a $(h, 1, N)$-broom has a vertex-minor isomorphic to a $(h+1, t, 1)$-broom or P_c.*

Proof. Let $N = g_3(c,t) = g_0(c, 2t)$ where g_0 is given in Theorem 8.28. Suppose that G has a $(h, 1, N)$-broom H and let x be the center of H. Let F be the fiber of H.

Since F is connected, by Theorem 8.28, F has an induced subgraph isomorphic to P_c, or F has a vertex-minor isomorphic to K_{2t+1}. We may assume that F has an induced subgraph F' isomorphic to K_{2t+1}. Let $P = p_1 p_2 \dots p_m$ be a shortest path from $p_1 = x$ to F' in H. Note that $m \geqslant 2$ and p_{m-1} is adjacent to at least one vertices of F'. Let $S = N_H(p_{m-1}) \cap V(F')$.

We claim that there exists a vertex $v \in V(F')$ such that $(G * v)[V(F) \cup \{x\}]$ has an induced path of length at least $m-1$ from x, and the last vertex of the path has t neighbors in F' which form a stable set in G.

If $|S| \leqslant t$, then choose $p_{m+1} \in V(F') \backslash S$ and we delete $S \backslash p_m$ from F'. And by applying local complementation at p_{m+1}, we obtain a path from x to p_{m+1} such that p_{m+1} has t neighbors in F' which form a stable set.

If $|S| \geqslant t+1$, then by applying local complementation at p_m, we obtain a path from x to p_m such that p_m has t neighbors in F' which form a stable set. Thus, we prove the claim.

Since $m \geqslant 2$, the union of the handle of H and the path in the claim form a path of length at least $h+1$, and the last vertex of the path has t neighbors which form a stable set in F'. Therefore, G has a vertex-minor isomorphic to a $(h+1, t, 1)$-broom. $\square$

Proposition 8.39. *For positive integers c, t, there exists $N = g_4(c,t)$ such that for all $h \geqslant 1$, every prime graph having a $(h, N, 1)$-broom has a vertex-minor isomorphic to a $(h+1, t, 1)$-broom, P_c, or $K_c \boxminus K_c$.*

Proof. By Proposition 8.38, there exists N_0 depending only on c and t such that every prime graph having a $(h, 1, N_0)$-broom has a vertex-minor isomorphic to a $(h+1, t, 1)$-broom or P_c. By applying Proposition 8.37 $(N_0 - 2)$ times, we deduce that there exists N_1 such that every prime graph having a $(h, N_1, 2)$-broom has a vertex-minor isomorphic to a $(h, 1, N_0)$-broom, P_c, or $K_c \boxminus K_c$. By Proposition 8.29, there exists N such that every prime graph having a $(h, N, 1)$-broom has a vertex-minor isomorphic to a $(h, N_1, 2)$-broom, P_c, or $K_c \boxminus K_c$. $\square$

We are now ready with all necessary lemmas to prove Proposition 8.27.

Proof of Proposition 8.27. By Theorem 4.2, every prime graph on at least 5 vertices has a vertex-minor isomorphic to C_5 and P_4 is a vertex-minor of C_5. Therefore we may assume that $c \geqslant 5$.

By applying Proposition 8.39 $(c-3)$ times, we deduce that there exists a big integer t depending only on c such that every prime graph G with a $(1, t, 1)$-broom has a vertex-minor isomorphic to a $(c-2, 1, 1)$-broom, P_c, or $K_c \boxminus K_c$. Since a $(c-2, 1, 1)$-broom is isomorphic to P_c and a $(1, t, 1)$-broom is isomorphic to $K_{1,t+1}$, we conclude that every prime graph having a vertex-minor isomorphic to $K_{1,t+1}$ has a vertex-minor isomorphic to P_c or $K_c \boxminus K_c$. By Theorem 8.4, there exists N such that every connected graph on at least N vertices has a vertex-minor isomorphic to $K_{1,t+1}$. This completes the proof. $\square$

8.6 Why optimal?

Theorem 8.1 states that sufficiently large prime graphs must have a vertex-minor isomorphic to C_n or $K_n \boxminus K_n$. But do we really need these two graphs? To justify why we need both, we should show that for some n, C_n is not a vertex-minor of $K_N \boxminus K_N$ for all N and similarly $K_n \boxminus K_n$ is not a vertex-minor of C_N for all N, because C_n and $K_n \boxminus K_n$ are also prime.

Proposition 8.40. *Let n be a positive integer.*

1. *$K_3 \boxminus K_3$ is not a vertex-minor of C_n.*
2. *C_7 is not a vertex-minor of $K_n \boxminus K_n$.*

Since C_7 is a vertex-minor of C_n for all $n \geqslant 7$, the above proposition implies that C_n is not a vertex-minor of $K_N \boxminus K_N$ when $n \geqslant 7$. Similarly $K_n \boxminus K_n$ is not a vertex-minor of C_N for all $n \geqslant 3$.

We can classify all non-trivial prime vertex-minors of a cycle graph.

Lemma 8.41. *If a prime graph H on at least 5 vertices is a vertex-minor of C_n, then H is locally equivalent to a cycle graph.*

Proof. We proceed by induction on n. If $n = 5$, then it is trivial. Let us assume $n > 5$. Suppose $|V(H)| < |V(C_n)|$. By Lemma 1.8, H is a vertex-minor of $C_n \backslash v$, $C_n * v \backslash v$, or $C_n \wedge vw \backslash v$ for a neighbor w of v.

If H is vertex-minor of $C_n * v \backslash v$, then we can apply the induction hypothesis because $C_n * v \backslash v$ is isomorphic to C_{n-1}.

By Lemma 4.3, H cannot be a vertex-minor of $C_n \backslash v$ because $C_n \backslash v$ has no prime induced subgraph on at least 5 vertices.

Thus we may assume that H is a vertex-minor of $C_n \wedge vw \backslash v$ for a neighbor w of v. Again, by Lemma 4.3, H is isomorphic to a vertex-minor of C_{n-2}. $\square$

Classifying prime vertex-minors of $K_n \boxminus K_n$ turns out to be more tedious. Instead of identifying prime vertex-minors of $K_n \boxminus K_n$, we focus on characterizing prime vertex-minors on 7 vertices to prove (2) of Proposition 8.40.

Instead of $K_n \boxminus K_n$, we will first consider H_n. Let H_n be the graph having two specified vertices called *roots* and n internally disjoint paths of length 3 joining the roots. Let J_n be the graph obtained from H_n by adding a common neighbor of the two roots. Then H_n has $2n + 2$ vertices and J_n has $2n + 3$ vertices, see Figure 8.8. It is easy to observe the following.

Lemma 8.42. *Let H be a prime vertex-minor of H_n on at least 5 vertices. If $|V(H_n)| - |V(H)| \geqslant 3$, then J_{n-1} has a vertex-minor isomorphic to H.*

Proof. We may assume $n \geqslant 3$. Since at most 2 vertices of H_n have degree other than 2, there exists $v \in V(H_n) \backslash V(H)$ of degree 2 in H_n. Let w be the neighbor of v having degree 2 in H_n. Let $av'w'b$ be a path of length 3 from a to b in H_n such that $\{v, w\} \neq \{v', w'\}$.

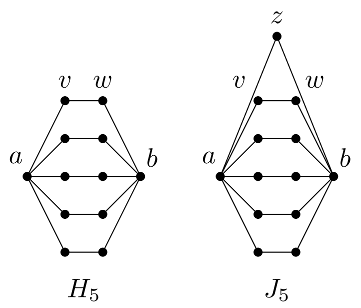


Figure 8.8: The graphs H_5 and J_5.

By Lemma 1.8, H is a vertex-minor of either $H_n \backslash v$, $H_n * v \backslash v$ or $H_n \wedge vw \backslash v$. If H is a vertex-minor of $H_n * v \backslash v$, then H is isomorphic to a vertex-minor of J_{n-1}, because $H_n * v \backslash v$ is isomorphic to J_{n-1}.

Since w has degree 1 in $H_n \backslash v$, by Lemma 4.3, if H is a vertex-minor of $H_n \backslash v$, then H is isomorphic to a vertex-minor of $H_n \backslash v \backslash w$. Since $H_n \backslash v \backslash w$ is isomorphic to H_{n-1} and H_{n-1} is an induced subgraph of J_{n-1}, H is isomorphic to a vertex-minor of J_{n-1}.

Similarly, if H is a vertex-minor of $H_n \wedge vw \backslash v$, then H is isomorphic to a vertex-minor of $H_n \wedge vw \backslash v \backslash w$. Clearly, $(H_n \wedge vw \backslash v \backslash w) \wedge v'w'$ is isomorphic to H_{n-1}. Since H_{n-1} is an induced subgraph of J_{n-1}, H is isomorphic to a vertex-minor of J_{n-1}, as required. □

Lemma 8.43. *Let H be a prime vertex-minor of J_n on at least 5 vertices. If $|V(J_n)| - |V(H)| \geqslant 2$, then H_n has a vertex-minor isomorphic to H.*

Proof. We may assume $n \geqslant 2$. Let a, b be the roots of J_n, azb be the path of length 2, and $avwb$ be a path of length 3 from a to b.

Case 1: Suppose that $V(J_n) \backslash V(H)$ has a degree-2 vertex on a path of length 3 from a to b. We may assume that it is v by symmetry. By Lemma 1.8, H is a vertex-minor of $J_n \backslash v$, $J_n * v \backslash v$, or $J_n \wedge vw \backslash v$.

If H is a vertex-minor of $J_n \backslash v$, then H is isomorphic to a vertex-minor of $J_n \backslash v \backslash w$ by Lemma 4.3, because w has degree 1 in $J_n \backslash v$. Similarly, if H is a vertex-minor of $J_n \wedge vw \backslash v$, then H is isomorphic to a vertex-minor of $J_n \wedge vw \backslash v \backslash w$. Clearly, $J_n \backslash v \backslash w$ and $(J_n \wedge vw \backslash v \backslash w) * z$ are isomorphic to J_{n-1}, and J_{n-1} is a vertex-minor of H_n.

If H is a vertex-minor of $J_n * v \backslash v$, then by Lemma 4.3, H is isomorphic to a vertex-minor of $J_n * v \backslash v \backslash w$, which is isomorphic to J_{n-1}, because w and z have the same set of neighbors in $J_n * v \backslash v$. Since J_{n-1} is a vertex-minor of H_n, H is isomorphic to a vertex-minor of H_n. This proves the lemma in Case 1.

Case 2: Suppose that $z \in V(J_n) \backslash V(H)$. Then by Lemma 1.8, H is a vertex-minor of $J_n \backslash z$, $J_n * z \backslash z$, or $J_n \wedge az \backslash z$. Since $J_n \backslash z$ and $(J_n * z \backslash z) \wedge vw$ are isomorphic to H_n, we may assume that H is a vertex-minor of $J_n \wedge az \backslash z$. However, $J_n \wedge az \backslash z$ has no prime induced subgraph on at least 5 vertices and therefore by Lemma 4.3, H cannot be a vertex-minor of $J_n \wedge az \backslash z$, contradicting our assumption.

Case 3: Suppose that a or b is contained in $V(J_n) \backslash V(H)$. By symmetry, let us assume $a \in V(J_n) \backslash V(H)$. By Lemma 1.8, H is a vertex-minor of $J_n \backslash a$, $J_n * a \backslash a$, or $J_n \wedge az \backslash a$.

Since $J_n \backslash a$ has no prime induced subgraph on at least 5 vertices, H cannot be a vertex-minor of $J_n \backslash a$ by Lemma 4.3.

Suppose H is a vertex-minor of $J_n \wedge az \backslash a$. By the definition of pivoting, b is adjacent to all vertices of $N_{J_n}(a) \backslash \{z\}$ in $J_n \wedge az \backslash a$. We can remove all these edges between b and $N_{J_n}(a) \backslash \{z\}$ by applying local

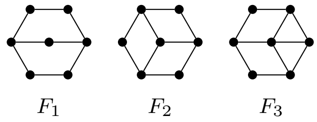


Figure 8.9: Graphs F_1, F_2 and F_3.

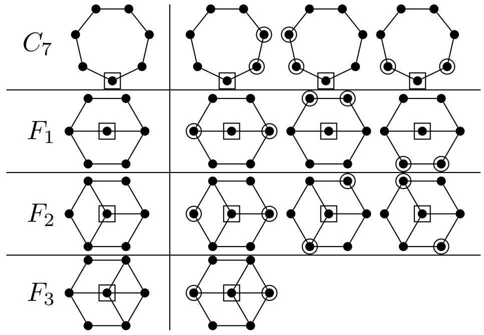


Figure 8.10: List of all 3-vertex sets having cut-rank 2 containing a fixed vertex x denoted by a square.

complementation on all vertices of $N_{J_n}(b)\backslash\{z\}$ in $J_n \wedge az\backslash a$. Thus, H_n is locally equivalent to $J_n \wedge az\backslash a$, and H is isomorphic to a vertex-minor of H_n.

Now suppose that H is a vertex-minor of $J_n * a\backslash a$. By the definition of local complementation, $N_{J_n}(a)$ forms a clique in $J_n * a\backslash a$. So, b is adjacent to all vertices of $N_{J_n}(a)\backslash\{z\}$ in $(J_n * a\backslash a) * z$. Similarly in the above case, by applying local complementation on all vertices of $N_{J_n}(b)\backslash\{z\}$ in $(J_n * a\backslash a) * z$, we can remove all edges between b and $N_{J_n}(a)\backslash\{z\}$ in $(J_n * a\backslash a) * z$. Finally, by pivoting vw, we can remove the edge bz, and therefore, $J_n * a\backslash a$ is locally equivalent to H_n. Thus, H is isomorphic to a vertex-minor of H_n. □

Let F_1, F_2, F_3 be the graphs in Figure 8.9.

Lemma 8.44. *Let $n \geqslant 3$ be an integer. If a prime graph H is a vertex-minor of H_n and $|V(H)| = 7$, then H is locally equivalent to F_1, F_2, or F_3.*

Proof. We proceed by induction on n. If $n = 3$, then let H be a prime 7-vertex vertex-minor of H_3. Let $axyb$ be a path from a root a to the other root b in H_3. By symmetry, we may assume that $V(H_3)\backslash V(H) = \{x\}$ or $\{a\}$. By Lemma 1.8, H is locally equivalent to $H_3\backslash x$, $H_3 * x\backslash x$, $H_3 \wedge xa\backslash x$, $H_3\backslash a$, $H_3 * a\backslash a$, or $H_3 \wedge ab\backslash a$. The conclusion follows because $H_3\backslash x$, $H_3 \wedge xy\backslash x$, $H_3\backslash a$ are not prime and $H_3 * x\backslash x$, $H_3 \wedge ax\backslash a$, and $H_3 * a\backslash a$ are isomorphic to F_1, F_2, and F_3, respectively.

Suppose $n > 3$. By Lemma 8.42, every 7-vertex prime vertex-minor is also isomorphic to a vertex-minor of J_{n-1}. By Lemma 8.43, it is isomorphic to a vertex-minor of H_{n-1}. The conclusion follows from the induction hypothesis. □

Lemma 8.45. *The graphs F_1, F_2, F_3 are not locally equivalent to C_7.*

Proof. Suppose that F_i is locally equivalent to C_7. Then $\rho_{F_i}(X) = \rho_{C_7}(X)$ for all $X \subseteq V(C_7)$ by Lemma 2.1. Let x be the vertex in the center of F_i, see Figure 8.10. By symmetry of C_7, we may assume that x is mapped to a particular vertex in C_7. Figure 8.10 presents all vertex subsets of size 3 having cut-rank 2 and containing x in graphs C_7, F_1, F_2, F_3. It is now easy to deduce that no bijection on the vertex set will map these subsets correctly. □

We are now ready to prove Proposition 8.40.

Proof of Proposition 8.40. (1) By Lemma 8.41, it is enough to check that $K_3 \boxminus K_3$ is not locally equivalent to C_6. This can be checked easily.

(2) By applying local complementation at roots, we can easily see that H_n has a vertex-minor isomorphic to $K_n \boxminus K_n$. Lemma 8.44 states that all 7-vertex prime vertex-minors of H_n are F_1, F_2, and F_3. Lemma 8.45 proves that none of them are locally equivalent to C_7. Thus H_n has no vertex-minor isomorphic to C_7 and therefore $K_n \boxminus K_n$ has no vertex-minor isomorphic to C_7. $\square$

Chapter 9. An exact algorithm to compute linear rank-width

We discuss exact algorithms to compute linear rank-width. We first verify that computing linear rank-width on bipartite graphs is NP-hard in Section 9.1. For this, we use the relation between matroid path-width and linear rank-width.

We naturally ask which graph classes allow a polynomial-time algorithm to compute linear rank-width. Previously, Adler and Kanté [2] proved a linear-time algorithm for trees, and it was the only known exact algorithm to compute linear rank-width which runs in polynomial time.

We provide a polynomial-time algorithm to compute the linear rank-width of distance-hereditary graphs.

Theorem 9.1. *The linear rank-width of distance-hereditary graphs with n vertices can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$.*

We remark that computing the path-width of distance-hereditary graphs is NP-hard [129]. Therefore, our result provides a difference between path-width and linear rank-width on distance-hereditary graphs. We use the notion of limbs and the characterization of linear rank-width on distance-hereditary graphs, developed in Chapter 5.

As a corollary of Theorem 9.1, we prove that the path-width of matroids of branch-width at most 2 can be computed in polynomial time, provided that the matroid is given by an independent set oracle.

Corollary 9.2. *The linear rank-width of n-element matroid branch-width at most 2 with a given independent set oracle can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$.*

9.1 NP-hardness of computing linear rank-width

Here, we prove that computing the linear rank-width of a graph is NP-hard.

Theorem 9.3. *The problem of computing the linear rank-width of a graph is NP-hard.*

We first remark that the computation of the path-width of a graph is NP-hard [6, 19].

Theorem 9.4 (Arnborg, Corneil, and Proskurowski [6])**.** *The problem of computing the path-width of a graph is NP-hard.*

Kashyap [123] showed an analogous result for a matroid path-width of graphic matroid.

Theorem 9.5 (Kashyap [123])**.** *For a fixed field $\mathbb{F}$, computing the path-width of a matroid representable over $\mathbb{F}$ is NP-hard.*

For linear rank-width, we may use a direct relation between the branch-width of a binary matroid and the rank-width of its fundamental graph, mentioned in Section 3.4. We recall the following.

Proposition 3.17 (Oum [150]). *Let G be a bipartite graph with a bipartition (A, B) and let $M := M(G, A, B)$. Then $\mathrm{rw}(G) = \mathrm{bw}(M) - 1$ and $\mathrm{lrw}(G) = \mathrm{pw}(M) - 1$.*

Proof of Theorem 9.3. Given a binary matroid M, we want to produce a bipartite graph G in polynomial time such that $\mathrm{pw}(M)$ can be computed from $\mathrm{lrw}(G)$. Let M be a binary matroid. We first run a greedy algorithm to find a base B of M [156, Section 1.8]. After choosing one base B, for each $e \in B$ and $e' \in E \backslash B$, we test whether $(B\backslash\{e\}) \cup \{e'\}$ is again a base, and we create the fundamental graph G with respect to M in polynomial time. By Proposition 3.17 and Theorem 9.5, $\mathrm{lrw}(G) = \mathrm{pw}(M) - 1$, and NP-hardness of computing path-width of binary matroids implies that computing linear rank-width of bipartite graphs is NP-hard. □

9.2 A polynomial-time algorithm for distance-hereditary graphs

We describe an algorithm to compute the linear rank-width of distance-hereditary graphs. Since the linear rank-width of a graph is equal to the maximum over all linear rank-width of its connected components, we will focus on connected distance-hereditary graphs.

The main idea consists of rooting the canonical split decomposition D of a connected distance-hereditary graph and associating with each bag B of D a canonical limb $\mathcal{L} := \widetilde{\mathcal{L}}_D[B', y]$ with its parent B' and computing the linear rank-width of $\widehat{\mathcal{L}}$. In order to compute the linear rank-width of $\widehat{\mathcal{L}}$, we will use the linear rank-width of the graphs $\widehat{\mathcal{L}_1}, \ldots, \widehat{\mathcal{L}_p}$ where $\mathcal{L}_1, \ldots, \mathcal{L}_p$ are the limbs associated with the children of B.

Let D be a canonical split decomposition of a connected distance-hereditary graph and let B be a bag of D, y be a vertex represented by a vertex of B. Let $\mathcal{L} := \mathcal{L}_D[B, y]$ be a limb and $T_{\mathcal{L}}$ be its split decomposition tree. Let B' be the bag in the component of $D\backslash V(B)$ containing y that has a neighbor in B, and let w be the node of T_D such that its corresponding bag is B'. One easily checks that the split decomposition tree $\widetilde{T}$ of $\widetilde{\mathcal{L}}_D[B, y]$ is obtained similarly as $\widetilde{\mathcal{L}}_D[B, y]$, namely,

1. $\widetilde{T} = T_{\mathcal{L}}$ if $|B'| \geqslant 4$ or $|V(T_{\mathcal{L}})| = 1$,
2. $\widetilde{T} = T_{\mathcal{L}}\backslash\{w\}$ if $|B'| = 3$ and w has two neighbors in T_D,
3. If $|B'| = 3$ and w has 3 neighbors in T_D, then let T' be obtained from $T_{\mathcal{L}}\backslash\{w\}$ by adding an edge e' between the two neighbors of w in T_D. In this case, either $\widetilde{T} = T'$ or $\widetilde{T}$ is obtained from T' by contracting the edge e'.

As a consequence, we may assume that every node of $\widetilde{T}$, but at most one, is also a node of T_D. Similarly, every edge of $\widetilde{T}$, but at most one, is an edge of T_D.

We now define the notion of *rooted split decomposition trees of limbs.* A split decomposition tree is *rooted* if we distinguish either a node or an edge and call it *root.* Let T be a rooted decomposition tree with root r. A node v is a descendant of a node v' if either $r = vv'$, or v' is in the unique path from the root to v; if moreover v and v' are adjacent we call v a *child* of v' and v' the *parent* of v. Observe from the definition of descendants that if $r = vv'$ then v is the parent of v' and also v' is the parent of v. Two nodes v and v' are called *comparable* if one node is a descendant of the other one. Otherwise, they are called *incomparable.* Recall that for each node v of T and each canonical split decomposition D with T as a split decomposition tree we write $\mathsf{bag}_D(v)$ to denote the bag of D with which it is in correspondence, and we let $\mathsf{pbag}_D(v)$ be $\mathsf{bag}_D(v')$ with the parent v' of v.

Let D be a canonical split decomposition of a connected distance-hereditary graph and let T be its split decomposition tree rooted at r. For each canonical limb $\widetilde{\mathcal{L}}$ we root its split decomposition tree $\widetilde{T}$ as follows.

1. If the root (node or edge) r of T exists in $\widetilde{\mathcal{L}}_D[B,y]$, then we assign it as the root of $\widetilde{T}$. In the other cases, we can easily see that either the root node is removed or a node incident with the root edge is removed.
2. If the removed node has one neighbor, then we assign this neighbor as the root of $\widetilde{T}$.
3. If the removed node has two neighbors in T and they are linked by a new edge in $\widetilde{T}$, then we assign the new edge as the root of $\widetilde{T}$.
4. If the removed node has two neighbors in T and they are identified in $\widetilde{T}$, then we assign the new node as the root of $\widetilde{T}$.

The following observation is easy to check from the definition of rooted split decomposition trees of canonical limbs.

Fact 1. *Let D be a canonical split decomposition and let T_D be its rooted split decomposition tree. If w is a non-root node of the rooted split decomposition tree $\widetilde{T}$ of a canonical limb $\widetilde{\mathcal{L}}_D[B,y]$, then w is also a non-root node of T_D with the property that $V(\mathsf{bag}_D(w)) = V(\mathsf{bag}_{\widetilde{\mathcal{L}}_D[B,y]}(w))$.*

For two disjoint bags B and B' we denote by $\mathcal{T}_D[B,B']$ the component of $D \backslash V(B)$ containing B'. For conciseness, for every non-root node v of T, we define that

$$f_1(D,v) := f_D(\mathsf{pbag}_D(v), \mathcal{T}_D[\mathsf{pbag}_D(v), \mathsf{bag}_D(v)])$$
$$f_2(D,v) := f_D(\mathsf{bag}_D(v), \mathcal{T}_D[\mathsf{bag}_D(v), \mathsf{pbag}_D(v)]).$$

A node v of T is called *k-critical* if $f_1(D,v) = k$ and v has two children v_1 and v_2 such that $f_1(D,v_1) = f_1(D,v_2) = k$.

From now on, let G be a fixed connected distance-hereditary graph and we fix a root r for the split decomposition tree T_D of the canonical split decomposition D of G. We remark that since G has rank-width at most 1, by Lemma 3.5, $\mathrm{lrw}(G) \leqslant \lfloor \log_2 |V(G)| \rfloor$. For convenience, we denote by $\mathsf{lrwbd} := \lfloor \log_2 |V(G)| \rfloor$.

For each non-root node v of T_D and each $1 \leqslant j \leqslant \mathsf{lrwbd}$, we define the following.

1. Let D^v_{lrwbd} be any canonical limb $\widetilde{\mathcal{L}}_D[\mathsf{pbag}_D(v), y]$ with an unmarked vertex y in

$$\mathcal{T}_D[\mathsf{pbag}_D(v), \mathsf{bag}_D(v)]$$

   represented by a marked vertex in $\mathsf{pbag}_D(v)$; and let T^v_{lrwbd} be the rooted split decomposition tree of D^v_{lrwbd}.
2. Let

$$\alpha^v_j := \max\{f_1(D^v_j, w) : w \text{ is a non-root node of } T^v_j\}.$$

   If $\alpha^v_j \neq j$, then let $D^v_{j-1} = D^v_j$ and $T^v_{j-1} := T^v_j$. If $\alpha^v_j = j$, then

(a) if (T_j^v has a node with at least 3 children w such that $f_1(D_j^v, w) = j$) or (T_j^v has two incomparable nodes v_1 and v_2 with a j-critical node v_1 and $f(D_j^v, v_2) = j$) or (T_j^v has no j-critical nodes), then let $D_{j-1}^v = D_j^v$ and $T_{j-1}^v := T_j^v$.

(b) if T_j^v has the unique j-critical node v_c, then let $D_{j-1}^v := \widetilde{\mathcal{L}}_{D_j^v}[\mathsf{bag}_{D_j^v}(v_c), y]$ with y any unmarked vertex in $\mathcal{T}_{D_j^v}[\mathsf{bag}_{D_j^v}(v_c), \mathsf{pbag}_{D_j^v}(v_c)]$ represented by a marked vertex in $\mathsf{bag}_{D_j^v}(v_c)$ and let T_{j-1}^v be the rooted split decomposition tree of D_{j-1}^v.

The following can be derived from Theorem 5.11 and Proposition 5.10.

Proposition 9.6. *Let $0 \leqslant j \leqslant \mathsf{lrwbd}$. Let v be a non-root node of T_D such that $\alpha_j^v \leqslant j$ and T_j^v contains neither*

- *a node with at least 3 children w such that $f_1(D_j^v, w) = \alpha_j^v$, nor*
- *two incomparable nodes v_1 and v_2 such that v_1 is a α_j^v-critical node and $f_1(D_j^v, v_2) = \alpha_j^v$.*

Let w be a α_j^v-critical node of T_j^v. Then w is the unique α_j^v-critical vertex of T_j^v. Moreover, $\mathrm{lrw}(\widehat{D_j^v}) = \alpha_j^v + 1$ if and only if $\mathrm{lrw}(\widehat{D_{j-1}^v}) = f_2(D_j^v, w) = \alpha_j^v$.

Proof. Let $k := \alpha_j^v$. We first show that w is the unique k-critical node of T_j^v. Let w' be a k-critical node of T_j^v that is distinct from w. From the second assumption, w and w' must be comparable in T_j^v. Without loss of generality, we may assume that w is a descendant of w' in T_j^v. Then by the definition of k-criticality, w' has a child w'' such that $f_1(D_j^v, w'') = k$ and w is not a descendant of w'' in T_j^v, contradicting to the second assumption.

Now we claim that $\mathrm{lrw}(\widehat{D_j^v}) = k + 1$ if and only if $f_2(D_j^v, w) = k$. By the assumption on k and by Theorem 5.11 $\mathrm{lrw}(\widehat{D_j^v}) \leqslant k + 1$. Also, by definition one can see that $D_\ell^v = D_j^v$ for all $k \leqslant \ell \leqslant j$. Let w_1 and w_2 be the two children of w such that $f_1(D_j^v, w_1) = f_1(D_j^v, w_2) = k$. By assumption for all the other children w' of w we have $f_1(D_j^v, w') \leqslant k - 1$. So, by Theorem 5.11 it remains to check $f_{D_j^v}(\mathsf{bag}_{D_j^v}(w), \mathsf{pbag}_{D_j^v}(w)) = f_2(D_j^v, w) = \mathrm{lrw}(\widehat{D_{j-1}^v})$ to conclude whether $\mathrm{lrw}(\widehat{D_j^v}) = k + 1$. Therefore, we can conclude that $\mathrm{lrw}(\widehat{D_{j-1}^v}) = f_2(D_j^v, w) = k$ implies that $\mathrm{lrw}(\widehat{D_j^v}) \geqslant k + 1$.

For the forward direction, suppose that $\mathrm{lrw}(\widehat{D_j^v}) \geqslant k + 1$. Since T_j^v contains no node having at least three children w such that $f_1(D_j^v, w) = k$, by Theorem 5.11, there should exist a k-critical node v_c of T_j^v such that $f_2(D_j^v, v_c) = k$. Since w is the unique k-critical node of T_j^v, $w = v_c$ and $f_2(D_j^v, w) = \mathrm{lrw}(\widehat{D_{j-1}^v}) = k$, as required. □

Let v be a non-root node of T_D and let $k := \max\{\mathrm{lrw}(\widehat{D_{\mathsf{lrwbd}}^v}) : v \text{ is a non-root vertex of } T_D\}$. From Theorem 5.11, we can easily observe that $k \leqslant \mathrm{lrw}(\widehat{D_{\mathsf{lrwbd}}^v})) \leqslant k + 1$. We discuss now how to determine it precisely. By Proposition 9.6, the computation of $\mathrm{lrw}(\widehat{D_{\mathsf{lrwbd}}^v}))$ can be reduced to the computation of $f_2(D_{\mathsf{lrwbd}}^v, v_c)$ where v_c is the unique k-critical node of D_{lrwbd}^v. In order to compute it, we can recursively call the algorithm on $\widehat{D_{k-1}^v}$. However, we will prove that these recursive calls are not needed if we compute more than the linear rank-width.

Lemma 9.7. *Let v be a non-root vertex of T_D. Let i be an integer such that $0 \leqslant i < \mathsf{lrwbd}$. If $\alpha_i^v \leqslant i$, then $\alpha_{i+1}^v \leqslant i + 1$.*

Proof. Suppose that $\alpha_{i+1}^v \geqslant i + 2$. By the definition of D_i^v, $D_i^v = D_{i+1}^v$ and therefore, $\alpha_i^v \geqslant i + 2$, which yields a contradiction. □

Before describing the algorithm we prove the following which states that the choice of canonical limbs in the definition of the D_i^v's is not important.

Proposition 9.8. *Let v be a non-root vertex of T_D. Let i be an integer such that $0 \leqslant i \leqslant \mathsf{lrwbd}$ and $\alpha_i^v \leqslant i$. Let w be a non-root node of T_i^v. Then, $\alpha_i^w \leqslant i$ and D_i^w is locally equivalent to $\widetilde{\mathcal{L}}_{D_i^v}[\mathsf{pbag}_{D_i^v}(w), y]$ for any unmarked vertex y in $\mathcal{T}_{D_i^v}[\mathsf{pbag}_{D_i^v}(w), \mathsf{bag}_{D_i^v}(w)]$ represented by a marked vertex in $\mathsf{pbag}_{D_i^v}(w)$.*

Proof. Let w be a non-root vertex of T_i^v. By Fact 1, for each $i+1 \leqslant j \leqslant \mathsf{lrwbd}$, $w \in V(T_j^v)$ and hence $w \in V(T_D)$. Moreover, since $\alpha_i^v \leqslant i$, by Lemma 9.7, $\alpha_j^v \leqslant j$ for all $i+1 \leqslant j \leqslant \lfloor \log_2 |V(G)| \rfloor$. Now, we claim that for each $i \leqslant j \leqslant \mathsf{lrwbd}$ and each unmarked vertex y in $\mathcal{T}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), \mathsf{bag}_{D_j^v}(w)]$ represented by a marked vertex in $\mathsf{pbag}_{D_j^v}(w)$, $\widetilde{\mathcal{L}}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), y]$ is locally equivalen to D_j^w. We prove it by induction on $(\mathsf{lrwbd} - j)$.

If $j = \mathsf{lrwbd}$, then D_{lrwbd}^v and D_{lrwbd}^w are both limbs in D, and hence by Proposition 5.17, we can conclude that D_{lrwbd}^w is locally equivalent to a canonical limb

$$\widetilde{\mathcal{L}}_{D_{\mathsf{lrwbd}}^v}[\mathsf{pbag}_{D_{\mathsf{lrwbd}}^v}(w), y]$$

with an unmarked vertex y in $\mathcal{T}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), \mathsf{bag}_{D_j^v}(w)]$ represented by a marked vertex in $\mathsf{pbag}_{D_j^v}(w)$.

Now let us assume that $i \leqslant j < \mathsf{lrwbd}$, and let y be an unmarked vertex in

$$\mathcal{T}_{D_{j+1}^v}[\mathsf{pbag}_{D_{j+1}^v}(w), \mathsf{bag}_{D_{j+1}^v}(w)]$$

represented by a marked vertex in $\mathsf{pbag}_{D_{j+1}^v}(w)$. By induction hypothesis D_{j+1}^w is locally equivalent to $\widetilde{\mathcal{L}}_{D_{\mathsf{lrwbd}}^v}[\mathsf{pbag}_{D_{\mathsf{lrwbd}}^v}(w), y]$. Assume first that $\alpha_{j+1}^v \leqslant j$. Then, by Proposition 5.17 and Lemma 4.8 we can conclude that $\alpha_{j+1}^w \leqslant j$. Since by definition in that case $D_j^v = D_{j+1}^v$ and $D_j^w = D_{j+1}^w$, we can conclude the statement.

Assume now that $\alpha_{j+1}^v = j+1$. Since $\alpha_{j+1}^v = j+1$ and $\alpha_j^v \leqslant j$, T_{j+1}^v should have a unique $(j+1)$-critical vertex v_c such that $D_j^v := \widetilde{\mathcal{L}}_{D_{j+1}^v}[\mathsf{bag}_{D_{j+1}^v}(v_c), y_c]$ with y_c some unmarked vertex in $\mathcal{T}_{D_{j+1}^v}[\mathsf{bag}_{D_{j+1}^v}(v_c), \mathsf{pbag}_{D_{j+1}^v}(v_c)]$ represented by a marked vertex in $\mathsf{bag}_{D_{j+1}^v}(v_c)$. Let y be any unmarked vertex in $\mathcal{T}_{D_{j+1}^v}[\mathsf{pbag}_{D_{j+1}^v}(w), \mathsf{bag}_{D_{j+1}^v}(w)]$ represented by a marked vertex in $\mathsf{pbag}_{D_{j+1}^v}(w)$ and let y' be any unmarked vertex in $\mathcal{T}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), \mathsf{bag}_{D_j^v}(w)]$ represented by a marked vertex in $\mathsf{pbag}_{D_j^v}(w)$. We distinguish two cases: either v_c is incomparable with w in T_{j+1}^v, or v_c is a descendant of w in T_{j+1}^v. Since w is a vertex of T_j^v, w cannot be a descendant of v_c.

Case 1. *v_c is incomparable with w in T_{j+1}^v.*

Since v_c is incomparable with w in T_{j+1}^v and v_c is the unique $(j+1)$-critical vertex in T_{j+1}^v, there is no $(j+1)$-critical vertex in T_{j+1}^w, which is by inductive hypothesis and Lemma 4.8 the split decomposition tree of $\widetilde{\mathcal{L}}_{D_{j+1}^v}[\mathsf{pbag}_{D_{j+1}^v}(w), y]$. Hence, $D_j^w = D_{j+1}^w$ by definition. By Proposition 5.17 $\widetilde{\mathcal{L}}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), y']$ is locally equivalent to $\widetilde{\mathcal{L}}_{D_{j+1}^v}[\mathsf{pbag}_{D_{j+1}^v}(w), y]$. Hence, we can conclude that D_j^w is locally equivalent to $\widetilde{\mathcal{L}}_{D_j^v}[\mathsf{pbag}_{D_j^v}(w), y']$ because $D_j^w = D_{j+1}^w$ and D_{j+1}^w is locally equivalent to $\widetilde{\mathcal{L}}_{D_{j+1}^v}[\mathsf{pbag}_{D_{j+1}^v}(w), y]$ by inductive hypothesis.

Case 2. *v_c is a descendant of w in T_{j+1}^v.*

If v_c is a child of w in T_{j+1}^v and the bag $\mathsf{bag}_{D_{j+1}^v}(w))$ has size 3, then T_j^v cannot contain w as a node, and this contradicts the assumption that w is a node of T_j^v. Therefore, we may assume that either

1. $|\mathsf{bag}_{D_{j+1}}(w)| \geqslant 4$, or
2. $|\mathsf{bag}_{D_{j+1}}(w))| = 3$ and v_c is not a child of w in T_{j+1}^v.

This implies that v_c is a node of the split decomposition tree of $D' := \widetilde{\mathcal{L}}_{D^v_{j+1}}[\mathsf{pbag}_{D^v_{j+1}}(w), y]$, and by Proposition 5.16, $\widetilde{\mathcal{L}}_{D^v_j}[\mathsf{pbag}_{D^v_j}(w), y']$ is locally equivalent to $\widetilde{\mathcal{L}}_{D'}[\mathsf{bag}_{D'}(v_c), z]$ where z is any unmarked vertex in $\mathcal{T}_{D'}[\mathsf{bag}_{D'}(v_c), \mathsf{pbag}_{D'}(v_c)]$ represented by a marked vertex in $\mathsf{bag}_{D'}(v_c)$. By inductive hypothesis, we know that D^w_{j+1} is locally equivalent to D', and by definition v_c is also the unique critical node of T^w_{j+1}, and moreover $D^w_j = \widetilde{\mathcal{L}}_{D^w_{j+1}}[\mathsf{bag}_{D^w_{j+1}}(v_c), z']$ for some unmarked vertex z' in $\mathcal{T}_{D^w_{j+1}}[\mathsf{bag}_{D^w_{j+1}}(v_c), \mathsf{pbag}_{D^w_{j+1}}(v_c)]$ represented by a marked vertex in $\mathsf{bag}_{D^w_{j+1}}(v_c)$. Hence, by Proposition 5.18 $\widetilde{\mathcal{L}}_{D'}[\mathsf{bag}_{D'}(v_c), z]$ is locally equivalent to D^w_j, that is, $\widetilde{\mathcal{L}}_{D'}[\mathsf{bag}_{D'}(v_c), z]$ is locally equivalent to D^w_j, and this concludes the proof. $\square$

Now we are ready to present and analyze our algorithm. We describe the algorithm explicitly in Algorithm 1. First, we modify the given decomposition as follows. For the canonical split decomposition D' of a distance-hereditary graph G, we modify D' into a canonical split decomposition D of a connected distance-hereditary graph by adding a bag R and making it adjacent to a bag R' of D' so that $f(D, R, D[V(D')]) = \operatorname{lrw}(G)$. So, if we root T_D at the node r such that $\mathsf{bag}_D(r) = R$, then $\operatorname{lrw}(G) = \operatorname{lrw}(\widehat{D^{r'}_{\mathsf{lrwbd}}})$ with $\mathsf{bag}_D(r') = R'$. We call (D, R) a *modified canonical split decomposition of* G. The basic strategy is to compute $\operatorname{lrw}(\widehat{D^v_i})$ for all non-root vertices v of T_D and all integers i such that $\alpha^v_i \leqslant i$, and we will use Proposition 9.6 and the canonical split decompositions D^v_j for $j \leqslant i$.

Proof of Theorem 9.1. Let G be a connected distance-hereditary graph. We recall that $\mathsf{lrwbd} = \lfloor \log_2 |V(G)| \rfloor$. We first show that Algorithm 1 correctly computes the linear rank-width of G. Let (D, R) be a modified canonical split decomposition of G and let r' be the unique neighbor node of the root of T_D. As we observed, we have that $\operatorname{lrw}(G) = \operatorname{lrw}(\widehat{D^{r'}_{\mathsf{lrwbd}}})$, and we want to prove that $\beta^{r'}_t = \operatorname{lrw}(\widehat{D^{r'}_{\mathsf{lrwbd}}})$. We claim that for each non-root node v of T_D and each $0 \leqslant i \leqslant \mathsf{lrwbd}$ such that $\alpha^v_i \leqslant i$, Algorithm 1 $\beta^v_i = \operatorname{lrw}(\widehat{D^v_i})$.

Suppose v is a non-root leaf node of T_D. Since every canonical limb is connected by Lemma 5.5, D^v_{lrwbd} is isomorphic to either a complete graph or a star and it has moreover at least two unmarked vertices. Thus, $\operatorname{lrw}(\widehat{D^v_{\mathsf{lrwbd}}}) = 1$, and by construction for each $0 \leqslant i \leqslant \mathsf{lrwbd}$, $D^v_i = D^v_{\mathsf{lrwbd}}$, and so Line 3 correctly puts these values.

We assume that v is a non-root node in T_D that is not a leaf, and for all its descendants v' and integers $0 \leqslant \ell \leqslant \mathsf{lrwbd}$ such that $\alpha^{v'}_\ell \leqslant \ell$, $\beta^{v'}_\ell$ is computed (*i.e.* $\beta^{v'}_\ell \neq 0$). We claim that Line 8- 12 recursively computes D^v_i for each i where $\alpha^v_i \leqslant i$. We first remark that for computing α^v_i of T^v_i, we use the fact that for each non-root node w of T^v_i, $\alpha^w_i \leqslant i$ and $\operatorname{lrw}(\widehat{D^w_i}) = f_1(D^v_i, w)$ from Proposition 9.8. So, $\alpha^v_i = \max\{\beta^w_i : w \text{ a non-root node } w \text{ of } T^v_i\}$.

Let $i \in \{0, 1, \ldots, \mathsf{lrwbd}\}$ such that $\alpha^v_i \leqslant i$. If $\alpha^v_i < i$, then by the definition, $T^v_{i-1} = T^v_i$ and thus, we take $D^v_{i-1} = D^v_i$. We may assume that $\alpha^v_i = i$. If either T^v_i has a node with at least 3 children v' such that $\beta^{v'}_i = i$, or T^v_i has two incomparable nodes v_1 and v_2 with v_1 an i-critical node and $\beta^{v_2}_i = i$, then from the definition of D^v_i, we have that $D^v_{i-1} = D^v_i$ and for all $0 \leqslant \ell \leqslant i-1$, $\alpha^v_\ell = i > \ell$. Since we do not need to evaluate β^v_ℓ when $\alpha^v_\ell > \ell$, we stop the loop. If T^v_i has no i-critical node, then $\beta^v_i = \alpha^v_i = i$, that is, the β^v_i value cannot be increased by one. In this case, we also stop the loop. These 3 cases are the conditions in Line 9.

Suppose neither of the conditions in Line 9 occur. Then by Proposition 9.6, T^v_i has a unique i-critical node v_c and D^v_{i-1} is equal to some canonical limb $\widetilde{\mathcal{L}}_{D^v_i}[\mathsf{bag}_{D^v_i}(v_c), y]$ where y is some unmarked vertex in $\mathcal{T}_{D^v_i}[\mathsf{bag}_{D^v_i}(v_c), \mathsf{pbag}_{D^v_i}(v_c)]$ represented by a marked vertex in $\mathsf{bag}_{D^v_i}(v_c)$. So, we compute D^v_{i-1} from D^v_i, the rooted split decomposition tree T^v_{i-1} of D^v_{i-1} and compute subsequently α^v_{i-1}. Notice that for all $\alpha^v_{i-1} \leqslant \ell \leqslant i-1$, $D^v_\ell = D^v_{i-1}$ and thus it is sufficient to deal with $D^v_{\alpha^v_{i-1}}$ in the next iteration. Thus, Line 8- 12 correctly computes canonical split decompositions D^v_i for each i where $\alpha^v_i = i$.

Algorithm 1: COMPUTE LRW OF CONNECTED DH GRAPHS

Input: A connected distance-hereditary graph G

Output: The linear rank-width of G

1 Compute a modified canonical split decomposition (D, R) of G.

2 Let $\beta_i^v \leftarrow 0$ for each non-root node v and each $0 \leqslant i \leqslant \mathsf{lrwbd} = \lfloor \log_2 |V(G)| \rfloor$

3 For each non-root leaf node v in T_D and each $0 \leqslant i \leqslant \mathsf{lrwbd}$, let $\beta_i^v \leftarrow 1$

4 **while** *T_D has a non-root node v such that β^v_{lrwbd} is not computed* **do**

5   Let v a non-root node in T_D such that $\beta^v_{\mathsf{lrwbd}} = 0$, but $\beta^{v'}_{\mathsf{lrwbd}} \neq 0$ for each child v' of v

6   Compute D^v_{lrwbd}, T^v_{lrwbd} and $\alpha^v_{\mathsf{lrwbd}}$

7     $k \leftarrow \alpha^v_{\mathsf{lrwbd}}$, $i \leftarrow k$, and let S be a stack

8   **while** *(true)* **do**

9   **if** *(T_i^v has a node with at least 3 children v' such that $\beta_i^{v'} = i$) or (T_i^v has two incomparable nodes v_1 and v_2 with v_1 an i-critical node and $\beta_i^{v_2} = i$) or (T_i^v has no i-critical nodes)* **then**

10   Stop this loop

11   Find the unique i-critical node v_c of T_i^v;

12   Compute D^v_{i-1}, T^v_{i-1} and α^v_{i-1}

13   push(S, i) and $i \leftarrow \alpha^v_{i-1}$

14   **if** *(T_i^v has a node with at least 3 children v' such that $\beta_i^{v'} = i$) or (T_i^v has two incomparable nodes v_1 and v_2 with v_1 an i-critical node and $\beta_i^{v_2} = i$)* **then** $\beta_i^v \leftarrow i+1$ **else** $\beta_i^v \leftarrow i$

15   **while** $(S \neq \varnothing)$ **do**

16         $j \leftarrow \text{pull}(S)$

17   **if** $\beta_i^v = j$ **then** $\beta_j^v \leftarrow j+1$ **else** $\beta_j^v \leftarrow j$

18   **for** $\ell \leftarrow i+1$ ***to*** $j-1$ **do**

19             $\beta_\ell^v \leftarrow \beta_i^v$

20         $i \leftarrow j$

21   **for** $j \leftarrow k+1$ ***to*** *lrwbd* **do**

22         $\beta_j^v \leftarrow \beta_k^v$

23 Let r' be the unique neighbor of the root and **return** $\beta_t^{r'}$

Now we verify the procedure of computing β_j^v in Line 14. Let $0 \leqslant \ell \leqslant t$ be the minimum integer such that $\alpha_\ell^v = \ell$. If $\ell = 0$, then $\beta_\ell^v = 1$. Suppose $\ell \geqslant 1$. Then since $\alpha^v_{\ell-1} > \ell - 1$, we have that

1. $\beta_\ell^v = \ell + 1$ if either T_ℓ^v has a node with at least 3 children v' such that $\beta_\ell^{v'} = \ell$ or T_ℓ^v has two incomparable nodes v_1 and v_2 with v_1 an i-critical node and $\beta_i^{v_2} = i$,
2. $\beta_\ell^v = \ell$ if otherwise.

So, Line 14 correctly computes it.

In the loop in Line 8, we use a stack to pile up the integers i such that T_i^v has the unique i-critical node. When T_i^v has the unique i-critical node, then by Proposition 9.6,

1. $\beta_i^v = i + 1$ if $\beta^v_{i-1} = i$, and
2. $\beta_i^v = i$ if $\beta^v_{i-1} \leqslant i - 1$.

So, from the lower value in the stack we can compute β_i^v recursively. From Line 15 to Line 21, Algorithm 1 computes all β_i^v correctly where $\alpha_i^v \leqslant i$, and in particular, it computes β_t^v. Therefore, at the end of the algorithm, it computes $\beta_t^{r'}$ that is equal to the linear rank-width of G.

Let us now analyze its running time. Let n and m be the number of vertices and edges of G. Its canonical split decomposition can be computed in time $\mathcal{O}(n+m)$ by Theorem 4.4, and one can of course a modified canonical split decomposition (D, R) in constant time.

For each node v and each $0 \leqslant j \leqslant \mathsf{lrwbd}$, β_j^v can be computed in time $\mathcal{O}(n \cdot \log_2 n)$. Line 5 can be done in time $\mathcal{O}(n)$. For computing α_i^v from T_i^v, for each non-root vertex w of T_i^v, we call the value β_i^w. Since α_i^v is the maximum β_i^w over all non-root nodes of T_i^v, Line 6 or 12 can be done in $\mathcal{O}(n)$ time.

The loop in Line 8 runs lrwbd times, and all the steps in Line 8 can be implemented in time $\mathcal{O}(n)$. Also, Lines 14-21 can be done in time $\mathcal{O}(n)$. Since the number of bags in D is bounded by $\mathcal{O}(n)$ (see [99, Lemma 2.2]), we conclude that this algorithm runs in time $\mathcal{O}(n^2 \cdot \log_2 n)$. ☐

Corollary 9.9. *For every connected n-vertex distance-hereditary graph G, we can compute in time $\mathcal{O}(n^2 \cdot \log_2 n)$ a layout of the vertices of G witnessing the linear rank-width of G.*

Proof. We establish a linear layout witnessing $\mathrm{lrw}(G) = k$. Let G be a connected distance-hereditary graph. Let D be a modified canonical split decomposition of G with the root bag R. We first run the algorithm computing $\mathrm{lrw}(G)$ and assume that for each non-root vertex v of T_D and each $0 \leqslant i \leqslant \mathsf{lrwbd}$ such that $\alpha_i^v \leqslant i$, β_i^v is computed.

Then using the values β_{lrwbd}^v, we can search for the path depicted in Lemma 5.15, and this can be done in linear time. Now for all the subtrees pending on that path, the linear rank-width of the corresponding limbs are at most $k-1$. We recursively apply the same algorithm on each of them. Then, similarly in the backward direction of Theorem 5.11, we can output a linear layout witnessing $\mathrm{lrw}(G) = k$.

Note that the total number of the recursive calls is bounded by the number of bags. Therefore, we make at most $\mathcal{O}(n)$ recursive calls and in each call, the path is found in $\mathcal{O}(n)$. So, if all of the β_i^v's are computed before, then we can compute an optimal layout in time $\mathcal{O}(n^2)$. ☐

9.3 Path-width of matroids with branch-width at most 2

Now we prove that the path-width of matroids of branch-width at most 2 can be computed in polynomial time, provided that the matroid is given by an independent set oracle. Note that by Corollary 1.14, every matroid of branch-width at most 2 is binary. We use the direct relation between binary matroids and bipartite graphs, mentioned in Section 2 [150].

Proof of Corollary 9.2. Let M be a matroid of branch-width at most 2 and assume that an independent oracle of M is given. We first run a greedy algorithm to find a base B of M [156, Section 1.8] in time $\mathcal{O}(|E(M)|)$. After choosing one base B, for each $e \in B$ and $e' \in E(M)\backslash B$, we test whether $(B\backslash\{e\}) \cup \{e'\}$ is again a base using the independent set oracle, and we create the fundamental graph G with respect to M in time $\mathcal{O}(|E(M)|^2)$. By Proposition 2.7, the rank-width of G is at most 1. Using Theorem 9.1, we can compute the linear rank-width of G in time $\mathcal{O}(|E(M)|^2 \cdot \log_2|E(M)|)$, which is the same as $\mathrm{pw}(M) - 1$. ☐

Chapter 10. Linear rank-width 1 vertex deletion

We discuss a graph modification problem related to graphs of linear rank-width 1, which are called *thread graphs*. We recall the problem.

Thread Vertex Deletion (Linear Rank-Width 1 Deletion)
Input : A graph G, an integer k
Parameter : k
Question : Is there a vertex subset $S \subseteq V(G)$ of size at most k such that $G \backslash S$ is a thread graph?

We prove the following.

Theorem 10.1. *For a fixed k and a given graph G with n vertices, the Thread Vertex Deletion problem can be solved in $\mathcal{O}(8^k \cdot n^8)$ time.*

Theorem 10.2. *There exists a polynomial-time algorithm that transforms a given instance (G, k) of the Thread Vertex Deletion problem into an instance (G', k') such that*

1. *(G, k) is a* Yes*-instance if and only if (G', k') is a* Yes*-instance,*
2. *$k' \leqslant k$ and $|V(G')| \leqslant \mathcal{O}(k^{33})$.*

We use the induced subgraph obstructions for thread graphs in Theorem 5.22. We recall that the obstructions consist of a house, a gem, a domino, and induced cycles of length at least 5 in Figure 5.1, which are the induced subgraph obstructions for distance-hereditary graphs [9], and 14 induced subgraph obstructions for thread graphs that are distance-hereditary, depicted in Figure 5.7. We define Ω_U as the set of graphs in Figure 5.7. Because the following sets are frequently used, we define that

- $\Omega_T :=$ {house, gem, domino, hole} $\cup\ \Omega_U$, and
- $\Omega_N :=$ {house, gem, domino, C_5, C_6, C_7, C_8} $\cup\ \Omega_U$.

One of the main ingredient is to investigate a new class of graphs, called *necklace graphs*, which are close to thread graphs. Briefly, necklace graphs are locally thread graphs, but they may have a long induced cycle. We show that every connected graph having no induced subgraph in Ω_N is either a necklace graph or a thread graph, and it is easy to find a minimum vertex set on Ω_N-free graphs whose removal makes a given graph a thread graph. We first use a simple branching algorithm to remove the obstructions in Ω_N with the time complexity $\mathcal{O}^*(8^k)$ because every graph in Ω_T has at most 8 vertices. (The $\mathcal{O}^*$ notation indicates that polynomial factors of an input size are suppressed.) If the instance does not have an obstruction in Ω_N, then it is an Ω_N-free graph, and we compare the remaining budget with the minimum deleting set in the Ω_N-free graph to decide whether it is a Yes-instance.

To obtain a polynomial size kernel, we adapt an idea used to obtain a polynomial size kernel for Proper Interval Vertex Deletion due to Fomin, Saurabh, and Villanger [90]. When a finite list of graphs is fixed, they use Sunflower lemma to find a small vertex set T in G satisfying that a set is a minimal hitting set for the list in G if and only if it is a minimal hitting set for the list in the subgraph of G induced on T. From the property of T, automatically, the remaining part obtained by removing T

has no induced subgraph in the list. For our purpose, this will be an Ω_N-free graph, and after adding at most one vertex from each component of $G\backslash T$, we obtain a vertex subset T' of size polynomial in k such that $G\backslash T'$ is a thread graph. We analyze how to shrink the remaining part.

For a graph G, a set $S \subseteq V(G)$ is called a *thread vertex deletion set* if $G\backslash S$ is a thread graph.

We recall thread blocks and thread graphs defined in Section 1.1. A *thread graph* is a graph that is either an one vertex graph or $G = P \odot \mathcal{B}_P$ for some directed path P and some set of thread blocks $\mathcal{B}_P$ mergeable with P. In the next subsection, we study graphs which are defined using directed cycles instead of directed paths.

10.1 Necklace graphs

We generalize the construction of thread graphs from directed paths to directed cycles. A connected graph G is a *necklace graph* if there exist a directed cycle C, called the *underlying directed cycle*, and some set of thread blocks $\mathcal{B}_C$ mergeable with C such that $G = C \odot \mathcal{B}_C$.

Our FPT algorithm and the construction of a polynomial size kernel rely deeply on the following characterization of Ω_N-free graphs.

Theorem 10.3. *A connected Ω_N-free graph is either a thread graph or a necklace graph whose underlying directed cycle has length at least* 9.

We use the following lemma several times to find one of the forbidden graphs in each induction step.

Lemma 10.4. *Let $k \geqslant 4$ be an integer. Let G be a graph and let $v \in V(G)$ such that $G\backslash v$ is a path $p_1p_2\cdots p_k$, and v is adjacent to both p_1 and p_k in G. Then G contains an induced subgraph isomorphic to a house, a gem, a domino, or an induced cycle of length at least* 5.

Proof. The neighbors of v divide $G\backslash v$ into edge-disjoint paths $I = \{P_1, \ldots, P_m\}$ where the end vertices of each P_i are two neighbors of v and all internal vertices of P_i have degree 2 in G. If one of the path in I has length at least 3, then together with v, G contains an induced cycle of length at least 5. We may assume that each path in I has length at most 2.

If there exist two consecutive paths P_i, P_{i+1} such that one has length 1 and the other has length 2, then G contains a house. So, we may assume that all paths $P_1, P_2, \ldots, P_m$ have the same length. If all paths in I have length 1, then G contains a gem because $k \geqslant 4$. If all paths in I have length 2, then G contains a domino. Therefore, we conclude that G has an induced subgraph isomorphic to either a house, a gem, a domino or an induced cycle of length at least 5. □

Let G be a connected Ω_N-free graph and suppose that G is not a thread graph. Since G is Ω_N-free and it is not a thread graph, by Theorem 5.22, G has an induced subgraph isomorphic to C_k for some $k \geqslant 9$. We prove by induction on $|V(G)|$ that if C is a shortest cycle among induced cycles of length at least 9 in G, then G is a necklace graph whose underlying directed cycle is C. Let $C := (v_1, v_2, \ldots, v_k, v_1)$ be a shortest cycle among induced cycles of length at least 9 in G and we regard it as a directed cycle where for each $1 \leqslant j \leqslant k$, v_jv_{j+1} is an arc.

If $G = C$, then we are done because C itself is a necklace graph with the underlying directed cycle C. We may assume that $|V(G)| > |V(C)|$. We may choose a vertex $v \in V(G)\backslash V(C)$ such that $G\backslash v$ is connected. Clearly, $G\backslash v$ is again Ω_N-free graph, and C is a shortest cycle among induced cycles of length at least 9 in $G\backslash v$. By the induction hypothesis, there exists some set of thread blocks $\mathcal{B}_C$ mergeable with

C such that $G \backslash v = C \odot \mathcal{B}_C$. The remaining part of this section devotes to prove that $G = C \odot \mathcal{B}'_C$ for some set of thread blocks $\mathcal{B}'_C$ mergeable with C.

For convenience, let $v_{k+1} := v_1$ and $v_{k+2} := v_2$. Let $\mathcal{B}_C := \{B(x, y) : xy \text{ is an arc of } C\}$ such that for each $1 \leqslant j \leqslant k$, $B(v_j, v_{j+1})$ is a thread block (B_j, σ_j, ℓ_j) with a linear layout σ_j and a labelling ℓ_j. We define that

$$\mathcal{S} := \{house, gem, domino, C_5, C_6, \ldots, C_{k-1}\} \cup \Omega_U,$$

$$\mathcal{M} := \{B(v_j, v_{j+1}) : v \text{ has a neighbor in } V(B_j)\backslash\{v_j, v_{j+1}\}\}.$$

The proof consists of three steps.

Lemma 10.5. *If there are two non-adjacent neighbors of v on C, then G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$.*

Lemma 10.6. *If $\mathcal{M}$ contains at least two thread blocks of $\mathcal{B}_C$, then either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$ or G is a necklace graph whose underlying directed cycle is C.*

Lemma 10.7. *If the neighbors of v are contained in one thread block of $\mathcal{B}_C$, then either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$ or G is a necklace graph whose underlying directed cycle is C.*

Using Lemmas 10.5 and 10.6 and excluding some small cases, we can show that the neighbors of v should be contained in one thread block. Then, with Lemma 10.7, we conclude our claim.

For an induced path P of a graph G and $v \in V(G)\backslash V(P)$, we say (P, v) is a *bad pair* in G if P has length ℓ where $3 \leqslant \ell \leqslant k-3$, and v is adjacent to the end vertices of P. Lemma 10.4 tells us that if G has a bad pair (P, v), then G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$.

We first prove Lemma 10.5.

Proof of Lemma 10.5. Suppose that v has two neighbors on C that are not consecutive. Let $I = \{i : vv_i \in E\}$. If there exist two vertices v_i, v_j in I such that one of the two paths from v_i to v_j in C has length ℓ where $3 \leqslant \ell \leqslant k-3$, then the path together with v is a bad pair in G.

By the assumption, there exists two distinct vertices v_i, v_j in I where the distance from v_i to v_j on C is 2. Therefore, G contains an induced subgraph isomorphic to either α_1 or α_4 which are in $\mathcal{S}$. $\square$

To prove Lemma 10.6, we need to analyze several cases.

Lemma 10.8. *If $B(v_j, v_{j+1}) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$ and v is adjacent to a vertex in $V(C)\backslash\{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$, then G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$.*

Proof. Let $z \in V(B_j)\backslash\{v_j, v_{j+1}\}$ be a neighbor of v and let $w \in V(C)\backslash\{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$ be a neighbor of v. Since the maximum distance between two vertices in C is $\lfloor k/2 \rfloor$, there exists an induced path P from w to z in $G\backslash v$ having length ℓ where $3 \leqslant \ell \leqslant \lfloor k/2 \rfloor + 1 \leqslant k-3$. Thus, (P, v) is a bad pair, and by Lemma 10.4, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. $\square$

Lemma 10.9. *If $B(v_j, v_{j+1}) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$ and $vv_{j-1} \in E$, $vv_{j+1} \notin E$, then G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$.*

Proof. Let $z \in V(B_j)\backslash\{v_j, v_{j+1}\}$ be a neighbor of v. See Figure 10.1 for the descriptions of cases. If z has a label $\{R\}$ or $\{L, R\}$, then $(v_{j-1}vzv_{j+1}, v_j)$ is a bad pair in G. Let us assume that z has a label $\{L\}$. Since v_jv_{j+1} is an arc of the directed cycle C and by definition of thread blocks which is

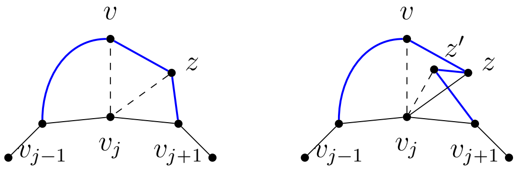


Figure 10.1: Two cases in Lemma 10.9.

$\ell_j(\sigma_j^{-1}(2)) \neq \{L\}$, z is not a pendant vertex adjacent to v_j in $G\backslash v$, and therefore, there exists at least one element $z' \in V(B_j)$ preceding z in the linear layout σ_j of $B(v_j, v_{j+1})$ such that it has label $\{R\}$ or $\{L, R\}$. From the previous case, we may assume that v is not adjacent to z'. Then $(v_{j-1}vzz'v_{j+1}, v_j)$ is a bad pair in G. Again, by Lemma 10.4, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. □

Lemma 10.10. *If $B(v_{j-1}, v_j), B(v_j, v_{j+1}) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$ and $vv_{j-1} \notin E$ and $vv_{j+1} \notin E$, then G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$.*

Proof. Let $z_{j-1} \in V(B_{j-1})\backslash\{v_{j-1}, v_j\}$ and $z_j \in V(B_j)\backslash\{v_j, v_{j+1}\}$ be neighbors of v. If z_j has a label $\{R\}$ or $\{L, R\}$, then either $(z_{j-1}vz_jv_{j+1}, v_j)$ or $(v_{j-1}z_{j-1}vz_jv_{j+1}, v_j)$ is a bad pair in G depending on the adjacency between v_{j-1} and z_{j-1}. We may assume that z_j has a label $\{L\}$. Since z_j is not a pendant vertex adjacent to v_j in $G\backslash v$, there exists at least one element $z_j' \in V(B_j)\backslash\{v_j, v_{j+1}\}$ preceding z_j in the linear layout of $B(v_j, v_{j+1})$ that has a label $\{R\}$ or $\{L, R\}$. In this case, we have either $(z_{j-1}vz_jz_j'v_{j+1}, v_j)$ or $(v_{j-1}z_{j-1}vz_jz_j'v_{j+1}, v_j)$ is a bad pair in G depending on the adjacency between v_{j-1} and z_{j-1}. Thus, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$, as required. □

Lemma 10.11. *If $B(v_{j-1}, v_j) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$, $vv_{j+1} \in E$ and $N_G(v) \subseteq V(B_{j-1}) \cup V(B_j)\backslash\{v_{j-1}\}$, then either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$ or G is a necklace graph whose underlying directed cycle is C.*

Proof. Let z_{j-1} be a neighbor of v in $V(B_{j-1})\backslash\{v_{j-1}, v_j\}$. If z_{j-1} has a label $\{L, R\}$ or $\{L\}$ in $B(v_{j-1}, v_j)$, then $(v_{j-1}z_{j-1}vv_{j+1}, v_j)$ is a bad pair in G, and we are done by Lemma 10.4. Thus, we may assume that all neighbors of v in $B(v_{j-1}, v_j)$ have a label $\{R\}$, and are all pairwise non-adjacent. If there exists a vertex z_{j-1}' with $z_{j-1} <_{\sigma_{j-1}} z_{j-1}'$ and a label $\{L\}$ or $\{L, R\}$, then $(v_{j-1}z_{j-1}'z_{j-1}vv_{j+1}, v_j)$ is a bad pair in G. Since we can reorder between the vertices having the same neighbors in G, we may also assume that all the neighbors of v in $B(v_{j-1}, v_j)$ are the last vertices in the order σ_{j-1} before v_j.

We forbid the following 4 configurations. See Figure 10.2 for the description of these configurations (in this order).

1. v has a neighbor w in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with $\ell_j(w) = \{R\}$.
   - $(z_{j-1}v_jv_{j+1}w, v)$ is a bad pair in G.

2. There exists a vertex w of $V(B_j)\backslash\{v_j, v_{j+1}\}$ with $\ell_j(w) = \{L, R\}$ such that $vw \notin E$.
   - $(z_{j-1}vv_{j+1}w, v_j)$ is a bad pair in G.

3. There exists a pair of vertices w_1 and w_2 in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with $vw_1, vw_2 \notin E$, $w_1 <_{\sigma_j} w_2$, $\ell_j(w_1) = \{R\}$, and $\ell_j(w_2) = \{L\}$.
   - $(z_{j-1}vv_{j+1}w_1w_2, v_j)$ is a bad pair in G.

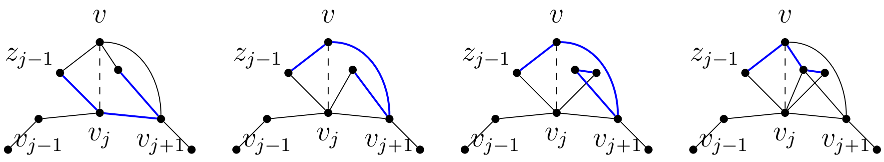


Figure 10.2: Forbidden configurations in Lemma 10.11.

4. There exists a pair of vertices w_1 and w_2 in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with $vw_1 \in E$, $vw_2 \notin E$, $w_1 <_{\sigma_j} w_2$, $\ell_j(w_1) = \{L, R\}$, and $\ell_j(w_2) = \{L\}$.
   - $(z_{j-1}vw_1w_2, v_j)$ is a bad pair in G.

In all of the four cases, by Lemma 10.4, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. Now we may assume that G does not have any of these 4 configurations. In particular, by the forbidden configurations (1) and (2), we may assume that all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{L, R\}$ are adjacent to v, and all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{R\}$ are not adjacent to v.

Let x be the first vertex of $V(B_j)\backslash\{v_j, v_{j+1}\}$ in the sequence ℓ_j. Note that x has no label $\{L\}$ because v_jv_{j+1} is an arc of the directed cycle C. If x has a label $\{R\}$, then $vw \notin E$, and all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{L\}$ must be adjacent to v because of the forbidden configuration (3). Similarly, if x has a label $\{L, R\}$, then $vw \in E$, and all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{L\}$ must be adjacent to v because of the forbidden configuration (4). It implies that all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{L\}$ or $\{L, R\}$ are adjacent to v.

Now we claim that

$$B'_{j-1} = G[V(B_{j-1})\backslash N_G(v) \cup \{v_j\}]$$

and

$$B'_j = G[V(B_j) \cup \{v\} \cup N_G(v)]$$

are new thread blocks with the same end vertices. Clearly, B'_{j-1} is a thread block with the end vertices v_{j-1} and v_j because we just remove some vertices from $V(B_{j-1})\backslash\{v_{j-1}, v_j\}$.

For B'_j, we define a linear layout σ'_j and a labeling ℓ'_j of B'_j as follows. We take any linear layout σ_a of the vertices of $N_G(v) \cap (V(B_{j-1})\backslash\{v_j\})$. Let σ_b be the linear layout obtained from σ_j by removing v_j and v_{j+1}. Let

$$\sigma'_j := (v_j, v) \oplus \sigma_a \oplus \sigma_b \oplus (v_{j+1}).$$

We define that

1. $\ell'_j(v_j) := \{R\}$ and $\ell'_j(v_{j+1}) := \{L\}$,
2. $\ell'_j(v) := \{L, R\}$ or $\{R\}$ depending on $vv_j \in E$ or not,
3. for all neighbors w of v in $B(v_{j-1}, v_j)$, $\ell'_j(w) = \{L\}$, and
4. for all $w \in V(B_j)\backslash\{v_j, v_{j+1}\}$, $\ell'_j(w) := \ell_j(w)$.

Since all vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ with a label $\{L\}$ or $\{L, R\}$ are adjacent to v, v has no conflict with vertices in the linear layout σ_j. Since all neighbors w of v in $B(v_{j-1}, v_j)$ are not adjacent to the vertices in $V(B_j)\backslash\{v_j, v_{j+1}\}$ and are pairwise non-adjacent, we conclude that B'_j is indeed a thread block with the end vertices v_j and v_{j+1}.

Since there are no edges between $V(B'_{j-1})\backslash\{v_{j-1}, v_j\}$ and $V(B'_j)\backslash\{v_j, v_{j+1}\}$, we conclude that G is a necklace graph whose underlying directed cycle is C. $\square$

Proof of Lemma 10.6. Suppose that $\mathcal{M}$ has at least two blocks.

If $B(v_j, v_{j+1}) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$ and v is adjacent to a vertex in $V(C)\backslash\{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$, then by Lemma 10.8, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. We may assume that if $B(v_j, v_{j+1}) \in \mathcal{M}$ for some $1 \leqslant j \leqslant k$, then v has no neighbors in $V(C)\backslash\{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$.

Case 1. There exist two blocks $B(v_p, v_{p+1}), B(v_q, v_{q+1}) \in \mathcal{M}$ such that $B(v_p, v_{p+1})$, $B(v_q, v_{q+1})$ are not consecutive thread blocks.

Let $z_p \in V(B_p)$ and $z_q \in V(B_q)$ such that z_p and z_q are neighbors of v in G. Since $B(v_p, v_{p+1})$ and $B(v_q, v_{q+1})$ are not consecutive, the distance between $\{v_p, v_{p+1}\}$ and $\{v_q, v_{q+1}\}$ on C is at least 1 and the longest possible distance between two vertices in C is at most $\lfloor k/2 \rfloor$. Thus, $G\backslash v$ has an induced path from z_p to z_q of length ℓ where $3 \leqslant \ell \leqslant \lfloor k/2 \rfloor + 2$. Since $k \geqslant 9$, by Lemma 10.4, G contains an induced subgraph isomorphic to a house, a gem, a domino, or an induced cycle of length ℓ where $5 \leqslant \ell \leqslant \lfloor k/2 \rfloor + 4 \leqslant k - 1$.

Case 2. $\mathcal{M}$ contains exactly two blocks that are consecutive in $G\backslash v$.

Suppose that $\mathcal{M} = \{B(v_p, v_{p+1}), B(v_{p+1}, v_{p+2})\}$. From the assumption, we may assume that v has no neighbors on $V(C)\backslash\{v_{p-1}, v_p, v_{p+1}, v_{p+2}, v_{p+3}\}$ in G. Also, if v is adjacent to both v_p and v_{p+2}, then G contains an induced subgraph isomorphic to either α_1 or α_4. So, we may assume that v is not adjacent to both of v_p and v_{p+2}. Let z_{p+1} be the first vertex of $V(B_{p+1})\backslash\{v_{p+1}, v_{p+2}\}$ that is adjacent to v. Let z_p be a neighbor of v in $V(B_p)\backslash\{v_p, v_{p+1}\}$.

If v is adjacent to v_p and not adjacent to v_{p+2} in G, then since v has a neighbor on $B(v_{p+1}, v_{p+2})$, by Lemma 10.9, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. If v is adjacent to neither v_p nor v_{p+2} in G, then since v has neighbors on $B(v_p, v_{p+1})$ and $B(v_{p+1}, v_{p+2})$, by Lemma 10.10, G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. At last, if v is not adjacent to v_p and adjacent to v_{p+2} in G, then

$$N_G(v) \subseteq V(B_p) \cup V(B_{p+1})\backslash\{v_p\},$$

and by Lemma 10.11, either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$, or G is a necklace graph whose underlying directed cycle is C. $\square$

Now we prove Lemma 10.7. We will use structural properties of distance-hereditary graphs in this proof. For detailed incremental characterization of distance-hereditary graphs, we refer to [104]. We denote by $G + v$ the graph obtained from G by adding a new vertex v and new edges incident with v.

Proof of Lemma 10.7. Suppose that $N_G(v) \subseteq V(B_j)$ for some $1 \leqslant j \leqslant k$, and G has no induced subgraph isomorphic to a graph in $\mathcal{S}$. If $N_G(v) = \{v_j\}$, then we can extend the thread block $B(v_{j-1}, v_j)$ into a thread block containing v_j by putting it as the last second vertex. Suppose that $N_G(v) \neq \{v_j\}$.

We claim that $G[V(B_j) \cup \{v\}]$ is a thread block with the same end vertices v_j and v_{j+1}. If it is true, then it directly implies that G is a necklace graph because v has no neighbors on the other thread blocks except the vertices of C.

We regard $G[V(B_{j-1}) \cup V(B_j) \cup V(B_{j+1}) \cup \{v\}]$ as a graph obtained from the union of three consecutive thread blocks by adding a vertex v. If it is not a thread graph, then it should have an induced subgraph isomorphic to an induced cycle H of length at least 9 that contains v. However, since v_j and v_{j+1} are cut vertices in $G[V(B_{j-1}) \cup V(B_j) \cup V(B_{j+1}) \cup \{v\}]$ and v has no neighbors on $(V(B_{j-1}) \cup V(B_{j+1}))\backslash V(B_j)$, H should be contained in B_j. But then $H\backslash v$ is an induced path of length at least 7, while the longest induced path in a thread block has length 3, it is a contradiction. Therefore, $G[V(B_{j-1}) \cup V(B_j) \cup V(B_{j+1}) \cup \{v\}]$ is a thread graph.

Let D_j be the canonical split decomposition of $G[V(B_{j-1}) \cup V(B_j) \cup V(B_{j+1})]$. We use one vertex incremental characterization of canonical split decompositions of distance-hereditary graphs, developed by Gioan and Paul [104]. For a distance-hereditary graph G and $v \notin V$, they characterize the conditions for $G + v$ being distance-hereditary. Since $G[V(B_{j-1}) \cup V(B_j) \cup V(B_{j+1}) \cup \{v\}]$ is a thread graph (especially, distance-hereditary graphs), its canonical split decomposition can be modified from D_j. The new vertex v is placed in either

- (Case 1) a bag of D_j,
- (Case 2) a new bag put between two bags of D_j, or
- (Case 3) a new bag put between two bags by splitting one bag of D_j.

If v is placed in a bag of D_j, then it implies that v and some vertex of $B(v_j, v_{j+1})$ have the same neighbors in G, and since $N_G(v) = \{v_j\}$, v is not placed in the bag containing v_j. Thus, we can naturally extend the linear layout and the labelling of $B(v_j, v_{j+1})$ into $G[V(B_j) \cup \{v\}]$. In Case 2 or 3, the new bag containing v cannot be a star bag whose center is an unmarked vertex because $G \backslash v$ is connected. Therefore, depending on the type of the bag containing v, we can also naturally extend the linear layout and the labelling of $B(v_j, v_{j+1})$ into $G[V(B_j) \cup \{v\}]$. For instance, if the new bag is a star bag and the center is adjacent to the previous bag, then we give a label $\{L\}$ on v. This extends the thread block $B(v_{j-1}, v_j)$ with the vertex v, and G is again a necklace graph whose underlying cycle is C. □

Now we prove the main result of this section.

Proof of Theorem 10.3. Let G be a connected Ω_N-free graph and suppose that G is not a thread graph. Since G is Ω_N-free and it is not a thread graph, by Theorem 5.22, G has an induced subgraph isomorphic to C_k for some $k \geqslant 9$. We prove by induction on $|V(G)|$ that if C is a shortest cycle among induced cycles of length at least 9 in G, then G is a necklace graph whose underlying directed cycle is C. Let $C := (v_1, v_2, \ldots, v_k, v_1)$ be a shortest cycle among induced cycles of length at least 9 in G and we regard it as a directed cycle where for each $1 \leqslant j \leqslant k$, $v_j v_{j+1}$ is an arc.

If $G = C$, then we are done because C itself is a necklace graph with the underlying directed cycle C. We may assume that $|V(G)| > |V(C)|$. We may choose a vertex $v \in V(G) \backslash V(C)$ such that $G \backslash v$ is connected. Clearly, $G \backslash v$ is again Ω_N-free graph, and C is a shortest cycle among induced cycles of length at least 9 in $G \backslash v$. By the induction hypothesis, there exists some set of thread blocks $\mathcal{B}_C$ mergeable with C such that $G \backslash v = C \odot \mathcal{B}_C$. We prove that G is a necklace graph whose underlying directed cycle is C.

For convenience, let $v_{k+1} := v_1$ and $v_{k+2} := v_2$. Let $\mathcal{B}_C := \{B(x, y) : xy \text{ is an arc of } C\}$ such that for each $1 \leqslant j \leqslant k$, $B(v_j, v_{j+1})$ is a thread block (B_j, σ_j, ℓ_j) with a linear layout σ_j and a labelling ℓ_j. We recall that

$$\mathcal{S} = \{house, gem, domino, C_5, C_6, \ldots, C_{k-1}\} \cup \Omega_U,$$

$$\mathcal{M} = \{B(v_j, v_{j+1}) : v \text{ has a neighbor in } V(B_j) \backslash \{v_j, v_{j+1}\}\}.$$

Since G is connected, v has at least one neighbor. Since G is $\mathcal{S}$-free, by Lemma 10.5, v is adjacent to at most two vertices of C and if v has two neighbors on C, then they must be consecutive. Also, by Lemma 10.6, the number of indices j such that v has a neighbor on $V(B_j) \backslash \{v_j, v_{j+1}\}$ is at most 1.

Suppose that v has a neighbor on $V(B_j) \backslash \{v_j, v_{j+1}\}$ for some $1 \leqslant j \leqslant k$. If v has a neighbor on $V(C) \backslash \{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$, then by Lemma 10.8, G_i contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. So, we may assume that v has no neighbors on $V(C) \backslash \{v_{j-1}, v_j, v_{j+1}, v_{j+2}\}$. If $vv_{j-1} \in E(G_i)$, then

$vv_{j+1} \notin E(G_i)$, and by Lemma 10.9, G_i contains an induced subgraph isomorphic to a graph in $\mathcal{S}$. If $vv_{j+2} \in E(G_i)$, then $vv_{j-1}, vv_j \notin E(G_i)$, and therefore,

$$N_{G_i}(v) \subseteq V(B_j) \cup V(B_{j+1}) \backslash \{v_j\}.$$

By Lemma 10.11, either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$, or G is a necklace graph whose underlying directed cycle is C.

Therefore, we may assume that the neighbors of v on C are contained in $\{v_j, v_{j+1}\}$. So, $N_{G_i}(v) \subseteq V(B_j)$ and by Lemma 10.7, either G contains an induced subgraph isomorphic to a graph in $\mathcal{S}$, or G is a necklace graph whose underlying directed cycle is C.

We conclude that G is a necklace graph whose underlying directed cycle is C. □

10.2 A fixed parameter tractable algorithm for Thread Vertex Deletion

We prove that Thread Vertex Deletion is fixed parameter tractable.

Theorem 10.12. *For a given graph G with n vertices,* Thread Vertex Deletion *can be solved in time $\mathcal{O}(8^k \cdot n^8)$.*

Our algorithm is a branching algorithm that reduces a given instance to an Ω_N-free graph. For this, it suffices to hit all induced subgraphs isomorphic to a graph in Ω_N by Theorem 10.3. As each graph of Ω_N has size at most 8, the announced complexity follows. It remains to prove that given an Ω_N-free graph, a minimum thread vertex deletion set can be found in polynomial time. In fact, we prove that such a set has size at most one per component and identifying such a vertex requires polynomial time.

Lemma 10.13. *Let G be a necklace graph. Then there exists a vertex v such that $G \backslash v$ is a disjoint union of connected thread graphs.*

Proof. Let C be a directed cycle $(v_1, v_2, \ldots, v_k, v_1)$ where for each $1 \leqslant j \leqslant k$, $v_j v_{j+1}$ is an arc. Suppose that $G = C \odot \mathcal{B}_C$ for some set of thread blocks $\mathcal{B}_C$ where $\mathcal{B}_C = \{B(v_j, v_{j+1}) = (B_j, \sigma_j, \ell_j) : 1 \leqslant j \leqslant k\}$. Let $v_{k+1} := v_1$.

We show that for each $1 \leqslant i \leqslant k$, $G \backslash v_i$ is a disjoint union of a thread graph and one vertex graphs. Without loss of generality, we assume that $i = 1$. Let S be the set of all pendant vertices adjacent to v_1 in G. We claim that $G[V(G) \backslash (S \cup \{v_1\})]$ is a connected thread graph. Since S is a disjoint union of one vertex graphs in $G \backslash v_1$, it is enough to show the claim. Since $v_k v_1$ is an arc of C, the vertices of S are contained in $B(v_k, v_1)$.

Suppose that $V(B_k) \backslash (S \cup \{v_1\}) \neq \{v_k\}$. Since $V(B_k) \backslash S$ has no pendant vertices adjacent to v_1 in G, the last vertex z in the linear layout σ_k except $S \cup \{v_1\}$ must have a label either $\{L, R\}$ or $\{L\}$. It is easy to check that $B(v_k, z) = (V(B_k) \backslash (S \cup \{v_1\}), \sigma'_k, \ell'_k)$ is again a thread block where σ'_k is the restriction of σ_k on $V(B_k) \backslash (S \cup \{v_1\})$, and

$$\ell'_k(x) = \begin{cases} \ell_k(x) & \text{if } x \neq z \\ \{L\} & \text{if } x = z \end{cases}$$

If $V(B_k) \backslash (S \cup \{v_1\}) = \{v_k\}$, then we can regard $B(v_{k-1}, v_k)$ as the last thread block of $G[V(G) \backslash (S \cup \{v_1\})]$.

Similarly, if $V(B_1) \backslash (S \cup \{v_1\}) = \{v_2\}$, then we can regard $B(v_2, v_3)$ as the first thread block of $G[V(G) \backslash (S \cup \{v_1\})]$. If otherwise, we regard $B(v_1, v_2)[V(B_1) \backslash (S \cup \{v_1\})]$ as the first thread block. Let y

be the first vertex in the linear layout σ_1 except $V(B_1)\backslash(S\cup\{v_1\})$. We conclude that $G[V(G)\backslash(S\cup\{v_1\})]$ is a thread graph on a directed path P where

$$P = \begin{cases} v_2v_3\cdots v_k & \text{if } V(B_k)\backslash S\backslash\{v_1\} = \{v_k\} \text{ and } V(B_1)\backslash S\backslash\{v_1\} = \{v_2\}, \\ yv_2v_3\cdots v_k & \text{if } V(B_k)\backslash S\backslash\{v_1\} = \{v_k\} \text{ and } V(B_1)\backslash S\backslash\{v_1\} \neq \{v_2\}, \\ v_2v_3\cdots v_kz & \text{if } V(B_k)\backslash S\backslash\{v_1\} \neq \{v_k\} \text{ and } V(B_1)\backslash S\backslash\{v_1\} = \{v_2\}, \\ yv_2v_3\cdots v_kz & \text{otherwise.} \end{cases} \qquad \square$$

From Lemma 10.13, we can find a minimum thread vertex deletion set on necklace graphs in polynomial time. To find such a set in polynomial-time, we use the following lemma.

Lemma 10.14. *Let G be a necklace graph whose underlying directed cycle has length at least 9 and let $v \in V(G)$. For a positive integer i, let $N_i(v)$ be the set of vertices z whose distance from v is exactly i in G. Then $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a thread graph containing two consecutive thread blocks of G. Moreover, every cut vertex of $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a vertex of the underlying directed cycle.*

Proof. Suppose that there exist a directed cycle $C := (v_1, v_2, \ldots, v_k, v_1)$ where for each $1 \leqslant j \leqslant k$, v_jv_{j+1} is an arc, and some set of thread blocks $\mathcal{B}_C$ where $\mathcal{B}_C = \{B(v_j, v_{j+1}) = (B_j, \sigma_j, \ell_j) : 1 \leqslant j \leqslant k\}$, such that $G = C \odot \mathcal{B}_C$. Without loss of generality, we may assume that $v \in V(B_1)$.

First assume that $v = v_1$. Then $N_1(v)$ contains all vertices with a label $\{L\}$ or $\{L, R\}$ in B_1. Also, $N_2(v)$ contains all vertices with a label $\{R\}$ in B_1, and therefore

$$B_1 \subseteq \{v_1\} \cup N_1(v_1) \cup N_2(v_1).$$

Similarly,

$$B_k \subseteq \{v_1\} \cup N_1(v_1) \cup N_2(v_1).$$

So, $G[\{v_1\} \cup N_1(v_1) \cup N_2(v_1)]$ contains two consecutive thread blocks of G. By the same reason, it is not hard to observe that $G[\{v_2\} \cup N_1(v_2) \cup N_2(v_2)]$ contains two consecutive thread blocks B_1 and B_2.

Now suppose that $v \in V(B_1)\backslash\{v_1, v_2\}$. Since $N_1(v)$ contains one of v_1 and v_2, by the previous case, we can observe that $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ contains two consecutive thread blocks in G. Since C has length at least 9, $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ cannot contain all vertices of C, and it implies that $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a thread graph.

Now we claim that every cut vertex of $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a vertex of the underlying directed cycle. Since $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a thread graph with at least two thread blocks, there exists a cut vertex w of it. Suppose that $w \in V(B_j)\backslash\{v_j, v_{j+1}\}$ for some $1 \leqslant j \leqslant k$. We first show that $v_j, v_{j+1} \in V(G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)])$. If $v \in V(B_j)$, then this is clear because $G[\{v\} \cup N_1(v) \cup N_2(v)]$ contains B_j as an induced subgraph. We may assume that $v \notin V(B_j)$. Since $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is connected, without loss of generality, $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ contains v_j. However, since $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ contains w, $G[\{v\} \cup N_1(v) \cup N_2(v)]$ contains v_j, and it implies that $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ contains v_{j+1} as well. We conclude that both v_j and v_{j+1} are contained in $V(G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)])$.

The vertices v_j and v_{j+1} are on the same component of $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]\backslash w$ as they are adjacent. Then all vertices of $V(B_j)\backslash w$ are contained in the same component of $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]\backslash w$, contradicting to the assumption that w is a cut vertex of $G[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$. Therefore, $w \in V(C)$. $\square$

Proposition 10.15. *Let G be an Ω_N-free graph with n vertices and m edges. We can compute the minimum thread vertex deletion set of G in time $\mathcal{O}(n + m)$.*

Proof. We remark that each component of G is either a thread graph or a necklace graph whose underlying directed cycle has length at least 9. For each component H of G, we test whether H is a thread graph or not in time $\mathcal{O}(|V(H)| + |E(H)|)$ using Theorem 5.23. If H is a thread graph, then we do not need to remove any vertex from it. If H is a necklace graph, then we need to remove at least one vertex from it to make G a thread graph. Also, by Lemma 10.13, it is sufficient to remove one vertex on the underlying cycle to make H a disjoint union of thread graphs. Thus, the number of non-thread components are exactly the minimum size of thread vertex deletion set of G.

To identify a deletion set, let H be a necklace graph whose underlying cycle has length at least 9. Choose any vertex v in H, and for a positive integer i, let $N_i(v)$ the set of vertices z whose distance from v is exactly i in H. Then by Lemma 10.14, $H[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ is a thread graph containing two consecutive thread blocks of G, and its cut vertex is a vertex of the underlying directed cycle. We compute $H[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ in time $\mathcal{O}(|V(H)| + |E(H)|)$ using breadth-first search. Then we can find a cut vertex of $H[\{v\} \cup N_1(v) \cup N_2(v) \cup N_3(v)]$ in time $\mathcal{O}(|V(H)| + |E(H)|)$; for instance we use the algorithm by Hopcroft and Tarjan [112]. Since we proceed in time $\mathcal{O}(|V(H)| + |E(H)|)$ per each component H, we can find a minimum thread deletion set in time $\mathcal{O}(|V(G)| + |E(G)|)$. □

Proof of Theorem 10.12. Let (G, k) be an instance of the Thread Vertex Deletion problem. The first phase of the algorithm is to find an induced subgraph of G that is isomorphic to a graph in Ω_N and branch by removing one of the vertices in the subgraph. Because the maximum size of graphs in Ω_N is 8, we can find such a vertex subset in time $\mathcal{O}(n^8)$ if exists. If no such vertex subset is found, the remaining graph is Ω_N-free and the algorithm proceeds to the next phase. After the branching algorithm, we transform the given instance (G, k) into at most 8^k sub-instances (G', k') such that each sub-instance consists of an Ω_N-free graph G' and a remaining budget k'. It totally takes a time $8^k \cdot n^8$. Clearly, (G, k) is a Yes-instance if and only if one of sub-instances (G', k') is a Yes-instance.

Let (G', k') be a sub-instance obtained from the branching algorithm. Since G' is Ω_N-free, by Theorem 10.3, each connected component of G' is either a connected thread graph or a necklace graph on a directed cycle of length at least 9. By Proposition 10.15, we can compute a minimum thread vertex deletion set of G' in time $\mathcal{O}(n + m)$. So, we can decide whether (G', k') is a Yes-instance in time $\mathcal{O}(n + m)$. Since (G, k) is a Yes-instance if and only if one of sub-instances (G', k') is a Yes-instance, by checking all sub-instances, we can decide whether (G, k) is a Yes-instance in time $8^k \cdot \mathcal{O}(n + m)$. Therefore, we conclude that the Thread Vertex Deletion problem can be solved in time $8^k \cdot \mathcal{O}(n^8)$. □

10.3 A polynomial kernel for Thread Vertex Deletion

In this section, we prove that Thread Vertex Deletion has a polynomial size kernel.

10.3.1 Hitting small obstructions

Let $\mathcal{F}$ be a family of sets over a universe U. A subset $U' \subseteq U$ is called *a hitting set* of $\mathcal{F}$ if for every set $F \in \mathcal{F}$, $F \cap U' \neq \emptyset$. For a graph G and a family of graphs $\mathcal{F}$, a set $S \subseteq V(G)$ is also called *a hitting set* for $\mathcal{F}$ if for every induced subgraph H of G that is isomorphic to a graph in $\mathcal{F}$, $V(H) \cap S \neq \emptyset$. The following is a crucial ingredient for the polynomial kernel.

Lemma 10.16 (Fomin, Saurabh and Villanger [90]). *Let $\mathcal{F}$ be a family of sets of size at most d over a universe U, and let k be a positive integer. Then there is an $\mathcal{O}(|\mathcal{F}|(k+|\mathcal{F}|))$ time algorithm that finds a nonempty set $\mathcal{F}' \subseteq \mathcal{F}$ such that*

1. *for every $U' \subseteq U$ of size at most k, U' is a minimal hitting set of $\mathcal{F}$ if and only if U' is a minimal hitting set of $\mathcal{F}'$, and*
2. *$|\mathcal{F}'| \leqslant d!(k+1)^d$.*

Let (G,k) be an instance of Thread Vertex Deletion. We apply Lemma 10.16 for the set $\mathcal{F}$ of obstructions of size at most 8 in G.

Lemma 10.17. *Let (G,k) be an instance of* Thread Vertex Deletion. *There is a polynomial time algorithm that either concludes that (G,k) is a* No*-instance or finds a nonempty set $T \subseteq V(G)$ such that*

1. *$G\backslash T$ is a thread graph,*
2. *for every set $S \subseteq V(G)$ of size at most k, S is a minimal hitting set for Ω_N in G if and only if it is a minimal hitting set for Ω_N contained in $G[T]$, and*
3. *$|T| \leqslant 8 \cdot 8!(k+1)^8 + k$.*

Proof. Let $\mathcal{F}$ be the set of vertex sets S of G such that $G[S]$ is isomorphic to a graph in Ω_N. Since the maximum size of a set in $\mathcal{F}$ is 8, using Lemma 10.16, we can find a subset $\mathcal{F}'$ of $\mathcal{F}$ such that

1. for every vertex subset $X \subseteq V$ of size at most k, X is a minimal hitting set of $\mathcal{F}$ if and only if X is a minimal hitting set of $\mathcal{F}'$, and
2. $|\mathcal{F}'| \leqslant 8!(k+1)^8$.

Let $T' := \bigcup_{S\in\mathcal{F}'} S$. From the condition 1, $G\backslash T'$ has no induced subgraph isomorphic to a graph in Ω_N and by Theorem 10.3, $G\backslash T'$ is a graph whose component is either a necklace graph with underlying directed cycle of length at least 9 or a thread graph. So, using Proposition 10.15, we can find a minimum thread vertex deletion set Y of $G\backslash T'$ in polynomial time. If $|Y| \geqslant k+1$, then we conclude that (G,k) is a No-instance. Otherwise, we add Y to T', increasing its size by at most k. We conclude that $T := T' \cup Y$ is a required set. ☐

10.3.2 Bounding the Size of $G\backslash T$

The goal now is to shrink $G\backslash T$ while preserving the solutions. Let us fix in this section an instance (G,k) of Thread Vertex Deletion and also a subset T of V satisfying the conditions in Lemma 10.17. Let us remark that for every minimal hitting set S for Ω_N in G, we have that $S \subseteq T$.

A vertex v of G is called *irrelevant* if (G,k) is a Yes-instance if and only if $(G\backslash v,k)$ is a Yes-instance. We first show that if a thread block in $G\backslash T$ is large, then we can always find an irrelevant vertex in there.

Lemma 10.18. *If $G\backslash T$ contains a thread block $(G_{xy},\sigma_{xy},\ell_{xy})$ of size at least $(k+2)((8\cdot 8!(k+1)^8 + k)+2)^2+1$, we can find an irrelevant vertex in G_{xy} in polynomial time.*

To find an irrelevant vertex, we use the following lemma.

Lemma 10.19. *Let G be a graph and let $v_1v_2v_3v_4v_5$ be an induced path of length 4 in G. If two distinct vertices w_1,w_2 in $V(G)\backslash\{v_1,v_2,\ldots,v_5\}$ have the neighbors v_2 and v_4 in G, then $G\backslash v_3$ contains an induced subgraph isomorphic to a graph in Ω_N.*

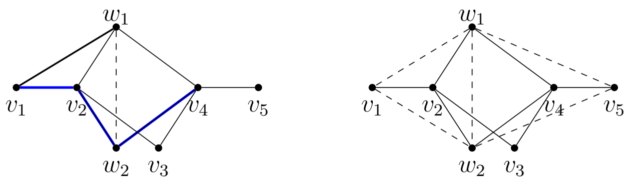


Figure 10.3: Cases in Lemma 10.19.

Proof. See Figure 10.3 for the following cases. If v_1 is adjacent to w_1 but not adjacent to w_2, then $(v_1v_2w_2v_4, w_1)$ is a bad pair in $G\backslash v_3$. Thus, by Lemma 10.4, $G\backslash v_3$ has an induced subgraph isomorphic to a graph in Ω_N. So, we may assume that for each $v \in \{v_1, v_5\}$, v is adjacent to both w_1, w_2 or neither of them. Depending on the adjacency between $\{v_1, v_5\}$ and $\{w_1, w_2\}$, and the adjacency between w_1 and w_2, we have one of the 6 graphs in Ω_N, which are $\alpha_1, \alpha_2, \ldots, \alpha_6$. □

Proof of Lemma 10.18. Suppose that $G\backslash T$ contains a thread block of size at least $(k+2)((8\cdot 8!(k+1)^8 + k)+2)^2+1$. We compute the canonical decomposition of each component of $G\backslash T$. Because thread blocks are divided by unmarked vertices that are the centers of star bags, we can compute the size of each thread block. Then we can find a thread block of size at least $(k+2)(8\cdot 8!(k+1)^8 + k + 2)^2 + 1$ in polynomial time. Let $B := B(x, y) = (B, \sigma, \ell)$ be a thread block of size at least $(k + 2)(8 \cdot 8!(k + 1)^8 + k + 2)^2 + 1$. For convenience, let σ' be the linear layout obtained from σ by removing the end vertices x and y.

In the following procedure, we mark some vertices of B in order to find an irrelevant vertex in B. We set $Z := \varnothing$.

1. For each v of T, choose the first $k + 2$ vertices z of σ' that are neighbors of v with $R \in \ell(z)$, and add them to Z. If there are at most $k + 1$ such vertices, then we add all of them into Z.
2. For each v of T, choose the last $k + 2$ vertices z of σ' that are neighbors of v with $L \in \ell(z)$, and add them to Z. If there are at most $k + 1$ such vertices, then we add all of them into Z.
3. For each pair of two vertices v, v' in T, choose $k + 2$ common neighbors of v and v' in B, and add them to Z. If there are at most $k + 1$ such vertices, then we add all of them into Z.
4. Choose the first $k + 2$ vertices z of σ' with $R \in \ell(z)$ and add them to Z. Choose the last $k + 2$ vertices z of σ' with $L \in \ell(z)$, and add them to Z.

Clearly, we can mark the set Z in polynomial time. The size of Z is bounded by

$$\begin{aligned} |T|(2k + 4) + |T|^2(k + 2) + (2k + 4) &= (k + 2)(|T|^2 + 2|T| + 2) \\ &\leqslant (k + 2)(8 \cdot 8!(k + 1)^8 + k + 2)^2 - 2. \end{aligned}$$

Since $|V(B)| \geqslant (k + 2)(8 \cdot 8!(k + 1)^8 + k + 2)^2 + 1$, there exists a vertex w in $V(B)\backslash Z\backslash\{x, y\}$. We claim that w is an irrelevant vertex.

If (G, k) is a YES-instance, then there exists a vertex set X of size at most k in G such that $G\backslash X$ is a thread graph. Since $(G\backslash w)\backslash X$ is a thread graph, $(G\backslash w, k)$ is a YES-instance. Now suppose that $(G\backslash w, k)$ is a YES-instance and let $X \subseteq V(G)\backslash\{w\}$ such that $|X| \leqslant k$ and $G\backslash(X \cup \{w\})$ is a thread graph. We may assume that $G\backslash X$ is not a thread graph. So, $G\backslash X$ must have an obstruction in Ω_T that contains the vertex w.

Since $X \cup \{w\}$ is a thread vertex deletion set of G, $X \cup \{w\}$ hits all induced subgraphs of Ω_N in G. Thus, there exists a vertex subset $Y \subseteq X \cap T$ that hits all induced subgraphs of Ω_N contained in $G[T]$.

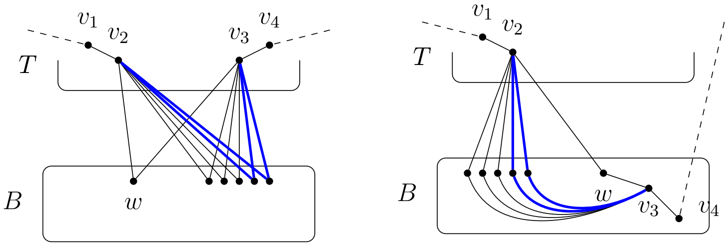


Figure 10.4: Case 1 and Case 2 in Lemma 10.18.

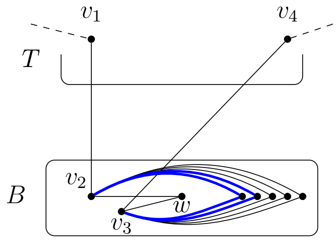


Figure 10.5: Case 3 in Lemma 10.18.

From the property of T, this set Y also hits all induced subgraphs of Ω_N in G. Since $Y \subseteq X$, $G\backslash X$ must have an induced cycle of length at least 9 that contains w.

Let C be an induced cycle of length at least 9 containing w in $G\backslash X$. We will find an induced subgraph of $G\backslash(X \cup \{w\})$ that is isomorphic to a graph in Ω_N, which leads a contradiction. Let v_1, v_2, w, v_3, v_4 be the consecutive vertices on C. Clearly, $v_1 - v_2 - w - v_3 - v_4$ is an induced path of length 4 in $G\backslash X$. We divide into cases depending on the places of v_2 and v_3.

Case 1. $v_2, v_3 \in T$.

Since v_2 and v_3 have a common neighbor w in $V(B)\backslash Z$, Z contains $k+2$ common neighbors of v_2 and v_3. Since $|X| \leqslant k$, there exist two vertices $w_1, w_2 \in Z\backslash X$ that are common neighbors of v_2 and v_3. By Lemma 10.19, $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N.

Case 2. Exactly one of v_2 *and* v_3 *is contained in* T.

From the symmetry, we may assume that $v_2 \in T$ and $v_3 \notin T$. Since $w \notin \{x, y\}$, v_3 is contained in B.

Case 2.1. $R \in \ell(w)$.

Since $R \in \ell(w)$, we have $L \in \ell(v_3)$ and $w <_\sigma v_3$, otherwise $L \in \ell(w)$ and we are in *Case 2.2.* From the construction of Z, Z contains the first $k+2$ vertices z of σ' that are neighbors of v_2 with $R \in \ell(z)$. Since $|X| \leqslant k$, we can choose two such vertices w_1, w_2 contained in $Z\backslash X$. Since w is not contained in Z, we have $w_1 <_\sigma w, w_2 <_\sigma w$, and they must be adjacent to v_3 in $G\backslash X$. Therefore, by Lemma 10.19, $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N.

Case 2.2. $L \in \ell(w)$.

Similar to Case 2.1, we may assume that $v_3 <_\sigma w$ and using the last $k+2$ vertices z of σ' that are neighbors of v_2 with $L \in \ell(z)$, we can verify that $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N.

Case 3. Neither v_2 *nor* v_3 *is contained in* T.

Since $w \notin \{x, y\}$, v_2 and v_3 are contained in B. If $v_2 <_\sigma w <_\sigma v_3$, then $R \in \ell(v_2)$, $L \in \ell(v_3)$ and it implies that $v_2v_3 \in E$. But this contradicts to the assumption that $v_1 - v_2 - w - v_3 - v_4$ is an induced path. Similarly, we may assume that $v_3 <_\sigma w <_\sigma v_2$, and thus, both of v_2 and v_3 appear either before w in σ or after w in σ.

By the symmetry, we may assume that v_2 and v_3 appear before w in σ. So, $R \in \ell(v_2)$, $R \in \ell(v_3)$, and $L \in \ell(w)$.

Since Z contains the last $k+2$ vertices z of σ' with $L \in \ell(z)$, there exist two vertices w_1, w_2 from those $k+2$ vertices that are not in X. Since w_1, w_2 appear after w and contain a label L, they are adjacent to both v_2 and v_3. Therefore, we have that $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N by Lemma 10.19.

In all cases, $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N. It contradicts to the assumption that $X \cup \{w\}$ is a thread vertex deletion set of G. Therefore, $G\backslash X$ is a thread graph, and we conclude that (G, k) is a YES-instance. $\square$

We say that (G, k) is *reduced* with respect to Lemma 10.18 if every thread block of $G\backslash T$ has size at most $(k+2)((8 \cdot 8!(k+1)^8 + k) + 2)^2$.

We now focus on the connected components of $G\backslash T$. By Lemma 10.18, a connected component of $G\backslash T$ is large if it contains a large number of thread blocks. We show that large components can be shrunk. The idea is that if a component is formed by a large number of thread blocks, then we can identify a sequence of consecutive thread blocks not touched by any obstruction. This allows us to contract one of these "safe" thread blocks, say $B(x, y)$, to a vertex v such that $N_{G\backslash T}(v) = (N_{G\backslash T}(x) \cup N_{G\backslash T}(y))\backslash B(x, y)$, hence reducing the input graph. We first prove that every obstruction in $\{\beta_1, \beta_2, \beta_3, \beta_4\}$, see Figure 5.7, either does not hit T or hits T in at least two vertices.

Lemma 10.20. *Let $U \subseteq T$ such that for every $u \in U$, there exists $S_u \subseteq V$ such that $S_u \cap T = \{u\}$ and $G[S_u]$ is a graph in $\{\beta_1, \beta_2, \beta_3, \beta_4\}$. If $|U| > k$, then (G, k) is a* NO*-instance; otherwise, (G, k) is a* YES*-instance if and only if $(G\backslash U, k - |U|)$ is a* YES*-instance.*

Proof. We claim that every minimal thread vertex deletion set in G contains U. Let S be a minimal thread vertex deletion set in G. Then there exists a vertex subset $S' \subseteq S$ such that S' is a minimal hitting set for graphs of Ω_N in $G[T]$. From the property of T, S' is also a minimal hitting set for graphs of Ω_N in G, and we must have $U \subseteq S' \subseteq S$ because S' hits the sets S_u for each $u \in U$, that induces a graph of $\{\beta_1, \beta_2, \beta_3, \beta_4\}$. It also implies that if $|U| > k$, then (G, k) is a NO-instance. Otherwise, since U is always contained in any minimal thread vertex deletion set of G, we have that (G, k) is a YES-instance if and only if $(G\backslash U, k - |U|)$ is a YES-instance. $\square$

By Lemmas 10.18 and 10.20 we can assume now that each thread block has size at most $(k+2)((8 \cdot 8!(k+1)^8 + k) + 2)^2$ and any obstruction from $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ either does not hit T or contains at least two vertices from T. We can with these assumptions prove that any connected component is small.

Lemma 10.21. *If $G\backslash T$ has a connected component with at least $19(6(8 \cdot 8!(k+1)^8 + k) + 1)$ thread blocks, then we can in polynomial time transform G into a graph G' with $|V(G')| < |V(G)|$ such that (G, k) is a* YES*-instance if and only if (G', k) is a* YES*-instance.*

Proof. Suppose that $G\backslash T$ has a component H such that H consists of at least $19(6(8 \cdot 8!(k+1)^8 + k) + 1)$ thread blocks. Let $\mathcal{L}$ be the sequence $B_1, B_2, \ldots, B_t$ of thread blocks of H.

We claim that every vertex v of T has neighbors in at most 6 thread blocks of H. Let $v \in T$ and for contradiction, suppose that v has neighbors in at least 7 thread blocks. Then we can choose three thread blocks $B_{t_1}, B_{t_2}, B_{t_3}$ having a neighbor of v in G such that

1. $B_{t_1}, B_{t_2}, B_{t_3}$ appear in this order in $\mathcal{L}$, and
2. $t_2 - t_1 \geqslant 3$, $t_3 - t_2 \geqslant 3$.

So, every vertex in B_{t_1} has no neighbors on B_{t_2} in H, and every vertex in B_{t_2} has no neighbors on B_{t_3} in H. For each $i \in \{1, 2, 3\}$, let p_i be a neighbor of v in B_{t_i}. Since each thread block of H has at least two vertices, we can choose a neighbor q_i of p_i in B_{t_i} for each $i \in \{1, 2, 3\}$. Depending on the adjacency between v and the vertices q_1, q_2, q_3, we have an induced subgraph of G that is isomorphic to a graph in $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ such that it has exactly one vertex of T. This contradicts to the assumption that (G, k) is an instance reduced with respect to Lemma 10.20.

Now, for each vertex v of T, we mark the thread blocks B of H if it has a neighbor in B. Since the number of thread blocks in H is at least $19(6(8 \cdot 8!(k+1)^8 + k) + 1)$ and $19(6(8 \cdot 8!(k+1)^8 + k) + 1) - 6(8 \cdot 8!(k+1)^8 + k) \geqslant 18(6(8 \cdot 8!(k+1)^8 + k) + 1)$, there exist consecutive non-marked thread blocks $B(v_{i_1}, v_{i_2}), B(v_{i_2}, v_{i_3}), \dots, B(v_{i_m}, v_{i_{m+1}})$ in $\mathcal{L}$ where $m \geqslant 18$.

We choose a thread block $B(v_{i_d}, v_{i_{d+1}}) = (B_{i_d}, \sigma_{i_d}, \ell_{i_d})$ where $d - 1 \geqslant 8$ and $m - d \geqslant 9$. Note that since $m \geqslant 18$, such a thread block exists. We transform the graph G into a smaller graph G' by removing the vertices of $V(B_{i_d})$ and adding a new vertex z such that $N_{G'}(z) = N_G(x) \cup N_G(y)$. Let H' be the component of $G' \backslash T$ that is modified from the component H of $G \backslash T$. Since we remove at least two vertices from G and add one vertex, we have $|V(G')| < |V(G)|$.

Now, we show that (G, k) is a YES-instance if and only if (G', k) is a YES-instance. Suppose that G has a minimal thread vertex set X. We first assume that $X \cap V(B_{i_d}) \neq \varnothing$ and let $q \in X \cap V(B_{i_d})$. Since X is a minimal thread vertex deletion set and all small obstructions of Ω_N are contained in $G \backslash V(B_{i_d})$, q must hit an induced cycle of length at least 9 in G, and the cycle must pass through the vertices x and y. Thus, $(X \backslash V(B_{i_d})) \cup \{z\}$ is a thread vertex deletion set of G' with $|(X \backslash V(B_{i_d})) \cup \{z\}| \leqslant k$.

Let us assume that $X \cap V(B_{i_d}) = \varnothing$. Suppose $G' \backslash X$ is not a thread graph, otherwise, (G', k) is a YES-instance. Then $G' \backslash X$ must have an induced cycle C of length at least 9 intersecting the new vertex z. The cycle obtained from C by replacing z with the edge xy is also an induced cycle of length at least 9 in $G \backslash X$. It contradicts to the assumption that $G \backslash X$ is a thread graph.

Now suppose that G' has a minimal thread vertex deletion set X. If $z \in X$, then z hits an induced cycle of length at least 9 in G' because of the minimality of X and the distance from x to the vertices of T. Because x hits all induced cycles of length at least 9 in G having a vertex of $V(B_{i_d})$, $(X \backslash \{z\}) \cup \{x\}$ is again a thread vertex deletion set of G.

Assume that $z \notin X$. Suppose $G \backslash X$ is not a thread graph, otherwise, (G, k) is a YES-instance. So, $G \backslash X$ has an induced subgraph isomorphic to an induced cycle C of length at least 9 passing through x and y. Let C' be the cycle obtained from C by replacing the edge xy with the vertex z. This cycle C' clearly exists in $G' \backslash X$ and it has length at least 9 because it should contain at least one vertex from the thread blocks $d - 1 \geqslant 8$ and $m - d \geqslant 9$. This contradicts to the assumption that $G' \backslash X$ is a thread graph. We conclude that (G, k) is a YES-instance if and only if (G', k) is a YES-instance. $\square$

We can now assume that every connected component of $G \backslash T$ has size bounded by

$$19(48 \cdot 8!(k+1)^8 + 6k + 1) \cdot (k+2)(8 \cdot 8!(k+1)^8 + k + 2)^2.$$

It remains now to bound the number of connected components which we show in the next two lemmas.

Lemma 10.22. *If $G \backslash T$ has at least $2(8 \cdot 8!(k+1)^8 + k) + 1$ connected components containing at least two vertices, then we can find an irrelevant vertex in polynomial time.*

Proof. If a component H of $G\backslash T$ contains no vertices having a neighbor in T, then we do not need to remove any vertex of H because H is a thread graph. Thus, (G,k) is a YES-instance if and only if $(G\backslash V(H),k)$ is a YES-instance. So, we may assume that every component of H contains a vertex having a neighbor in T.

Let $\mathcal{C}$ be the set of components of $G\backslash T$ which consist of at least two vertices. Since every component of H has a vertex having a neighbor in T, if $|\mathcal{C}| > 2(8 \cdot 8!(k+1)^8 + k)$, then there exists a vertex $u \in T$ such that u has neighbors in three distinct components of $\mathcal{C}$. Since each component of $\mathcal{C}$ has at least two vertices, there exists a vertex subset S of G such that S induces a graph in $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ and $S \cap T = \{u\}$. It contradicts to the assumption that (G,k) is an instance reduced with respect to Lemma 10.20. $\square$

Lemma 10.23. *If $G\backslash T$ has at least $(8 \cdot 8!(k+1)^8 + k)^2 \cdot (k+2) + 1$ isolated vertices, then we can find an irrelevant vertex in polynomial time.*

Proof. Let S be the union of isolated vertices in $G\backslash T$. If a vertex in S has no neighbors in T, then it is an irrelevant vertex. We may assume that every vertex in S has a neighbor in T.

We define a set Z, similarly in the proof of Lemma 10.18. For each pair of two vertices in T, choose $k+2$ common neighbors in S, and add them to Z. If there are at most $k+1$ common neighbors, then we add all of them into Z. Since $|S| > (8 \cdot 8!(k+1)^8 + k)^2 \cdot (k+2)$, there is a vertex w in $S\backslash Z$.

We claim that w is an irrelevant vertex of the problem. If (G,k) is a YES-instance, then there exists a vertex subset X of size at most k in G such that $G\backslash X$ is a thread graph. Since $G\backslash(X \cup \{w\})$ is also a thread graph, $(G\backslash w, k)$ is a YES-instance.

Suppose that $(G\backslash w, k)$ is a YES-instance. We choose a minimal vertex set X in $G\backslash w$ such that $|X| \leqslant k$ and $G\backslash(X \cup \{w\})$ is a thread graph. We may assume that $G\backslash X$ is not a thread graph. Let $X' \subseteq X \cup \{w\}$ be a hitting set for Ω_N in $G[T]$. Then by the property of T, X' also hits all induced subgraphs in G that are isomorphic to a graph of Ω_N. Since X already hits all small obstructions in G, there exists an induced cycle C of length at least 9 in $G\backslash X$ containing w.

Let w_1, w_2 be the neighbors of w on the cycle C. Since w_1, w_2 have $k+2$ common neighbors in Z, we may choose two vertices $z_1, z_2 \in Z\backslash X$ that are common neighbors of w_1 and w_2. By Lemma 10.19, we have that $G\backslash(X \cup \{w\})$ has an induced subgraph isomorphic to a graph in Ω_N, which implies that $G\backslash(X \cup \{w\})$ is not a thread graph. It is a contradiction, and we conclude that (G,k) is a YES-instance. $\square$

10.3.3 Kernel size

Let us now piece everything together and analyze the kernel size.

Proof of Theorem 10.2. Let (G,k) be an instance of THREAD VERTEX DELETION. We may safely assume that G has at most k connected components and that none of them is a thread graph. Let $T \subset V$ be a vertex subset satisfying Lemma 10.17.

By Lemma 10.20, we may assume that for every vertex subset $S \subseteq V$ such that $G[S]$ is a graph of $\{\beta_1, \beta_2, \beta_3, \beta_4\}$, $|S \cap T| \geqslant 2$. Combining Lemma 10.18 and Lemma 10.21, we can assume that every connected component of $G\backslash T$ has size at most $(k+2)(8 \cdot 8!(k+1)^8 + k + 2)^2 \cdot 19(6(8 \cdot 8!(k+1)^8 + k) + 1)$ (otherwise the instance can be reduced in polynomial time). Finally by Lemma 10.22 and Lemma 10.23, we can assume that the number of non-trivial components of $G\backslash T$ is at most $2(8 \cdot 8!(k+1)^8 + k)$ and the number of isolated vertices in $G \backslash T$ is at most $(8 \cdot 8!(k+1)^8 + k)^2(k+2)$. It follows that

$$|V \backslash T| \leqslant 2(8 \cdot 8!(k+1)^8 + k) \cdot 19(6(8 \cdot 8!(k+1)^8 + k) + 1) \cdot (k+2)((8 \cdot 8!(k+1)^8 + k) + 2)^2 + (8 \cdot 8!(k+1)^8 + k)^2 \cdot (k+2) = \mathcal{O}(k^{32})$$

Considering the number of components of G, we conclude that the kernel size is $\mathcal{O}(k^{33})$. $\square$

Chapter 11. Rank-width 1 vertex deletion

We discuss a graph modification problem related to distance-hereditary graphs. We recall the problem.

DISTANCE-HEREDITARY VERTEX DELETION (RANK-WIDTH 1 DELETION)
Input : A graph G, an integer k
Parameter : k
Question : Is there a vertex subset $S \subseteq V(G)$ of size at most k such that $G \backslash S$ is distance-hereditary?

We investigate a fixed parameter tractable algorithm for DISTANCE-HEREDITARY VERTEX DELETION.

Theorem 11.1. *For fixed k and an input graph G with n vertices, the* DISTANCE-HEREDITARY VERTEX DELETION *problem can be solved in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.*

Because the following sets are frequently used, we let

- $\Omega_D :=$ {house, gem, domino, hole}.

We recall that G is distance-hereditary if and only if G has no induced subgraph isomorphic to a graph in Ω_D. For a graph G, a set $S \subseteq V(G)$ is called a *DH deletion set* if $G \backslash S$ is distance-hereditary.

We use the technique, called *iterative compression*. The iterative compression tool was firstly developed for the Odd Cycle Transversal problem (vertex deleting to bipartite graphs) by Reed, Smith and Vetta [164], and further developed for various problems [48, 52, 70, 108, 116, 91]. Especially, we use a similar idea of Cao and Marx [45] to prove that Chordal Vertex Deletion (vertex deleting to chordal graphs) can be solved in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.

We formulate a new problem, usually called a *compression problem*, as follows.

DISTANCE-HEREDITARY COMPRESSION (G, t, S)
Input : A graph G, an integer $t \leqslant k$, $S \subseteq V(G)$ of size at most $k+1$ where $G \backslash S$ is distance-hereditary
PARAMETER : k
Question : Is there a vertex subset $S' \subseteq V(G) \backslash S$ of size at most t such that $G \backslash S'$ is distance-hereditary?

We prove the following.

Theorem 11.2. *The* DISTANCE-HEREDITARY COMPRESSION *problem can be solved in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ where n is the number of vertices of G.*

We first prove Theorem 11.1 assuming Theorem 11.2.

Proof of Theorem 11.1. Let G be a graph such that $V(G) = \{v_1, v_2, \dots, v_n\}$ and let k be a positive integer. For each $1 \leqslant i \leqslant n$, let $G_i := G[\{v_1, v_2, \dots, v_i\}]$. Note that $\varnothing$ is a DH deletion set of G_1 because G_1 is distance-hereditary.

Assume S_i is a DH deletion set of size at most k for G_i. Then clearly $G_{i+1} \backslash (S_i \cup \{v_{i+1}\})$ is also distance-hereditary. If there exists a DH deletion set of size at most k in G, then G_{i+1} must have a DH deletion set of size at most k. We show how to find one for G_{i+1}, if exists.

We guess a subset S' of $S_i \cup \{v_{i+1}\}$ that will be included in a DH deletion set of size at most k for G_{i+1}. This means that we will not remove the vertices of $(S_i \cup \{v_{i+1}\}) \backslash S'$. We have at most 2^{k+1} many branchings from this point, and for each subset S', we solve the Distance-Hereditary Compression problem with the instances

$$(G_{i+1} \backslash S', k - |S'|, (S_i \cup \{v_{i+1}\}) \backslash S').$$

We can easily see that G has a DH deletion set of size at most k if and only if one of the Distance-Hereditary Compression problems is a Yes-instance.

Since the Distance-Hereditary Compression problem is solved in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ by assumption, the Distance-Hereditary Vertex Deletion problem can be solved in time $n \cdot 2^{k+1} \cdot (2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}) = 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. □

From now on, we concentrate on the Distance-Hereditary Compression problem. For each compression, we will branch at most $\mathcal{O}(k^4)$ subproblems of the Distance-Hereditary Compression problem with smaller k. Since each branching will decrease k, these branchings appear $(\mathcal{O}(k^4))^{k+1}$ in total. Therefore, we can solve the Distance-Hereditary Compression problem in time $(\mathcal{O}(k^4))^{k+1} \cdot 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)} = 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.

We have three steps to design an algorithm for the Distance-Hereditary Compression problem. Let k be a positive integer and let $t \leqslant k+1$, and let G be a graph with n vertices and let $S \subseteq V(G)$ of size at most t where $G \backslash S$ is distance-hereditary,

1. We first search a house, a gem, or a domino in G in time $\mathcal{O}(n^6)$ and if we find one of the subgraph, then we branch by removing one of the vertices in the subgraph, that are not contained in S. Since we are not allowed to remove vertices in S, if the obstruction is contained in S, then it is a No-instance. By this preprocessing, we may assume that G is {house, gem, domino}-free.

2. We find a shortest hole X in G, and branch by removing some vertex on the hole. To do this, we pick some vertices of X, and we will call them *junctions*. We find a shortest hole X in time $\mathcal{O}(n^4(n+m))$ as follows. We guess four consecutive vertices v_1, v_2, v_3, v_4 in G and find a shortest path from v_1 to v_4 in
$$G \backslash \{v_2, v_3\} \backslash ((N(v_2) \cup N(v_3)) \backslash \{v_1, v_4\}).$$

3. We show that the number of junctions in the hole is bounded by $\mathcal{O}(k^3)$. If each interval has length at most 6, then we branch along all vertices of H. Otherwise, we have some intervals with length longer than 6. For such a long interval, we will show that we can find a same solution from other solution by replacing the part on that interval with some vertex separator between two end junctions. We will show that in a distance-hereditary graph, we can find a minimum vertex separator of two distinct vertices in polynomial time. We branch along a minimum vertex separator for each long interval to have a smaller instance. This finalizes the algorithm.

For easier discussion, we introduce one notion in canonical split decompositions. A *walk* in a graph G is a nonempty sequence $v_1 v_2 \cdots v_k$ of vertices in G such that $v_i v_{i+1} \in E(G)$ for each $1 \leqslant i \leqslant k-1$. Let D be the canonical split decomposition of a connected graph G. For two unmarked vertices v, w in

D, a walk in D from v to w is called a *semi-alternating walk* W in D if there are no two consecutive marked edges in W. We sometimes remove vertices to break a certain structure, and it makes a given decomposition disconnected. Note that two vertices are connected by a path in the original graph, then there exists a semi-alternating walk between the two vertices (they can use the same edges even if the corresponding structure in the original graph is a path), and if two vertices are disconnected, then there is no semi-alternating walk between the vertices.

We prove that a shortest hole in {house, gem, domino}-free graphs has a special property that for every vertex v outside the hole, the number of the neighbors of v on this hole is at most 3 and they are close to each other. This is an essential property to obtain an FPT algorithm for the Distance-Hereditary Vertex Deletion problem. We again use Lemma 10.4 to show it.

Lemma 11.3. *Let G be a {house, gem, domino}-free graph and let X be a shortest hole in G. If $v \in V(G)\backslash V(X)$, then v has at most 3 neighbors on X that have pairwise distance at most 2 on X.*

Proof. Let $X = v_1v_2 \cdots v_tv_1$ and let $I = \{i : vv_i \in E(G)\}$. Suppose there exist two vertices v_i, v_j in I such that one of the two paths from v_i to v_j in C has length ℓ where $3 \leqslant \ell \leqslant t-3$. By Lemma 10.4, G contains an induced subgraph isomorphic to either a house, a gem, a domino, or an induced cycle of length ℓ where $5 \leqslant \ell \leqslant t-1$. It contradicts to that G is {house, gem, domino}-free and X is the shortest hole of G. ☐

Let X be a shortest hole in G. Since we are solving the Distance-Hereditary Compression problem, we have $V(X) \cap S \neq \varnothing$, and $V(X) \cap S$ forms a disjoint union of induced paths in G. Note that the number of paths of $V(X) \cap S$ is at most $k+1$ because $|S| \leqslant k+1$.

In the next section, we define junctions on $V(X) \cap S$.

11.1 Junctions on the shortest hole

As we discussed before, we will mark some vertices of the hole to divide it into intervals, and call them junctions. We first analyze general properties of induced paths between two vertices in a distance-hereditary graph using its canonical split decomposition, and we will adapt those properties into $G\backslash S$.

11.1.1 Induced paths between two vertices in a distance-hereditary graph

Let H be a connected distance-hereditary graph with at least two vertices and let D_H be the canonical split decomposition of H, and x, y be two distinct vertices of H.

Let B_x, B_y be the bags of H such that $x \in B_x$ and $y \in B_y$. Then there is a path P of bags

$$B_x = B_1, B_2, \ldots, B_t = B_y$$

of D connecting B_x and B_y because H is connected. In the order of bags from B_1 to B_t on P, let

$$B'_1, B'_2, \ldots, B'_r$$

be the star bags whose centers are neither adjacent to a vertex of the previous bag nor the next bag. For each $1 \leqslant i \leqslant r$, we let w_i be the center of B'_i and let U_i be the set of unmarked vertices of D_H represented by w_i.

We characterize induced paths from x to y in H.

Lemma 11.4. *Let Q be an induced subgraph of H. Then the following are equivalent.*

1. *Q is an induced path from x to y in H.*

2. *For each $1 \leqslant i \leqslant r$, $|V(Q) \cap U_i| = 1$ and $V(Q)\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i) = \{x, y\}$.*

Proof. For convenience, let $q_0 := x$ and $q_{r+1} := y$. First suppose that for each $1 \leqslant i \leqslant r$, $V(Q) \cap U_i = \{q_i\}$ and $V(Q)\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i) = \{x, y\}$. Note that for each $0 \leqslant i \leqslant r$, q_i is adjacent to all vertices in U_{i+1} because they are linked by alternating paths in D_H. Thus, it is easy to observe that $xq_1q_2 \cdots q_r y$ is an induced path in H.

If Q is an induced path from x to y in H, then $|V(Q) \cap U_i| \geqslant 1$ otherwise, there is no semi-alternating walk from x to y in D_H, which means that x and y are disconnected in H. If $|V(Q) \cap U_j| \geqslant 2$ for some $1 \leqslant j \leqslant r$, then two vertices in $V(Q) \cap U_j$ will be adjacent to at least two of x, y, a vertex of $V(Q) \cap U_{j-1}$ and a vertex of $V(Q) \cap U_{j+1}$. It implies that Q contains a subgraph isomorphic to C_4. Thus, $|V(Q) \cap U_i| = 1$ for each $1 \leqslant i \leqslant r$, and since the set

$$\left(\bigcup_{1 \leqslant i \leqslant r} (V(Q) \cap U_i) \right) \cup \{x, y\}$$

already forms an induced path from x to y in H, there are no other vertices of Q in H. ☐

As a corollary, we can easily determine the distance from x to y in H, and minimal separators of x and y in H.

Lemma 11.5. *The distance from x to y in H is $r + 1$.*

Proof. This is clear from Lemma 11.4. ☐

Lemma 11.6. *A vertex set S of H is a minimal separator between x and y if and only if $S = U_i$ for some $1 \leqslant i \leqslant m$.*

Proof. If $S = U_i$, then there is no semi-alternating walk from x to y in D_H, and therefore, S is a separator between x and y in H. For $S' \subsetneq S$, we can choose a vertex s in $S \backslash S'$ so that we can link from x to y by a semi-alternating walk using s. This means that S is a minimal separator between x and y in H.

Suppose S is a minimal separator between x and y in H. If $|V(G\backslash S) \cap U_i| = 1$ for each $1 \leqslant i \leqslant r$, then by Lemma 11.4, there exists an induced path from x to y in $G\backslash S$, which contradicts to the assumption. Thus, $U_i \subseteq S$ for some i, and by the above argument, $U_i = S$. ☐

Note that $(\bigcup_{1 \leqslant i \leqslant r} U_i)$ separates H into several connected components, where two of them, say C_x and C_y, contain x and y, respectively. Since each component of $H\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i)$ may connect at most two sets U_i directly, we can naturally partition the components C of $H\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i)$ except C_x and C_y as follows.

1. For each $1 \leqslant i \leqslant r$, let $\mathcal{C}_i$ be the set of components C of $H\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i)$ such that C has a vertex linked to a vertex of U_i, and no vertex linked to a vertex of U_j where $j \neq i$.

2. For each $1 \leqslant j \leqslant r - 1$, let $\mathcal{D}_i$ be the set of components C of $H\backslash(\bigcup_{1 \leqslant i \leqslant r} U_i)$ such that C has a vertex linked to a vertex of U_i or U_{i+1}, and no vertex linked to a vertex of U_j where $j \neq i, i + 1$.

It is not hard to observe that for $1 \leqslant i \leqslant r$ and $C \in \mathcal{C}_i$, $N_H(C) \subseteq U_i$.

11.1.2 Junctions

We recall that we are solving the Distance-Hereditary Compression problem with an instance (G, t, S) where $t \leqslant k$ and G is {house, gem, domino}-free, and X is a shortest hole of G. Let D be the canonical split decomposition of $G\backslash S$. We remind that $(G\backslash S)[X]$ is a disjoint union of paths.

Let $P := p_0p_1 \cdots p_{m+1}$ be one of the paths of $(G\backslash S)[V(X)]$. Consider the two end vertices p_0 and p_{m+1} in $G\backslash S$, and let $B_{p_0}, B_{p_{m+1}}$ be the bags of $G\backslash S$ such that $p_0 \in B_{p_0}$ and $p_m \in B_{p_{m+1}}$. In the order of bags from B_{p_0} to $B_{p_{m+1}}$ in D, let

$$B'_1, B'_2, \ldots, B'_r$$

be the star bags whose centers are neither adjacent to a vertex of the previous bag or the next bag. For each $1 \leqslant i \leqslant r$, we let w_i be the center of B'_i and let U_i be the set of unmarked vertices of D represented by w_i, and we define sets $\mathcal{C}_i$, $\mathcal{D}_i$ for each $1 \leqslant i \leqslant r$ similarly in the previous section. From Lemma 11.5, we can easily deduce that $r = m$.

For each $1 \leqslant i \leqslant r-1$, we define

$$M_i := U_i \cup \left(\bigcup_{C \in \mathcal{C}_i} C \right) \cup \left(\bigcup_{C \in \mathcal{D}_i} C \right),$$

and

$$M_r := U_r \cup \left(\bigcup_{C \in \mathcal{C}_r} C \right).$$

Note that if $i \neq j$, then $M_i \cap M_j = \varnothing$.

A vertex p on the path P is called a *junction* if $p = p_0$, or p_{m+1}, or there exists a vertex $v \in M_i$ having a neighbor in S where $1 \leqslant i \leqslant m$. We say a vertex $s \in S$ *witnesses* a junction p_i if either s is adjacent to p_0 or p_{m+1}, or there exists a vertex $v \in M_i$ such that v is adjacent to s.

The following lemma gives a bound on the number of junctions.

Lemma 11.7. *If a vertex $s \in S$ witnesses at least $5k+5$ junctions on P, then G has $k+1$ holes where the intersection of them is exactly s. This implies that (G, t, S) is a* No*-instance.*

Proof. Let $s \in S$, and let $p_{j_1}, p_{j_2}, \ldots, p_{j_\ell}$ be the junctions on P in the order that are witnessed by s. We assume that $\ell \geqslant 5k+5$. For each $1 \leqslant m \leqslant \ell$, we choose a vertex $w_{j_m} \in M_{j_m}$ adjacent to s.

For each $0 \leqslant m \leqslant k$, we choose an induced path P_m from $w_{j_{5m+1}}$ to $w_{j_{5m+4}}$ in $G\backslash S$. Since the two bags $B_{j_{5m+2}}$ and $B_{j_{5m+3}}$ are on the path from the bag containing $w_{j_{5m+1}}$ to the bag containing $w_{j_{5m+4}}$, by Lemma 11.4, this path P_m has length at least 3. Note that for each $0 \leqslant m \leqslant k-1$, $V(P_m) \cap V(P_{m+1}) = \varnothing$, because $B_{j_{5m+5}}$ is a separator between $V(P_m)$ and $V(P_{m+1})$, and moreover, if $i \neq j$, then $V(P_i) \cap V(P_j) \neq \varnothing$.

Since G is {house, gem, domino}-free, by Lemma 10.4, each graph $G[V(P_m) \cup \{s\}]$ contains a hole having s. So, G has $k+1$ holes whose pairwise intersection is exactly the vertex s. Since we are solving the Distance-Hereditary Compression problem, it is not allowed to remove this vertex s, and therefore, (G, t, S) is a No-instance. □

From Lemma 11.7, we may assume that the total number of junctions on P is at most $(5k+4)(k+1)$ because $|S| \leqslant k+1$. Since $(G\backslash S)[X]$ has at most $k+1$ components, the total number of junctions on X is at most $(5k+4)(k+1)^2$.

Now we describe the branching step based on these junctions. If we do not break the hole using the vertices near junctions, we need to remove some minimum separator between two consecutive junctions, otherwise, we still have a hole of same size. Lemma 11.6 gives a way to find a minimum separator between two vertices in $G\backslash S$.

11.1.3 Breaking long intervals

Lemma 11.8. *Let p_i and p_j be two consecutive junctions on P such that $j - i \geqslant 4$. Every hole in G containing a vertex of U_ℓ for some $i + 2 \leqslant \ell \leqslant j - 2$ also contains one vertex from each $U_{\ell'}$ for $i + 1 \leqslant \ell' \leqslant j - 1$.*

Proof. Let Y be a hole in G, and suppose that Y contains a vertex of U_ℓ for some $i+2 \leqslant \ell \leqslant j-2$, but does not contain a vertex from $U_{\ell'}$ for some $i+1 \leqslant \ell' \leqslant j-1$. Since U_ℓ is a separator between U_{i+1} and U_{j-1} in $G\backslash S$ and the hole Y cannot end inside of $G\backslash S$, this hole Y should intersect on one of U_{i+1} and U_{j-1} with at least two vertices. By symmetry, we may assume that $\ell < \ell'$ and thus $|V(Y) \cap U_{i+1}| \geqslant 2$.

If $|Y \cap M_i| \leqslant 1$, then Y is contained in $G\backslash S$, which is a contradiction. Therefore, we may choose two vertices x, y in U_{i+1} that have distinct two neighbors x', y' in M_i. However, by the definition of splits, x, y should be completely adjacent to x', y', which contradicts that they are vertices on an induced cycle.

It is clear that Y cannot have two vertices from some $U_{\ell'}$, otherwise, H contains a subgraph isomorphic to a cycle of length 4 with two vertices in $U_{\ell'-1}$ and $U_{\ell'+1}$. □

From the previous lemma, if we do not remove some vertex on X that has distance $\leqslant 1$ from junctions, we need to remove some minimal separator between two junctions in $G\backslash S$. Therefore, we guess one of the intervals between two consecutive junctions p_i and p_j to clear, and for that interval, it is enough to find a minimal separator between U_{i+1} and U_{j-1} in $G\backslash S$, and remove it. Note that if (G, t, S) is a Yes-instance, then the number of total intervals is at most $(5k+5)(k+1)^2$ by Lemma 11.7.

11.2 Distance-Hereditary Compression problem

Now we prove the main result of this chapter.

Proof of Theorem 11.2. Let (G, t, S) be an instance of the Distance-Hereditary Compression problem. We first search an induced subgraph isomorphic to either a house, a gem, or a domino and if we find one, then we branch into instances $(G\backslash v, t-1, S\backslash\{v\})$ by removing one of the vertices v in the subgraph. It takes $\mathcal{O}(n^6)$ time. By this processing, we may assume that G is {house, gem, domino}-free. We find a canonical split decomposition D of $G\backslash S$ in time $\mathcal{O}(n+m)$. Next, we find a shortest hole X of G in time $\mathcal{O}(n^4 \cdot (n+m))$ as follows. We guess four consecutive vertices v_1, v_2, v_3, v_4 as a part of X and find a shortest path from v_1 to v_4 in $G\backslash\{v_2, v_3\}\backslash((N(v_2) \cup N(v_3))\backslash\{v_1, v_4\})$.

Now we mark the junctions of the hole X. Let $P := p_0 p_1 \cdots p_{m+1}$ be a component of $(G\backslash S)[X]$. Let $B_{p_0}, B_{p_{m+1}}$ be the bags of $G\backslash S$ such that $p_0 \in B_{p_0}$ and $p_m \in B_{p_{m+1}}$. In the order of bags from B_{p_0} to $B_{p_{m+1}}$ in D, we mark all bags

$$B'_1, B'_2, \ldots, B'_r$$

that are the star bags whose centers are neither adjacent to a vertex of the previous bag or the next bag. It can be done in time $\mathcal{O}(n)$. For each $1 \leqslant i \leqslant r$, we let w_i be the center of B'_i and let U_i be the set of unmarked vertices of D represented by w_i, and we define sets $\mathcal{C}_i$, $\mathcal{D}_i$, M_i for each $1 \leqslant i \leqslant m$ as defined in Sections 11.1.1 and 11.1.2. Since the set of components of $(G\backslash S)\backslash(\bigcup_{1\leqslant i\leqslant m} U_i)$ can be computed in polynomial time, we can also compute all of $\mathcal{C}_i$, $\mathcal{D}_i$, M_i in polynomial time, for each component $(G\backslash S)[X]$. If M_i contains a neighbor of some vertex of S, then we mark p_i as a junction. If the number of all junctions in X is at least $(5k+4)(k+1)^2+1$, then there exists a vertex s that witnesses at least $5k+5$ junctions on one of the components of $(G\backslash S)[X]$, and by Lemma 11.7, it is a No-instance. Thus, we may assume that X contains at most $(5k+4)(k+1)^2$ junctions.

We first branch along all vertices on X that have distance at most 1 from some junction in X. Since the number of all junctions on X is at most $(5k+4)(k+1)^2$, it needs $\mathcal{O}(k^3)$ branchings. Now we may assume that any solution of the instance (G,t,S) does not contain a vertex in X that has distance at most 1 from some junction in X.

Let S' be a minimum DH deletion set of (G,t,S) where $S' \subseteq V(G)\backslash S$. If S' does not contain any vertex separator fully between two junctions p and p' of X for all pairs of two consecutive junctions of X, then $G\backslash S'$ contains another hole by choosing remaining vertices from each U_i. It implies that S' should contain at least one vertex separator T between two consecutive junctions p and p'. Let T' be a minimum separator between p and p' in $G\backslash S$. Note that it can be computed easily from Lemma 11.6.

We claim that $S'\backslash T \cup T'$ is again a minimum DH deletion set of (G,t,S). Let t be the vertex of $T \cap V(X)$. Since S' is a minimum DH deletion set of (G,t,S), $G\backslash(S'\backslash\{t\})$ contains a hole X' having the vertex t. From our assumption, t has distance at least two from p and p'. By Lemma 11.8, this hole X' should contain a vertex from each set U_i between p and p' in $G\backslash S$. However, $X' \cap T' \neq \varnothing$ because T' is a separator between p and p' in $G\backslash S$, thus $(S'\backslash T) \cup T'$ is again a DH deletion set of (G,t,S). Since $|T'| \leqslant |T|$, $(S'\backslash T) \cup T'$ is a minimum DH deletion set of (G,t,S).

So, we choose one of the intervals between two consecutive junctions that have distance at least 4 and find a minimum separator T between them, and branch into instances $(G\backslash T, t-|T|, S\backslash T)$ by removing it. Since the number of intervals are at most $\mathcal{O}(k^3)$, it needs $\mathcal{O}(k^3)$ branchings.

Each branching decrease k by at least 1, and the number of all subproblems will be at most $\mathcal{O}(k^3)^{k+1}$. This would give that total running time $\mathcal{O}(k^3)^{k+1} \cdot 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)} = 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. $\square$

References

[1] I. Adler, A. M. Farley, and A. Proskurowski. Obstructions for linear rank-width at most 1. *Discrete Applied Mathematics*, 168(0):3–13, 2014.

[2] I. Adler and M. M. Kanté. Linear rank-width and linear clique-width of trees. *Theoretical Computer Science*, http://dx.doi.org/10.1016/j.tcs.2015.04.021, 2015.

[3] I. Adler, M. M. Kanté, and O. Kwon. Linear rank-width of distance-hereditary graphs. In D. Kratsch and I. Todinca, editors, *Graph-Theoretic Concepts in Computer Science*, volume 8747 of *Lecture Notes in Computer Science*, pages 42–55. Springer International Publishing, 2014.

[4] D. Adolphson and T. C. Hu. Optimal linear ordering. *SIAM J. Appl. Math.*, 25:403–423, 1973. Collection of articles dedicated to William Prager on the occasion of his seventieth birthday.

[5] E. Amir. Approximation algorithms for treewidth. *Algorithmica*, 56(4):448–479, 2010.

[6] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[7] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Appl. Math.*, 23(1):11–24, 1989.

[8] R. Balasubramanian, M. R. Fellows, and V. Raman. An improved fixed-parameter algorithm for vertex cover. *Inform. Process. Lett.*, 65(3):163–168, 1998.

[9] H. Bandelt and H. M. Mulder. Distance-hereditary graphs. *J. Comb. Theory, Ser. B*, 41(2):182–208, 1986.

[10] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, June 2004.

[11] Y. Z. Ben-Dor A, Shamir R. Clustering gene expression patterns. *J Comput Biol.*, 6(3-4):281–297, 1999.

[12] M. W. Bern, E. L. Lawler, and A. L. Wong. Linear-time computation of optimal subgraphs of decomposable graphs. *J. Algorithms*, 8(2):216–235, 1987.

[13] D. Bienstock, N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a forest. *J. Comb. Theory, Ser. B*, 52(2):274–283, 1991.

[14] S. Böcker. A golden ratio parameterized algorithm for cluster editing. *J. Discrete Algorithms*, 16:79–89, 2012.

[15] S. Böcker, S. Briesemeister, and G. W. Klau. Exact algorithms for cluster editing: evaluation and experiments. *Algorithmica*, 60(2):316–334, 2011.

[16] H. Bodlaender and T. Kloks. Approximating treewidth and pathwidth of some classes of perfect graphs. In *Algorithms and computation (Nagoya, 1992)*, volume 650 of *Lecture Notes in Comput. Sci.*, pages 116–125. Springer, Berlin, 1992.

[17] H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Automata, languages and programming (Tampere, 1988)*, volume 317 of *Lecture Notes in Comput. Sci.*, pages 105–118. Springer, Berlin, 1988.

[18] H. L. Bodlaender. On disjoint cycles. In *Proceedings of the 17th International Workshop*, WG '91, pages 230–238, London, UK, UK, 1992. Springer-Verlag.

[19] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernet.*, 11(1-2):1–21, 1993.

[20] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[21] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. An $O(c^k n)$ 5-approximation algorithm for treewidth. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science—FOCS 2013*, pages 499–508. IEEE Computer Soc., Los Alamitos, CA, 2013.

[22] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, and minimum elimination tree height. *J. Algorithms*, 18:238–255, 1991.

[23] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996.

[24] H. L. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discrete Math.*, 8(4):606–616, 1995.

[25] H. L. Bodlaender and A. M. C. A. Koster. Combinatorial optimization on graphs of bounded treewidth. *Comput. J.*, 51(3):255–269, May 2008.

[26] H. L. Bodlaender and R. H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Discrete Math.*, 6(2):181–188, 1993.

[27] A. Bouchet. Caractérisation des symbols croiśś de genre nul. *C.R. Acad. Sci. Paris*, 274:724–727, 1972.

[28] A. Bouchet. Isotropic systems. *European J. Combin.*, 8(3):231–244, 1987.

[29] A. Bouchet. Reducing prime graphs and recognizing circle graphs. *Combinatorica*, 7(3):243–254, 1987.

[30] A. Bouchet. Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B*, 45(1):58–76, 1988.

[31] A. Bouchet. Transforming trees by successive local complementations. *J. Graph Theory*, 12(2):195–207, 1988.

[32] A. Bouchet. Connectivity of isotropic systems. In *Combinatorial Mathematics: Proceedings of the Third International Conference (New York, 1985)*, volume 555 of *Ann. New York Acad. Sci.*, pages 81–93, New York, 1989. New York Acad. Sci.

[33] A. Bouchet. An efficient algorithm to recognize locally equivalent graphs. *Combinatorica*, 11(4):315–329, 1991.

[34] A. Bouchet. Recognizing locally equivalent graphs. *Discrete Math.*, 114(1-3):75–86, 1993. Combinatorics and algorithms (Jerusalem, 1988).

[35] A. Bouchet. Circle graph obstructions. *J. Combin. Theory Ser. B*, 60(1):107–144, 1994.

[36] A. Bouchet, W. H. Cunningham, and J. F. Geelen. Principally unimodular skew-symmetric matrices. *Combinatorica*, 18(4):461–486, 1998.

[37] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM, 1999.

[38] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 517–526. IEEE Computer Soc., Los Alamitos, CA, 2009.

[39] B. Bui-Xuan, M. M. Kanté, and V. Limouzy. A note on graphs of linear rank-width 1. *preprint*, arxiv.org/abs/1306.1345, 2013.

[40] B. Bui-Xuan, J. A. Telle, and M. Vatshelle. Boolean-width of graphs. *Theoret. Comput. Sci.*, 412(39):5187–5204, 2011.

[41] M. Burlet and J.-P. Uhry. Parity graphs. In *Topics on perfect graphs*, volume 88 of *North-Holland Math. Stud.*, pages 253–277. North-Holland, Amsterdam, 1984.

[42] J. F. Buss and J. Goldsmith. Nondeterminism within P. *SIAM J. Comput.*, 22(3):560–572, 1993.

[43] A. Cabello, M. G. Parker, G. Scarpa, and S. Severini. Exclusivity structures and graph representatives of local complementation orbits. *Journal of Mathematical Physics*, 54(7):–, 2013.

[44] Y. Cao, J. Chen, and Y. Liu. On feedback vertex set new measure and new structures. In *Algorithm theory—SWAT 2010*, volume 6139 of *Lecture Notes in Comput. Sci.*, pages 93–104. Springer, Berlin, 2010.

[45] Y. Cao and D. Marx. Chordal editing is fixed-parameter tractable. In *31st International Symposium on Theoretical Aspects of Computer Science*, volume 25 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 214–225. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2014.

[46] Y. Cao and D. Marx. Interval deletion is fixed-parameter tractable. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 122–141. SIAM, 2014.

[47] C. Chekuri and J. Chuzhoy. Polynomial bounds for the grid-minor theorem. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 60–69, New York, NY, USA, 2014. ACM.

[48] J. Chen, F. V. Fomin, Y. Liu, S. Lu, and Y. Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. System Sci.*, 74(7):1188–1198, 2008.

[49] J. Chen, I. A. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.

[50] J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theoret. Comput. Sci.*, 411(40-42):3736–3756, 2010.

[51] J. Chen, L. Liu, and W. Jia. Improvement on vertex cover for low-degree graphs. *Networks*, 35(4):253–259, 2000. Selected papers from the Workshop on Computational Graph Theory (Kowloon, 1999).

[52] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 177–186, New York, NY, USA, 2008. ACM.

[53] C. Chun, G. Ding, B. Oporowski, and D. Vertigan. Unavoidable parallel minors of 4-connected graphs. *J. Graph Theory*, 60(4):313–326, 2009.

[54] V. Chvátal and P. L. Hammer. Aggregation of inequalities in integer programming. In *Studies in integer programming (Proc. Workshop, Bonn, 1975)*, pages 145–162. Ann. of Discrete Math., Vol. 1. North-Holland, Amsterdam, 1977.

[55] S. Cicerone and G. D. Stefano. On the extension of bipartite to parity graphs. *Discrete Applied Mathematics*, 95(1–3):181–195, 1999.

[56] D. Corneil, E. Gioan, C. Paul, and M. Tedder. Practical and efficient circle graph recognition. *Algorithmica*, 69(4):759–788, 2014.

[57] D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3(3):163–174, 1981.

[58] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inform. and Comput.*, 85(1):12–75, 1990.

[59] B. Courcelle. Graph transformations expressed in logic and applications to structural graph theory. Report of Banff workshop in Graph Minors (08w5079), 2008. http://www.birs.ca/events/2008/5-day-workshops/08w5079/report08w5079.pdf.

[60] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic*, volume 138 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2012. A language-theoretic approach, With a foreword by Maurice Nivat.

[61] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[62] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

[63] B. Courcelle and S. Oum. Vertex-minors, monadic second-order logic, and a conjecture by seese. 97(1):91–126, 2007.

[64] W. H. Cunningham. Decomposition of directed graphs. *SIAM J. Algebraic Discrete Methods*, 3(2):214–228, 1982.

[65] W. H. Cunningham and J. Edmonds. A combinatorial decomposition theory. *Canad. J. Math.*, 32(3):734–765, 1980.

[66] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time (extended abstract). In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*, pages 150–159. IEEE Computer Soc., Los Alamitos, CA, 2011.

[67] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. Wojtaszczyk. An improved fpt algorithm and a quadratic kernel for pathwidth one vertex deletion. *Algorithmica*, 64(1):170–188, 2012.

[68] E. Dahlhaus. Parallel algorithms for hierarchical clustering, and applications to split decomposition and parity graph recognition. *Journal of Algorithms*, 36(2):205–240, 2000.

[69] H. de Fraysseix. Local complementation and interlacement graphs. *Discrete Math.*, 33(1):29–35, 1981.

[70] F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens. An o(2o(k)n3) fpt algorithm for the undirected feedback vertex set problem. *Theory of Computing Systems*, 41(3):479–492, 2007.

[71] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[72] E. D. Demaine and M. Hajiaghayi. Bidimensionality: new connections between FPT algorithms and PTASs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 590–601. ACM, New York, 2005.

[73] E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008.

[74] J. S. Dharmatilake. A min-max theorem using matroid separations. In *Matroid theory (Seattle, WA, 1995)*, volume 197 of *Contemp. Math.*, pages 333–342. Amer. Math. Soc., Providence, RI, 1996.

[75] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34(3):313–356, Sept. 2002.

[76] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg, fourth edition, 2010.

[77] R. Diestel, T. R. Jensen, K. Y. Gorbunov, and C. Thomassen. Highly connected sets and the excluded grid theorem. *J. Combin. Theory Ser. B*, 75(1):61–73, 1999.

[78] G. Ding and P. Chen. Unavoidable doubly connected large graphs. *Discrete Math.*, 280(1-3):1–12, 2004.

[79] G. Ding, B. Oporowski, J. Oxley, and D. Vertigan. Unavoidable minors of large 3-connected binary matroids. *J. Combin. Theory Ser. B*, 66(2):334–360, 1996.

[80] G. Ding, B. Oporowski, J. Oxley, and D. Vertigan. Unavoidable minors of large 3-connected matroids. *J. Combin. Theory Ser. B*, 71(2):244–293, 1997.

[81] F. Dorn, F. V. Fomin, and D. M. Thilikos. Subexponential parameterized algorithms. In *Automata, languages and programming*, volume 4596 of *Lecture Notes in Comput. Sci.*, pages 15–27. Springer, Berlin, 2007.

[82] R. Downey and M. Fellows. Fixed-parameter tractability and completeness. III. Some structural aspects of the W hierarchy. In *Complexity theory*, pages 191–225. Cambridge Univ. Press, Cambridge, 1993.

[83] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness. In *Proceedings of the Twenty-first Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, MB, 1991)*, volume 87, pages 161–178, 1992.

[84] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*. Texts in Computer Science. Springer, London, 2013.

[85] J. A. Ellis, I. H. Sudborough, and J. S. Turner. The vertex separation and search number of a graph. *Inf. Comput.*, 113(1):50–79, 1994.

[86] S. Even and A. Itai. Queues, stacks and graphs. *Theory of Machines and Computations*, pages 71–86, 1971.

[87] U. Feige, M. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.

[88] M. R. Fellows and M. A. Langston. Layout permutation problems and well-partially-ordered sets. In *Advanced research in VLSI (Cambridge, MA, 1988)*, pages 315–327. MIT Press, Cambridge, MA, 1988.

[89] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Clique-width is NP-complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009.

[90] F. Fomin, S. Saurabh, and Y. Villanger. A polynomial kernel for proper interval vertex deletion. In L. Epstein and P. Ferragina, editors, *Algorithms -ESA 2012*, volume 7501 of *Lecture Notes in Computer Science*, pages 467–478. Springer Berlin Heidelberg, 2012.

[91] F. V. Fomin, S. Gaspers, D. Kratsch, M. Liedloff, and S. Saurabh. Iterative compression and exact algorithms. *Theoretical Computer Science*, 411(7–9):1045–1053, 2010.

[92] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 470–479, 2012.

[93] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 503–510. SIAM, Philadelphia, PA, 2010.

[94] A. Frank. Packing paths, circuits, and cuts—a survey. In *Paths, flows, and VLSI-layout (Bonn, 1988)*, volume 9 of *Algorithms Combin.*, pages 47–100. Springer, Berlin, 1990.

[95] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1–3):3–31, 2004. Papers presented at the 2002 IEEE Symposium on Logic in Computer Science (LICS).

[96] C. P. Gabor, K. J. Supowit, and W. L. Hsu. Recognizing circle graphs in polynomial time. *J. Assoc. Comput. Mach.*, 36(3):435–473, 1989.

[97] R. Ganian. Thread graphs, linear rank-width and their algorithmic applications. In *Combinatorial Algorithms*, volume 6460 of *Lecture Notes in Comput. Sci.*, pages 38–42. Springer, 2011.

[98] R. Garbe. Tree-width and path-width of comparability graphs of interval orders. In *Graph-theoretic concepts in computer science (Herrsching, 1994)*, volume 903 of *Lecture Notes in Comput. Sci.*, pages 26–37. Springer, Berlin, 1995.

[99] C. Gavoille and C. Paul. Distance labeling scheme and split decomposition. *Discrete Mathematics*, 273(1-3):115–130, 2003.

[100] J. Geelen, B. Gerards, and G. Whittle. Excluding a planar graph from GF(q)-representable matroids. *J. Combin. Theory Ser. B*, 97(6):971–998, 2007.

[101] J. Geelen and S. Oum. Circle graph obstructions under pivoting. *J. Graph Theory*, 61(1):1–11, 2009.

[102] J. F. Geelen. *Matchings, matroids and unimodular matrices.* PhD thesis, University of Waterloo, 1995.

[103] J. F. Geelen and S. Oum. Circle graph obstructions under pivoting. *Journal of Graph Theory*, 61(1):1–11, 2009.

[104] E. Gioan and C. Paul. Split decomposition and graph-labelled trees: characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Appl. Math.*, 160(6):708–733, 2012.

[105] M. C. Golumbic. *Algorithmic graph theory and perfect graphs.* Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London-Toronto, Ont., 1980. With a foreword by Claude Berge, Computer Science and Applied Mathematics.

[106] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica*, 39(4):321–347, 2004.

[107] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: fixed-parameter algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005.

[108] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386 – 1396, 2006.

[109] J. Gustedt. On the pathwidth of chordal graphs. *Discrete Appl. Math.*, 45(3):233–248, 1993. ARIDAM IV and V (New Brunswick, NJ, 1988/1989).

[110] P. L. Hammer and F. Maffray. Completely separable graphs. *Discrete Appl. Math.*, 27(1-2):85–99, 1990. Computational algorithms, operations research and computer science (Burnaby, BC, 1987).

[111] L. H. Harper. Optimal numberings and isoperimetric problems on graphs. *J. Combinatorial Theory*, 1:385–393, 1966.

[112] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973.

[113] E. Howorka. A characterization of distance-hereditary graphs. *Quart. J. Math. Oxford Ser. (2)*, 28(112):417–420, 1977.

[114] E. Howorka. On metric properties of certain clique graphs. *J. Combin. Theory Ser. B*, 27(1):67–74, 1979.

[115] E. Howorka. A characterization of Ptolemaic graphs. *J. Graph Theory*, 5(3):323–331, 1981.

[116] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.*, 47(1):196–217, 2010.

[117] J. Javelle, M. Mhalla, and S. Perdrix. New protocols and lower bounds for quantum secret sharing with graph states. In *Theory of quantum computation, communication, and cryptography*, volume 7582 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, Heidelberg, 2013.

[118] V. Jelínek. The rank-width of the square grid. *Discrete Appl. Math.*, 158(7):841–850, 2010.

[119] J. Jeong, O. Kwon, and S. Oum. Excluded vertex-minors for graphs of linear rank-width at most k. *European J. Combin.*, 41:242–257, 2014.

[120] I. Kanj, M. Pelsmajer, and M. Schaefer. Parameterized algorithms for feedback vertex set. In R. Downey, M. Fellows, and F. Dehne, editors, *Parameterized and Exact Computation*, volume 3162 of *Lecture Notes in Computer Science*, pages 235–247. Springer Berlin Heidelberg, 2004.

[121] M. M. Kanté, E. J. Kim, O. Kwon, and C. Paul. Fpt algorithm and polynomial kernel for linear rank-width one vertex deletion. *preprint*, arxiv.org/abs/1504.05905, 2015.

[122] H. Kaplan and R. Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. *SIAM J. Comput.*, 25(3):540–561, 1996.

[123] N. Kashyap. Matroid pathwidth and code trellis complexity. *SIAM J. Discrete Math.*, 22(1):256–272, 2008.

[124] K. Kawarabayashi and Y. Kobayashi. Linear min-max relation between the treewidth of H-minor-free graphs and its largest grid minor. In *29th International Symposium on Theoretical Aspects of Computer Science*, volume 14 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 278–289. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012.

[125] K. Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. *J. Combin. Theory Ser. B*, 102(2):424–435, 2012.

[126] D. C. Kay and G. Chartrand. A characterization of certain ptolemaic graphs. *Canad. J. Math.*, 17:342–346, 1965.

[127] E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 613–624, 2013.

[128] N. G. Kinnersley. The vertex separation number of a graph equals its path-width. *Inform. Process. Lett.*, 42(6):345–350, 1992.

[129] T. Kloks, H. Bodlaender, H. Müller, and D. Kratsch. Computing treewidth and minimum fill-in: All you need are the minimal separators. In T. Lengauer, editor, *Algorithms–ESA '93*, volume 726 of *Lecture Notes in Computer Science*, pages 260–271. Springer Berlin Heidelberg, 1993.

[130] T. Kociumaka and M. Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556 – 560, 2014.

[131] A. Kotzig. Eulerian lines in finite 4-valent graphs and their transformations. In *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, pages 219–230. Academic Press, New York, 1968.

[132] A. Kotzig. Quelques remarques sur les transformations κ. séminaire Paris. 1977.

[133] O. Kwon. Connecting rank-width and tree-width via pivot-minors. *M.S. thesis, KAIST*, 2012.

[134] O. Kwon and S. Oum. Graphs of small rank-width are pivot-minors of graphs of small tree-width. *Discrete Appl. Math.*, 168:108–118, 2014.

[135] O. Kwon and S. Oum. Unavoidable vertex-minors in large prime graphs. *European J. Combin.*, 41:100–127, 2014.

[136] J. Lagergren. Efficient parallel algorithms for graphs of bounded tree-width. *J. Algorithms*, 20(1):20–44, 1996.

[137] J. Lagergren. Upper bounds on the size of obstructions and intertwines. *J. Combin. Theory Ser. B*, 73(1):7–40, 1998.

[138] A. Leaf and P. Seymour. Treewidth and planar minors. *Manuscript*, 2012.

[139] C. Leiserson. Area-efficient graph layouts. In *Foundations of Computer Science, 1980., 21st Annual Symposium on*, pages 270–281, Oct 1980.

[140] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36(2):177–189, 1979.

[141] F. Makedon and I. H. Sudborough. On minimizing width in linear layouts. *Discrete Appl. Math.*, 23(3):243–265, 1989.

[142] D. Markham and B. C. Sanders. Graph states for quantum secret sharing. *Phys. Rev. A (3)*, 78(4):042309, 17, 2008.

[143] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 35(1):18–44, 1988.

[144] H. Nagamochi. Linear layouts in submodular systems. In K.-M. Chao, T.-s. Hsu, and D.-T. Lee, editors, *Algorithms and Computation*, volume 7676 of *Lecture Notes in Computer Science*, pages 475–484. Springer Berlin Heidelberg, 2012.

[145] W. Naji. Reconnaissance des graphes de cordes. *Discrete Math.*, 54(3):329–337, 1985.

[146] J. Nešetřil and R. Thomas. A note on spatial representation of graphs. *Comment. Math. Univ. Carolin.*, 26(4):655–659, 1985.

[147] R. Niedermeier and P. Rossmanith. Upper bounds for vertex cover further improved. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 561–570. Springer, Berlin, 1999.

[148] R. Niedermeier and P. Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *J. Algorithms*, 47(2):63–77, 2003.

[149] B. Oporowski, J. Oxley, and R. Thomas. Typical subgraphs of 3- and 4-connected graphs. *J. Combin. Theory Ser. B*, 57(2):239–257, 1993.

[150] S. Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005.

[151] S. Oum. Rank-width and well-quasi-ordering. *SIAM J. Discrete Math.*, 22(2):666–682, 2008.

[152] S. Oum. Rank-width is less than or equal to branch-width. *J. Graph Theory*, 57(3):239–244, 2008.

[153] S. Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):Art. 10, 20, 2009.

[154] S. Oum. Excluding a bipartite circle graph from line graphs. *J. Graph Theory*, 60(3):183–203, 2009.

[155] S. Oum and P. Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006.

[156] J. Oxley. *Matroid Theory.* Number 21 in Oxford Graduate Texts in Mathematics. Oxford University Press, second edition, 2012.

[157] G. Philip, V. Raman, and Y. Villanger. A quartic kernel for pathwidth-one vertex deletion. In D. Thilikos, editor, *Graph Theoretic Concepts in Computer Science*, volume 6410 of *Lecture Notes in Computer Science*, pages 196–207. Springer Berlin Heidelberg, 2010.

[158] Y. Rabinovich, J. A. Telle, and M. Vatshelle. Upper bounds on boolean-width with applications to exact algorithms. In *Parameterized and exact computation*, volume 8246 of *Lecture Notes in Comput. Sci.*, pages 308–320. Springer, Cham, 2013.

[159] V. Raman and S. Saurabh. Parameterized complexity of directed feedback set problems in tournaments. In *Algorithms and data structures*, volume 2748 of *Lecture Notes in Comput. Sci.*, pages 484–492. Springer, Berlin, 2003.

[160] V. Raman, S. Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Algorithms and computation*, volume 2518 of *Lecture Notes in Comput. Sci.*, pages 241–248. Springer, Berlin, 2002.

[161] F. P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc.*, s2-30:264–286, 1930.

[162] M. Rao. Solving some NP-complete problems using split decomposition. *Discrete Appl. Math.*, 156(14):2768–2780, 2008.

[163] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[164] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299 – 301, 2004.

[165] B. A. Reed. Finding approximate separators and computing tree width quickly. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 221–228, New York, NY, USA, 1992. ACM.

[166] N. Robertson and P. Seymour. Graph minors. XXI. Graphs with unique linkages. *J. Combin. Theory Ser. B*, 99(3):583–616, 2009.

[167] N. Robertson and P. Seymour. Graph minors. XXII. Irrelevant vertices in linkage problems. *J. Combin. Theory Ser. B*, 102(2):530–563, 2012.

[168] N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Combin. Theory Ser. B*, 62(2):323–348, 1994.

[169] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *J. Comb. Theory, Ser. B*, 35(1):39–61, 1983.

[170] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[171] N. Robertson and P. D. Seymour. Graph minors V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–144, 1986.

[172] N. Robertson and P. D. Seymour. Graph minors. VIII. A Kuratowski theorem for general surfaces. *J. Combin. Theory Ser. B*, 48(2):255–288, 1990.

[173] N. Robertson and P. D. Seymour. An outline of a disjoint paths algorithm. In *Paths, flows, and VLSI-layout (Bonn, 1988)*, volume 9 of *Algorithms Combin.*, pages 267–292. Springer, Berlin, 1990.

[174] N. Robertson and P. D. Seymour. An outline of a disjoint paths algorithm. In *Paths, flows, and VLSI-layout (Bonn, 1988)*, volume 9 of *Algorithms Combin.*, pages 267–292. Springer, Berlin, 1990.

[175] N. Robertson and P. D. Seymour. Graph minors. X. obstructions to tree-decomposition. 52(2):153–190, 1991.

[176] N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995.

[177] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004.

[178] P. D. Seymour. Decomposition of regular matroids. *J. Combin. Theory Ser. B*, 28(3):305–359, 1980.

[179] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Appl. Math.*, 144(1-2):173–182, 2004.

[180] J. Spinrad. Recognition of circle graphs. *J. Algorithms*, 16(2):264–282, 1994.

[181] K. Suchan and I. Todinca. Pathwidth of circular-arc graphs. In *Graph-theoretic concepts in computer science*, volume 4769 of *Lecture Notes in Comput. Sci.*, pages 258–269. Springer, Berlin, 2007.

[182] A. Takahashi, S. Ueno, and Y. Kajitani. Minimal acyclic forbidden minors for the family of graphs with bounded path-width. *Discrete Mathematics*, 127(1-3):293–304, 1994.

[183] S. Thomassé. A quadratic kernel for feedback vertex set. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 115–119. SIAM, Philadelphia, PA, 2009.

[184] W. T. Tutte. A homotopy theorem for matroids. I, II. *Trans. Amer. Math. Soc.*, 88:144–174, 1958.

[185] W. T. Tutte. Lectures on matroids. *J. Res. Nat. Bur. Standards Sect. B*, 69B:1–47, 1965.

[186] W. T. Tutte. Connectivity in matroids. *Canad. J. Math.*, 18:1301–1324, 1966.

[187] R. van Bevern, C. Komusiewicz, H. Moser, and R. Niedermeier. Measuring indifference: Unit interval vertex deletion. In D. Thilikos, editor, *Graph Theoretic Concepts in Computer Science*, volume 6410 of *Lecture Notes in Computer Science*, pages 232–243. Springer Berlin Heidelberg, 2010.

[188] P. van't Hof and Y. Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013.

[189] T. Wittkop, J. Baumbach, F. Lobo, and S. Rahmann. Large scale clustering of protein sequences with force –a layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, 8(1):396, 2007.

[190] J. Zhang. Local complementation rule for continuous-variable four-mode unweighted graph states. *Phys. Rev. A*, 78:034301, Sep 2008.

Index

Summary

On the structural and algorithmic properties of linear rank-width

이 논문에서는 linear rank-width의 성질들에 대해서 알아보았다. 알려진 path-width와 tree-width의 관계와 같이 linear rank-width는 rank-width의 선형형태로 볼 수 있다. Path-width에 대해 알려진 결과들로부터 동기를 부여받아, linear rank-width와 관련된 질문들을 던지고 그 질문들에 답을 하였다.

첫 번째로, linear rank-width와 연관된 그래프의 구조적 성질에 대해 관찰하였다. 본 저자는 linear rank-width가 k 이하인 그래프 모임에 대해 vertex-minor 관계로 최소인 그래프들의 개수가 적어도 $2^{\Omega(3^k)}$ 개 이상 있음을 증명하였다. 주어진 그래프 G와 상수 k에 대해 G의 linear rank-width가 k이하임을 테스트하는 문제에 대한 constructive한 fixed parameter tractable 알고리즘이 알려진 것이 없는데, vertex-minor 관계로 최소인 그래프들을 모두 구할 수 있으면, 이런 알고리즘을 제시할 수 있다. Linear rank-width가 k 이하인 그래프 모임에 대한 vertex-minor 관계로 최소인 그래프의 크기의 상한을 구하는 것에 대해 알려진 것이 없으며, 이를 알아내는 것은 흥미로운 미해결 문제이다.

Split과 prime 그래프 개념은 이 논문에서 가장 중요하게 쓰인 도구 중 하나이다. 본 저자는 고정된 트리 그래프 T에 대해 충분히 linear rank-width가 큰 그래프는 T를 반드시 vertex-minor로 가진다는 질문을 던졌다. 이에 대해 이 질문을 증명하는 것은 이 질문을 prime 그래프 상에서 증명하는 것과 동치임을 증명하였다. 또한, 고정된 상수 n에 대해, 다른 상수 N이 존재하여 점의 개수가 N개 이상인 prime 그래프는 반드시 길이 n인 원 그래프를 가지거나 아니면 완전 이분그래프 $K_{2,n}$의 선 그래프를 vertex-minor로 가짐을 증명하였다.

두 번째는, linear rank-width와 관련된 그래프의 알고리즘을 개발하였다. 먼저, 본 저자는 처음으로 n개의 점을 가지는 distance-hereditary 그래프 상에서 linear rank-width를 다항식 시간에 계산하는 알고리즘을 고안하였다. 이 결과를 이용하면 independent set 오라클이 있다는 가정 하에 branch-width가 2 이하인 매트로이드의 path-width도 다항식 시간에 계산할 수 있음을 증명하게 된다. 이를 일반화하는 문제로서 rank-width가 상한된 그래프에 대해서도 linear rank-width를 계산하는 다항식 알고리즘을 생각해 볼 수 있는데, 이에 대해 아직 알려진 바가 없다.

마지막으로, 본 저자는 Linear rank-width w 점 지우기과 Rank-width w 점 지우기 문제를 w가 1인 경우에 대하여 연구하였다. 다음으로 연구해볼 수 있는 것은 1보다 큰 w에 대해서 비슷한 fixed parameter tractable 알고리즘과 다항식 kernel의 존재 여부이다. 본 저자는 특히, Rank-width 1 점 지우기 문제를 $2^{\mathcal{O}(k \log_2 k)} \cdot n^{\mathcal{O}(1)}$ 시간에 해결할 수 있음을 증명하였는데, 이를 어떤 상수 c에 대해 $c^k \cdot n^{\mathcal{O}(1)}$ 시간에 풀 수 있음을 미해결 문제로 제시한다.

감 사 의 글

먼저 이 논문이 있기까지 많은 도움을 주신 엄상일 교수님께 진심으로 감사의 말을 드립니다. 교수님께서는 그래프 이론이라는 분야에 대해 조금의 흥미가 있었을 뿐이었던 저에게 좋은 수업으로서 영감을 주셨었고, 논문을 찾아보면서 연구자의 길을 택해도 흥미로울 것 같다는 느낌을 충분히 가지게 해 주셨습니다. 교수님의 늘 부지런하심과 연구자로서의 바른 자세는 저에게는 언제나 따르고자 하는 잣대가 되어주셨고, 그 덕분에 많이 배우고 조그만 결과들을 내서 졸업할 수 있었던 것 같습니다. 또한, 좋은 연구자들과 연결고리들을 많이 만들어주셨고, 한국에 있지만 다른 연구자들과의 교류를 지속적으로 할 수 있게 아낌없이 지원해주신 것에 감사하게 생각하고 있습니다. 함부르크라는 좋은 곳에서 6개월동안 같이 생활했었던 시간들은 잊지 못할 것 같습니다.

이 논문을 열심히 심사해주신 진교택 교수님, Otfried Cheong 교수님, Andreas Holmsen 교수님, 김석진 교수님께 감사의 말씀을 드립니다. 특별히 많은 연구를 같이 진행한 Mamadou Kantè 교수님께 감사의 글을 남깁니다. 연구자로서나 동료로서, 그는 저에게 많은 아이디어들을 서슴없이 제안하여 주었고, 같이 토론하고 연구하는 것을 좋아해 주었습니다. 특히 Clermont-Ferrand에 두 번이나 방문할 수 있는 기회를 가지게 해주었고, 연구하는 즐거움을 많이 불어넣어 주셔서 감사하게 생각합니다. 그 외에 같이 연구를 진행하였던 김은정 박사님과 Isolde Adler, Chrsitophe Paul, Hans Bodlaender, Stefan Kratsch, Vincent Kreuzen, Petr Hliněný, Jan Obdržálek와 Sebastian Ordyniak 모두에게 감사의 말씀을 드립니다. 정말 같이 연구할 수 있는 기회들을 가질 수 있어서 즐거웠고, 이 분들의 도움이 없이는 논문을 완성하지 못하였을 것입니다.

사랑하는 부모님과 이 논문을 쓴 기쁨을 함께 하고 싶습니다. 어렸을 때부터 수학을 좋아했었던 저를 언제나 인정해주시고 비록 다른 공부는 좋아하지 않아서 스트레스를 많이 받으실 때에도 좋아하는 것을 하게끔 지켜봐주신 부모님께 감사드립니다. 제가 조금만 더 현명했더라면 과학고를 통해서 카이스트에 왔을 지도 모르겠지만, 어쩌면 그렇지 않았었기에 아직도 수학을 즐겁게 하고 있는 게 아닌가 생각도 듭니다. 초등학교 때 올림피아드 문제집을 다 풀었을 때 처음부터 끝까지 손수 지워주셔서 다시 풀게끔 해 주셨던 것을 비롯해서 여러가지로 어머니께서 저에게 해 주셨던 정성이 없었더라면 저는 여기까지 올 생각을 못 했을 것이라고 생각합니다. 아버지는 제가 어릴 적에 가족 여행처럼 경시대회 같은 곳에 같이 가는 것을 좋아하셨었는데, 옛날에 그렇게 다니던 것이 추억으로 많이 남아 있습니다. 늘 건강하게 키워주시고 아껴주시고 응원해 주신 것에 감사드리고, 오래오래 건강하게 계셨으면 좋겠습니다. 사랑하는 동생 오영이와 어릴 때 오랫동안 같이 지낸 혜원이 누나에게도 고맙다는 말을 보냅니다.

무엇보다도 사랑하는 저의 친구 윤희에게 소중한 논문을 바칩니다. 교수님과 같이 함부르크에 있던 시절, 교회에서 우리는 만났고, 박사기간 3년동안 저에게 모든 힘이 되어주었습니다. 수학에만 몰입하면서 힘들고 외로우면서 감정적으로 지칠 때면 저를 위해 아낌없이 기도해 주고, 꾸준히 공부를 할 수 있게 도와 주었습니다. 비록 2년이라는 오랜 기간동안 멀리 떨어져 있음에도 항상 응원해 주었고, 많은 부분들을 지혜롭게 해쳐나갈 수 있게 도와주어서 고맙다는 말 마음 깊이 전하고 싶습니다.

우리 연구실을 지나쳤던 모든 친구들에게도 감사하다는 말을 드립니다. 석사 시절에 같이 보냈던 린기에게 처음 왔을 때 많이 해 준 조언에 대해 고맙게 생각하고 있습니다. 덴마크에 있으면서도 같이 자주 만나서 연구도 하고, 같이 연구하는 게 즐겁다고 생각하게 해 주는 성민이에게도 고맙다는 얘기를 남기고 싶습니다. 카이스트에 있었던 5년이라는 긴 시간동안 항상 같이 있었던 지수에게도 고맙다고 얘기하고 싶고, 늘 칠칠맞지 못했던 형과 같이 있느라고 고생했다고 전하고 싶습니다. 락용이와, 그리고 호진이와 기원이, 동엽이, 또 지금 포닥으로 와 있는 일규와 김연진 박사님에게도, 모두에게 고맙고 같이 있었던 시간들을 기억하고 있겠습니다.

같이 꼭 박사 하자고 한양대에서 대학원 같이 준비했었고, 계속해서 열심히 하고 있는 승로, 영훈이, 재광이에게 고맙다는 얘기하고 싶습니다. 2010년에 같이 카이스트에 입학해서 시간을 같이 보냈었던 미희, 초롱이, 철광이, 준하 등 모두 고맙고, 한양대에서 같이 왔던 기림이와 병준이, 영민이에게 좋은 시간들을 보냈던 것에 감사하게 생각합니다. 처음 그래프 이론 수업 때 만나서 재밌는 시간을 같이 보냈던 소희와 상돈이에게도 고맙다는 얘기 하고 싶고, 같이 얘기 많이 나누었던 우상이에게도 고맙다고 전하고 싶습니다. 그 외 모두 나열하지는 못해도 한양대와 카이스트에서 만나서 같이 얘기를 나누고 시간을 보냈던 친구들에게 모두 감사의 말을 드립니다.

이 력 서

이 름 : 권 오 정

생 년 월 일 : 1984년 9월 12일

출 생 지 : 경상북도 안동시

본 적 지 : 경상북도 안동시 용상동 1192-6번지

주 소 : 대전시 유성구 구성동 373-1번지 한국과학기술원 수리과학과

E-mail 주 소 : ojoungkwon@gmail.com

학 력

2000. 3. – 2003. 2. 안동고등학교

2003. 3. – 2010. 8. 한양대학교 수학과 (B.S.)

2010. 9. – 2012. 8. 한국과학기술원 수리과학과 (M.S.)

2012. 9. – 2015. 8. 한국과학기술원 수리과학과 (Ph.D.)

경 력

2011. 3. – 2014. 12. 한국과학기술원 수리과학과 일반조교 (Graph Theory, Discrete Mathematics)

2015. 1. – 2015. 3. RWTH Aachen University Research Fellowship

학 회 활 동

1. I. Adler, M. M. Kanté, and **O. Kwon**, *Linear rank-width of distance-hereditary graphs*, 40th International Workshop on Graph-Theoretic Concepts in Computer Science, Le Domaine de Chales, near Orleans (France), June., 2014.DOI: 10.1007/978-3-319-12340-0-4.
2. J. Jeong, **O. Kwon**, and S. Oum, *Excluded vertex-minors for graphs of linear rank-width at most k*, in Proc. 30th International Symposium on Theoretical Aspects of Computer Science, Kiel (Germany), Mar., 2013. DOI: 10.4230/LIPIcs.STACS.2013.221.

연 구 업 적

1. H. L. Bodlaender, S. Kratsch, V. Kreuzen, **O. Kwon**, and S. Ok, *Characterizing width two for variants of treewidth*, Discrete Applied Math. Accepted in Jan, 2015. `http://dx.doi.org/10.1016/j.dam.2015.01.015`

2. J. Jeong, **O. Kwon** and S. Oum. *Excluded vertex-minors for graphs of linear rank-width at most k*, European J. Combin. 41(October, 2014), pp. 242-257. `http://dx.doi.org/10.1016/j.ejc.2014.04.010`

3. **O. Kwon** and S. Oum. *Unavoidable vertex-minors in large prime graphs*, European J. Combin. 41(October, 2014), pp. 100-127. `http://dx.doi.org/10.1016/j.ejc.2014.03.013`

4. **O. Kwon** and S. Oum. *Graphs of small rank-width are pivot-minors of graphs of small tree-width*, Discrete Applied Math. 168(May 11, 2014), pp. 108-118. `http://dx.doi.org/10.1016/j.dam.2013.01.007`

5. M. M. Kanté, E. Kim, **O. Kwon**, and C. Paul, *FPT algorithm and polynomial kernel for linear rank-width one vertex deletion*, arXiv:1504.05905, preprint.

6. P. Hliněný, **O. Kwon**, J. Obdržálek, and S. Ordyniak, *Tree-depth and Vertex-minors*, arXiv:1403.7024, submitted.

7. I. Adler, M. M. Kanté, and **O. Kwon**, *Linear rank-width of distance-hereditary graphs*, arXiv:1403.1081, preprint.