論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	The Path-Width of Graphs and Its Applications
著者(和文)	高橋篤司
Author(English)	Atsushi Takahashi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:乙第2853号, 授与年月日:1996年2月29日, 学位の種別:論文博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:乙第2853号, Conferred date:1996/2/29, Degree Type:Thesis doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

The Path-Width of Graphs and Its Applications

Atsushi TAKAHASHI

Department of Electrical and Electronic Engineering
Tokyo Institute of Technology

October 1995

Contents

Chapter 1

Introduction

1.1 Applications

The *path-width* of a graph was introduced by Robertson and Seymour [71], and it was shown that path-width is an important parameter with regards to minor containment of forests. It has been shown that a wide variety of concepts important from an application point of view have close connection with path-width. In what follows, we review some of those applications.

1.1.1 VLSI Gate Matrix Layout

Gate matrix layout problem [29, 62] is a combinatorial problem that arises in several VLSI layout styles, including gate matrix layout, PLAs with multiple folding, Weinberger arrays, and others.

In the most general form, an instance consists of an $n \times m$ Boolean matrix M, whose rows and columns represent the nets and the gates of the circuit, respectively. The gates may be thought of as the basic electronic devices that are arranged linearly in a row, and the nets as realizing connections between them. Connections are realized horizontally by reserving for a given permutation of the gates for every net the part of the row from the leftmost to the rightmost gate to which a connection must be established. The connections may share the same row, called track, if they have no column in common. Minimizing the layout area, that is, minimizing the number of tracks, leads to the following gate matrix layout problem. We are given an $n \times m$ Boolean matrix M and an integer k, and are asked whether we can permute the columns of M so that, if in each row we change to $*$ every 0 lying between the row's leftmost and rightmost 1, then no column contains more than k 1's and $*$'s. The cost of a permutation is the maximum number of 1's and $*$'s in any column of the corresponding matrix. The *cost* of a gate matrix layout is the minimum cost over all its permutations.

As described in [36] an instance of the problem can be mapped to an equivalent instance with only two 1's per column and then modeled as a graph in which vertices correspond to the rows and edges to the columns. Fellows and Langston [37] showed that the path-width of a graph corresponding to a gate matrix layout with cost k is $k-1$.

1.1.2 VLSI Channel Routing

An *interval graph* is the intersection graph of a set of intervals on a real line, in which the vertices correspond to intervals and two vertices are connected by an edge if and only if the corresponding intervals have nonempty intersection. Interval graphs have been studied with regard to their application to a variety of subjects. For an overview see [42].

The concept of VLSI *channel routing* was introduced by Hashimoto and Stevens in [45]. A channel is a rectangle routing region with two rows of terminals along its top and bottom. A connection between terminals is routed by line segments on orthogonal grids which are called horizontal and vertical tracks. Horizontal tracks are isolated from vertical tracks, and connections between them are made through via holes on grid points. A net is the set of terminals to be connected. The terminals of each net in the channel are connected without connecting terminals of distinct nets. The major object of the problem is to minimize the channel height, that is, the number of horizontal tracks used to connect the terminals of every net in the channel. The *density* of a channel is the maximum number of nets that can be cut by a vertical line through the channel, where we say a net is cut by a vertical line if the net contains one or more terminals on each side of the line or if the line passes through a terminal belonging to the net (and the net is not a two terminal net both of whose terminals are on the line). A trivial lower bound for the channel height is determined by the channel density. Let $\mathcal{I}$ be the set of intervals corresponding to nets in the channel such that the left (respectively, right) endpoint of an interval correspond to the position of the leftmost (respectively, rightmost) terminal of the corresponding net. Then the density of the channel is equal to the number of vertices in the maximum clique of the interval graph of $\mathcal{I}$.

Assume that terminals of nets can be moved with some constraints so that the density of a channel is minimized. Let G be a graph such that the vertices correspond to nets of a channel and two vertices are connected by an edge if and only if there exists a constraint that the connections of the corresponding nets must be cross the same vertical line in the channel. Although it is easy to see that not every graph is an interval graph, every graph G is a subgraph of an interval graph since the complete graph with the same number of vertices of G is an interval graph of G. An interval graph obtained from G by adding edges is called an *interval supergraph* of G. Each interval supergraph of G corresponds to a channel routing problem with two rows of terminals satisfying the constraints. The problem is stated as follows: find an interval supergraph graph of G such that the corresponding channel has minimum density. Kirousis and Papadimitriou defined the *interval thickness* of a graph G which is the number of vertices in the smallest maximum clique over all interval supergraphs of G. This formulation has already existed in [65] for gate matrix layout problem. Möhring [62] and Scheffler [91] showed that the interval thickness of a graph is equal to its path-width plus one.

1.1.3 VLSI Linear Layout

A *separator* of a connected graph is a set of vertices whose removal disconnects the remainder of the graph. A separator whose removal separates the graph into two connected components of nearly equal size has applications in VLSI layout [55] and divide and conquer algorithms [59]. Lengauer [56] called this a "static" definition of separator and introduced a corresponding "dynamic" notion of a separator, called *vertex separation game*. We consider the same concept as Lengauer but describe it in terms of linear layouts.

A *linear layout* of a graph $G = (V, E)$ is a one-to-one mapping $L{:}V \rightarrow \{1, 2, \ldots, |V|\}$. For any layout L, define $V_L(i)$ be the set of vertices of G mapped into integers less than or equal to i that are adjacent to vertices mapped into integers greater than i. The vertex separation number of G with respect to L, denoted by $vs_L(G)$, is the maximum number of vertices in any $V_L(i)$. The *vertex separation number* of G, denoted by $vs(G)$, is the minimum $vs_L(G)$ over all possible layouts L of G. Kinnersley [48] showed that the vertex separation number is identical to the path-width.

The concept of the vertex separation is closely related to a well known problem on undirected graphs, called the *min-cut linear arrangement* problem [40]. Indeed, if we use edge separators instead of (vertex) separators in the above definition, we have the min-cut linear arrangement problem. For any layout L, define $E_L(i)$ be the set of edges of G that connect vertices mapped to integers less than or equal to i and vertices mapped to integers greater than i. The cut-width of G with respect to L, denoted by $cw_L(G)$, is the maximum number of edges in any $E_L(i)$. The *cut-width* of G, denoted by $cw(G)$, is the minimum $cw_L(G)$ over all possible layouts L of G. The min-cut linear arrangement problem is the problem of finding the cut-width of a graph. This problem formalizes the cost measure for certain approaches to VLSI layout and have been extensively studied so far [57, 26, 100, 60]. Chung and Seymour [25] showed a relation between cut-width and path-width.

Pebbling is a technique that allows relationships between time and space to be studied by means of a game played on directed acyclic graphs. Computation time is modeled by the length of the pebbling strategy. Storage space is modeled by the maximum number of pebbles used at any instant in the game. For an overview see [67]. The *black pebble game* models deterministic sequential computation using pebbles of only one color (black) which represent computations. The *black and white pebble game* models nondeterministic sequential computation. In the black and white pebble game, we allow a second kind of pebbles called white pebbles which represent nondeterministic computations. The *progressive pebble game* does not allow repebbling which has been proposed as a model of register allocation for the computation of arithmetic expressions, where recomputation is not deemed realistic. Lengauer [56] showed that progressive black and white pebble games and vertex separation games are polynomially reducible from one to the other.

1.1.4 Network Survivability

Consider the behavior of a rogue program, such as a computer worm or virus, in a network. As soon as it is decided that the network is indeed infected, then all locations must be suspected of being infected, and must be systematically tested and cleared. Suppose that only a few copies of the vaccine program are available, and that it is impossible or impractical to generate more copies. Then a clearing strategy is needed to use this limited resource. Further, a poor strategy may cause some nodes to become reinfected. It is irrelevant whether the rogue program is malicious or not, one must always assume that it will spread whenever it can.

This type of model of damage spread, worst case spread, was formulated by Breisch [17] and Parsons [66]. They, motivated by the problem of locating a lost explorer in a maze of caves, invented a game on graphs, called *edge-searching*. We will define this search game precisely in Chapter 4, but an informal description is as follows. A graph is thought of as a system of tunnels, and is supposed that an infinitely fast and cunning intruder is hiding in the graph. This intruder must be captured by searchers which slide along the edges at finite speed and cannot see the intruder until they capture him. The *edge-search number* of a graph is the minimum number of searchers necessary to guarantee the capture of the intruder that cannot move through edges and vertices with searchers. Kirousis and Papadimitriou [51] introduced another variant called *node-searching.* In node-searching, we do not slide searchers along the edges, but only place them as guards on the vertices. An intruder on an edge is captured by placing searchers at both end vertices of the edge simultaneously. *Mixed-searching* is a natural generalization of edge-searching and node-searching which have been extensively studied so far. In mixed-searching, an intruder on an edge is captured by placing searchers at both end vertices of the edge simultaneously or by sliding a searcher along the edge. For an overview see [10].

Kirousis and Papadimitriou [50] proved that the *node-search number* of a graph is equal to its interval thickness. Relations between node-searching, pebbling, and vertex separation have been studied in [51], and edge-searching and vertex separation have been studied in [32]. It was shown that the node-search number of a graph is equal to its path-width plus one [62, 11].

1.1.5 Network Reliability

A *k-tree* is a graph obtained from a complete graph on k vertices by recursively adding a new vertex which is adjacent to all vertices of an existing complete subgraph on k vertices. A partial k-tree is a subgraph of a k-tree. Our interest in the family of partial k-trees is motivated by some practical questions about reliability of communication networks in the presence of isolated failures [33, 34, 99, 64], concurrent broadcasting in a common medium network [27], reliability evaluation in complex systems [2]. For an overview see [3]. In many of these problems, the class of k-trees accurately captures the structure of the application. Furthermore, many problems which are NP-complete for general graphs have linear time algorithms for partial k-trees when k is fixed [5].

A k-tree Q is called a *k-path* if either the number of vertices of Q is at most $k+1$ or Q has exactly two vertices of degree k. A k-tree Q is called a *k-intercat* if there exists a k-path obtained from Q by deleting some vertices of degree k. These concepts of k-path and k-intercat are useful to analyze problems related to k-tree. For example, given two non-adjacent vertices u and v of a k-tree, it was shown that the union of complete subgraphs on k vertices separating u and v is a $(k-1)$-intercat and the vertices of minimal separators which disconnect u and v induce a k-path [68]. A k-cable is a collection of k vertex disjoint paths between vertices u and v in a k connected graph. The k-path plays a major role in the analysis of k-cables in k-trees [43]. Moreover, Neufeld and Colbourn [64] showed that 2-paths are most reliable 2-trees. That is, a 2-path has larger probability than any other 2-tree that the network is connected under the assumption that the probability that a communication link is up is the same for every edge.

In Chapter 2, we show that a simple graph is a partial k-intercat if and only if the path-width of the graph is at most k. Moreover we introduce the *proper-path-width* of a graph which is a slightly different from the path-width, and show that a simple graph is a partial k-path if and only if the proper-path-width of the graph is at most k.

1.1.6 Linguistics

In the *syntactic theory* [94], the structural description of sentences is given as directed graphs in which the vertices correspond to words and the directed edges correspond to dependencies. In the model shown in [52], a grammatical derivation starts with a *dependency graph* which encodes the major syntactic relations that can be obtained among words by labeled directed edges. A grammatical derivation of a sentence begins with a dependency graph and ends with linear sequence of vertices, corresponding to the temporal order in which the words of the sentence are uttered or written.

The essential feature of the model is a relatively small storage unit called the *shack*. The shack has the following feature: the shack is finite, unordered and random access memory; elementary memory cells of the shack are indistinguishable. It is assumed that the shack can hold at most six or seven vertices at any given moment and can not store two or more copies of the same element. In modeling the production of actual sentences, the shack hardly ever contains more than four items, and a research on human sentences production suggests that in a realistic model overloading the shack results in the loss of the entire memory content, rather than in the loss of the last item. This property is successfully captured in connectionist symbol manipulation models.

During the derivation the graph is moved from a permanent storage space, called the inner memory, to the outer memory via the shack. The order in which items are moved from the inner memory to the shack is arbitrary. But a vertex can be moved from the shack to the outer memory only if all of the vertices connected to it are also in the shack or already in the outer memory. This constraint captures the idea that the structural relations obtained between those parts of the sentence which are already spoken and those which are not must be kept in the short term memory of the speaker. Similarly, in order to understand the full content of the sentence, the listener has to remember all words having dependencies to the unspoken part.

The sequence $S = \{v_1, v_2, \ldots, v_n\}$ of the vertices of a graph G, when viewed as a sequence from the inner memory to the shack, together with the constraint defines a minimum demand on the shack capacity in the following way. In the i-th step, put vertex v_i from the inner memory to the shack; move all v_j $(j \leq i)$ with no neighbors v_k $(k > i)$ from the shack to the outer memory. The narrowness of G with respect to S, denoted by $na_S(G)$, is the maximum number of vertices in the shack during this process. This gives a lower bound for the capacity of the shack needed for a particular sequence. The *narrowness* of G, denoted by $na(G)$, is the maximum $na_S(G)$ over all sequences of the vertices of G. Kornai and Tuza [52] showed that the narrowness of a graph is equal to its path-width plus one.

1.2 Motivation

As shown in the previous section, the path-width and proper-path-width have been studied in various fields of computer science, and have a number of applications. The purpose of this thesis is to investigate the path-width and proper-path-width mainly from the computational point of view.

The problem of computing the path-width of a graph is NP-hard. This was independently shown for interval thickness [47], vertex separation number [56], and node-search number [51]. It was also shown that the path-width can be computed in linear time for trees [62, 91, 32]. However, no complexity results are known for proper-path-width. In this thesis, we prove that the problem of computing the proper-path-width of a graph is NP-hard for general graphs, but can be solved in linear time for trees. Many applications require a proper-path-decomposition to describe an explicit algorithm for the graphs with bounded proper-path-width. Similar to the path-width, we show that a proper-path-decomposition of a tree can be obtained in linear time.

Robertson and Seymour [86, 83] proved that the problem of testing membership for any minor-closed family of graphs can be solved in polynomial time provided that we know all the minimal forbidden minors for the family. Path-width plays an important role in graph minor theory and the family of graphs with path-width at most k is minor-closed. However the minimal forbidden minors are known only for the family of graphs with path-width at most one or two [36, 49]. Similarly the family of graphs with proper-path-width at most k is minor-closed, but no minimal forbidden minors are known for the family of graphs with bounded proper-path-width. In this thesis, we list the minimal acyclic forbidden minors for the family of graphs with bounded path-width or proper-path-width. Moreover we list all 36 minimal forbidden minors for the family of graphs with proper-path-width at most two. This give us the first explicit membership test algorithm for the family of graphs with proper-path-width at most two using a minor test algorithm.

Mixed-searching is a natural generalization of edge-searching and node-searching which have been extensively studied so far, as mentioned in the previous section. In this thesis, we prove that the mixed-search number of a simple graph is equal to the proper-path-width of the graph. This also shows that the problem of computing mixed-search number is NP-hard for general graphs but can be solved in linear time for trees. The optimal mixed-search strategy for a tree is obtained from a proper-path-decomposition of the tree.

Given a family $\mathcal{F}$ of graphs, a graph G is said to be *universal* for $\mathcal{F}$ if G contains every graph in $\mathcal{F}$ as a subgraph. A *minimum universal graph* for $\mathcal{F}$ is a universal graph for $\mathcal{F}$ with the minimum number of edges. We denote the number of edges in a minimum universal graph for $\mathcal{F}$ by $f(\mathcal{F})$. Determining $f(\mathcal{F})$ has been known to have applications to the circuit design [97], data representation [24, 89], and parallel computing [8]. Because, in the context of parallel computing, for example, the edges in the universal graphs correspond to the communication links of the parallel computing machine that simulates many parallel algorithms without communication overhead. For general families of (unbounded-degree) graphs, $f(\mathcal{F})$ was investigated for the family of all planar graphs [7], trees [20], and 2-paths [96]. Although the order of $f(\mathcal{F})$ was known for trees and 2-paths (both of which are subsets of planar graphs), there exists a gap between upper and lower bounds for the family of all planar graphs. In this thesis, we show that the number of edges in a minimum universal graph for the family of all graphs on n vertices with path-width at most k is $\Theta(kn\log(n/k))$ $(k \geq 1, n \geq 12k)$. This is a generalization of the results in [96], and it follows that the number of edges in a minimum universal graph for the family of all planar graphs on n vertices with bounded path-width is $\Theta(n \log n)$.

1.3 Thesis Outline

We describe the outline of this thesis.

In Chapter 2, we define the path-width and proper-path-width of graphs, and discuss properties of them. First, we show that the path-width and proper-path-width of a graph may differ by at most one. To simplify the argument that follows, we introduce various forms of decomposition each of which is equivalent to path-decomposition or proper-path-decomposition. We show basic properties of (proper-)path-width of graphs such as upper bounds or lower bounds. The complexity of computing the path-width or proper-path-width of graphs is discussed.

In Chapter 3, we attempt to characterize the family of graphs with bounded path-width or proper-path-width by means of the set of minimal forbidden minors. We show that some compositions and equivalence relations on graphs by which we can obtain minimal forbidden minors for the family of graphs with bounded (proper-)path-width k from those with bounded (proper-)path-width k' ($k \geq k'$). We list all the acyclic forbidden minors for bounded (proper-)path-width and the forbidden minors for the family of graphs with proper-path-width at most two. That is, we characterize the family of trees for bounded (proper-)path-width and the family of graphs with proper-path-width at most two.

In Chapter 4, relations between path-width and search game are discussed. We introduce a new version of the search game, called mixed-searching, which is a natural generalization of edge-searching and node-searching. Relations between mixed-searching and two preceding searchings, and a monotonicity in mixed-searching are discussed. It is known that the node-search number of a graph G is equal to the path-width of G plus one. We show that the mixed-search number of a simple graph G is equal to the proper-path-width of G.

In Chapter 5, we give a universal graph for the family of graphs with bounded path-width, and show that the number of edges in a minimum universal graph for the family of n vertex graphs with bounded path-width is $\Theta(n \log n)$. We also give an embedding algorithm of a graph with a path-decomposition on the universal graph.

In the last chapter, we conclude this thesis, and describe further future researches.

Chapter 2

Path-Width

2.1 Introduction of Path-Width and Proper-Path-Width

2.1.1 Definitions

The path-width of a graph was introduced by Robertson and Seymour in the first paper of the Graph Minors series [71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86]. Graphs we consider are finite and undirected, but may have loops and multiple edges unless otherwise specified. A graph with no edges is said to be *empty*. A graph is *simple* if it has no loops and multiple edges. We denote the vertex set and edge set of a graph G by $V(G)$ and $E(G)$, respectively. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a sequence of subsets of $V(G)$. The *width* of $\mathcal{X}$ is $\max_{1 \leq i \leq r} |X_i| - 1$.

Definition 2.1 *A sequence $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of subsets of $V(G)$ is called a* path-decomposition *of G if the following conditions are satisfied:*

(i) *For any distinct i and j $(1 \leq i, j \leq r)$, $X_i \not\subseteq X_j$;*

(ii) $\bigcup_{i=1}^{r} X_i = V(G)$;

(iii) *For any edge $(u, v) \in E(G)$, there exists an i such that $u, v \in X_i$;*

(iv) *For any a, b, and c $(1 \leq a \leq b \leq c \leq r)$, $X_a \cap X_c \subseteq X_b$.*

The path-width *of G is the minimum width over all path-decompositions of G, and denoted by $pw(G)$.*

The definition of proper-path-width is slightly different from that of path-width.

Definition 2.2 *A sequence* $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ *of subsets of* $V(G)$ *is called a* proper-path-decomposition *of* G *if the following conditions are satisfied:*

(i) *For any distinct* i *and* j $(1 \leq i, j \leq r)$, $X_i \not\subseteq X_j$;

(ii) $\bigcup_{i=1}^{r} X_i = V(G)$;

(iii) *For any edge* $(u, v) \in E(G)$, *there exists an* i *such that* $u, v \in X_i$;

(iv) *For any* a, b, *and* c $(1 \leq a \leq b \leq c \leq r)$, $X_a \cap X_c \subseteq X_b$.

(v) *For any* a, b, *and* c $(1 \leq a < b < c \leq r)$, $|X_a \cap X_c| \leq |X_b| - 2$ *if* $|X_b| \geq 2$.

The proper-path-width *of* G, *denoted by* $ppw(G)$, *is the minimum width over all proper-path-decompositions of* G.

As an example, path-decomposition and proper-path-decomposition of the graph shown in Fig. 2.1 are shown in Figs. 2.2 and 2.3, respectively. The width of the path-decomposition of G shown in Fig. 2.2 is two, and the width of the proper-path-decomposition of G shown in Fig. 2.3 is three. It is easy to see that the width of the path-decomposition is minimum over all path-decompositions of G, and $pw(G) = 2$. Similarly, $ppw(G) = 3$. Notice that both the path-width and the proper-path-width of an empty graph are zero, and both the path-width and the proper-path-width of a graph with single edge are one.

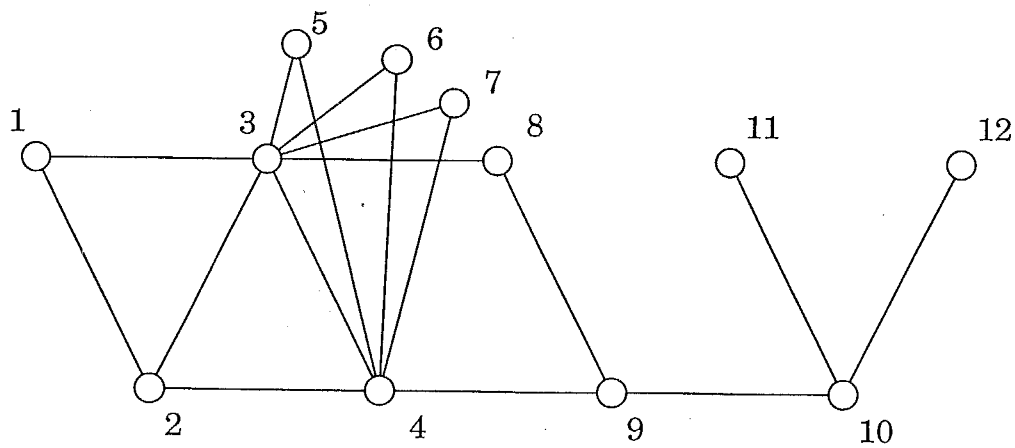


Figure 2.1: A graph G.

$$\left(\overbrace{\underbrace{\begin{matrix}1\\2\\3\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}2\\3\\4\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\5\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\6\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\7\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\8\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}4\\8\\9\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}9\\10\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}10\\11\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}10\\12\end{matrix}}} \right)$$

Figure 2.2: A path-decomposition of the graph shown in Fig. 2.1.

$$\left(\overbrace{\underbrace{\begin{matrix}1\\2\\3\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}2\\3\\4\\5\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\5\\6\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\6\\7\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\8\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}4\\8\\9\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}9\\10\\11\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}10\\12\end{matrix}}} \right)$$

Figure 2.3: A proper-path-decomposition of the graph shown in Fig. 2.1.

The sequence obtained by concatenating sequences $\mathcal{X}_i$ $(1 \leq i \leq r)$ is denoted by $(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_r)$. For a sequence $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of subsets of $V(G)$ and a set $S \subseteq V(G)$, we denote the sequence $(X_1 \cup S, X_2 \cup S, \ldots, X_r \cup S)$ by $(\mathcal{X} \cup S)$, and $(X_1 \cap S, X_2 \cap S, \ldots, X_r \cap S)$ by $(\mathcal{X} \cap S)$ for simplicity. $G \backslash S$ denotes the graph obtained from G by deleting the vertices in S.

2.1.2 Consecutiveness in Path-Decomposition

We first show that vertices appear consecutive members in path-decomposition, and give brief observations.

Lemma 2.1 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a sequence of subsets of $V(G)$. Then the following are equivalent:*

1. *For any a, b, and c $(1 \leq a \leq b \leq c \leq r)$, $X_a \cap X_c \subseteq X_b$.*
2. *For any vertex $v \in V(G)$, if $v \in X_s$ and $v \in X_t$ $(1 \leq s \leq t \leq r)$, then $v \in X_i$ for any i $(s \leq i \leq t)$, that is, v appears in consecutive X_i's.*

Proof: Assume that $X_a \cap X_c \not\subseteq X_b$ for some distinct integers a, b, and c $(a < b < c)$. Then there exists a vertex $v \in (X_a \cap X_c) - X_b$, and v does not appear in consecutive X_i's. That is, $v \in X_a$, $v \in X_c$, and $v \notin X_b$. Conversely, assume that a vertex $v \in V(G)$ does not appear in consecutive X_i's. Then there exist distinct integers a, b, and c ($a < b < c$) such that $v \in X_a$, $v \in X_c$ and $v \notin X_b$, and $X_a \cap X_c \not\subseteq X_b$. □

Lemma 2.2 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a sequence of subsets of $V(G)$ such that $X_a \cap X_c \subseteq X_b$ for any a, b, and c ($1 \leq a \leq b \leq c \leq r$). Then the following are equivalent:*

1. *For any distinct i and j ($1 \leq i, j \leq r$), $X_i \not\subseteq X_j$;*
2. *For any i ($1 \leq i < r$), $X_i \not\subseteq X_{i+1}$ and $X_i \not\supseteq X_{i+1}$.*

Proof: It is trivial that if $X_i \not\subseteq X_j$ for any distinct i and j ($1 \leq i, j \leq r$), then $X_i \not\subseteq X_{i+1}$ and $X_i \not\supseteq X_{i+1}$ for any i ($1 \leq i < r$).

Assume that $X_i \not\subseteq X_{i+1}$ and $X_i \not\supseteq X_{i+1}$ for any i ($1 \leq i < r$). Suppose that $X_a \subseteq X_b$ for some distinct integers a and b ($1 \leq a, b \leq r$). If $a < b$ then $X_a \cap X_b \subseteq X_{a+1}$ by the assumption of the lemma. Thus $X_a = X_a \cap X_b \subseteq X_{a+1}$, and contradicting the assumption that $X_a \not\subseteq X_{a+1}$. Similarly, if $a > b$ then $X_{a-1} \supseteq X_a$, and contradicting the assumption that $X_{a-1} \not\supseteq X_a$. □

Lemma 2.3 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a sequence of subsets of $V(G)$ such that $X_a \cap X_c \subseteq X_b$ for any a, b, and c ($1 \leq a \leq b \leq c \leq r$). Then the following are equivalent:*

1. *For any a, b, and c ($1 \leq a < b < c \leq r$), $|X_a \cap X_c| \leq |X_b| - 2$ if $|X_b| \geq 2$;*
2. *For any i such that $|X_i| \geq 2$ ($1 < i < r$), $|X_{i-1} \cap X_{i+1}| \leq |X_i| - 2$.*

Proof: It is trivial that if $|X_a \cap X_c| \leq |X_b| - 2$ for any a, b, and c such that $|X_b| \geq 2$ ($1 \leq a < b < c \leq r$), then $|X_{i-1} \cap X_{i+1}| \leq |X_i| - 2$ for any i such that $|X_i| \geq 2$ ($1 < i < r$).

Assume that $|X_{i-1} \cap X_{i+1}| \leq |X_i| - 2$ for any i such that $|X_i| \geq 2$ ($1 < i < r$). Suppose that $|X_a \cap X_c| > |X_b| - 2$ for some distinct a, b, and c such that $|X_b| \geq 2$ ($1 \leq a < b < c \leq r$). Since $X_a \cap X_c \subseteq X_{b-1} \cap X_{b+1}$ by the assumption of the lemma, $|X_{b-1} \cap X_{b+1}| \geq |X_a \cap X_c| > |X_b| - 2$, and contradicting the assumption that $|X_{b-1} \cap X_{b+1}| \leq |X_b| - 2$. □

2.1.3 Path-Width and Proper-Path-Width

In this section, we show that the path-width and proper-path-width of a graph may differ by at most one.

Theorem 2.1 *For any graph G, $pw(G) \leq ppw(G) \leq pw(G) + 1$.*

Proof: The first inequality follows from the definition.

To prove the second inequality, we show that a proper-path-decomposition of G with width at most $k+1$ can be obtained from a k-path-decomposition of G. Let $(X_1, X_2, \ldots, X_r)$ be a k-path-decomposition of G. For any i $(1 \leq i \leq r)$, let $X'_i = X_{i-1} \cup X_i$ if $|X_{i-1} \cap X_{i+1}| = |X_i| - 1$ $(1 < i < r)$, let $X'_i = X_i$ otherwise. It is trivial that the sequence $\mathcal{X}' = (X'_1, X'_2, \ldots, X'_r)$ satisfies conditions (ii) and (iii) in Definition 2.2. Since each vertex appears in consecutive X'_i's, $\mathcal{X}'$ also satisfies condition (iv) in Definition 2.2 by Lemma 2.1. Let $\mathcal{X}$ be the sequence obtained from $\mathcal{X}'$ by deleting every X'_i such that $X'_i \subseteq X'_{i+1}$. We show that $\mathcal{X}$ is a proper-path-decomposition of G with width at most $k+1$. It is easy to see that $\mathcal{X}$ also satisfies conditions (ii), (iii), and (iv) in Definition 2.2.

To verify the condition (i), assume that $X'_i \subseteq X'_j$ for some distinct i and j. Notice that $X_i \subseteq X'_i = X'_i \cap X'_j$. Since $X_i \not\subseteq X_j$, $X'_j = X_{j-1} \cup X_j$. If $i > j$ then $X_{j-1} \cap X_{i-1} \subseteq X_j$ and $X_{j-1} \cap X_i \subseteq X_j$ by the condition (iv) in Definition 2.1. Thus $X_{j-1} \cap X'_i \subseteq X_{j-1} \cap (X_{i-1} \cup X_i) \subseteq X_j$. Since $X_j \cap X'_i \subseteq X_j$, $X'_j \cap X'_i = (X_{j-1} \cup X_j) \cap X'_i \subseteq X_j$. Hence, we have $X_i \subseteq X'_j \cap X'_i \subseteq X_j$. However, this is contradicting to $X_i \not\subseteq X_j$. Similarly, if $i < j$ then $X_i \subseteq X'_i \subseteq X_{j-1}$, and we have $i = j-1$. Thus, $X'_i \subseteq X'_{i+1}$. However $\mathcal{X}$ does not contain such X'_i. Hence $\mathcal{X}$ satisfies the condition (i) in Definition 2.2.

To verify the condition (v), first, assume that $X'_i = X_{i-1} \cup X_i$ $(1 < i < r)$. By the condition (i) in Definition 2.1, $|X'_i| = |X_{i-1} \cup X_i| \geq |X_i| + 1$. By the condition (iv) in Definition 2.1, $X'_a \cap X'_c \subseteq X_a \cap X_{c-1} \subseteq X_i$ for any a and c $(1 \leq a < i < c \leq r)$. Moreover, $X_a \cap X_{c-1} \neq X_i$, for otherwise $X_a \supseteq X_i$. Thus $|X'_a \cap X'_c| \leq |X_i| - 1$, and $|X'_a \cap X'_c| \leq |X'_i| - 2$ for any a and c $(1 \leq a < i < c \leq r)$. Next, assume that $X'_i = X_i$ $(1 < i < r)$. Notice that $|X_{i-1} \cap X_{i+1}| \leq |X_i| - 2$ by the definition of X'_i. If $X'_{i+1} = X_i \cup X_{i+1}$ then X'_i is not contained in $\mathcal{X}$. Thus we assume that $X'_{i+1} = X_{i+1}$. Since $X'_a \cap X'_c \subseteq X'_{i-1} \cap X'_{i+1} \subseteq X_{i-1} \cap X_{i+1}$, we have $|X'_a \cap X'_c| \leq |X_i| - 2 = |X'_i| - 2$ for any a and c $(1 \leq a < i < c \leq r)$. Thus, $\mathcal{X}$ satisfies the condition (v) in Definition 2.2.

Finally, we show that the width of $\mathcal{X}$ is at most $k+1$. If $X'_i = X_{i-1} \cup X_i$ then $|X_i| - |X_{i-1} \cap X_{i+1}| = 1$ by the definition of X'_i. By the condition (iv) in Definition 2.1, $X_{i-1} \cap X_{i+1} \subseteq X_{i-1} \cap X_i \subseteq X_i$. Moreover, $X_{i-1} \cap X_{i+1} = X_{i-1} \cap X_i$, for otherwise $X_{i-1} \cap X_i = X_i$ and $X_{i-1} \supseteq X_i$. Thus, $X_i - X_{i-1} = X_i - (X_{i-1} \cap X_i) = X_i - (X_{i-1} \cap X_{i+1})$. Hence, $|X_i - X_{i-1}| = 1$ and $|X'_i| = |X_{i-1} \cup X_i| = |X_{i-1}| + 1 \leq k+2$. If $X'_i = X_i$ then $|X'_i| = |X_i| \leq k+1$. Hence, the width of $\mathcal{X}$ is at most $k+1$. □

The difference between the path-width and the proper-path-width of the graph shown in Fig. 2.1 is one. The path-width and the proper-path-width of the graph shown in

Fig. 2.4 are two.

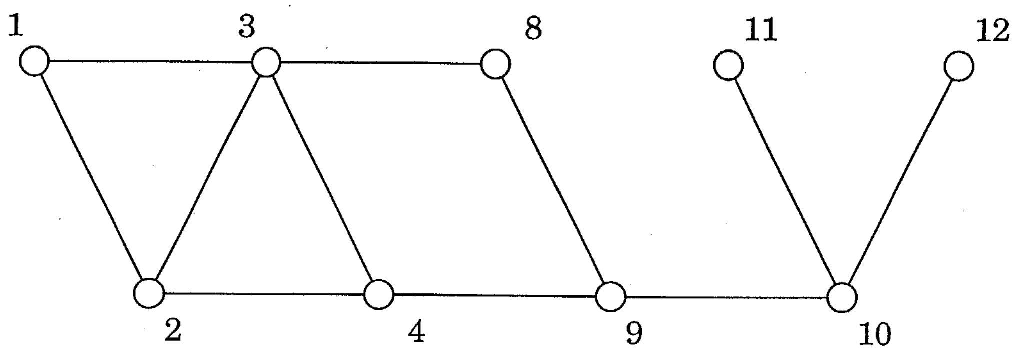


Figure 2.4: A graph G'.

2.2 Path-Width

2.2.1 Path-Decomposition

A path-decomposition with width k is called a k-path-decomposition. A k-path-decomposition $(X_1, X_2, \ldots, X_r)$ is said to be *full* if $|X_i| = k+1$ $(1 \leq i \leq r)$ and $|X_j \cap X_{j+1}| = k$ $(1 \leq j \leq r-1)$.

Lemma 2.4 *For any graph G with path-width k, there exists a full k-path-decomposition of G.*

Proof: It is trivial when $k = 0$. Thus we assume that $k \geq 1$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a k-path-decomposition of G such that $\sum_{i=1}^{r}(|X_i| - k)$ is maximum. We shall show that $\mathcal{X}$ is a full k-path-decomposition of G. If $r = 1$ then $\mathcal{X}$ is trivially a full k-path-decomposition of G. Thus we assume that $r \geq 2$.

Suppose that $|X_i| \leq k$ for some i $(2 \leq i \leq r)$. Let $v \in X_{i-1} - X_i$. The sequence $\mathcal{X}' = (X_1, X_2, \ldots, X_{i-1}, X_i \cup \{v\}, X_{i+1}, \ldots, X_r)$ satisfies conditions (ii), (iii), and (iv) in Definition 2.1. To verify the condition (i), assume that $X_j \subseteq X_i \cup \{v\}$ for some $j(\neq i)$. If $j > i$ then $X_j \subseteq X_i$ since $v \notin X_j$, contradicting the condition (i) in Definition 2.1. Thus $j = i-1$ since $X_j = X_j \cap (X_i \cup \{v\}) \subseteq X_{i-1}$. Therefore, $(X_1, X_2, \ldots, X_{i-2}, X_i \cup \{v\}, X_{i+1}, \ldots, X_r)$ is a k-path-decomposition of G. But this is contradicting the choice of $\mathcal{X}$ since $|X_{i-1}| \leq k$, for otherwise $X_{i-1} = X_i \cup \{v\}$ and $X_{i-1} \supseteq X_i$. Thus $\mathcal{X}'$ satisfies the condition (i) in Definition 2.1, and $\mathcal{X}'$ is a k-path-decomposition of G. But again this is contradicting the choice of $\mathcal{X}$. Thus $|X_i| = k+1$ for any i $(2 \leq i \leq r)$. Since $(X_r, \ldots, X_1)$ is also a path-decomposition of G, $|X_i| = k+1$ for any i $(1 \leq i \leq r)$.

Suppose next that $|X_i \cap X_{i+1}| \leq k-1$ for some i $(1 \leq i \leq r-1)$. Let $v \in X_{i+1} - X_i$ and $w \in X_i - X_{i+1}$. The sequence $\mathcal{X}' = (X_1, \ldots, X_i, (X_i \cup \{v\}) - \{w\}, X_{i+1}, \ldots, X_r)$ satisfies conditions (ii), (iii), and (iv) in Definition 2.1. Assume that $X_j \subseteq (X_i \cup \{v\}) - \{w\}$ or $(X_i \cup \{v\}) - \{w\} \subseteq X_j$ for some j $(1 \leq j \leq r)$. Since $|(X_i \cup \{v\}) - \{w\}| = |X_j| = k+1$, $X_j = (X_i \cup \{v\}) - \{w\}$. Then $j = i$ or $j = i+1$ since $X_j = X_j \cap ((X_i \cup \{v\}) - \{w\}) \subseteq X_i$ if $j \leq i$, $X_j = ((X_i \cup \{v\}) - \{w\}) \cap X_j \subseteq X_{i+1}$ otherwise. But this is contradicting the assumption that $|X_i \cap X_{i+1}| \leq k-1$. Thus $\mathcal{X}'$ satisfies the condition (i) in Definition 2.1, and $\mathcal{X}'$ is a k-path-decomposition of G. But this is contradicting the choice of $\mathcal{X}$ since $|(X_i \cup \{v\}) - \{w\}| = k+1$. Thus $|X_i \cap X_{i+1}| = k$ for any i $(1 \leq i \leq r-1)$.

Therefore $\mathcal{X}$ is a full k-path-decomposition of G. □

As an example, full path-decomposition of the graph shown in Fig. 2.1 is shown in Fig. 2.5.

$$\left(\overbrace{\underbrace{\begin{matrix}1\\2\\3\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}2\\3\\4\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\5\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\6\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\7\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\8\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}4\\8\\9\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}8\\9\\10\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}9\\10\\11\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}9\\10\\12\end{matrix}}} \right)$$

Figure 2.5: A full path-decomposition of the graph shown in Fig. 2.1.

Although the condition (i) in Definition 2.1 makes a path-decomposition easy to handle, it is not essential to characterize the path-width of graph.

Lemma 2.5 *If there exists a sequence* $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ *of subsets of* $V(G)$ *with width* k *satisfying the conditions (ii), (iii), and (iv) in Definition 2.1, then* $pw(G) \leq k$.

Proof: Let $\mathcal{X}'$ be a sequence obtained from $\mathcal{X}$ by the following procedure:

1. Let $i = 1$, and $j = 1$;

2. While $X_i \subseteq X_{i+1}$ and $i < r$, let $i = i+1$.

3. Define $X'_j = X_i$, and let $i = i+1$ and $j = j+1$;

4. While $X_{i-1} \supseteq X_i$ and $i \leq r$, let $i = i + 1$.

5. If $i \leq r$ then return to 2.

6. Let $r' = j - 1$ and define $\mathcal{X}' = (X'_1, X'_2, \ldots, X'_{r'})$.

We will show that $\mathcal{X}'$ is a path-decomposition of G with width k.

Since $\mathcal{X}'$ is obtained from $\mathcal{X}$ by deleting some X_i's, the width of $\mathcal{X}'$ is k and each vertex appears in consecutive X'_i's. By Lemma 2.1, $\mathcal{X}'$ satisfies the condition (iv) in Definition 2.1. To verify the condition (ii) in Definition 2.1, assume that a vertex $v \in V(G)$ is not contained in any X'_i $(1 \leq i \leq r')$. By the condition (ii) in Definition 2.1, there exists an integer i such that $v \in X_i$. Let a and b be such minimum and maximum integers, respectively. Since $v \in X_a - X_{a-1}$, $X_{a-1} \not\supseteq X_a$. Thus $X_a \subseteq X_{a+1}$ since X_a is not contained in $\mathcal{X}'$. Similarly, $X_i \subseteq X_{i+1}$ for any i $(a \leq i \leq b)$ since X_i is not contained in $\mathcal{X}'$. Thus we have $X_b \subseteq X_{b+1}$. However this is contradicting the assumption that $v \notin X_{b+1}$. Hence $\mathcal{X}'$ satisfies the condition (ii) in Definition 2.1. Similarly, we can verify the condition (iii) in Definition 2.1. To verify the condition (i) in Definition 2.1, assume that $X'_i \subseteq X'_{i+1}$ for some i $(1 \leq i < r')$. Let $X_s = X'_i$ and $X_t = X'_{i+1}$ $(1 \leq s < t \leq r)$. Then $X_s = X_s \cap X_t \subseteq X_{s+1}$ by the condition (iv) in Definition 2.1, and contradicting that $X_s \not\subseteq X_{s+1}$. Thus $X'_i \not\subseteq X'_{i+1}$ for any i $(1 \leq i < r')$. Similarly, we can show that $X'_i \not\supseteq X'_{i+1}$ for any i $(1 \leq i < r')$. By Lemma 2.2, $\mathcal{X}'$ satisfies the condition (i) in Definition 2.1.

Thus $\mathcal{X}'$ is a k-path-decomposition of G, and $pw(G) \leq k$. □

Lemma 2.6 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a path-decomposition of G, and H be a subgraph of G with path-width k. Let $X'_i = X_i \cap V(H)$. Then there exists an integer i $(1 \leq i \leq r)$ such that $|X'_i| \geq k + 1$.*

Proof: For otherwise, $\mathcal{X} \cap V(H)$ is a sequence of subsets of $V(H)$ with width $k - 1$ satisfying the conditions (ii), (iii), and (iv) in Definition 2.1, and $pw(H) \leq k - 1$ by Lemma 2.5. Contradicting to the assumption that $pw(H) = k$. □

2.2.2 Upper and Lower Bounds for Path-Width

In this section, we show upper and lower bounds for the path-width of graphs and present the characterization for trees with path-width at least $k + 1$.

Lemma 2.7 *Let H be a connected subgraph of a graph G, and $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a path-decomposition of G. Let $X'_i = X_i \cap V(H)$ $(1 \leq i \leq r)$. If $X'_a \neq \emptyset$ and $X'_b \neq \emptyset$ $(1 \leq a \leq b \leq r)$, then $X'_i \neq \emptyset$ for any i $(a \leq i \leq b)$. That is, the vertices of H appear in consecutive X_i's. Moreover, if $|X'_i| = 1$ for some i $(a < i < b)$, then $X'_i = X'_{i-1} \cap X'_{i+1}$.*

Proof: Suppose that $X'_c = \emptyset$ for some c $(a < c < b)$. Each vertex of H appears in consecutive X_i's by Lemma 2.1. Thus, if $P = \bigcup_{i=1}^{c-1} X'_i$ and $Q = \bigcup_{i=c+1}^{r} X'_i$, $P \cap Q = \emptyset$. Since $V(H)$ is partitioned into P and Q, and H is connected, there exist $u \in P$ and $v \in Q$ such that $(u, v) \in E(H)$. However, $\{u, v\} \not\subseteq X_i$ for any i $(1 \leq i \leq r)$, contradicting the condition (iii) in Definition 2.1.

Suppose that $|X'_c| = 1$ for some c $(a < c < b)$. Let $v \in X'_c$. If $v \notin X'_{c-1}$ then $V(H)$ is partitioned into $\bigcup_{i=1}^{c-1} X'_i$ and $\bigcup_{i=c}^{r} X'_i$. This is contradicting the condition (iv) in Definition 2.1. Thus $v \in X'_{c-1}$. Similarly, $v \in X'_{c+1}$. □

The following lemma gives a lower bound for the width of a path-decomposition.

Lemma 2.8 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a path-decomposition of a graph G, and G_1, $G_2, \ldots, G_s$, and H be disjoint connected subgraphs of G. Let $X'_i = X_i \cap V(H)$ $(1 \leq i \leq r)$. If there exists an integer b such that $|X'_b| \geq k + 1$ and there exist integers a_i and c_i $(a_i < b < c_i)$ such that $|X_{a_i} \cap V(G_i)| \neq \emptyset$ and $|X_{b_i} \cap V(G_i)| \neq \emptyset$ for any i $(1 \leq i \leq s)$, then the width of $\mathcal{X}$ is at least $k + s$.*

Proof: By Lemma 2.7, there exists at least one vertex in $X_b \cap V(G_i)$ for any i $(1 \leq i \leq s)$. Thus $|X_b| \geq |X'_b| + |X_b \cap V(G_1)| + \cdots + |X_b \cap V(G_s)| \geq k + s + 1$, and the width of $\mathcal{X}$ is at least $k + s$. □

The following lemma shows a lower bound for the path-width of a graph.

Lemma 2.9 *Let G be a connected graph and k be a positive integer. If G has a vertex v such that $G \backslash \{v\}$ has at least three connected components with path-width k or more, then $pw(G) \geq k + 1$.*

Proof: We may assume that the path-widths of connected components of $G \backslash \{v\}$ are at most k, for otherwise trivially $pw(G) \geq k + 1$. Let H_1, H_2, and H_3 be connected components of $G \backslash \{v\}$ with path-width k, and $v_1 \in V(H_1)$, $v_2 \in V(H_2)$, and $v_3 \in V(H_3)$ be vertices adjacent to v in G.

Suppose contrary that $pw(G) \leq k$ and there exists a path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of G with width $\leq k$. There exists some i_j such that $|X_{i_j} \cap V(H_j)| \geq k+1$

for $j = 1, 2$, and 3 by Lemma 2.6. Without loss of generality we assume that $i_1 < i_2 < i_3$. It is trivial that $G \backslash V(H_2)$ is a connected subgraph of G. Since $X_{i_1} \cap V(G \backslash V(H_2)) \neq \emptyset$ and $X_{i_3} \cap V(G \backslash V(H_2)) \neq \emptyset$, the width of $\mathcal{X}$ is at least $k+1$ by Lemma 2.8. This contradict the assumption that the width of $\mathcal{X}$ is at most k. Thus $pw(G) \geq k+1$. □

The following two lemmas show upper bounds for the path-width of a graph.

Lemma 2.10 (Robertson and Seymour[71]) *If every connected component of G has path-width $\leq k$, then $pw(G) \leq k$.*

Lemma 2.11 (Robertson and Seymour[71]) *If $S \subseteq V(G)$ and $pw(G \backslash S) \leq k$, then $pw(G) \leq k + |S|$.*

The following lemma shows an upper bound for the path-width of a tree.

Lemma 2.12 *Let T be a tree and k be a positive integer. Suppose that for any $v \in V(T)$, $T \backslash \{v\}$ has no connected component with path-width $k+1$ or more and at most two connected components with path-width k. Then $pw(T) \leq k$.*

Proof: Let T_0 be T, and let v_0 be a vertex such that $T_0 \backslash \{v_0\}$ has the maximum number of connected components with path-width k.

If $T_0 \backslash \{v_0\}$ has no connected component with path-width k, then $pw(T) \leq k$ by Lemmas 2.10 and 2.11.

If $T_0 \backslash \{v_0\}$ has two connected components with path-width k, let T_1 be one of these components and $v_1 \in V(T_1)$ be a vertex adjacent to v_0 in T_0. We recursively define T_i and $v_i \in V(T_i)$ $(1 < i \leq a)$ while $T_{i-1} \backslash \{v_{i-1}\}$ has a component with path-width k as follows: Let T_i be a connected component of $T_{i-1} \backslash \{v_{i-1}\}$ with path-width k and $v_i \in V(T_i)$ be a vertex adjacent to v_{i-1} in T_{i-1}. $T_a \backslash \{v_a\}$ has no connected component with path-width k. Let T_{a+1} be the other connected component of $T_0 \backslash \{v_0\}$ with path-width k, and $v_{a+1} \in V(T_{a+1})$ be a vertex adjacent to v_0 in T_0. Define recursively T_i and $v_i \in V(T_i)$ $(a+1 < i \leq b)$ as above. Notice that $T_i \backslash \{v_i\}$ $(1 \leq i \leq b)$ has at most one connected component with path-width k, for otherwise $T_0 \backslash \{v_i\}$ has three or more connected components with path-width k, contradicting the assumption of the lemma.

Let H'_i $(0 \leq i \leq b)$ be the union of components of $T_i \backslash \{v_i\}$ with path-width $\leq k-1$, and H_i $(0 \leq i \leq b)$ be the induced subgraph of T on $V(H'_i) \cup \{v_i\}$. By Lemma 2.10, $pw(H'_i) \leq k-1$ $(0 \leq i \leq b)$. By Lemma 2.11, $pw(H_i) \leq k$ $(0 \leq i \leq b)$. Let $\mathcal{X}'_{(i)}$ be a path-decomposition of H'_i with width $\leq k-1$. Then $\mathcal{X}_{(i)} = (\mathcal{X}'_{(i)} \cup \{v_i\})$ is a path-decomposition of H_i with width $\leq k$. We define sequences $\mathcal{L}$ and $\mathcal{R}$ as follows.

$$\begin{aligned}\mathcal{L} &= (\mathcal{X}_{(a)}, \{v_a, v_{a-1}\}, \mathcal{X}_{(a-1)}, \{v_{a-1}, v_{a-2}\}, \ldots, \mathcal{X}_{(2)}, \{v_2, v_1\}, \mathcal{X}_{(1)}, \{v_1, v_0\}).\\ \mathcal{R} &= (\{v_0, v_{a+1}\}, \mathcal{X}_{(a+1)}, \{v_{a+1}, v_{a+2}\}, \mathcal{X}_{(a+2)}, \ldots, \{v_{b-2}, v_{b-1}\}, \mathcal{X}_{(b-1)}, \{v_{b-1}, v_b\}, \mathcal{X}_{(b)}).\end{aligned}$$

It is easy to see that $(\mathcal{L}, \mathcal{X}_{(0)}, \mathcal{R})$ is a sequence of subsets of $V(T)$ with width at most k satisfying the conditions (ii), (iii), and (iv) in Definition 2.1, and $pw(T) \leq k$ by Lemma 2.5.

If $T_0 \backslash \{v_0\}$ has just one connected component with path-width k, the sequence $\mathcal{R}$ above is empty, and $(\mathcal{L}, \mathcal{X}_{(0)})$ is a sequence of subsets of $V(T)$ with width at most k satisfying the conditions (ii), (iii), and (iv) in Definition 2.1, and we also have $pw(T) \leq k$ by Lemma 2.5.

□

The following theorem shows necessary and sufficient conditions for trees with path-width at least $k+1$.

Theorem 2.2 *For any tree T and integer $k \geq 1$, $pw(T) \geq k+1$ if and only if T has a vertex v such that $T\backslash\{v\}$ has at least three connected components with path-width k or more.*

Proof: Suppose that $pw(T) \geq k+1$. Let T' be a minimal subgraph of T with $pw(T') \geq k+1$. Since T' is minimal, $T'\backslash\{w\}$ has no connected component with path-width $\geq k+1$ for any $w \in V(T')$. Thus there exists a vertex $v \in V(T')$ such that $T'\backslash\{v\}$ has at least three connected components with path-width k, for otherwise $pw(T') \leq k$ by Lemma 2.12. Hence $T\backslash\{v\}$ has at least three connected components with path-width $\geq k$. The converse follows from Lemma 2.9. □

Theorem 2.2 was independently obtained by Scheffler [91], by Kinnersley [48], by Kornai and Tuza [52], and by Ellis, Sudborough and Turner [32]. Similar results can be also found in the literature. Parsons [66] obtained a similar results on edge-search number, and Chung, Makedon, Sudborough, and Turner [26] obtained on cut-width.

2.3 Proper-Path-Width

2.3.1 Proper-Path-Decomposition

A proper-path-decomposition with width k is called a k-proper-path-decomposition. A k-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ is said to be *full* if $|X_i| = k+1$ $(1 \leq i \leq r)$ and $|X_j \cap X_{j+1}| = k$ $(1 \leq j \leq r-1)$.

Lemma 2.13 *Let k be a positive integer. If a graph G has a k-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ such that*

$$|X_{i-1} \cap X_{i+1}| \leq k-1 \qquad (2.1)$$

for any i $(1 < i < r)$, then G has a full k-proper-path-decomposition.

Proof: Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a k-path-decomposition of G satisfying inequality (2.1) such that $\sum_{i=1}^{r}(|X_i| - k)$ is maximum. We shall show that $\mathcal{X}$ is a full k-proper-path-decomposition of G. In the following, $X_j = \emptyset$ if $j \leq 0$ or $j > r$.

Assume that $|X_i| \leq k$ for some i $(2 \leq i \leq r)$. If $|X_{i-2} \cap X_i| = k-1$, let $v \in X_{i-1} - X_{i-2}$, otherwise let $v \in X_{i-1} - X_i$. In former case, if $v \in X_i$ then $v \in (X_{i-1} \cap X_i) - (X_{i-2} \cap X_i)$. Since $X_{i-2} \cap X_i \subseteq X_{i-1} \cap X_i$, $|X_{i-1} \cap X_i| \geq k$ and $X_{i-1} \supseteq X_i$, contradicting the condition (i) in Definition 2.2. Thus $v \notin X_i$. In either case, we have $v \notin X_i$ and $|X_{i-2} \cap (X_i \cup \{v\})| \leq k-1$. By Lemma 2.1, $v \notin X_{i+2}$, and so $|(X_i \cup \{v\}) \cap X_{i+2}| \leq k-1$. Thus, the sequence $\mathcal{X}' = (X_1, X_2, \ldots, X_{i-1}, X_i \cup \{v\}, X_{i+1}, \ldots, X_r)$ satisfies inequality (2.1) and conditions (ii), (iii), and (iv) in Definition 2.2. To verify the condition (i), assume that $X_j \subseteq X_i \cup \{v\}$ for some $j (\neq i)$. If $j > i$ then $X_j \subseteq X_i$ since $v \notin X_j$, contradicting the condition (i) in Definition 2.2. Thus $j = i-1$ since $X_j = X_j \cap (X_i \cup \{v\}) \subseteq X_{i-1}$. Therefore, $(X_1, X_2, \ldots, X_{i-2}, X_i \cup \{v\}, X_{i+1}, \ldots, X_r)$ is a k-path-decomposition of G satisfying inequality (2.1). But this is contradicting the choice of $\mathcal{X}$ since $|X_{i-1}| \leq k$, for otherwise $X_{i-1} \supseteq X_i$. Thus $\mathcal{X}'$ is a k-path-decomposition of G. But again this is contradicting the choice of $\mathcal{X}$. Thus $|X_i| = k+1$ for any i $(2 \leq i \leq r)$. Since $(X_r, \ldots, X_1)$ is also a path-decomposition of G, $|X_i| = k+1$ for any i $(1 \leq i \leq r)$.

Assume next that $|X_i \cap X_{i+1}| \leq k-1$ for some i $(1 \leq i \leq r-1)$. If $|X_{i-1} \cap X_{i+1}| = k-1$, let $v \in X_i - X_{i-1}$; otherwise let $v \in X_i - X_{i+1}$. In either case, we have $v \notin X_{i+1}$ and $|X_{i-1} \cap (X_{i+1} \cup \{v\})| \leq k-1$. If $|X_{i+1} \cap X_{i+2}| = k$, let $u \in (X_{i+1} \cap X_{i+2}) - X_i$. Note that $(X_{i+1} \cap X_{i+2}) - X_i \neq \emptyset$ since $|X_{i+1} \cap X_{i+2}| = k > k-1 \geq |X_i \cap X_{i+1}|$. If $|X_{i+1} \cap X_{i+2}| < k$, let $u \in X_{i+1} - X_i$. In either case, we have $|(X_{i+1} - \{u\}) \cap X_{i+2}| \leq k-1$. Since $v \notin \bigcup_{j=i+1}^{r} X_j$ and $u \notin \bigcup_{j=1}^{i} X_j$, the sequence $\mathcal{X}' = (X_1, \ldots, X_i, (X_{i+1} \cup \{v\}) - \{u\}, X_{i+1}, \ldots, X_r)$ satisfies inequality (2.1) and conditions (ii), (iii), and (iv) in Definition 2.2. To verify the condition (i), assume that $X_j \subseteq (X_i \cup \{v\}) - \{u\}$ or $(X_i \cup \{v\}) - \{u\} \subseteq X_j$ for some j $(1 \leq j \leq r)$. Since $|(X_i \cup \{v\}) - \{u\}| = |X_j| = k+1$, $X_j = (X_i \cup \{v\}) - \{u\}$. Then $j = i$ or $j = i+1$, since if $j \leq i$, $X_j = X_j \cap ((X_i \cup \{v\}) - \{u\}) \subseteq X_i$; otherwise, $X_j = ((X_i \cup \{v\}) - \{u\}) \cap X_j \subseteq X_{i+1}$. But this is contradicting the assumption that $|X_i \cap X_{i+1}| \leq k-1$. Thus $\mathcal{X}'$ satisfies the condition (i) in Definition 2.2, and $\mathcal{X}'$ is a

k-path-decomposition of G satisfying inequality (2.1). But this is contradicting the choice of $\mathcal{X}$ since $|(X_i \cup \{v\}) - \{u\}| = k+1$. Thus $|X_i \cap X_{i+1}| = k$ for any i $(1 \le i \le r-1)$.

Therefore $\mathcal{X}$ is a full k-path-decomposition of G satisfying inequality (2.1), and so a full k-proper-path-decomposition of G by Lemma 2.3 since $|X_{i-1} \cap X_{i+1}| \le k-1 = |X_i|-2$ $(1 < i < r)$.

□

Lemma 2.14 *For any graph G with $ppw(G) = k$, there exists a full k-proper-path-decomposition of G.*

Proof: It is trivial when $k = 0$. Thus we assume that $k \ge 1$. A k-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ of G is a k-path-decomposition satisfying inequality (2.1). Thus we obtain the lemma from Lemma 2.13.

□

As an example, full proper-path-decomposition of the graph shown in Fig. 2.1 is shown in Fig. 2.6.

$$\left(\overbrace{\underbrace{\begin{matrix}1\\2\\3\\4\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}2\\3\\4\\5\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\5\\6\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\6\\7\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}3\\4\\7\\8\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}4\\7\\8\\9\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}7\\8\\9\\10\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}8\\9\\10\\11\end{matrix}}}, \overbrace{\underbrace{\begin{matrix}9\\10\\11\\12\end{matrix}}} \right)$$

Figure 2.6: A full proper-path-decomposition of the graph shown in Fig. 2.1.

Similar to path-width, the condition (i) in Definition 2.2 is not essential to characterize the proper-path-width of graph.

Lemma 2.15 *Let k be a positive integer. If there exists a sequence $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of subsets of $V(G)$ with width k satisfying the conditions (ii), (iii), and (iv) in Definition 2.2, and $|X_a \cap X_b \cap X_c| \le k-1$ for any X_a, X_b, and X_c such that each one is not a subset of the others $(1 \le a < b < c \le r)$, then $ppw(G) \le k$.*

Proof: Let $\mathcal{X}' = (X'_1, X'_2, \ldots, X'_{r'})$ be a sequence obtained from $\mathcal{X}$ by the procedure mentioned in Lemma 2.5. By Lemma 2.5, $\mathcal{X}'$ is a k-path-decomposition of G. Moreover $|X'_{i-1} \cap X'_{i+1}| \le k-1$ for any i $(1 < i < r')$. Thus by Lemma 2.13, there exists a k-proper-path-decomposition of G, and $ppw(G) \le k$.

□

Lemma 2.16 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a proper-path-decomposition of G, and H be a subgraph of G with proper-path-width k ($k \geq 1$). Let $X_i' = X_i \cap V(H)$. Then there exists an integer i ($1 \leq i \leq r$) such that either (a) $|X_i'| \geq k+1$, or (b) $|X_i'| = k$ and $|X_{i-1}' \cap X_{i+1}'| \geq k-1$.*

Proof: For otherwise, $\mathcal{X} \cap V(H)$ is a sequence with width $k-1$ satisfying the condition of Lemma 2.15 if $k \geq 2$, contradicting to the assumption that $ppw(H) = k$. If $k = 1$ then there exists an integer i such that X_i' contains both ends of an edge of H, that is, $|X_i'| \geq 2$. □

2.3.2 Upper and Lower Bounds for Proper-Path-Width

In this section, we show upper and lower bounds for the proper-path-width of graphs and present the characterization for trees with proper-path-width at least $k+1$.

Lemma 2.17 *Let H be a connected subgraph of a graph G, and $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a proper-path-decomposition of G. Let $X_i' = X_i \cap V(H)$ ($1 \leq i \leq r$). If $X_a' \neq \emptyset$ and $X_b' \neq \emptyset$ ($1 \leq a \leq b \leq r$), then $X_i' \neq \emptyset$ for any i ($a \leq i \leq b$). That is, the vertices of H appear in consecutive X_i's. Moreover, if $|X_i'| = 1$ for some i ($a < i < b$), then $X_i' = X_{i-1}' \cap X_{i+1}'$.*

Proof: Suppose that $X_c' = \emptyset$ for some c ($a < c < b$). Each vertex of H appears in consecutive X_i's by Lemma 2.1. Thus, if $P = \bigcup_{i=1}^{c-1} X_i'$ and $Q = \bigcup_{i=c+1}^{r} X_i'$, $P \cap Q = \emptyset$. Since $V(H)$ is partitioned into P and Q, and H is connected, there exist $u \in P$ and $v \in Q$ such that $(u, v) \in E(H)$. However, $\{u, v\} \not\subseteq X_i$ for any i ($1 \leq i \leq r$), contradicting the condition (iii) in Definition 2.2.

Suppose that $|X_c'| = 1$ for some c ($a < c < b$). Let $v \in X_c'$. If $v \notin X_{c-1}'$ then $V(H)$ is partitioned into $\bigcup_{i=1}^{c-1} X_i'$ and $\bigcup_{i=c}^{r} X_i'$. This is contradicting the condition (iv) in Definition 2.2. Thus $v \in X_{c-1}'$. Similarly, $v \in X_{c+1}'$. □

The following lemma gives a lower bound for the width of a proper-path-decomposition.

Lemma 2.18 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a proper-path-decomposition of a graph G, and $G_1, G_2, \ldots, G_s$, and H be disjoint connected subgraphs of G. Let $X_i' = X_i \cap V(H)$ ($1 \leq i \leq r$). Assume that there exists an integer b such that either $|X_b'| \geq k+1$ or $|X_b'| = k$ and $|X_{b-1}' \cap X_{b+1}'| \geq k-1$. If there exist integers a_i and c_i ($a_i < b < c_i$) such that $|X_{a_i} \cap V(G_i)| \neq \emptyset$ and $|X_{c_i} \cap V(G_i)| \neq \emptyset$ for any i ($1 \leq i \leq s$), then the width of $\mathcal{X}$ is at least $k+s$.*

Proof: By Lemma 2.17, there exists at least one vertex in $X_b \cap V(G_i)$ $(1 \leq i \leq s)$. If either $|X'_b| \geq k+1$ or $|X_b \cap V(G_i)| \geq 2$ for some i, then $|X_b| \geq |X'_b| + |X_b \cap V(G_1)| + \cdots + |X_b \cap V(G_s)| \geq k+s+1$, and the width of $\mathcal{X}$ is at least $k+s$. Thus we assume that $|X'_b| = k$ and $|X_b \cap V(G_i)| = 1$ for any i. Let v_i be a vertex in $X_b \cap V(G_i)$ $(1 \leq i \leq s)$. By Lemma 2.17, $v_i \in X_{b-1} \cap X_{b+1}$. Since $X_{b-1} \cap X_{b+1} \supseteq (X'_{b-1} \cap X'_{b+1}) \cup \{v_1, v_2, \ldots, v_s\}$, we have that $|X_b| \geq |X_{b-1} \cap X_{b+1}| + 2 \geq |(X'_{b-1} \cap X'_{b+1})| + |\{v_1, v_2, \ldots, v_s\}| + 2 = k+s+1$ by Definition 2.2(v). Thus the width of $\mathcal{X}$ is at least $k+s$. □

The following lemma shows a lower bound for the proper-path-width of a graph.

Lemma 2.19 *Let G be a connected graph and k be a positive integer. If G has a vertex v such that $G \backslash \{v\}$ has at least three connected components with proper-path-width k or more, then $ppw(G) \geq k+1$.*

Proof: We may assume that the proper-path-widths of connected components of $G\backslash\{v\}$ are at most k, for otherwise trivially $ppw(G) \geq k+1$. Let H_1, H_2, and H_3 be connected components of $G\backslash\{v\}$ with proper-path-width k, and $v_1 \in V(H_1)$, $v_2 \in V(H_2)$, and $v_3 \in V(H_3)$ be vertices adjacent to v in G.

Suppose contrary that $ppw(G) \leq k$ and G has a proper-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ with width $\leq k$. There exists some i_j such that either $|X_{i_j} \cap V(H_j)| \geq k+1$ or $|X'_{i_j}| = k$ and $|X'_{i_j-1} \cap X'_{i_j+1}| \geq k-1$ for $j = 1, 2$, and 3 by Lemma 2.16. Without loss of generality we assume that $i_1 < i_2 < i_3$. It is trivial that $G\backslash V(H_2)$ is a connected subgraph of G. Since $X_{i_1} \cap V(G\backslash V(H_2)) \neq \emptyset$ and $X_{i_3} \cap V(G\backslash V(H_2)) \neq \emptyset$, the width of $\mathcal{X}$ is at least $k+1$ by Lemma 2.18. This contradict the assumption that the width of $\mathcal{X}$ is at most k. Thus $ppw(G) \geq k+1$. □

The following two lemmas show upper bounds for the proper-path-width of a graph.

Lemma 2.20 *If every connected component of G has proper-path-width $\leq k$, then $ppw(G) \leq k$.*

Proof: Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_s$ be proper-path-decompositions of each connected component of G with width at most k. Then the sequence $(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_s)$ is a proper-path-decomposition of G with width at most k. □

Lemma 2.21 *For any positive integer k, if $S \subseteq V(G)$ and $ppw(G\backslash S) \leq k$, then $ppw(G) \leq k + |S|$.*

Proof: Let $\mathcal{X}$ be a proper-path-decomposition of $G\backslash S$ with width at most k. Then $(\mathcal{X} \cup S)$ is a path-decomposition of G with width at most $k + |S|$ satisfying inequality (2.1). Thus G has a proper-path-decomposition with width at most $k + |S|$ by Lemma 2.13. □

The following lemma shows an upper bound for the proper-path-width of a tree.

Lemma 2.22 *Let T be a tree and k be an integer ($k \geq 2$). Suppose that for any $v \in V(T)$, $T\backslash\{v\}$ has no connected component with proper-path-width $k+1$ or more and at most two connected components with proper-path-width k. Then $ppw(T) \leq k$.*

Proof: Let T_0 be T, and let v_0 be a vertex such that $T_0\backslash\{v_0\}$ has the maximum number of connected components with proper-path-width k.

If $T_0\backslash\{v_0\}$ has no connected component with proper-path-width k, then $ppw(T) \leq k$ by Lemmas 2.20 and 2.21.

If $T_0\backslash\{v_0\}$ has two connected components with proper-path-width k, let T_1 be one of these components and $v_1 \in V(T_1)$ be a vertex adjacent to v_0 in T_0. We recursively define T_i and $v_i \in V(T_i)$ $(1 < i \leq a)$ while $T_{i-1}\backslash\{v_{i-1}\}$ has a component with proper-path-width k as follows: Let T_i be a connected component of $T_{i-1}\backslash\{v_{i-1}\}$ with proper-path-width k and $v_i \in V(T_i)$ be a vertex adjacent to v_{i-1} in T_{i-1}. $T_a\backslash\{v_a\}$ has no connected component with proper-path-width k. Let T_{a+1} be the other connected component of $T_0\backslash\{v_0\}$ with proper-path-width k, and $v_{a+1} \in V(T_{a+1})$ be a vertex adjacent to v_0 in T_0. Define recursively T_i and $v_i \in V(T_i)$ $(a+1 < i \leq b)$ as above. Notice that $T_i\backslash\{v_i\}$ $(1 \leq i \leq b)$ has at most one connected component with proper-path-width k, for otherwise $T_0\backslash\{v_i\}$ has three or more connected components with proper-path-width k, contradicting the assumption of the lemma.

Let H'_i $(0 \leq i \leq b)$ be the union of components of $T_i\backslash\{v_i\}$ with proper-path-width $\leq k-1$, and H_i $(0 \leq i \leq b)$ be the induced subgraph of T on $V(H'_i) \cup \{v_i\}$. By Lemma 2.20, $pw(H'_i) \leq k-1$ $(0 \leq i \leq b)$. By Lemma 2.21, $pw(H_i) \leq k$ $(0 \leq i \leq b)$. It is easy to see that there exists a proper-path-decomposition $(\mathcal{X}'_{(i)} \cup \{v_i\})$ of H_i with width at most k such that $\mathcal{X}'_{(i)}$ is a proper-path-decomposition of H'_i with width $\leq k-1$ $(0 \leq i \leq b)$. Let $\mathcal{X}_{(i)}$ be a such proper-path-decomposition of H_i $(0 \leq i \leq b)$. We define sequences $\mathcal{L}$ and $\mathcal{R}$ as follows.

$$\begin{aligned}
\mathcal{L} &= (\mathcal{X}_{(a)}, \{v_a, v_{a-1}\}, \mathcal{X}_{(a-1)}, \{v_{a-1}, v_{a-2}\}, \ldots, \mathcal{X}_{(2)}, \{v_2, v_1\}, \mathcal{X}_{(1)}, \{v_1, v_0\}). \\
\mathcal{R} &= (\{v_0, v_{a+1}\}, \mathcal{X}_{(a+1)}, \{v_{a+1}, v_{a+2}\}, \mathcal{X}_{(a+2)}, \ldots, \{v_{b-2}, v_{b-1}\}, \mathcal{X}_{(b-1)}, \{v_{b-1}, v_b\}, \mathcal{X}_{(b)}).
\end{aligned}$$

It is easy to see that $(\mathcal{L}, \mathcal{X}_{(0)}, \mathcal{R})$ is a sequence of subsets of $V(T)$ with width at most k satisfying the conditions in Lemma 2.15, and $ppw(T) \leq k$ by Lemma 2.15.

If $T_0 \backslash \{v_0\}$ has just one connected component with proper-path-width k, the sequence $\mathcal{R}$ above is empty, and $(\mathcal{L}, \mathcal{X}_{(0)})$ is a sequence of subsets of $V(T)$ with width at most k satisfying the conditions in Lemma 2.15, and we also have $ppw(T) \leq k$ by Lemma 2.15. □

The following theorem shows necessary and sufficient conditions for trees with proper-path-width at least $k+1$.

Theorem 2.3 *For any tree T and integer $k \geq 2$, $ppw(T) \geq k+1$ if and only if T has a vertex v such that $T \backslash \{v\}$ has at least three connected components with proper-path-width k or more.*

Proof: Suppose that $ppw(T) \geq k+1$. Let T' be a minimal subgraph of T with $ppw(T') \geq k+1$. Since T' is minimal, $T' \backslash \{w\}$ has no connected component with proper-path-width $\geq k+1$ for any $w \in V(T')$. Thus there exists a vertex $v \in V(T')$ such that $T' \backslash \{v\}$ has at least three connected components with proper-path-width k, for otherwise $ppw(T') \leq k$ by Lemma 2.22. Hence $T \backslash \{v\}$ has at least three connected components with proper-path-width $\geq k$. The converse follows from Lemma 2.19. □

2.4 Complexity of Computing Path-Width

2.4.1 k-Path and k-Intercat

A *clique* of a graph G is a complete subgraph of G. A clique on k vertices is called *k-clique*. For a positive integer k, *k-trees* are defined recursively as follows:

1. The complete graph on k vertices is a k-tree;
2. Given a k-tree Q on n vertices ($n \geq k$), a graph obtained from Q by adding a new vertex adjacent to the vertices of a k-clique of Q is a k-tree on $n+1$ vertices.

A k-tree Q is called a *k-path* [68] or *k-chordal path* [4] if either $|V(Q)| \leq k+1$ or Q has exactly two vertices of degree k. A *k-separator* S of a connected graph G is an induced subgraph of G on k vertices such that $G \backslash V(S)$ has at least two connected components. It is well-known that a k-separator of a k-tree Q is a k-clique of Q. For a positive integer k, *k-intercats (interior k-caterpillars)* [68] or *k-interval graph* [4] are defined as follows:

1. A k-path is a k-intercat;

2. Given a k-intercat Q on n vertices ($n \geq k+2$), a graph obtained from Q by adding a new vertex adjacent to the vertices of a k-separator of Q is also a k-intercat on $n+1$ vertices.

A 1-path, 1-intercat, and 1-tree are an ordinary path, caterpillar, and tree, respectively. A subgraph of a k-path, k-intercat, and k-tree are called a *partial k-path*, *partial k-intercat*, and *partial k-tree*, respectively.

It is well-known that any k-intercat H on n vertices ($n \geq k$) can be obtained as follows:

1. Define that Q_k is the complete graph on k vertices C_k;

2. Given Q_i and C_i ($k \leq i \leq n-1$), define that Q_{i+1} is the k-intercat obtained from Q_i by adding vertex $v_{i+1} \notin V(Q_i)$ adjacent to the vertices in C_i, and let $C_{i+1} = (C_i \cup \{v_{i+1}\}) - \{w_i\}$ where $w_i \in C_i \cup \{v_{i+1}\}$;

3. Define $H = Q_n$.

In the following, to determine the complexity of computing (proper-)path-width, we show that path-width characterizes partial k-intercat and proper-path-width characterizes partial k-path.

2.4.2 Path-Width and k-Intercat

Theorem 2.4 *For any simple graph G and an integer k ($k \geq 1$), $pw(G) \leq k$ if and only if G is a partial k-intercat.*

Proof: Suppose that $pw(G) = h \leq k$. If $h = 0$ then G is trivially a k-intercat. Thus we assume that $h \geq 1$. There exists a full h-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of G by Lemma 2.4. If $r = 1$ then G is a subgraph of a complete graph on $h+1$ vertices, and so we conclude that G is a partial h-intercat. Thus we assume that $r \geq 2$. We construct an h-intercat H from $\mathcal{X}$ as follows:

1. Let v_1 be a vertex in $X_1 \cap X_2$. Define that Q_0 is the complete graph on $X_1 - \{v_1\}$;

2. Define that Q_1 is the h-intercat obtained from Q_0 by adding v_1 and the edges connecting v_1 and the vertices in $X_1 - \{v_1\}$;

3. Given Q_i $(1 \leq i \leq r-1)$, define that Q_{i+1} is the h-intercat obtained from Q_i by adding $v_{i+1} \in X_{i+1} - X_i$ and the edges connecting v_{i+1} and the vertices in $X_{i+1} - \{v_{i+1}\}$;

4. Define $H = Q_r$.

Since $|X_{i+1} - X_i| = 1$ from the definition of full h-path-decomposition, v_{i+1} is uniquely determined $(1 \leq i \leq r-1)$. Since $X_{i+1} - \{v_{i+1}\} = ((X_i - \{v_i\}) \cup \{v_i\}) - \{w_i\}$ where $w_i \in X_i - X_{i+1}$ $(1 \leq i \leq r-1)$, H is an h-intercat. Furthermore, we have $V(H) = V(G)$ and $E(H) \supseteq E(G)$ from the definitions of path-decomposition and Q_i. Thus G is a partial h-intercat, and so a partial k-intercat.

Conversely, suppose, without loss of generality, that G is a partial h-intercat $(1 \leq h \leq k)$ with n' $(n' > h)$ vertices and H is an h-intercat such that $V(H) \supseteq V(G)$ and $E(H) \supseteq E(G)$. Let $n = |V(H)|$. As we mentioned before, we can assume that H can be obtained as follows:

1. Define that Q_h is the complete graph on h vertices C_h;

2. Given Q_i and C_i $(h \leq i \leq n-1)$, define that Q_{i+1} is the h-intercat obtained from Q_i by adding vertex $v_{i+1} \notin V(Q_i)$ adjacent to the vertices in C_i, and let $C_{i+1} = (C_i \cup \{v_{i+1}\}) - \{w_i\}$ where $w_i \in C_i \cup \{v_{i+1}\}$;

3. Define $H = Q_n$.

We define that $X_i = C_i \cup \{v_{i+1}\}$ $(h \leq i \leq n-1)$ and $\mathcal{X} = (X_h, X_{h+1}, \ldots, X_{n-1})$. It is easy to see that $\bigcup_{i=h}^{n-1} X_i = V(H)$ and each vertex appears in consecutive X_i's. Thus $\mathcal{X}$ satisfies conditions (ii) and (iv) in Definition 2.1. Since $w_i \in X_i - X_{i+1}$ and $v_{i+2} \in X_{i+1} - X_i$, $X_i \not\subseteq X_{i+1}$ and $X_{i+1} \not\subseteq X_i$ $(h \leq i \leq n-2)$. Hence $\mathcal{X}$ satisfies condition (i) in Definition 2.1 by Lemma 2.2. Since each edge of H either connects v_{i+1} and a vertex in C_i for some i $(h \leq i \leq n-1)$ or connects vertices in C_h, both ends of each edge of H are contained in some X_i. Thus $\mathcal{X}$ satisfies condition (iii) in Definition 2.1. It is easy to see that $|X_i| = h+1$ $(h \leq i \leq n-1)$ and $|X_i \cap X_{i+1}| = |C_{i+1}| = h$ $(h \leq i \leq n-2)$. Thus the sequence $\mathcal{X}$ is a full h-path-decomposition of H. Therefore, we have that $pw(G) \leq pw(H) \leq h \leq k$. □

2.4.3 Proper-Path-Width and k-Path

Theorem 2.5 *For any simple graph G and an integer k $(k \geq 1)$, $ppw(G) \leq k$ if and only if G is a partial k-path.*

Proof: Suppose that $ppw(G) = h \leq k$. If $h = 0$ then G is trivially a k-path. Thus we assume that $h \geq 1$. There exists a full h-proper-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of G by Lemma 2.14. If $r = 1$ then G is a subgraph of a complete graph on $h+1$ vertices, and so we conclude that G is a partial h-path. Thus we assume that $r \geq 2$. We construct an h-path H from $\mathcal{X}$ as follows:

1. Let v_1 be a vertex in $X_1 \cap X_2$. Define that Q_1 is the complete graph on $X_1 - \{v_1\}$.
2. Define that Q_2 is the h-path obtained from Q_1 by adding v_1 and the edges connecting v_1 and the vertices in $X_1 - \{v_1\}$.
3. Given Q_i $(2 \leq i \leq r)$, define that Q_{i+1} is the h-path obtained from Q_i by adding $v_i \in X_i - X_{i-1}$ and the edges connecting v_i and the vertices in $X_i - \{v_i\}$.
4. Define $H = Q_{r+1}$.

From the definition of full h-proper-path-decomposition, v_i is uniquely determined $(2 \leq i \leq r)$. Since $X_i - \{v_i\} \subset X_{i-1}$ $(2 \leq i \leq r)$, the induced subgraph of Q_i on $X_i - \{v_i\}$ is an h-clique of Q_i $(2 \leq i \leq r)$, and the induced subgraph of Q_{i+1} on X_i is $(h+1)$-clique. Thus H is an h-tree. Notice that $v_i \in X_{i+1}$ $(2 \leq i \leq r-1)$, for otherwise $|X_{i-1} \cap X_{i+1}| = h$. Since only the vertex in $X_2 - X_1$ and v_r have degree h, H is an h-path. Furthermore, we have $V(H) = V(G)$ and $E(H) \supseteq E(G)$ from the definitions of proper-path-decomposition and Q_i. Thus G is a partial h-path, and so a partial k-path.

Conversely, suppose, without loss of generality, that G is a partial h-path $(1 \leq h \leq k)$ with n $(n > h)$ vertices and H is an h-path such that $V(H) \supseteq V(G)$ and $E(H) \supseteq E(G)$. Let $n = |V(H)|$. Since a graph obtained from an h-path by deleting a vertex of degree h, if exists, is also an h-path, H can be obtained as follows:

1. Denote by $Q_1 = R_1$ the complete graph on h vertices.
2. Given Q_i and R_i $(1 \leq i \leq n-h)$, denote by Q_{i+1} the h-path obtained from Q_i by adding vertex $v_i \notin Q_i$ and the edges connecting v_i and the vertices of R_i, and let R_{i+1} be an h-clique of Q_{i+1} that contains v_i.
3. Define $H = Q_{n-h+1}$.

We define $X_i = V(R_i) \cup \{v_i\}$ $(1 \leq i \leq n-h)$ and $\mathcal{X} = (X_1, X_2, \ldots, X_{n-h})$. It is easy to see that $|X_i| = h+1$ for any i, $\bigcup_{i=1}^{n-h} X_i = V(H)$, and each vertex appears in consecutive X_i's. Thus $\mathcal{X}$ satisfies conditions (ii) and (iv) in Definition 2.2, and the width of $\mathcal{X}$ is h.

Since $v_i \in X_i - X_{i-1}$ and $\emptyset \neq V(R_{i-1}) - V(R_i) \subseteq X_{i-1} - X_i$, $X_i \not\subseteq X_{i-1}$ and $X_{i-1} \not\subseteq X_i$ for any i. Hence $\mathcal{X}$ satisfies condition (i) in Definition 2.2 by Lemma 2.2. Each edge of H either connects v_i with a vertex in $V(R_i)$ for some i or connects vertices in $V(R_1)$. So, both ends of each edge of H is contained in some X_i. Thus $\mathcal{X}$ satisfies condition (iii) in Definition 2.2. Since $V(R_{i+1}) = X_i \cap X_{i+1}$, $|X_i \cap X_{i+1}| = |V(R_{i+1})| = h$ for any i with $1 \leq i < n - h$. Since $X_{i+1} - X_{i-1} = \{v_i, v_{i+1}\}$, $|X_{i-1} \cap X_{i+1}| = h - 1 = |X_i| - 2$ $(1 < i < n - h)$. Thus the sequence $\mathcal{X}$ is a full h-proper-path-decomposition of H from Lemma 2.13. Therefore, we have that $ppw(G) \leq ppw(H) \leq h \leq k$. □

2.4.4 NP-hardness of Computing Path-Width and Proper-Path-Width

Arnborg, Corneil, and Proskurowski [4] proved that the problem of deciding, given a graph G and an integer k, whether G is a partial k-intercat is NP-complete, and whether G is a partial k-path is NP-complete. Thus we immediately have the following by Theorems 2.4 and 2.5.

Theorem 2.6 *The problem of computing $pw(G)$ is NP-hard.*

Theorem 2.7 *The problem of computing $ppw(G)$ is NP-hard.*

There are many other parameters which are equivalent to the path-width. The path-width of a graph G is equal to its vertex separation number $vs(G)$ [48], its gate matrix layout cost $gml(G)$ minus one [37], its interval thickness $it(G)$ minus one [62, 91], and its narrowness $na(G)$ minus one [52]. Kirousis and Papadimitriou proved that the node-search number $ns(G)$ of a graph G is equal to its interval thickness [50] and its vertex separation number plus one [51]. The relation between path-width and node-search number was mentioned in [62, 11]. Thus, for any graph G, $pw(G) = vs(G) = gml(G) - 1 = it(G) - 1 = na(G) - 1 = ns(G) - 1$.

As mentioned above, the problem computing these parameters are NP-hard. In fact, NP-hardness was independently shown for interval thickness [47], vertex separation number [56], and node-search number [51]. Although this problem is NP-hard for chordal graphs [44], and for planar graphs with vertex degree at most three by combining the results in [63] and [51], it can be solved in linear time for trees [62, 91, 32]. It is also known that for any fixed integer k, a k-path-decomposition of G with path-width at most k can be obtained, if exists, in $O(n \log n)$ time for general graphs by combining the results in [15] and [69], and in $O(n+e)$ time for cographs [16], where $n = |V(G)|$ and $e = |E(G)|$.

In the next section, we show that a proper-path-width of a tree T can be computed in linear time, and a proper-path-decomposition of T with width $ppw(T)$ can be obtained in linear time.

2.5 Proper-Path-Width of Trees

2.5.1 A Linear Time Algorithm for the Proper-Path-Width of Trees

Although the problem computing proper-path-width is NP-hard, it can be solved in linear time for trees. We show a practical algorithm to compute $ppw(T)$ for trees T based on Theorem 2.3, and prove the following.

Theorem 2.8 *For any tree T, the problem of computing $ppw(T)$ is solvable in linear time.*

Proof: For any tree T, $ppw(T) \geq 2$ if and only if T has a vertex v such that $T \backslash \{v\}$ has at least three connected components. That is, if we regard the proper-path-width of a single vertex as one, then Theorem 2.3 is true when $k = 1$. Thus we regard the proper-path-width of a single vertex as one if a tree has an edge in the algorithm.

Our algorithm to compute $ppw(T)$ is shown in Figs. 2.7 and 2.8. The outline of the algorithm is as follows.

We define the path-vector $\overline{pv}(v,T) = (p,c,n)$ for any tree T with a vertex $v \in V(T)$ as the root to compute $ppw(T)$. p describes the proper-path-width of T. c and n describe the condition of T as follows: If there exists $u \in V(T) - \{v\}$ such that $T \backslash \{u\}$ has two connected components with proper-path-width $ppw(T)$ and without v, then $c = 3$ and n is the path-vector of the connected component of $T \backslash \{u\}$ containing v; otherwise, c is the number of the connected components of $T \backslash \{v\}$ with proper-path-width $ppw(T)$ and $n = nul$. It should be noted that for any vertex $u \in V(T)$ the number of connected components of $T \backslash \{u\}$ with proper-path-width $ppw(T)$ is at most two from Theorem 2.3. Notice also that if there exists u such that $T \backslash \{u\}$ has two connected components with proper-path-width $ppw(T)$ and without v then u is uniquely determined. If there is no such u then the number of connected components of $T \backslash \{w\}$ with proper-path-width $ppw(T)$ and without v is not more than the number of connected components of $T \backslash \{v\}$ with proper-path-width $ppw(T)$. In the following, we denote an element x in $\overline{pv}(v,T)$ by $\overline{pv}(v,T)|x$, where x is either p, c or n.

Let T_0 be a tree with root $v \in V(T_0)$ and P_0 be the path-vector of T_0. We recursively define T_i and P_i $(1 \leq i \leq l)$ while $P_{i-1}|c = 3$ as follows: Let $u_{i-1} \in V(T_{i-1}) - \{v\}$ be the vertex such that $T_{i-1} \backslash \{u_{i-1}\}$ has two connected components with proper-path-width $ppw(T_{i-1})$ and without v, T_i be the connected component of $T_{i-1} \backslash \{u_{i-1}\}$ containing v as the root, and P_i be the path-vector of T_i. Assume that $P_l|c \neq 3$. We call such path-vectors $P_0, P_1, \ldots, P_l$ the chain of the path-vector P_0. We define b, n^*, b^*, and btm in the chain of P_0 as follows: Define that $P_i|b = P_{i-1}$ $(1 \leq i \leq l)$; define that $P_i|n^* = P_j$ if either $i = 0$ or $P_i|p < P_{i-1}|p - 1$ $(1 \leq i \leq l)$ where j is the maximum integer such that $j - i = P_i|p - P_j|p$; define that $P_i|b^* = P_j$ if $P_j|n^*$ is defined and $P_j|n^* = P_i$; define that $P_0|btm = P_l$. Thus we extend a path-vector as $\overline{pv}(v,T) = (p, c, n, b, n^*, b^*, btm)$ to reduce the time to traverse the chain as used in [61].

In the procedure, we omit the description of substitutions for b, n^*, b^*, and btm in the path-vector because no confusion is caused. Moreover, after substitutions, we can update n^*, b^*, and btm in the path-vectors in the chain in constant time. So we also omit the description of these operations. For the simplicity, if the substitution for P uses $P|x$, we abbreviate $P|x$ to x.

Suppose that a tree T_0 rooted at s is obtained from tree T_s rooted at s and tree T_t rooted at t by adding an edge (s,t). Based on Theorem 2.3, Procedure MERGE shown in Fig. 2.7 recursively calculates the path-vector of T_0 from the path-vector P_s of T_s and the path-vector P_t of T_t. Since the larger proper-path-width of two merged trees is reduced by at least two whenever Procedure MERGE is recursively called, the number of recursive calls is at most $\max(ppw(T_s), ppw(T_t)) - 1$. Since the time complexity of Procedure MERGE is $O(1)$ except for recursive calls, Procedure MERGE calculates the path-vector of T_0 in $O(\max(ppw(T_s), ppw(T_t)))$ time.

In Procedure LMERGE shown in Fig. 2.8, we can determine P' by using btm and b^* in the chain of the path-vector in $O(\min(ppw(T_s), ppw(T_t)))$ time. If P' is determined at either 1b or 2b in Procedure LMERGE then the number of recursive calls of Procedure MERGE is at most $P'|n^*|n|p < \min(ppw(T_s), ppw(T_t))$. Otherwise Procedure MERGE returns the path-vector in $O(1)$ time. Thus Procedure LMERGE calculates the path-vector of the join of two subtrees in $O(\min(ppw(T_s), ppw(T_t)))$ time. Procedure DFS shown in Fig. 2.8 computes the path-vector of a maximal subtree rooted at s in T from the path-vectors of maximal subtrees rooted at children of s in T by using Procedure LMERGE. Procedure MAIN shown in Fig. 2.8 obtains the proper-path-width of T from the path-vector of T obtained by Procedure DFS. The algorithm starts with the isolated vertices obtained from T by deleting all edges in T and reconstruct T by adding edge by edge while computing path-vectors of connected components.

Let $S(T)$ denote the time required to compute the path-vector of T, $M(T_1, T_2)$ denote the time required to obtain the path-vector of T from the path-vectors of T_1 and T_2 by Procedure LMERGE. From Corollary 3.4 on page 49, we have $ppw(T) = O(\log |V(T)|)$. Thus we have the following,

$$\begin{aligned} S(T) &\leq S(T_1) + S(T_2) + M(T_1, T_2) \\ &\leq S(T_1) + S(T_2) + O(\min(ppw(T_1), ppw(T_2))) \\ &\leq S(T_1) + S(T_2) + O(\log(\min(|V(T_1)|, |V(T_2)|))). \end{aligned}$$

Notice that the recurrence defined by $f(1) = 1$ and, for $n \geq 2$,

$$f(n) = \max_{1 \leq i < n}(f(i) + f(n-i) + \lceil \log_2(\min(i, n-i)) \rceil)$$

satisfies $f(n) = O(n)$. An easy way to verify this is to prove that, for $n \geq 1$, $f(n) \leq 2n - 1 - \lceil \log_2 n \rceil$ by a straightforward induction. Thus we can prove that the time complexity of the algorithm is $O(n)$ where $n = |V(T)|$. □

2.5.2 Interval Set and Proper-Path-Decomposition

Let Z be the set of integers. We denote an *interval* on integers by I. Two intervals I_1 and I_2 on integers are said to be *adjacent* if there exist integers $i \in I_1$ and $j \in I_2$ such that $|i - j| \leq 1$, and said to be *independent* if there exists no integer $i \in I_1 \cap I_2$ such that $\{i-1, i+1\} \not\subseteq I_1$ and $\{i-1, i+1\} \not\subseteq I_2$. A set $\mathcal{I}$ of distinct non-singleton intervals on integers such that any two distinct intervals are independent is called an *interval set* of a graph G if there exists a one-to-one correspondence $J : V(G) \to \mathcal{I}$ such that $J(u)$ and $J(v)$ are adjacent if $(u, v) \in E(G)$. For any $i \in Z$ and a set $\mathcal{I}$ of intervals on integers, define $\mathcal{I}(i) = \{I \mid i \in I, I \in \mathcal{I}\}$. The *density* of $\mathcal{I}$ is $\max_{i \in Z} |\mathcal{I}(i)|$. An interval set $\mathcal{I}$ of G is said to be *optimal* if the density of $\mathcal{I}$ is minimum over all interval sets of G.

In the following, we denote $a \in A$ if a is a member of a sequence A.

Suppose that $\mathcal{I}$ is an interval set of G with a one-to-one correspondence $J : V(G) \to \mathcal{I}$. For any vertex $v \in V(G)$, define that $l(v)$ (respectively, $r(v)$) is the integer i such that $i \in J(v)$ and $i - 1 \notin J(v)$ (respectively, $i + 1 \notin J(v)$). A sequence $(v_1, v_2, \ldots, v_{|\mathcal{I}|})$ of $V(G)$ is called the left (respectively, right) terminal sequence of $\mathcal{I}$ if $l(v_1) < l(v_2) < \cdots < l(v_{|\mathcal{I}|})$

(respectively, $r(v_1) < r(v_2) < \cdots < r(v_{|\mathcal{I}|})$). A sequence $(L_1, R_1, L_2, R_2, \ldots, L_r, R_r)$ is called the *terminal sequence* of $\mathcal{I}$ if the following conditions are satisfied: $(L_1, L_2, \ldots, L_r)$ and $(R_1, R_2, \ldots, R_r)$ are the left and right terminal sequences of $\mathcal{I}$, respectively; both L_i and R_i are nonempty $(1 \leq i \leq r)$; for any vertices $u \in L_i$ and $v \in R_i$ $(1 \leq i \leq r)$, $l(u) < r(v)$; for any vertices $v \in R_i$ and $u \in L_{i+1}$ $(1 \leq i \leq r-1)$, $r(v) < l(u)$. Notice that $l(u) \neq l(v)$, $r(u) \neq r(v)$, and $l(u) \neq r(v)$ for any distinct vertices $u, v \in V(G)$.

Before proving Theorem 2.9 below, we need the following lemmas.

Lemma 2.23 *For any graph G with at least two vertices, there exists an optimal interval set of G with the terminal sequence $(L_1, R_1, \ldots, L_r, R_r)$ such that $|L_r| = 1$ and $r \geq 2$.*

Proof: Suppose that $\mathcal{I}$ is an optimal interval set of G with a one-to-one correspondence $J : V(G) \to \mathcal{I}$ and the terminal sequence $(L_1, R_1, \ldots, L_r, R_r)$. Since $|V(G)| \geq 2$, if $|L_r| = 1$ then $r \geq 2$. Thus we assume that $|L_r| \geq 2$. Let v be the vertex in $V(G)$ such that $l(v) = \max_{w \in L_r} l(w)$, and u be the vertex in $V(G)$ such that $r(u) = \min_{w \in R_r - \{v\}} r(w)$. Define that $J'(v) = \{i | l(v)+1 \leq i \leq \max_{w \in R_r} r(w)+1, i \in Z\}$, $J'(u) = \{i | l(u) \leq i \leq l(v), i \in Z\}$, and $J'(w) = J(w)$ for any $w \in V(G) - \{u, v\}$. Let L'_r be the sequence obtained from L_r by deleting v, and R'_r be the sequence obtained from R_r by deleting u and moving v into the last. Then it is not difficult to see that $\{J'(w) | w \in V(G)\}$ is an optimal interval set of G with the terminal sequence $(L_1, R_1, \ldots, R_{r-1}, L'_r, u, v, R'_r)$. Thus we have this lemma. □

Lemma 2.24 *For any proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ of a connected graph G with at least two vertices, $|X_i| \geq 2$ $(1 \leq i \leq r)$.*

Proof: Suppose that $X_l = \{v\}$ for some l $(1 \leq l \leq r)$. Since G is connected and contains at least two vertices, there exists $u \in V(G) - \{v\}$ such that $(v, u) \in E(G)$. Thus $\{u, v\} \in X_i$ for some i $(1 \leq i \leq r)$ by condition (iii) in Definition 2.2. But this is contradicting to condition (i) in Definition 2.2 since $X_l \subset X_i$. □

Theorem 2.9 *For any non empty graph G and an integer k $(k \geq 1)$, there exists a proper-path-decomposition of G with width k if and only if there exists an interval set of G with density k.*

Proof: If G is not connected, it is sufficient to confirm the lemma for each connected component. It is trivial for a graph with one vertex. Thus we assume that G has at least two vertices and connected.

Suppose that $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ is a k-proper-path-decomposition of G. Let $V_1 = X_1$, $V_i = X_i - X_{i-1}$ $(2 \leq i \leq r)$, $U_i = X_i - X_{i+1}$ $(1 \leq i \leq r-1)$, and $U_r = X_r$. Let $v_i \in V_i$ and $u_i \in U_i$ such that $v_i \neq u_i$ $(1 \leq i \leq r)$. Notice that $V_i \neq \emptyset$, $U_i \neq \emptyset$, and $|V_i \cup U_i| \geq 2$ by Lemma 2.24 and conditions (i) and (v) in Definition 2.2. Let $\mathcal{I}$ be the set of intervals defined as follows:

1. Let $i = 1$ and $j = 1$;

2. For each vertex $w \in V_i - \{v_i\}$, define $l(w) = j$ and let $j = j + 1$;

3. Define $r(u_i) = j$ and $l(v_i) = j + 1$, and let $j = j + 2$;

4. For each vertex $w \in U_i - \{u_i\}$, define $r(w) = j$ and let $j = j + 1$;

5. If $i < r$ then let $i = i + 1$ and return to 2;

6. Define $J(w) = \{i | l(w) < i < r(w), i \in Z\}$ for any $w \in V(G)$, and let $\mathcal{I} = \{J(w) \mid w \in V(G)\}$.

First, we show that the intervals in $\mathcal{I}$ are well-defined. Since both of $(V_1, V_2, \ldots, V_r)$ and $(U_1, U_2, \ldots, U_r)$ are partitions of $V(G)$, both $l(w)$ and $r(w)$ are defined for any vertex $w \in V(G)$. Assume that $w \in V_i$ $(1 \leq i \leq r)$ and $w \in U_j$ $(1 \leq j \leq r)$. If $j < i$ then $w \in X_j \cap X_i$ and $w \notin X_{j+1}$. But this is contradicting to condition (iv) in Definition 2.2 since $X_j \cap X_i \not\subseteq X_{j+1}$. Thus $i \leq j$. If $i < j$ then trivially $l(w) < r(w)$ by the definition of $l(w)$ and $r(w)$. If $i = j$ then also $l(w) < r(w)$ since $v_i \neq u_i$. Thus $J(w)$ is a non-singleton interval on integers for any vertex $w \in V(G)$. Hence $\mathcal{I}$ is a set of distinct non-singleton intervals on integers such that any two distinct intervals in $\mathcal{I}$ are independent, and $J : V(G) \rightarrow \mathcal{I}$ is a one-to-one correspondence. Next, we show that $\mathcal{I}$ is an interval set of G. For some edge $(u, v) \in E(G)$, assume that $\{u, v\} \subseteq X_i$ by condition (iii) in Definition 2.2. If $\{u, v\} \subseteq X_i - \{v_i\}$ then intervals $J(u)$ and $J(v)$ are adjacent to each other since $\{J(u), J(v)\} \subseteq \mathcal{I}(r(u_i))$. Similarly, if $\{u, v\} \subseteq X_i - \{u_i\}$ then intervals $J(u)$ and $J(v)$ are adjacent to each other since $\{J(u), J(v)\} \subseteq \mathcal{I}(l(v_i))$. Otherwise ($\{u, v\} = \{u_i, v_i\}$) intervals $J(u)$ and $J(v)$ are adjacent to each other since $l(v_i) - r(u_i) = 1$. Thus for any edge $(u, v) \in E(G)$, intervals $J(u)$ and $J(v)$ are adjacent to each other. That is, $\mathcal{I}$ is an interval set of G. Finally, we show that the density of $\mathcal{I}$ is k. It is easy to see that $\max_{w \in V_i} |\mathcal{I}(l(w))| = |\mathcal{I}(r(u_i))| = |\mathcal{I}(l(v_i))| = \max_{w \in U_i} |\mathcal{I}(r(w))|$ for any i $(1 \leq i \leq r)$. Since $\max_{1 \leq i \leq r} |\mathcal{I}(l(v_i))| = \max_{1 \leq i \leq r} |X_i - \{u_i\}| = k$, the density of $\mathcal{I}$ is k. Thus $\mathcal{I}$ is an interval set of G with density k.

Conversely, suppose that $\mathcal{I}$ is an interval set of G with the terminal sequence $(L_1, R_1, \ldots, L_r, R_r)$ and density k. By Lemma 2.23, without loss of generality, we assume that $r \geq 2$ and $|L_r| = 1$. Let v_i be the vertex such that $l(v_i) = \min_{w \in L_i} l(w)$ for any i $(1 \leq i \leq r)$. We define a sequence $\mathcal{X} = (X_1, X_2, \ldots, X_{r-1})$ as follows:

1. Define $X_1 = L_1 \cup \{v_2\}$;

2. Given X_i $(1 \leq i \leq r-2)$, define $X_{i+1} = (X_i \cup L_{i+1} \cup \{v_{i+2}\}) - R_i$;

Since $R_i \cap L_{i+1} = \emptyset$ $(1 \leq i \leq r-2)$ and $L_r = \{v_r\}$, $\mathcal{X}$ satisfies conditions (ii) and (iv) in Definition 2.2. Since $v_{i+2} \in X_{i+1} - X_i$ and $X_i - X_{i+1} = R_i \neq \emptyset$ $(1 \leq i \leq r-2)$, $X_i \not\subseteq X_{i+1}$ and $X_{i+1} \not\subseteq X_i$. Thus $X_i \not\subseteq X_j$ for any distinct i and j, for otherwise $X_i = X_i \cap X_j \subseteq X_{i+1}$ $(i < j)$ or $X_i = X_i \cap X_j \subseteq X_{i-1}$ $(i > j)$. Hence $\mathcal{X}$ satisfies condition (i) in Definition 2.2. Let v'_i be the vertex such that $l(v'_i) = \max_{w \in L_i} l(w)$, and u'_i be the vertex such that $r(u'_i) = \max_{w \in R_i} r(w)$ $(1 \leq i \leq r)$. Let $J : V(G) \to \mathcal{I}$ be a one-to-one correspondence. Since $\bigcup_{w \in L_i} \mathcal{I}(l(w)) = \bigcup_{w \in R_i} \mathcal{I}(r(w)) = \mathcal{I}(l(v'_i))$ for any i $(1 \leq i \leq r)$, if two intervals $I_1, I_2 \in \mathcal{I}$ are adjacent then $I_1, I_2 \in \mathcal{I}(l(v'_i))$ or $\{I_1, I_2\} = \{J(u'_i), J(v_{i+1})\}$ $(1 \leq i \leq r-1)$. Notice that $\mathcal{I}(l(v'_r)) \subseteq \{J(v) | v \in X_{r-1}\}$ since $v_r = v'_r \in X_{r-1}$. Since $\{J(v) | v \in X_i\} = \mathcal{I}(l(v'_i)) \cup \{J(v_{i+1})\}$ $(1 \leq i \leq r-1)$, if two intervals $J(u), J(v) \in \mathcal{I}$ are adjacent then $u, v \in X_i$. Notice that $u'_i \in X_i$ $(1 \leq i \leq r-1)$. Thus by definition of an interval set, $\mathcal{X}$ satisfies condition (iii) in Definition 2.2. Since $v_{i+1} \notin X_{i-1} \cup R_i$ and $\emptyset \neq R_i \not\subseteq X_{i+1}$, we have $|X_{i-1} \cap X_{i+1}| \leq |X_i| - 2$ $(2 \leq i \leq r-2)$. Thus $\mathcal{X}$ satisfies condition (v) in Definition 2.2. Since $\max_{1 \leq i \leq r-1} |X_i| = \max_{1 \leq i \leq r-1} |\mathcal{I}(l(v'_i)) \cup \{J(v_{i+1})\}| = k+1$, the width of $\mathcal{X}$ is k. Therefore $\mathcal{X}$ is a k-proper-path-decomposition of G. □

Corollary 2.1 *For any graph G on n vertices, a k-proper-path-decomposition of G can be obtained in $O(kn)$ time if the terminal sequence of an interval set of G with density k is given.*

Notice that $r \leq n - k$ for any k-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ of G on n vertices.

2.5.3 A Linear Time Algorithm for Proper-Path-Decomposition of Trees

In the following, we show a practical algorithm to construct a proper-path-decomposition with optimal width for trees. As shown in Figs. 2.9 and 2.10, we modify the algorithm in Figs. 2.7 and 2.8 to construct the terminal sequence of an optimal interval set of a tree.

Theorem 2.10 *For any tree T with proper-path-width k ($k \geq 1$), the terminal sequence of an interval set of T with density k can be obtained in linear time.*

Proof: Let T_0 be a tree with root $v_0 \in V(T_0)$ and proper-path-width k. Suppose that $\overline{pv}(v_0, T_0)|c = 2$. Let T_1 be a connected component of $T_0 \backslash \{v_0\}$ with proper-path-width k, and $v_1 \in V(T_1)$ be the vertex adjacent to v_0 in T_0. We recursively define T_i and $v_i \in V(T_i)$ ($2 \leq i \leq a$) while $T_{i-1} \backslash \{v_{i-1}\}$ has a component with proper-path-width k as follows: Let T_i be a connected component of $T_{i-1} \backslash \{v_{i-1}\}$ with proper-path-width k and $v_i \in V(T_i)$ be the vertex adjacent to v_{i-1} in T_{i-1}. $T_a \backslash \{v_a\}$ has no connected component with proper-path-width k. Let T_{a+1} be the other connected component of $T_0 \backslash \{v_0\}$ with proper-path-width k, and $v_{a+1} \in V(T_{a+1})$ be the vertex adjacent to v_0 in T_0. Define recursively T_i and $v_i \in V(T_i)$ ($a+2 \leq i \leq b$) as above. Notice that $T_i \backslash \{v_i\}$ ($1 \leq i \leq b$) has at most one connected component with proper-path-width k, for otherwise $T_0 \backslash \{v_i\}$ has three or more connected components with proper-path-width k. Let H'_i ($0 \leq i \leq b$) be the union of components of $T_i \backslash \{v_i\}$ with proper-path-width $\leq k-1$, and H_i ($0 \leq i \leq b$) be the induced subgraph of T_0 on $V(H'_i) \cup \{v_i\}$. Notice that there is no connected component of $T_i \backslash \{v_i\}$ if $k = 1$. Let W'_i be the terminal sequence of an optimal interval set of H'_i. Since $ppw(H'_i) \leq k-1$ ($0 \leq i \leq b$), $W_i = (v_i, W'_i, v_i)$ is the terminal sequence of an interval set of H_i with density at most k by Theorem 2.9. It is easy to see that there exists an interval set $\mathcal{I}$ of T_0 with density k such that the terminal sequence of $\mathcal{I}$ is $(W_a, W_{a-1}, \ldots, W_1, W_0, W_{a+1}, W_{a+2}, \ldots, W_b)$.

Thus, if $\overline{pv}(v_0, T_0)|c = 2$, we assume that the terminal sequence of an interval set of T_0 with density k is $(W_L, v_0, W'_0, v_0, W_R)$ where $W_L = (W_a, W_{a-1}, \ldots, W_1)$ and $W_R = (W_{a+1}, W_{a+2}, \ldots, W_b)$. If $\overline{pv}(v_0, T_0)|c = 1$ then $T_0 \backslash \{v_0\}$ has just one connected component with proper-path-width k, the sequence W_R above is empty, and we assume that the terminal sequence of an interval set of T_0 with density k is (W_L, v_0, W'_0, v_0). Similarly, if $\overline{pv}(v_0, T_0)|c = 0$ then $T_0 \backslash \{v_0\}$ has no connected component with proper-path-width k, and we assume that the terminal sequence of an interval set of T_0 with density k is (v_0, W'_0, v_0).

If $\overline{pv}(v_0, T_0)|c \leq 2$ then we denote a terminal sequence, W_L, W'_0, W_R, v, and (W_L, W'_0, W_R) by W, L, C, R, r, and D, respectively.

Suppose that $\overline{pv}(v_0, T_0)|c = 3$. Let $u \in V(T_0) - \{v_0\}$ be the vertex such that $T_0 \backslash \{u\}$ has two connected components with proper-path-width k. Let T_L and T_R be two connected components of $T_0 \backslash \{u\}$ with proper-path-width k, T^* be the connected component of $T_0 \backslash \{u\}$ containing v_0, and T' be the union of the other connected components of $T_0 \backslash \{u\}$. Let $u_l \in T_L$ and $u_r \in T_R$ be the vertices adjacent to u in T_0. Since $T_L \backslash \{u_l\}$

has at most one connected component with proper-path-width k, $\overline{pv}(u_l, T_L)|c \leq 1$ and $\overline{pv}(u_r, T_R)|c \leq 1$. Thus we assume that the terminal sequences of optimal interval sets of T_L and T_R are $W_L = (W'_L, u_l)$ and $W_R = (u_r, W'_R)$, respectively. Then it is easy to see that there exists an interval set $\mathcal{I}$ of T_0 with density k such that the terminal sequence of $\mathcal{I}$ is $(W_L, u, W^*, W', u, W_R)$ where W^* and W' are the terminal sequences of optimal interval sets of T^*, and T', respectively.

If $\overline{pv}(v_0, T_0)|c = 3$ then we denote a terminal sequence, W_L, W^*, W', W_R, and u by W, L, N, C, R, and r, respectively. Moreover, the sequence obtained from the terminal sequence by deleting v is denoted by D.

We extend a path-vector as $\overline{pv}(v, T) = (p, c, n, b, n^*, b^*, btm, \{L, r, N, C, r, R\}, D)$. Notice that $W = (L, r, N, C, r, R)$. In the procedure, we omit the description of operations for b, n^*, b^*, and btm in the path-vector. Thus we denote the path-vector $\overline{pv}(v, T) = (p, c, n, \{L, r, N, C, r, R\}, D)$. The reverse of a terminal sequence is denoted by $\overline{W}$, and maintained in the procedure together with the reverses of L, N, C, R, and D. But we also omit the description of these operations.

Procedure MERGE-D shown in Fig. 2.9 recursively calculates the path-vector of T_0 from the path-vector P_s of T_s and the path-vector P_t of T_t in $O(\max(ppw(T_s), ppw(T_t)))$ time. Note that the time complexity of Procedure MERGE-D is $O(1)$ except for recursive calls. In Procedure LMERGE-D shown in Fig. 2.10, we can determine P' in $O(\min(ppw(T_s), ppw(T_t)))$ time by using btm and b^* in the chain of the path-vector. If P' is determined at 1.2 or 2.2 in Procedure LMERGE-D then the number of recursive calls of Procedure MERGE-D is at most $P'|n^*|n|p < \min(ppw(T_s), ppw(T_t))$. Otherwise Procedure MERGE-D returns the path-vector in $O(1)$ time. Thus Procedure LMERGE-D calculates the path-vector of the join of two subtrees in $O(\min(ppw(T_s), ppw(T_t)))$ time. Procedure DFS-D shown in Fig. 2.10 computes the path-vector of a maximal subtree rooted at s in T from the path-vectors of maximal subtrees rooted at children of s in T by using Procedure LMERGE-D. Procedure MAIN-D shown in Fig. 2.10 obtains the proper-path-width of T from the path-vector of T obtained by Procedure DFS-D. The algorithm starts with the isolated vertices obtained from T by deleting all edges in T and reconstruct T by adding edge by edge while computing path-vectors of connected components. Thus we can obtained the terminal sequence of an interval set of T with width $ppw(T)$ in linear time. □

By Corollary 2.1 and Theorem 2.10, we obtain the following theorem.

Theorem 2.11 *For any tree T with proper-path-width k, a k-proper-path-decomposition of T can be obtained in $O(n \log n)$ time.*

Notice that $ppw(T) = O(\log n)$ for any tree T on n vertices. It should be noted that a k-proper-path-decomposition of T, if exists, can be obtained in linear time if k is fixed. By a similar argument, a $pw(T)$-path-decomposition can be obtained in $O(n \log n)$ time for any tree T with n vertices.

```
Procedure MERGE( Ps, Pt )
[ Input:    Ps (path-vector of tree Ts rooted at s )
            Pt (path-vector of tree Tt rooted at t )
  Output:   the path-vector of tree rooted at s obtained from Ts and Tt by adding an edge (s,t). ]

  1. if Ps|p > Pt|p then
     (a) if Ps|c ≤ 2 then Ps := (p,c,nul);
     (b) else if Ps|n*|p < Pt|p then Ps := (p+1,0,nul);
     (c) else if Ps|n*|p = Pt|p then
          i. if Ps|n*|c ≥ 2 or Pt|c ≥ 2 then Ps := (p+1,0,nul);
         ii. else Ps|n* := (p,c+1,nul);
     (d) else if Ps|n*|c ≤ 2 then Ps|n* := (p,c,nul);
     (e) else if Ps|n*|c = 3 then
          i. Ps|n*|n := MERGE( Ps|n*|n, Pt );
         ii. if Ps|n*|n|p = Ps|n*|p then Ps := (p+1,0,nul);
         endif
     (f) return( Ps );
  2. else if Ps|p = Pt|p then
     (a) if Ps|c ≥ 2 or Pt|c ≥ 2 then Ps := (p+1,0,nul);
     (b) else Ps := (p,c+1,nul);
     (c) return( Ps );
  3. else if Ps|p < Pt|p then
     (a) if Pt|c ≤ 1 then Pt := (p,1,nul);
     (b) else if Pt|c = 2 then Pt := (p,3,Ps);
     (c) else if Ps|p > Pt|n*|p then Pt := (p+1,0,nul);
     (d) else if Ps|p = Pt|n*|p then
          i. if Ps|c ≥ 2 or Pt|n*|c ≥ 2 then Pt := (p+1,0,nul);
         ii. else Pt|n* := (p,Ps|c+1,nul);
     (e) else if Pt|n*|c ≤ 1 then Pt|n* := (p,1,nul);
     (f) else if Pt|n*|c = 2 then Pt|n* := (p,3,Ps);
     (g) else if Pt|n*|c = 3 then
          i. Pt|n*|n := MERGE( Ps, Pt|n*|n );
         ii. if Pt|n*|n|p = Pt|n*|p then Pt := (p+1,0,nul);
         endif
     (h) return( Pt );
     endif

END
```

Figure 2.7: Procedure MERGE: The algorithm to compute the path-vector of the join of two subtrees.

```
Procedure LMERGE( P_s, P_t )
[ Input:   P_s (path-vector of tree T_s rooted at s )
           P_t (path-vector of tree T_t rooted at t )
  Output:  the path-vector of tree rooted at s obtained from T_s and T_t by adding an edge (s,t). ]

  1. if P_s|p > P_t|p and P_s|c = 3 then
     (a) if P_s|btm|b*|p ≥ P_t|p then let P' be P_s|btm|b*;
     (b) else
             let P' be the path-vector P in the chain of P_s such that P|n* is defined and
             P|p ≥ P_t|p > P|n*|n|p;
     (c) P' := MERGE( P', P_t );
     (d) return( P_s );
     endif
  2. if P_s|p < P_t|p and P_t|c = 3 then
     (a) if P_t|btm|b*|p ≥ P_s|p then let P' be P_t|btm|b*;
     (b) else
             let P' be the path-vector P in the chain of P_t such that P|n* is defined and
             P|p ≥ P_s|p > P|n*|n|p;
     (c) P' := MERGE( P_s, P' );
     (d) return( P_t );
     endif
  3. return( MERGE( P_s, P_t ) );
END
Procedure DFS( s )
[ Input:   a vertex s
  Output:  the path-vector of the maximal subtree rooted at s ]

  1. P_s := (1, 0, nul); /* path-vector of a tree with one vertex s */
  2. for all children t of s in T do
     (a) P_t := DFS( t );
     (b) P_s := LMERGE( P_s, P_t );
     endfor
  3. return( P_s );
END
Procedure MAIN( T )
[ Input:   a tree T
  Output:  the proper-path-width of T ]

  1. Let r be a vertex in V(T);
  2. if T has no edge then return( 0 );
  3. else pv(r,T) := DFS( r );     (pv overlined)
  4. return( pv(r,T)|p );
END
```

Figure 2.8: The algorithm to compute $ppw(T)$.

Procedure MERGE-D(P_s, P_t)

[Input: P_s (path-vector of tree T_s rooted at s)
P_t (path-vector of tree T_t rooted at t)
Output: the path-vector of tree rooted at s obtained from T_s and T_t by adding an edge (s,t).]

1. if $P_s|p > P_t|p$ then
   (a) if $P_s|c \leq 2$ then $P_s := (p, c, -, \{L, r, -, (P_t|W, C), r, R\}, (L, P_t|W, C, R))$;
   (b) else if $P_s|n^*|p < P_t|p$ then $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
   (c) else if $P_s|n^*|p = P_t|p$ then
      i. if $P_s|n^*|c \geq 2$ or $P_t|c \geq 2$ then
         $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
      ii. else if $P_s|n^*|c = 0$ then $P_s|n^* := (p, 1, -, \{P_t|W, r, -, C, r, -\}, (P_t|W, D))$;
      iii. else if $P_s|n^*|c = 1$ then $P_s|n^* := (p, 2, -, \{L, r, -, C, r, P_t|\overline{W}\}, (D, P_t|\overline{W}))$;
         endif
   (d) else if $P_s|n^*|c \leq 2$ then $P_s|n^* := (p, c, -, \{L, r, -, (P_t|W, C), r, R\}, (L, P_t|W, C, R))$;
   (e) else if $P_s|n^*|c = 3$ then
      i. $P_s|n^*|n :=$ MERGE-D($P_s|n^*|n$, P_t);
      ii. if $P_s|n^*|n|p = P_s|n^*|p$ then $P_s := (p+1, 0, -, \{-, r, -, D, r, -\}, D)$;
      endif
   (f) return(P_s);
2. else if $P_s|p = P_t|p$ then
   (a) if $P_s|c \geq 2$ or $P_t|c \geq 2$ then $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
   (b) else if $P_s|c = 0$ then $P_s := (p, 1, -, \{P_t|W, r, -, C, r, -\}, (P_t|W, D))$;
   (c) else if $P_s|c = 1$ then $P_s := (p, 2, -, \{L, r, -, C, r, P_t|\overline{W}\}, (D, P_t|\overline{W}))$;
      endif
   (d) return(P_s);
3. else if $P_s|p < P_t|p$ then
   (a) if $P_t|c \leq 1$ then $P_t := (p, 1, -, \{W, P_s|r, -, P_s|D, P_s|r, -\}, (W, P_s|D))$;
   (b) else if $P_t|c = 2$ then $P_t := (p, 3, P_s, \{L, r, P_s|W, C, r, R\}, (L, r, P_s|D, C, r, R))$;
   (c) else if $P_s|p > P_t|n^*|p$ then
      $P_t := (p+1, 0, -, \{-, P_s|r, -, (W, P_s|D), P_s|r, -\}, (W, P_s|D))$;
   (d) else if $P_s|p = P_t|n^*|p$ then
      i. if $P_s|c \geq 2$ or $P_t|n^*|c \geq 2$ then
         $P_t := (p+1, 0, -, \{-, P_s|r, -, (W, P_s|D), P_s|r, -\}, (W, P_s|D))$;
      ii. else if $P_s|c = 0$ then $P_t|n^* := (p, 1, -, \{W, P_s|r, -, P_s|C, P_s|r, -\}, (W, P_s|D))$;
      iii. else if $P_s|c = 1$ then $P_t|n^* := (p, 2, -, \{P_s|L, P_s|r, -, P_s|C, P_s|r, \overline{W}\}, (P_s|D, \overline{W}))$;
         endif
   (e) else if $P_t|n^*|c \leq 1$ then $P_t|n^* := (p, 1, -, \{W, P_s|r, -, P_s|C, P_s|r, -\}, (W, P_s|D))$;
   (f) else if $P_t|n^*|c = 2$ then $P_t|n^* := (p, 3, P_s, \{L, r, P_s|W, C, r, R\}, (L, r, P_s|D, C, r, R))$;
   (g) else if $P_t|n^*|c = 3$ then
      i. $P_t|n^*|n :=$ MERGE-D(P_s, $P_t|n^*|n$);
      ii. if $P_t|n^*|n|p = P_t|n^*|p$ then $P_t := (p+1, 0, -, \{-, P_s|r, -, D, P_s|r, -\}, D)$;
      endif
   (h) return(P_t);
   endif

END

Figure 2.9: Procedure MERGE-D.

Procedure LMERGE-D(P_s, P_t)

Input:	P_s (path-vector of tree T_s rooted at s)
	P_t (path-vector of tree T_t rooted at t)
Output:	the path-vector of tree rooted at s obtained from T_s and T_t by adding an edge (s,t).

1. **if** $P_s|p > P_t|p$ **and** $P_s|c = 3$ **then**
   - (a) **if** $P_s|btm|b^*|p \geq P_t|p$ **then** let P' be $P_s|btm|b^*$;
   - (b) **else**
     let P' be the path-vector P in the chain of P_s such that $P|n^*$ is defined and $P|p \geq P_t|p > P|n^*|n|p$;
   - (c) $P' :=$ MERGE-D(P', P_t);
   - (d) **return**(P_s);

   endif
2. **if** $P_s|p < P_t|p$ **and** $P_t|c = 3$ **then**
   - (a) **if** $P_t|btm|b^*|p \geq P_s|p$ **then** let P' be $P_t|btm|b^*$;
   - (b) **else**
     let P' be the path-vector P in the chain of P_t such that $P|n^*$ is defined and $P|p \geq P_s|p > P|n^*|n|p$;
   - (c) $P' :=$ MERGE-D(P_s, P');
   - (d) **return**(P_t);

   endif
3. **return**(MERGE-D(P_s, P_t));

END

Procedure DFS-D(s)

Input:	a vertex s
Output:	the path-vector of the maximal subtree rooted at s

1. $P_s := (1, 0, -, \{-, s, -, -, s, -\}, -)$; /* path-vector of a tree with one vertex s */
2. **for all** children t of s in T **do**
   - (a) $P_t :=$ DFS-D(t);
   - (b) $P_s :=$ LMERGE-D(P_s, P_t);

   endfor
3. **return**(P_s);

END

Procedure MAIN-D(T)

Input:	a tree T
Output:	the proper-path-width of T

1. Let r be a vertex in $V(T)$;
2. **if** T has no edge **then return**($\{-, s, -, -, s, -\}$);
3. **else** $\overline{pv}(r, T)$:= DFS-D(r);
4. **return**($\overline{pv}(r, T)|W$);

END

Figure 2.10: The algorithm to construct the terminal sequence of an interval set of a tree.

Chapter 3

Path-Width and Graph Minor Theory

3.1 Introduction of Graph Minor Theory

A graph H is a *minor* of a graph G if H is isomorphic to a graph obtained from a subgraph of G by contracting edges. A family $\mathcal{F}$ of graphs is said to be *minor-closed* if the following condition holds: If $G \in \mathcal{F}$ and H is a minor of G then $H \in \mathcal{F}$. A graph G is a *minimal forbidden minor* for a minor-closed family $\mathcal{F}$ of graphs if $G \notin \mathcal{F}$ and any proper minor of G is in $\mathcal{F}$. $\mathcal{F}$ is characterized by the minimal forbidden minors for $\mathcal{F}$.

Robertson and Seymour developed Graph Minors Theory in the series of papers [71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86]. Surveies of the theory [87, 88, 38, 46, 35] or improvements of some results [11, 70] are also found in the literature. In the series, Robertson and Seymour proved the following deep theorems.

Theorem A (Robertson and Seymour[86]) *Every minor-closed family of graphs has a finite number of minimal forbidden minors.*

Theorem B (Robertson and Seymour[83]) *The problem of deciding if a fixed graph is a minor of an input graph can be solved in polynomial time.*

It follows that the problem of testing membership for any minor-closed family $\mathcal{F}$ of graphs can be solved in polynomial time provided that we know all the minimal forbidden minors for $\mathcal{F}$. As many important problems can be reduced to the problem, it is important to find all the minimal forbidden minors for minor-closed families of graphs. Although it is known that there is no general method to find all the minimal forbidden minors for any minor-closed family of graphs [37, 38], a special method could be applied to each minor-closed family of graphs. In fact, the minimal forbidden minors are known for minor-closed families of planar graphs [98], graphs embeddable on the projective plane [41, 1], partial 2-trees [30, 6], partial 3-trees [6, 90, 28], partial 4-trees [58], Δ-Y graphs [31], graphs with path-width at most 1 [36], and graphs with path-width at most 2 [49].

We investigate the family $\mathcal{F}_k$ of graphs with path-width at most k for any $k \geq 0$, and the family $\mathcal{P}_k$ of graphs with proper-path-width at most k for any $k \geq 1$. The families $\mathcal{F}_k$ and $\mathcal{P}_k$ are minor-closed families. $\mathcal{F}_k$ and $\mathcal{P}_k$ are known to have applications to VLSI layout, linguistics, and games on graphs [62, 52]. Thus, it is important to list the minimal forbidden minors for $\mathcal{F}_k$ and $\mathcal{P}_k$. Although Kinnersley and Langston [49] list all 110 minimal forbidden minors for $\mathcal{F}_2$, it is open to list all minimal forbidden minors for $\mathcal{F}_k$ ($k \geq 3$) and $\mathcal{P}_k$ ($k \geq 2$). We show that every minimal acyclic forbidden minor for $\mathcal{F}_k$ (respectively $\mathcal{P}_k$) can be obtained from those for $\mathcal{F}_{k-1}$ (respectively $\mathcal{P}_{k-1}$) by a simple composition, and we list all 36 minimal forbidden minors for $\mathcal{P}_2$. Our proof contains many general methods that would be useful to characterize minimal forbidden minors for $\mathcal{P}_k$ ($k \geq 3$). We also give estimates for the numbers of minimal forbidden minors for $\mathcal{F}_k$ and $\mathcal{P}_k$, and the numbers of vertices of the largest minimal forbidden minors for $\mathcal{F}_k$ and $\mathcal{P}_k$.

3.2 Minimal Forbidden Minors for Graphs with Bounded Path-Width

3.2.1 Minimal Acyclic Forbidden Minors

We introduce the star-composition of graphs which plays an important role in the following.

Definition 3.1 *Let H_1, H_2, and H_3 be graphs. A graph obtained from H_1, H_2, and H_3 by the following construction is called a* star-composition *of H_1, H_2, and H_3:*

(i) *choose a vertex $v_i \in V(H_i)$ for $i = 1, 2,$ and 3;*

(ii) *let v be a new vertex not in $H_1, H_2,$ or H_3;*

(iii) *connect v to v_i by an edge (v, v_i) for $i = 1, 2,$ and 3.*

The vertex v is called the center *of the star-composition.* □

In this section, we prove the following theorem.

Theorem 3.1 *Let $k \geq 1$. A tree T is in $\Omega_a(\mathcal{F}_k)$ if and only if T is a star-composition of (not necessarily distinct) three trees in $\Omega_a(\mathcal{F}_{k-1})$.*

Proof: We prove this theorem by a series of lemmas.

Lemma 3.1 *If $k \geq 1$ and H_1, H_2, and H_3 are (not necessarily distinct) graphs in $\Omega(\mathcal{F}_{k-1})$ then any star-composition of H_1, H_2, and H_3 is in $\Omega(\mathcal{F}_k)$.*

Proof: Let G be a star-composition of H_1, H_2, and H_3, and v be the center of the star-composition. Since $H_i \in \Omega(\mathcal{F}_{k-1})$ ($i = 1, 2$, and 3), $G \backslash \{v\}$ has three connected components with path-width k. Thus $pw(G) \geq k+1$ by Lemma 2.9. On the other hand, $pw(G \backslash \{v\}) \leq k$ by Lemma 2.10, and so $pw(G) \leq k+1$ by Lemma 2.11. Hence we have $pw(G) = k+1$.

Next we show that G is minimal. Let $v_i \in V(H_i)$ be a vertex adjacent to v in G. Since $H_i \in \Omega(\mathcal{F}_{k-1})$, $pw(H_i \backslash \{v_i\}) = k-1$. Let $\mathcal{X}_{(i)}$ be a path-decomposition of $H_i \backslash \{v_i\}$ with width $k-1$ ($i = 1, 2$, and 3). It is sufficient to show that the path-width of a minor G' of G obtained by deleting or contracting an edge e is at most k.

Case 1. $e \in \{(v, v_1), (v, v_2), (v, v_3)\}$.

Without loss of generality we assume that $e = (v, v_1)$. If G' is obtained by deleting edge (v, v_1), the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \mathcal{X}_{(2)} \cup \{v_2\}, \{v, v_2\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a path-decomposition of G' and $pw(G') \leq k$. If G' is obtained by contracting edge (v, v_1), the sequence $(\mathcal{X}_{(2)} \cup \{v_2\}, \{v, v_2\}, \mathcal{X}_{(1)} \cup \{v\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a path-decomposition of G' and again $pw(G') \leq k$.

Case 2. $e \notin \{(v, v_1), (v, v_2), (v, v_3)\}$.

Without loss of generality we assume that $e \in E(H_2)$. G' is a star-composition of H_1, H_2', and H_3 where H_2' is a minor of H_2 obtained by deleting or contracting e. Let $\mathcal{X}'$ be a path-decomposition of H_2' with width $\leq k-1$. Then the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \{v, v_1\}, \mathcal{X}' \cup \{v\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a path-decomposition of G' and $pw(G') \leq k$.

Thus the path-width of any proper minor of G is at most k, and G is minimal.

Hence $G \in \Omega(\mathcal{F}_k)$. □

Corollary 3.1 *If $k \geq 1$ and T_1, T_2, and T_3 are (not necessarily distinct) trees in $\Omega_a(\mathcal{F}_{k-1})$ then any star-composition of T_1, T_2, and T_3 is in $\Omega_a(\mathcal{F}_k)$.*

Proof: Any star-composition of trees is also a tree. □

Lemma 3.2 *If $k \geq 1$ and T is any tree in $\Omega_a(\mathcal{F}_k)$ then T is a star-composition of some (not necessarily distinct) trees T_1, T_2, and T_3 in $\Omega_a(\mathcal{F}_{k-1})$.*

Proof: There exists a vertex v such that $T\backslash\{v\}$ has three or more connected components with path-width k or more by Theorem 2.2. Because T is minimal, $T\backslash\{v\}$ has exactly three connected components with path-width k. Let T_1, T_2, and T_3 be connected components of $T\backslash\{v\}$. Suppose $T_1 \notin \Omega_a(\mathcal{F}_{k-1})$. Let T_1' be a proper minor of T_1 with path-width k and T' be a star-composition of T_1', T_2, and T_3. Then $pw(T') = k + 1$, contradicting that $T \in \Omega_a(\mathcal{F}_k)$. Thus $T_1 \in \Omega_a(\mathcal{F}_{k-1})$. Similarly T_2 and T_3 are in $\Omega_a(\mathcal{F}_{k-1})$. □

By Corollary 3.1 and Lemma 3.2, we obtain Theorem 3.1. □

Theorem 3.1 obtained independently by Kinnersley [48].

It is easy to see that $\Omega(\mathcal{F}_0) = \{K_2\}$. The graphs in $\Omega(\mathcal{F}_1)$ and $\Omega_a(\mathcal{F}_2)$ are shown in Figs. 3.1 and 3.2, respectively.

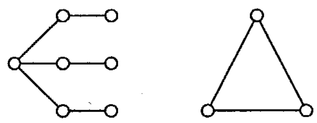


Figure 3.1: The graphs in $\Omega(\mathcal{F}_1)$.

The following corollaries can easily be proved by induction on k.

Corollary 3.2 *The number of vertices of a tree in $\Omega_a(\mathcal{F}_k)$ is $\frac{5\cdot 3^k - 1}{2}$ ($k \geq 0$).*

Corollary 3.3 $|\Omega_a(\mathcal{F}_k)| \geq k!^2$ *($k \geq 0$).*

We counted $|\Omega_a(\mathcal{F}_k)|$ for $k = 0, 1, 2, 3,$ and 4 as follows: $|\Omega_a(\mathcal{F}_0)| = |\Omega_a(\mathcal{F}_1)| = 1$, $|\Omega_a(\mathcal{F}_2)| = 10$, $|\Omega_a(\mathcal{F}_3)| = 117{,}480$, $|\Omega_a(\mathcal{F}_4)| = 14{,}403{,}197{,}619{,}396{,}707{,}660$.

3.3 Minimal Forbidden Minors for Graphs with Bounded Proper-Path-Width

3.3.1 Minimal Acyclic Forbidden Minors

We have the following lemma and theorem for $\mathcal{P}_k$ corresponding to Lemma 3.1 and Theorem 3.1, respectively. Proofs are almost same as those for $\mathcal{F}_k$.

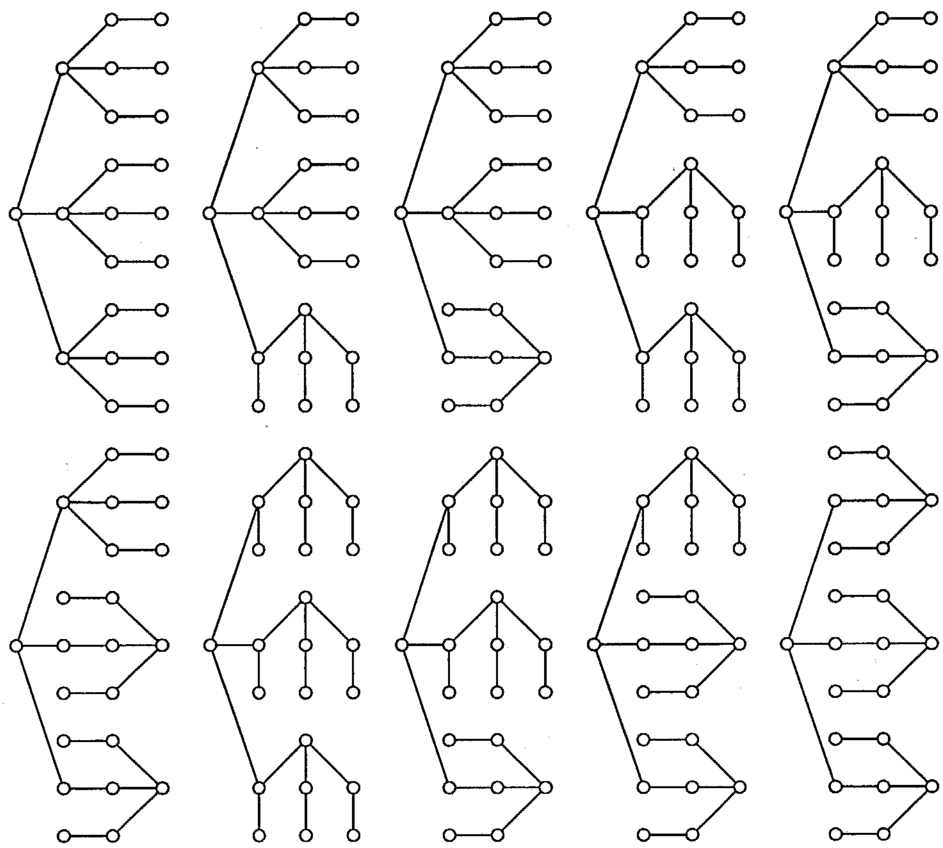


Figure 3.2: The trees in $\Omega_a(\mathcal{F}_2)$.

Lemma 3.3 *If $k \geq 2$ and H_1, H_2, and H_3 are (not necessarily distinct) graphs in $\Omega(\mathcal{P}_{k-1})$ then any star-composition of H_1, H_2, and H_3 is in $\Omega(\mathcal{P}_k)$.*

Proof: Let G be a star-composition of H_1, H_2, and H_3, and v be the center of the star-composition. Since $H_i \in \Omega(\mathcal{P}_{k-1})$ ($i = 1,2$, and 3), $G\backslash\{v\}$ has three connected components with proper-path-width k. Thus $ppw(G) \geq k+1$ by Lemma 2.19. On the other hand, $pw(G\backslash\{v\}) \leq k$ by Lemma 2.20, and so $pw(G) \leq k+1$ by Lemma 2.21. Hence we have $ppw(G) = k+1$.

Next we show that G is minimal. Let $v_i \in V(H_i)$ be a vertex adjacent to v in G. Since $H_i \in \Omega(\mathcal{P}_{k-1})$, $pw(H_i\backslash\{v_i\}) = k-1$. Let $\mathcal{X}_{(i)}$ be a proper-path-decomposition of $H_i\backslash\{v_i\}$ with width $k-1$ ($i = 1,2$, and 3). It is sufficient to show that the proper-path-width of a minor G' of G obtained by deleting or contracting an edge e is at most k.

Case 1. $e \in \{(v,v_1),(v,v_2),(v,v_3)\}$.

Without loss of generality we assume that $e = (v, v_1)$. If G' is obtained by deleting edge (v, v_1), the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \mathcal{X}_{(2)} \cup \{v_2\}, \{v, v_2\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a proper-path-decomposition of G' and $ppw(G') \leq k$. If G' is obtained by contracting edge (v, v_1), the sequence $(\mathcal{X}_{(2)} \cup \{v_2\}, \{v, v_2\}, \mathcal{X}_{(1)} \cup \{v\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a proper-path-decomposition of G' and again $ppw(G') \leq k$.

Case 2. $e \notin \{(v, v_1), (v, v_2), (v, v_3)\}$.

Without loss of generality we assume that $e \in E(H_2)$. G' is a star-composition of H_1, H_2', and H_3 where H_2' is a minor of H_2 obtained by deleting or contracting e. Let $\mathcal{X}'$ be a proper-path-decomposition of H_2' with width $\leq k-1$. Then the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \{v, v_1\}, \mathcal{X}' \cup \{v\}, \{v, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a proper-path-decomposition of G' and $ppw(G') \leq k$.

Thus the proper-path-width of any proper minor of G is at most k, and G is minimal.

Hence $G \in \Omega(\mathcal{P}_k)$. □

Theorem 3.2 *Let $k \geq 2$. A tree T is in $\Omega_a(\mathcal{P}_k)$ if and only if T is a star-composition of (not necessarily distinct) three trees in $\Omega_a(\mathcal{P}_{k-1})$.*

Proof: Let T be a tree in $\Omega_a(\mathcal{P}_k)$. There exists a vertex v such that $T\backslash\{v\}$ has three or more connected components with proper-path-width k or more by Theorem 2.3. Because T is minimal, $T\backslash\{v\}$ has exactly three connected components with proper-path-width k. Let T_1, T_2, and T_3 be connected components of $T\backslash\{v\}$. Suppose $T_1 \notin \Omega_a(\mathcal{P}_{k-1})$. Let T_1' be a proper minor of T_1 with proper-path-width k and T' be a star-composition of T_1', T_2, and T_3. Then $ppw(T') = k+1$, contradicting that $T \in \Omega_a(\mathcal{P}_k)$. Thus $T_1 \in \Omega_a(\mathcal{P}_{k-1})$. Similarly T_2 and T_3 are in $\Omega_a(\mathcal{F}_{k-1})$.

Converse follows from Lemma 3.3 □

It is easy to see that $\Omega(\mathcal{P}_1) = \{K_3, K_{1,3}\}$. The trees in $\Omega_a(\mathcal{P}_2)$ are shown in Fig 3.3.

Corollary 3.4 *The number of vertices of a tree in $\Omega_a(\mathcal{P}_k)$ is $\frac{3^{k+1}-1}{2}$ $(k \geq 1)$.*

Corollary 3.5 $|\Omega_a(\mathcal{P}_k)| \geq k!^2$ $(k \geq 1)$.

We counted $|\Omega_a(\mathcal{P}_k)|$ for $k = 1, 2, 3$, and 4 as follows: $|\Omega_a(\mathcal{P}_1)| = 1$, $|\Omega_a(\mathcal{P}_2)| = 4$, $|\Omega_a(\mathcal{P}_3)| = 1,330$, $|\Omega_a(\mathcal{P}_4)| = 2,875,919,312,080$.

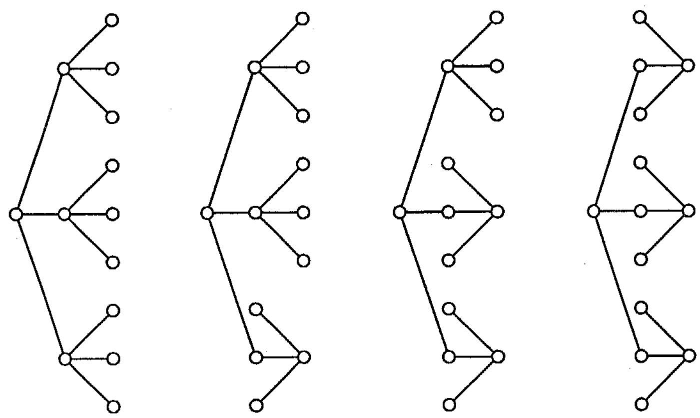


Figure 3.3: The trees in $\Omega_a(\mathcal{P}_2)$.

3.3.2 Delta Composition

Another kind of composition is possible for $\Omega(\mathcal{P}_k)$. That is, graphs in $\Omega(\mathcal{P}_k)$ can be obtained by the composition of minimal forbidden minors for $\mathcal{P}_{k-1}$.

Definition 3.2 *A* delta-composition *of graphs H_1, H_2, and H_3 is a graph obtained from H_1, H_2, and H_3 by the following construction:*

(i) choose a vertex $v_i \in V(H_i)$ for $i = 1, 2,$ and 3;

(ii) connect v_1 to v_2, v_2 to v_3, and v_3 to v_1 by edges (v_1, v_2), (v_2, v_3), and (v_3, v_1), respectively. □

Theorem 3.3 *If $k \geq 2$ and H_1, H_2, and H_3 are (not necessarily distinct) graphs in $\Omega(\mathcal{P}_{k-1})$ then any delta-composition of H_1, H_2, and H_3 is in $\Omega(\mathcal{P}_k)$.*

Proof: Let G be a delta-composition of H_1, H_2, and H_3. Let $v_i \in V(H_i)$ be the chosen vertex for $i = 1, 2,$ and 3. Because $H_i \in \Omega(\mathcal{P}_{k-1})$, $ppw(H_i) = k$ and $ppw(H_i \backslash \{v_i\}) = k - 1$. Let $\mathcal{X}_{(i)}$ be the proper-path-decomposition of $H_i \backslash \{v_i\}$ with width $k - 1$ ($i = 1, 2,$ and 3).

First, we show that $ppw(G) = k+1$. Suppose that $ppw(G) \leq k$ and G has a full proper-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ with width $\leq k$. Assume that $X_i \not\subseteq V(H_1)$ for any i ($1 \leq i \leq r$). Then $|X_i \cap V(H_1)| \leq k$ for any i ($1 \leq i \leq r$). Since H_1 is a connected subgraph of G, there exists an integer a such that $|X_a \cap V(H_1)| = k$ and $|X_{a-1} \cap X_{a+1} \cap V(H_1)| \geq k-1$ by Lemma 2.16. Since $G \backslash V(H_1)$ is a connected subgraph of G and each X_{a-1} and X_{a+1} contains a vertex of $G \backslash V(H_1)$, respectively, a vertex in $X_a - V(H_1)$

is contained both X_{a-1} and X_{a+1} by Lemma 2.17. Thus $|X_{a-1} \cap X_{a+1}| \geq k = |X_a| - 1$, and contradicting that $\mathcal{X}$ is a proper-path-decomposition. Thus there exists some i_1 such that $X_{i_1} \subseteq V(H_1)$. Similarly, there exist some i_2 and i_3 such that $X_{i_2} \subseteq V(H_2)$ and $X_{i_3} \subseteq V(H_3)$. Without loss of generality we assume that $i_2 < i_1 < i_3$. Then $X_{i_1} - V(H_1) = \emptyset$, and $X_{i_2} - V(H_1) \neq \emptyset$ and $X_{i_3} - V(H_1) \neq \emptyset$, contradicting Lemma 2.17. Hence we have $ppw(G) \geq k+1$. It is easy to see that the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \mathcal{X}_{(2)} \cup \{v_1, v_2\}, \{v_1, v_2, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a proper-path-decomposition of G, and $ppw(G) = k+1$.

Next we show that G is minimal. It is sufficient to show that the proper-path-width of a minor G' of G obtained by deleting or contracting an edge e is at most k.

Case 1. $e \in \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$.

Without loss of generality we assume that $e = (v_1, v_2)$. If G' is obtained by deleting edge (v_1, v_2), the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \{v_1, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\}, \{v_3, v_2\}, \mathcal{X}_{(2)} \cup \{v_2\})$ is a proper-path-decomposition of G' and $ppw(G') \leq k$. If G' is obtained by contracting edge (v_1, v_2), the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \mathcal{X}_{(2)} \cup \{v_1\}, \{v_1, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a proper-path-decomposition of G' and again $ppw(G') \leq k$.

Case 2. $e \notin \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$.

Without loss of generality we assume that $e \in E(H_2)$. G' is a delta-composition of H_1, H_2', and H_3 where H_2' is a minor of H_2 obtained by deleting or contracting e. Let $\mathcal{X}^* = (X_1^*, X_2^*, \ldots, X_{r^*}^*)$ of H_2' be a full proper-path-decomposition of H_2'. If the width of $\mathcal{X}^*$ is at most $k-2$ then the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \mathcal{X}^* \cup \{v_1, v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a k-proper-path-decomposition of G'. Thus we assume that the width of $\mathcal{X}^*$ is $k-1$. Suppose that $v_2 \in X_1^*$. Then the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, \{v_1, v_2, v_3\}, \mathcal{X}^* \cup \{v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a k-proper-path-decomposition of G'. Similarly, there exists a k-proper-path-decomposition of G' if $v_2 \in X_{r^*}^*$. Thus we suppose that $v_2 \in X_i^*$ $(1 < i < r^*)$. If $v_2 \in X_i^*$ and $v_2 \notin X_{i-1}^* \cup X_{i+1}^*$ then $X_i^* - \{v_2\} = X_{i-1}^* \cap X_i^* = X_i^* \cap X_{i+1}^*$ and $|X_{i-1}^* \cap X_{i+1}^*| = |X_i^*| - 1$ This contradict the assumption that $\mathcal{X}^*$ is a proper-path-decomposition. Thus, without loss of generality, we assume that $v_2 \in X_{i+1}^*$. Then the sequence $(\mathcal{X}_{(1)} \cup \{v_1\}, X_1^* \cup \{v_1\}, \ldots, X_i^* \cup \{v_1\}, (X_i^* \cap X_{i+1}^*) \cup \{v_1, v_3\}, X_{i+1}^* \cup \{v_3\}, \ldots, X_{r^*}^* \cup \{v_3\}, \mathcal{X}_{(3)} \cup \{v_3\})$ is a k-proper-path-decomposition of G'. Hence $ppw(G') \leq k$.

Thus the proper-path-width of any proper minor of G is at most k, and $G \in \Omega(\mathcal{P}_k)$. □

Notice that the above theorem does not hold for $\Omega(\mathcal{F}_k)$. Although a graph shown in Fig. 3.4 is a delta-composition of graphs in $\Omega(\mathcal{F}_1)$, it is not in $\Omega(\mathcal{F}_2)$ because its minor shown in Fig. 3.5 is in $\Omega(\mathcal{F}_2)$. Notice also that the star- and delta-compositions are not sufficient to characterize minimal forbidden minors for $\mathcal{P}_k$. A graph in $\Omega(\mathcal{P}_2)$ shown in Fig. 3.6 is neither a star-composition nor a delta-composition of graphs in $\Omega(\mathcal{P}_1)$.

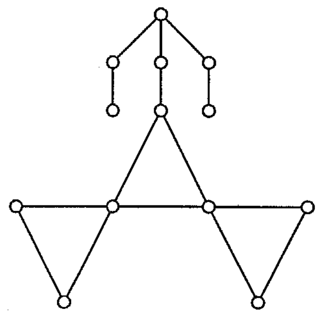


Figure 3.4: A graph not in $\Omega(\mathcal{F}_2)$ that is a delta-composition of minors in $\Omega(\mathcal{F}_1)$.

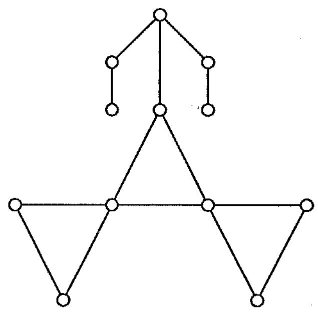


Figure 3.5: A graph in $\Omega(\mathcal{F}_2)$.

3.3.3 Equivalence Relations on Graphs

In this section, we introduce two equivalence relations on graphs such that a graph equivalent to a minimal forbidden minor for $\mathcal{P}_k$ is also a minimal forbidden minor for $\mathcal{P}_k$.

Lemma 3.4 *Let G be a graph and G' be an underlying simple graph of G. Then $ppw(G) = ppw(G')$.*

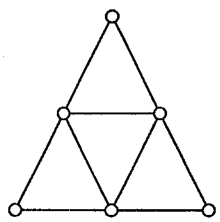


Figure 3.6: A graph in $\Omega(\mathcal{P}_2)$.

Proof: Since G' is a minor of G and a proper-path-decomposition of G' is also a proper-path-decomposition of G, $ppw(G) = ppw(G')$. □

For any vertex $v \in V(G)$, $d_G(v)$ is the number of vertices adjacent to v. If an edge (u, v) is contracted from a graph, the vertex obtained by identifying u and v is denoted by u or v.

Lemma 3.5 *Let G be a graph satisfying the following: there exist vertices $u, v \in V(G)$ such that $(u, v) \in E(G)$, $d_G(u) = 2$, and $d_G(v) = 1$. If G' is a graph obtained from G by contracting edge (u, v), then $ppw(G) = ppw(G')$.*

Proof: Since G' is a minor of G, $ppw(G) \geq ppw(G')$. We will show that $ppw(G) \leq ppw(G')$. Assume that $ppw(G') = k$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a k-proper-path-decomposition of G', and w be a unique vertex adjacent to u in G'. By Lemma 2.14, we assume that $\mathcal{X}$ is full. By Definition 2.2(iii), there exists an integer a such that $\{u, w\} \subseteq X_a$. If $\{u, w\} \subseteq X_1$, then $((X_1 - \{w\}) \cup \{v\}, \mathcal{X})$ is a k-proper-path-decomposition of G, and $ppw(G) \leq k$. Notice that $(X_r, X_{r-1}, \ldots, X_1)$ is also a k-proper-path-decomposition of G. Thus, we assume that $\{u, w\} \subseteq X_a$ $(1 < a < r)$. Suppose that $u \notin X_{a-1} \cup X_{a+1}$. Since $\mathcal{X}$ is full, $X_{a-1} \cap X_a = X_a \cap X_{a+1} = X_a - \{u\}$. However, we have $|X_a| \geq |X_{a-1} \cap X_{a+1}| + 2 = k+2$ by Definition 2.2(v), contradicting to the assumption that the width of $\mathcal{X}$ is k. Thus we assume, without loss of generality, that $u \in X_{a+1}$. For any integer i $(a+1 \leq i \leq r)$, let $X'_i = (X_i - \{u\}) \cup \{v\}$ if $u \in X_i$, $X'_i = X_i$ otherwise. Then $(X_1, \ldots, X_a, (X_a \cap X_{a+1}) \cup \{v\}, X'_{a+1}, \ldots, X'_r)$ is a k-proper-path-decomposition of G. Thus $ppw(G) \leq ppw(G')$. □

Lemma 3.6 *Let G be a graph satisfying the following: there exist vertices $u, v \in V(G)$ such that $(u, v) \in E(G)$, $d_G(u) = d_G(v) = 2$, and u and v have no common adjacent vertex. If G' is a graph obtained from G by contracting edge (u, v), then $ppw(G) = ppw(G')$.*

Proof: Since G' is a minor of G, $ppw(G) \geq ppw(G')$. We will show that $ppw(G) \leq ppw(G')$. Assume that $ppw(G') = k$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a full k-proper-path-decomposition of G', and x and y be vertices adjacent to u in G'. If $\{u, x, y\} \subseteq X_1$, then $((X_1 - \{x\}) \cup \{v\}, \mathcal{X})$ is a k-proper-path-decomposition of G, and $ppw(G) \leq k$. If $\{u, x, y\} \subseteq X_i$ $(1 < i < r)$ then we can assume that $\{u, y\} \subseteq X_{i+1}$ since $\mathcal{X}$ is full. Thus we assume that there exist distinct integers a and b such that $\{u, x\} \subseteq X_a, \{u, y\} \subseteq X_b$ by Definition 2.2(iii) $(1 \leq a < b \leq r)$. For any integer i $(a+1 \leq i \leq r)$, let $X_i' = (X_i - \{u\}) \cup \{v\}$ if $u \in X_i$, $X_i' = X_i$ otherwise. Then $(X_1, \ldots, X_a, (X_a \cap X_{a+1}) \cup \{v\}, X_{a+1}', \ldots, X_r')$ is a k-proper-path-decomposition of G. Thus $ppw(G) \leq ppw(G')$. □

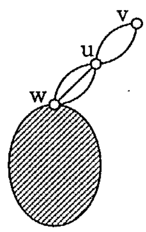


Figure 3.7: An example of graph G such that $(u, v) \in E(G)$, $d_G(u) = 2$, and $d_G(v) = 1$.

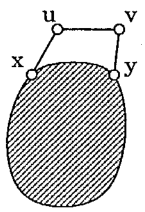


Figure 3.8: An example of graph G such that $(u, v) \in E(G)$ and $d_G(u) = d_G(v) = 2$.

Notice that a minimal forbidden minor for $\mathcal{P}_k$ is connected, since the proper-path-width of a graph G is the maximum proper-path-width over all connected components of G. From Lemmas 3.4, 3.5, and 3.6, we have the following theorem.

Theorem 3.4 *If a graph G is in $\Omega(\mathcal{P}_k)$ then G is connected and simple, and there are no adjacent vertices u and v such that $d_G(v) \leq 2$, $d_G(u) \leq 2$, and they have no common adjacent vertex.*

Lemma 3.7 *Let G be a connected graph and v be a vertex of G. Let H_i be a connected graph and u_i be a vertex of H_i $(1 \leq i \leq 2)$. Define that G_i be the graph obtained from G and H_i by identifying v and u_i $(1 \leq i \leq 2)$. If $H_1, H_2 \in \Omega(\mathcal{P}_k)$ then $ppw(G_1) = ppw(G_2)$.*

Proof: Assume that $ppw(G_1) = h$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a h-proper-path-decomposition of G_1. Let a and b be integers such that $v \in X_a \cap X_b$ and $v \notin X_{a-1} \cup X_{b+1}$ $(a \leq b)$. Let $X_i^H = X_i \cap V(H_1)$ and $X_i^G = X_i - V(H_1)$ for any i $(1 \leq i \leq r)$. From Lemma 2.16, there exists an integer i such that either (a) $|X_i^H| \geq k+2$ $(1 \leq i \leq r)$, or (b) $|X_i^H| = k+1$ and $|X_{i-1}^H \cap X_{i+1}^H| \geq k$ $(1 < i < r)$. Let c be such an integer. Let $X_i' = X_i^G \cup \{v\}$ if $\min\{a,c\} \leq i \leq \max\{b,c\}$, $X_i' = X_i^G$ otherwise. Since H_1 is connected, $|X_i'| \leq h+1$ by Lemma 2.17 $(1 \leq i \leq r)$. Since $H_2 \in \Omega(\mathcal{P}_k)$, $ppw(H_2 \backslash \{u_2\}) = k$. Let $\mathcal{X}^*$ be a k-proper-path-decomposition of $H_2 \backslash \{u_2\}$.

If $|X_c^H| \geq k+2$ then $(X_1', \ldots, X_{c-1}', X_c' \cup \mathcal{X}^*, X_{c+1}', \ldots, X_r')$ is a sequence with width h satisfying the conditions of Lemma 2.15, and $ppw(G_2) \leq h$. Notice that $|X_c'| = |X_c| - |X_c^H| + 1 \leq h-k$. Thus we assume that $|X_c^H| = k+1$ and $|X_c^G| \leq h-k$. If $|X_{c-1}^G \cap X_c^G| = |X_c^G \cap X_{c+1}^G| = h-k$ then $X_{c-1}^G \cap X_{c+1}^G = X_c^G$. We have that $|X_{c-1} \cap X_{c+1}| = |X_{c-1}^G \cap X_{c+1}^G| + |X_{c-1}^H \cap X_{c+1}^H| = (h-k)+k$. However, we have $|X_c| \geq h+2$ by Definition 2.2(v), contradicting to the assumption that the width of $\mathcal{X}$ is h. Thus, we assume, without loss of generality, $|X_c^G \cap X_{c+1}^G| \leq h-k-1$. Then, $(X_1', \ldots, X_c', (X_c' \cap X_{c+1}') \cup \{v\} \cup \mathcal{X}^*, X_{c+1}', \ldots, X_r')$ is a sequence with width h satisfying the conditions of Lemma 2.15, and $ppw(G_2) \leq h$.

Therefore, we have $ppw(G_2) \leq h = ppw(G_1)$. Similarly, we can prove that $ppw(G_1) \leq ppw(G_2)$, and we have $ppw(G_1) = ppw(G_2)$. □

The following definition plays an important role to characterize $\Omega(\mathcal{P}_k)$.

Definition 3.3 *Let H be a graph and x be a vertex of H. $R_1^1(H,x)$ is the graph obtained from H by adding vertices u, v, and w and edges (x,u), (x,v), and (x,w). $R_2^1(H,x)$ is the graph obtained from $R_1^1(H,x)$ by adding edge (u,w) and deleting v. $R_3^1(H,x)$ is the graph obtained from $R_1^1(H,x)$ by adding edges (u,v) and (v,w) and deleting edges (x,u) and (x,w). Graphs G_1 and G_2 are said to be* semi-1-equivalent, *denoted by $G_1 \overset{1}{\cong} G_2$, if there exist a graph H and a vertex $x \in V(H)$ such that $G_1 = R_i^1(H,x)$ $(1 \leq i \leq 3)$ and $G_2 = R_j^1(H,x)$ $(1 \leq j \leq 3)$. Graphs G_1 and G_i are said to be* 1-equivalent *if there exists a sequence of graphs $G_1, G_2, \ldots, G_i$ $(i \geq 1)$ such that $G_1 \overset{1}{\cong} G_2 \overset{1}{\cong} \cdots \overset{1}{\cong} G_i$.*

It is easy to see that the 1-equivalence is an equivalence relation on the graphs. An example of 1-equivalent graphs is shown in Fig. 3.9.

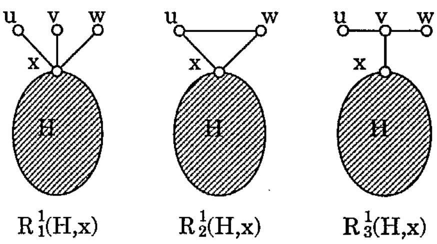


Figure 3.9: 1-equivalent graphs.

Theorem 3.5 *If $G \in \Omega(\mathcal{P}_k)$ then a graph 1-equivalent to G is in $\Omega(\mathcal{P}_k)$.*

Proof: Let H be a graph and $x \in V(H)$. Let $R_i = R^1_i(H,x)$ $(1 \leq i \leq 3)$. Assume that $R_a \in \Omega(\mathcal{P}_k)$ for some a $(1 \leq a \leq 3)$. To prove the lemma, it is sufficient to show that $R_i \in \Omega(\mathcal{P}_k)$ for any i $(1 \leq i \leq 3)$.

Let $H_i = R_i \backslash (V(H) - \{x\})$ $(1 \leq i \leq 3)$. Since H_1, H_2, $H_3 \in \Omega(\mathcal{P}_1)$, $ppw(R_1) = ppw(R_2) = ppw(R_3) = k+1$ by Lemma 3.7. Thus it is sufficient to show that R_i is minimal for any i, that is, the proper-path-width of a graph obtained from R_i by deleting or contracting an edge is at most k. Let G'_i be a graph obtained from R_i by deleting or contracting an edge $e \in E(H)$. We have $ppw(G'_a) = ppw(G'_i) = k$ for any i $(1 \leq i \leq 3)$ by Lemma 3.7. Let G''_i be a graph obtained from R_i by deleting or contracting an edge $e \notin E(H)$. Proper-path-widths of connected components of G''_i not containing x are at most one $(\leq k)$. A connected component of G''_i containing x is isomorphic to either graphs G^*_1, G^*_2, G^*_3, or G^*_4 obtained from R_2 by deleting edge (x,u), contracting edge (x,u), deleting edge (u,w), or deleting vertices u and w, respectively (See Fig. 3.10). Since both G^*_3 and G^*_4 are proper minors of R_a, $ppw(G^*_3) \leq k$ and $ppw(G^*_4) \leq k$. Since the underlying simple graph of G^*_2 is a proper minor of G^*_3, $ppw(G^*_2) \leq ppw(G^*_3) \leq k$ by Lemma 3.4. By Lemma 3.5, we have $ppw(G^*_1) = ppw(G^*_2)$. Thus $ppw(G''_i) \leq k$ for any i $(1 \leq i \leq 3)$. Hence we have $R_i \in \Omega(\mathcal{P}_k)$ $(1 \leq i \leq 3)$. □

The following definition also plays an important role to characterize $\Omega(\mathcal{P}_k)$.

Definition 3.4 *Let H be a graph, and x and y be non-adjacent vertices of H. $R^2_1(H,x,y)$ is the graph obtained from H by adding vertices u and v and edges (x,u), (y,u), and (u,v). $R^2_2(H,x,y)$ is the graph obtained from $R^2_1(H,x,y)$ by adding edge (x,y) and deleting v. Graphs G_1 and G_2 are said to be semi-2-equivalent, denoted by $G_1 \overset{2}{\cong} G_2$, if there exist a graph H and non-adjacent vertices $x \in V(H)$ and $y \in V(H)$ such that $G_1 = R^2_i(H,x,y)$*

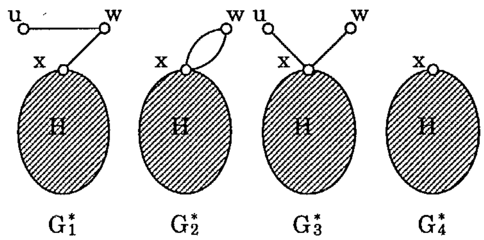


Figure 3.10: Connected components of G_i'' containing x.

$(i = 1,2)$, $G_2 = R_j^2(H,x,y)$ $(j = 1,2)$. *Graphs* G_1 *and* G_i *are said to be 2-equivalent if there exists a sequence of graphs* $G_1, G_2, \ldots, G_i$ $(i \geq 1)$ *such that* $G_1 \overset{2}{\cong} G_2 \overset{2}{\cong} \cdots \overset{2}{\cong} G_i$.

It is easy to see that the 2-equivalence is an equivalence relation on the graphs. An example of 2-equivalent graphs is shown in Fig. 3.11.

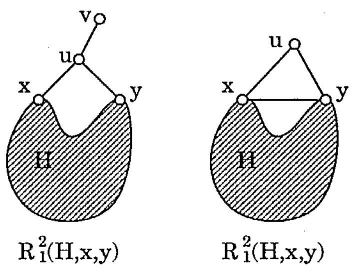


Figure 3.11: 2-equivalent graphs.

Lemma 3.8 *If a graph G is 2-equivalent to G′ then* $ppw(G) = ppw(G')$.

Proof: Let H be a graph and $x, y \in V(H)$ be non-adjacent vertices. Let $R_i = R_i^2(H,x,y)$ $(1 \leq i \leq 2)$. To prove the lemma, it is sufficient to show that $ppw(R_1) = ppw(R_2)$.

First, assume that $ppw(R_1) = k$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a full k-proper-path-decomposition of R_1. Let $X_i' = X_i - \{u, v\}$. If $\{u, x, y\} \subseteq X_i$ for some i $(1 \leq i \leq r)$, $\mathcal{X}$ is a k-proper-path-decomposition of a graph Q obtained from R_1 by adding edge (x, y). Since R_2 is a minor of Q, $ppw(R_2) \leq k$. Thus we assume that $\{u, x, y\} \not\subseteq X_i$ for any i $(1 \leq i \leq r)$. Suppose that $\{u, x\} \in X_{a-1} \cap X_a$ and $\{u, y\} \in X_b$ for some a and b $(a < b)$. Notice that $|X_{a-1}' \cap X_a'| \leq k - 1$. Since $(X_1', \ldots, X_{a-1}', (X_{a-1}' \cap X_a') \cup \{y, u\}, X_a' \cup$

$\{y\},\ldots,X'_{b-1}\cup\{y\},X'_b,\ldots,X'_r)$ is a sequence satisfying the conditions of Lemma 2.15, $ppw(R_2)\le k$. Thus we assume that $\{u,x\}\subseteq X_a$, $\{u,y\}\subseteq X_b$, $x\notin X_{a+1}$, $y\notin X_{b-1}$, and $u\notin X_{a-1}\cup X_{b+1}$ for some a and b $(1\le a<b\le r)$. There exists an integer c such that $\{u,v\}\subseteq X_c$ $(a\le c\le b)$. Moreover, since $\mathcal{X}$ is full, we assume that $\{u,v\}\subseteq X_c\cap X_{c+1}$ for some c $(a\le c<b)$. Notice that $|X'_c\cap X'_{c+1}|\le k-2$. Since $(X'_1,\ldots,X'_a,X'_{a+1}\cup\{x\},\ldots,X'_c\cup\{x\},(X'_c\cap X'_{c+1})\cup\{x,y,u\},X'_{c+1}\cup\{y\},\ldots,X'_{b-1}\cup\{y\},X'_b,\ldots,X'_r)$ is a sequence satisfying the conditions of Lemma 2.15, $ppw(R_2)\le k$. Thus we have that $ppw(R_2)\le ppw(R_1)$.

Next, assume that $ppw(R_2)=k$ and let $\mathcal{X}=(X_1,X_2,\ldots,X_r)$ be a full k-proper-path-decomposition of R_2. There exists an integer a such that $\{u,x,y\}\subseteq X_a$. If $\{u,x,y\}\subseteq X_1$, then $((X_1-\{x\})\cup\{v\},\mathcal{X})$ is a k-proper-path-decomposition of R_1, and $ppw(R_1)\le k$. If $\{u,x,y\}\subseteq X_a$ $(1<a<r)$ then we assume without loss of generality that $\{u,x\}\subseteq X_{a+1}$ since $\mathcal{X}$ is full. For any integer i $(a+1\le i\le r)$, let $X'_i=(X_i-\{u\})\cup\{v\}$ if $u\in X_i$, $X'_i=X_i$ otherwise. Then $(X_1,\ldots,X_a,(X_a\cap X_{a+1})\cup\{v\},X'_{a+1},\ldots,X'_r)$ is a k-proper-path-decomposition of R_1. Thus $ppw(R_1)\le ppw(R_2)$.

Therefore we have $ppw(R_1)=ppw(R_2)$. □

Theorem 3.6 *If $G\in\Omega(\mathcal{P}_k)$ then a graph 2-equivalent to G is in $\Omega(\mathcal{P}_k)$.*

Proof: Let H be a graph and $x,y\in V(H)$ be non-adjacent vertices. Let $R_i=R_i^2(H,x,y)$ $(1\le i\le 2)$. It is sufficient to show that $R_1\in\Omega(\mathcal{P}_k)$ if and only if $R_2\in\Omega(\mathcal{P}_k)$.

First, assume that $R_1\in\Omega(\mathcal{P}_k)$. By Lemma 3.8, we have $ppw(R_2)=ppw(R_1)=k+1$. It is sufficient to show that R_2 is minimal, that is, the proper-path-width of a graph obtained from R_2 by deleting or contracting an edge is at most k. Let G'_1 and G'_2 be graphs obtained from R_1 and R_2 by deleting or contracting an edge $e\in E(H)$, respectively. We have $ppw(G'_2)=ppw(G'_1)=k$ by Lemma 3.8. Let G''_2 be a graph obtained from R_2 by deleting or contracting an edge $e\in\{(x,u),(y,u),(x,y)\}$. Since either G''_2 or the underlying simple graph of G''_2 is a proper minor of R_1, $ppw(G''_2)\le k$. Thus we have $R_2\in\Omega(\mathcal{P}_k)$.

Next, assume that $R_2\in\Omega(\mathcal{P}_k)$. Similar to argument above, we have $ppw(R_1)=ppw(R_2)=k+1$, and the proper-path-width of a graph obtained from R_1 by deleting or contracting an edge $e\in E(H)$ is k. Let G''_1 be a graph obtained from R_1 by deleting or contracting an edge $e\in\{(x,u),(y,u),(u,v)\}$. Proper-path-widths of connected components of G''_1 not containing x is at most one. A connected component of G''_1 containing x is isomorphic to either of graphs G^*_1, G^*_2, or G^*_3 obtained from R_1 by contracting (x,u), deleting (x,u), or contracting (u,v), respectively (See Fig. 3.12). Let G^*_4 be a graph obtained from G^*_2 by contracting edge (u,v). We have $ppw(G^*_2)=ppw(G^*_4)$ by Lemma 3.5.

Since both G_1^*, G_3^*, and G_4^* are proper minors of R_2, proper-path-widths of these graphs are at most k. Then the proper-path-width of G_1'' is at most k. Thus we have $R_1 \in \Omega(\mathcal{P}_k)$.

□

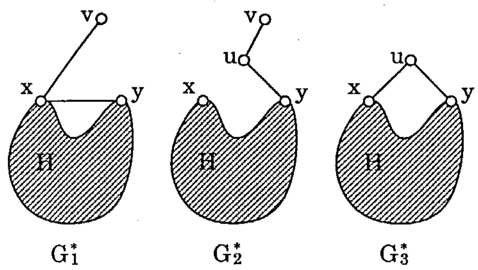


Figure 3.12: Connected components of G_1'' containing x.

3.3.4 Graphs with Cut-Vertices or Bridges

In this section, we show necessary conditions for a graph obtained from two disjoint graphs with proper-path-width k by either identifying some vertex or adding an edge to be a minimal forbidden minor for $\mathcal{P}_k$ ($k \geq 1$). A vertex v is called a cut-vertex of a connected graph G if $G \backslash \{v\}$ has at least two connected components. An edge (u, v) is called a bridge of a connected graph G if the graph obtained from G by deleting (u, v) has at least two connected components.

Lemma 3.9 *Let G be a connected graph with a cut-vertex v. Let H be a union of connected components of $G\backslash\{v\}$, H_1 be the induced subgraph of G on $V(H) \cup \{v\}$, and H_2 be the induced subgraph of G on $V(G) - V(H)$. If $ppw(G) = ppw(H_1) = ppw(H_2) = k$ and there exists at most one connected component of H with proper-path-width k, then there exists a k-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ of H_1 such that $v \in X_1$.*

Proof: Assume that the proper-path-width of each connected component of H is at most $k-1$. Since $ppw(H) = k - 1$, there exists a $(k-1)$-proper-path-decomposition $\mathcal{X}$ of H. Then $(\mathcal{X} \cup \{v\})$ is a k-proper-path-decomposition of H_1 satisfying the condition of this lemma. Thus we assume that there exists a connected component of H with proper-path-width k.

Let H' be a connected component of H with proper-path-width k and G' be the induced subgraph of G on $V(H') \cup V(H_2)$. Since H_2 is a minor of G' and G' is a minor of G, we have

$ppw(G') = k$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a k-proper-path-decomposition of G'. From Lemma 2.16, there exists an integer c such that either (a) $|X_c \cap V(H_2)| \geq k+1$ $(1 \leq c \leq r)$, or (b) $|X_c \cap V(H_2)| = k$ and $|X_{c-1} \cap X_{c+1} \cap V(H_2)| \geq k-1$ $(1 < c < r)$. Suppose that there exist integers a and b such that $X_a \cap V(H') \neq \emptyset$, $X_b \cap V(H') \neq \emptyset$ $(a < c < b)$. Since H' is connected, $|X_c \cap V(H')| \geq 1$ by Lemma 2.17. If $|X_c \cap V(H_2)| \geq k+1$ then $|X_c| \geq |X_c \cap V(H_2)| + |X_c \cap V(H')| \geq k+2$, contradicting to that the width of $\mathcal{X}$ is k. Thus we suppose that $|X_c \cap V(H_2)| = k$ and $|X_c \cap V(H')| = 1$. However, we have $|X_{c-1} \cap X_{c+1}| \geq k$ since $|X_{c-1} \cap X_{c+1} \cap V(H')| \geq 1$ by Lemma 2.17 and $|X_{c-1} \cap X_{c+1} \cap V(H_2)| \geq k-1$. Again this is contradicting to that the width of $\mathcal{X}$ is k since $|X_c| \geq |X_{c-1} \cap X_{c+1}| + 2 \geq k+2$. Thus, without loss of generality, we assume that $\bigcup_{i=1}^{c-1} X_i \cap V(H') = \emptyset$. Let d be the integer such that $v \in X_d$ and $v \notin X_{d+1}$. Notice that $c \leq d$ since there exists a vertex $v' \in V(H')$ such that $(v, v') \in E(G')$. Let $X_i' = (X_i \cap V(H')) \cup \{v\}$ for any integer i $(c \leq i \leq d)$, and $X_i' = X_i \cap V(H')$ for any integer i $(d < i)$. Notice that $|X_i'| \leq k+1$ since H_2 is connected.

Let H^* be the induced subgraph of H_1 on $V(H') \cup \{v\}$. Then $\mathcal{X}' = (X_c', X_{c+1}', \ldots, X_r')$ is a sequence of subsets of $V(H^*)$ with width k satisfying the conditions of Lemma 2.15. Since the proper-path-width of $H_1 \backslash V(H^*)$ is at most $k-1$, there exists a $(k-1)$-proper-path-decomposition $\mathcal{X}''$ of $H_1 \backslash V(H_1^*)$. Then $(\mathcal{X}'' \cup \{v\}, \mathcal{X}')$ is a sequence of subsets of $V(H_1)$ with width k satisfying the conditions of Lemma 2.15. It is not difficult to see that we can obtain a k-proper-path-decomposition of H_1 satisfying the condition of this lemma from the sequence $(\mathcal{X}'' \cup \{v\}, \mathcal{X}')$. □

Theorem 3.7 *Let G be a connected graph with a cut-vertex v. Let H be a union of connected components of $G \backslash \{v\}$, H_1 be the induced subgraph of G on $V(H) \cup \{v\}$, and H_2 be the induced subgraph of G on $V(G) - V(H)$. If $ppw(H_1) = ppw(H_2) = k$ and $G \in \Omega(\mathcal{P}_k)$, then either G is a star-composition of (not necessarily distinct) graphs in $\Omega(\mathcal{P}_{k-1})$, or $H_1 \in \Omega(\mathcal{P}_{k-1})$, or $H_2 \in \Omega(\mathcal{P}_{k-1})$.*

Proof: Assume contrary that G is not a star-composition of three graphs in $\Omega(\mathcal{P}_{k-1})$, $H_1 \notin \Omega(\mathcal{P}_{k-1})$, and $H_2 \notin \Omega(\mathcal{P}_{k-1})$. Notice that $G \backslash \{v\}$ has at most two connected components with proper-path-width k, for otherwise, G is not minimal.

Assume first that $H_1 \backslash \{v\}$ and $H_2 \backslash \{v\}$ have at most one connected component with proper-path-width k, respectively. Let H' be a proper minor of H_2 with proper-path-width k. Let G' be the minor of G such that $V(G') = V(H_1) \cup V(H')$ and $E(G') = E(H_1) \cup E(H')$. Since $G \in \Omega(\mathcal{P}_k)$, we have $ppw(G') = k$. By Lemma 3.9, there exists a k-proper-path-decomposition $\mathcal{X}^1 = (X_1^1, X_2^1, \ldots, X_{r_1}^1)$ of H_1 such that $v \in X_{r_1}^1$. Similarly, there exists a k-proper-path-decomposition $\mathcal{X}^2 = (X_1^2, X_2^2, \ldots, X_{r_2}^2)$ of H_2 such that $v \in X_1^2$. Then $(\mathcal{X}^1, \mathcal{X}^2)$ is a k-proper-path-decomposition of G, contradicting to that $ppw(G) \geq k+1$.

Assume next that $H_1\backslash\{v\}$ has exactly two connected components with proper-path-width k. Let R_1 and R_2 be connected components of $H_1\backslash\{v\}$ with proper-path-width k, $H_a = H_1\backslash V(R_2)$, and H_b be the induced subgraph of G on $V(H_2) \cup V(R_2)$. Note that H_a and H_b are connected, $V(H_a) \cap V(H_b) = \{v\}$, and $E(H_a) \cup E(H_b) = E(G)$. Since $H_a\backslash\{v\}$ and $H_b\backslash\{v\}$ have exactly one connected components with proper-path-width k, respectively, $ppw(H_a) = ppw(H_b) = k$, and H_a and H_b are not contained in $\Omega(\mathcal{P}_{k-1})$. Thus, by similar argument, there exists a k-proper-path-decomposition of G, contradicting to that $ppw(G) \geq k+1$. □

Theorem 3.8 *Let G be a connected graph with a bridge (u, v). Let H_1 be the connected component containing u of the graph obtained from G by deleting (u, v), H_2 be the other component. If $ppw(H_1) = ppw(H_2) = k$ and $G \in \Omega(\mathcal{P}_k)$, then G is a star-composition of (not necessarily distinct) graphs in $\Omega(\mathcal{P}_{k-1})$.*

Proof: Assume contrary that G is not a star-composition of three graphs in $\Omega(\mathcal{P}_{k-1})$. Since G is minimal, $H_1\backslash\{u\}$ and $H_2\backslash\{v\}$ have at most one connected component with proper-path-width at most k, respectively. Let G' be the graph obtained from G by contracting edge (u, v). Then $ppw(G') = k$, and graphs G', H_1, and H_2 satisfy the condition of Lemma 3.9. Thus there exist k-proper-path-decompositions $\mathcal{X}^1 = (X_1^1, X_2^1, \ldots, X_{r_1}^1)$ of H_1 and $\mathcal{X}^2 = (X_1^2, X_2^2, \ldots, X_{r_2}^2)$ of H_2 such that $u \in X_{r_1}^1$ and $v \in X_1^2$. Then $(\mathcal{X}^1, \{u, v\}, \mathcal{X}^2)$ is a k-proper-path-decomposition of G, contradicting to that $ppw(G) = k+1$. □

3.3.5 Minimal Forbidden Minors for Graphs with Proper-Path-Width at Most Two

In this section, we characterize the minimal forbidden minors for $\mathcal{P}_2$.

Let $P(a_0 a_1 \cdots a_{n-1})$ be the graph obtained from a cycle with vertices $\{v_0, v_1, \ldots, v_{n-1}\}$ and edges $\{(v_i, v_{(i+1) \bmod n}) | 0 \leq i \leq n-1\}$ by adding a_i vertices and connecting them with v_i by edges for all i $(0 \leq i \leq n-1)$.

Theorem 3.9 *A graph isomorphic to either $P(333)$, $P(3202)$, $P(2221)$, $P(22010)$, $P(11111)$, $P(101010)$, K_4, or $K_{2,3}$ (see Fig. 3.13) is a minimal forbidden minor for $\mathcal{P}_2$.*

Proof: Let G be a graph isomorphic to $P(11111)$. Let $u_0, u_1, \ldots, u_4$ be vertices of G such that $(u_i, u_{(i+1)\bmod 5}) \in E(G)$, and $v_0, v_1, \ldots, v_4$ be vertices of G such that $(v_i, u_i) \in E(G)$ $(0 \leq i \leq 4)$. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a proper-path-decomposition of G. Let s_i be an integer such that $v_i, u_i \in X_{s_i}$ $(0 \leq i \leq 4)$. Let $t_0, t_1, \ldots, t_4$ be a permutation of 0, 1, 2, 3, and 4 such that $s_{t_0} \leq s_{t_1} \leq \cdots \leq s_{t_4}$. Without loss of generality, we assume that $t_2 = 0$ and (t_0, t_1) is a combination of either $(1,2)$, $(1,3)$, or $(1,4)$. Assume first that (t_0, t_1) is a combination of $(1,2)$. Let H, G_1 and G_2 be induced subgraphs of G on $\{v_0\}$, $\{v_1, v_4, u_0, u_1, u_4\}$ and $\{v_2, v_3, u_2, u_3\}$, respectively. Notice that H, G_1 and G_2 are disjoint connected subgraphs of G. If $s_1 < s_0 < s_4$ and $s_2 < s_0 < s_3$, then $|X_{s_0}| \geq 4$ by Lemma 2.18, and the width of $\mathcal{X}$ is at least 3. Otherwise $s_0 = s_i$ for some i $(1 \leq i \leq 4)$ and $\{v_0, u_0, v_i, u_i\} \subseteq X_{s_0}$, again, $|X_{s_0}| \geq 4$ and the width of $\mathcal{X}$ is at least 3. Assume next that (t_0, t_1) is a combination of either $(1,3)$ or $(1,4)$. Let H, G_1 and G_2 be induced subgraphs of G on $\{v_0\}$, $\{v_1, v_2, u_1, u_2\}$ and $\{v_3, v_4, u_3, u_4\}$, respectively. Similarly we have the width of $\mathcal{X}$ is at least 3. Thus the width of $\mathcal{X}$ is at least 3 and we have $ppw(G) \geq 3$. It is easy to see that there exists a 3-proper-path-decomposition of G, and there exists a 2-proper-path-decomposition for each proper minor of G. Thus G is a minimal forbidden minor for $\mathcal{P}_2$.

Although we can prove all the other cases similarly, we omit the proof for the sake of space. □

Let $P(a_0 a_1 \cdots a_{n-1})$ be a graph such that $a_i \leq 3$ $(0 \leq i \leq n-1)$. A graph 1-equivalent to $P(a_0 a_1 \cdots a_{n-1})$ is denoted by $P(z_0 z_1 \cdots z_{n-1})$ where z_i is a_i, "b", or "c" if $a_i = 3$, $z_i = a_i$ otherwise $(1 \leq i \leq n-1)$. A graph 2-equivalent to $P(a_0 a_1 \cdots a_{n-1})$ is denoted by $P(z_0 z_1 \cdots z_{n-1})$ where z_i is a_i or "x" if $a_i = 1$, $z_i = a_i$ otherwise $(1 \leq i \leq n-1)$.

Corollary 3.6 *36 Graphs shown in Fig. 3.13 are minimal forbidden minors for $\mathcal{P}_2$.*

Proof: The eight graphs stated in Theorem 3.9 are minimal forbidden minors for $\mathcal{P}_2$. There exist new eleven (seven) graphs 1-equivalent (2-equivalent) to the graphs stated in Theorem 3.9. These graphs are minimal forbidden minors for $\mathcal{P}_2$ by Theorems 3.5 and 3.6. Finally, ten graphs are minimal forbidden minors for $\mathcal{P}_2$ by Theorem 3.2. □

Notice that any minimal forbidden minor for $\mathcal{P}_2$ stated in Theorem 3.3 is isomorphic to a graph 1-equivalent to $P(333)$.

In the following , we show that a graph G is a minimal forbidden minor for $\mathcal{P}_2$ if and only if G is isomorphic to one of 36 graphs shown in Fig. 3.13.

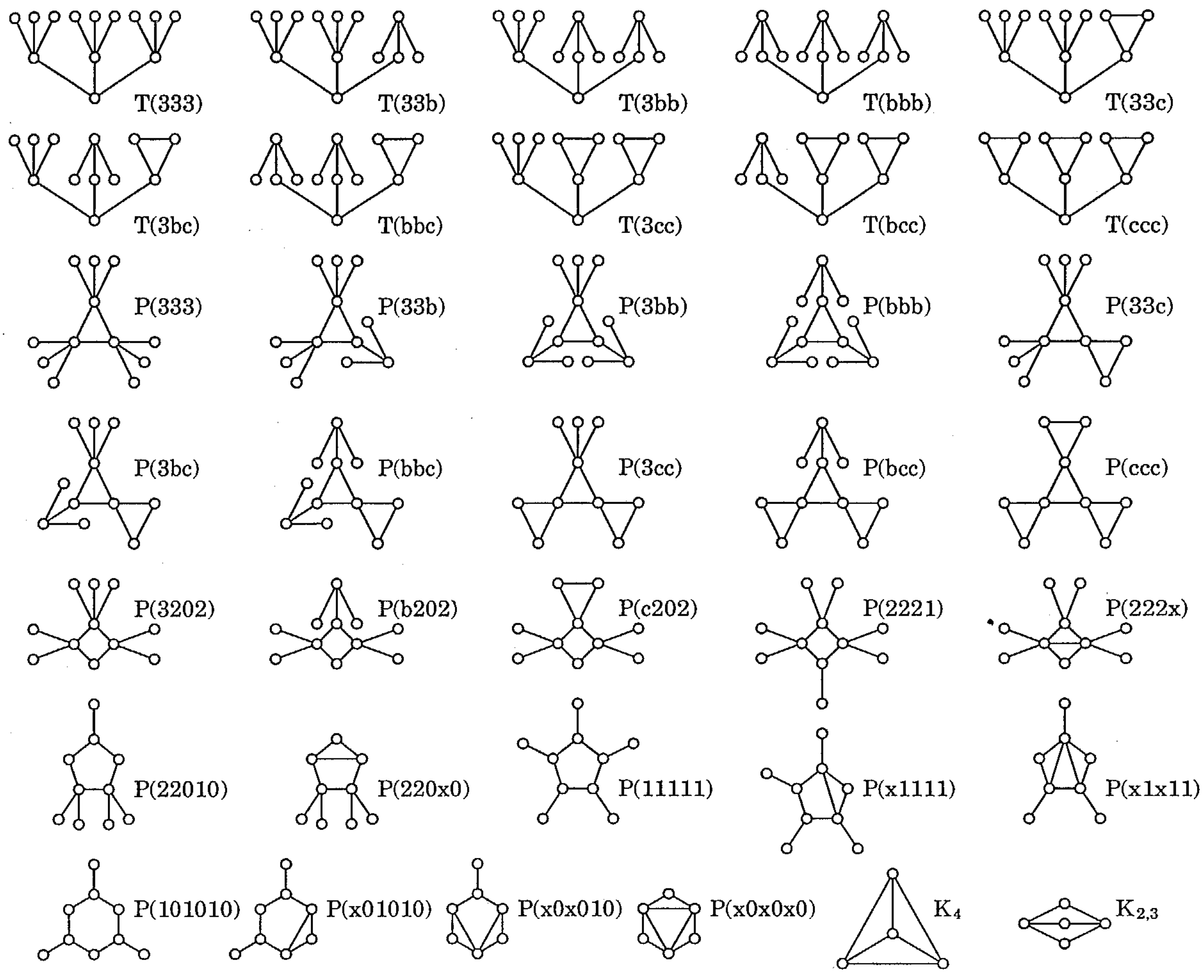


Figure 3.13: Minimal forbidden minors for $\mathcal{P}_2$.

A *block* of a graph G is a nontrivial connected subgraph of G, which contains no cut-vertices and which is maximal with respect to that property. A block with at least three vertices is called a *cyclic block*. Let G be a graph with a cyclic block B. For any vertex v of a cyclic block B in a graph G, $L(v)$ is the connected component of $G - E(B)$ containing v and $N(v)$ is the number of vertices of $L(v)\backslash\{v\}$ if $L(v)\backslash\{v\}$ consists of isolate vertices, infinite otherwise.

Lemma 3.10 *If a graph G is in $\Omega(\mathcal{P}_2)$, then G is connected and simple, and satisfies one of the following conditions:*

1. *G is a star-composition of (not necessarily distinct) three graphs in $\Omega(\mathcal{P}_1)$; or*
2. *G has exactly one cyclic block B, and $N(v) \leq 3$ for any vertex $v \in V(B)$; or*

3. *G is isomorphic to a graph 1-equivalent to a graph satisfying 2.*

Proof: By Theorem 3.4, a minimal forbidden minor for $\mathcal{P}_k$ is connected and simple.

Let G be a graph in $\Omega(\mathcal{P}_2)$. Assume that G is not a star-composition of (not necessarily distinct) three graphs in $\Omega(\mathcal{P}_1)$. If G has no cyclic block then G is acyclic, contradicting to Theorem 3.2. Thus we assume that G has at least one cyclic block.

Assume that there exist distinct cyclic blocks B_1 and B_2 of G. Note that $ppw(B_1) = ppw(B_2) = 2$. If $V(B_1) \cap V(B_2) = \emptyset$ then there exists a bridge (u, v) of G such that the graph obtained from G by deleting (u, v) has two connected components H_1 and H_2, and B_1 and B_2 are subgraphs of H_1 and H_2, respectively. Notice also that $ppw(H_1) = ppw(H_2) = 2$. However, this is contradicting to Theorem 3.8. Thus $V(B_1) \cap V(B_2) \neq \emptyset$. Let $v \in V(B_1) \cap V(B_2)$. Let H_1 be the connected component of $G \backslash (V(B_2) - \{v\})$ such that B_1 is a subgraph of H_1, and $H_2 = G \backslash (V(H_1) - \{v\})$. Notice that $V(H_1) \cap V(H_2) = \{v\}$, $E(H_1) \cup E(H_2) = E(G)$, and $ppw(H_1) = ppw(H_2) = 2$. By Theorem 3.7, $H_1 \in \Omega(\mathcal{P}_1)$ or $H_2 \in \Omega(\mathcal{P}_1)$. Without loss of generality, we assume that $H_1 \in \Omega(\mathcal{P}_1)$. Since H_1 has a cyclic block, H_1 is isomorphic to a complete graph $K_3 \in \Omega(\mathcal{P}_1)$. Then G is isomorphic to a graph 1-equivalent to the minimal forbidden minor for $\mathcal{P}_2$ such that the number of connected components is one less than that of G.

Thus we assume, without loss of generality, G has exactly one cyclic block. Let B be the cyclic block of G and w be a vertex in $V(B)$. If $ppw(L(w)) \geq 2$ then either $G \backslash (V(L(w)) - \{w\})$ or $L(w)$ is in $\Omega(\mathcal{P}_1)$ by Theorem 3.7. If $G \backslash (V(L(w)) - \{w\}) \in \Omega(\mathcal{P}_1)$ then G is 1-equivalent to a tree. Thus G is a star-composition of three graphs in $\Omega(\mathcal{P}_1)$, a contradiction. Thus $L(w) \in \Omega(\mathcal{P}_1)$. Hence $N(w) = 3$ or G is 1-equivalent to a graph with $N(w) = 3$. If $ppw(L(w)) = 1$, then $d_G(u) \leq 2$ for any vertex $u \in L(w)$. Since $d_G(u) = 1$ for any vertex $u \in L(w) - \{w\}$ by Theorem 3.4, $N(w) \leq 2$. Thus $N(v) \leq 3$ for any vertex $v \in V(B)$, or G is isomorphic to a graph 1-equivalent to a graph with $N(v) \leq 3$ for any vertex $v \in V(B)$. □

Since it is easy to characterize the graphs in $\Omega(\mathcal{P}_2)$ obtained by star-compositions of (not necessarily distinct) three graphs in $\Omega(\mathcal{P}_1)$, we consider, in the following, graphs with exactly one cyclic block. A graph is said to be *outer planar* if it can be embedded on the plane so that its edges intersect only at their ends and its vertices lie on the boundary on the exterior region. In the following, we call inner region of outer planar graph, simply, region.

Lemma 3.11 *For any minimal forbidden minor G for $\mathcal{P}_2$, G is not outer planar if and only if G is isomorphic to either K_4 or $K_{2,3}$.*

Proof: It is well-known that the class of all outer planar graphs is minor-closed, and K_4 and $K_{2,3}$ are the minimal forbidden minors for the class. Thus any non-outer planar graph G in $\Omega(\mathcal{P}_2)$ is either K_4 or $K_{2,3}$ by Corollary 3.6. □

To show that a graph $G \in \Omega(\mathcal{P}_2)$ is isomorphic to one of 36 graphs shown in Fig. 3.13, it is sufficient to consider graphs with one outer planar cyclic block. Let $P(310000)$ be the graph shown in Fig. 3.14.

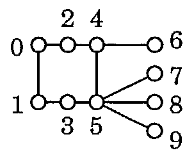


Figure 3.14: $P(310000)$.

Lemma 3.12 *For any minor G of $P(310000)$, there exists a 2-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ of G such that $0, 1 \in X_1$.*

Proof: ($\{0,1,2\}$, $\{1,2,3\}$, $\{2,3,4\}$, $\{3,4,5\}$, $\{4,5,6\}$, $\{5,6,7\}$, $\{5,7,8\}$, $\{5,8,9\}$) is a 2-proper-path-decomposition of $P(310000)$, and satisfies the condition of this lemma. Thus there exists such a proper-path-decomposition for any minor of $P(310000)$. □

Corollary 3.7 *The proper-path-widths of $P(310310)$, $P(310130)$, $P(33101)$, $P(31310)$, $P(31130)$, $P(3311)$, $P(3230)$, $P(3131)$ (shown in Fig. 3.15) are two.*

First, we consider the case that there exists exactly one region in outer planar cyclic block.

Lemma 3.13 *Let G be a simple connected outer planar graph such that G has an outer planar cyclic block B with one region and $N(v) \leq 3$ for any vertex $v \in V(B)$. If $G \in \Omega(\mathcal{P}_2)$ then G is isomorphic to either $P(333)$, $P(3202)$, $P(2221)$, $P(22010)$, $P(11111)$, or $P(101010)$.*

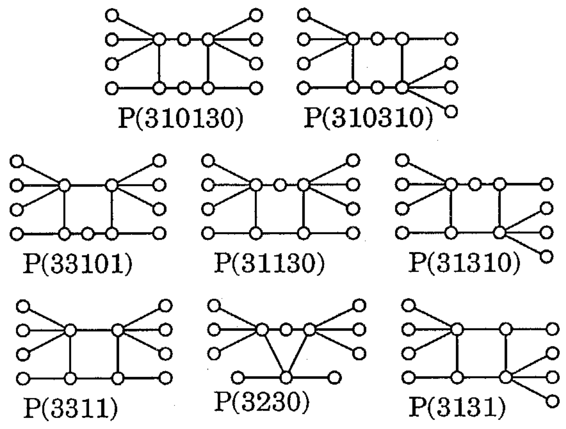


Figure 3.15: Graphs with proper-path-widths two.

Proof: Assume that G is not isomorphic to either $P(333)$, $P(3202)$, $P(2221)$, $P(22010)$, $P(11111)$, or $P(101010)$. We show that $G \notin \Omega(\mathcal{P}_2)$. To complete the proof, it is sufficient to show that either $ppw(G) = 2$ or there exists a proper minor of G that contained in $\Omega(\mathcal{P}_2)$.

In the following, we consider several cases according to the number of vertices in B.

1. $|V(B)| = 3$.

   Since G is a proper minor of $P(333) \in \Omega(\mathcal{P}_2)$, $ppw(G) = 2$.

2. $|V(B)| = 4$.

   Assume that a vertex $v_1 \in V(B)$ such that $N(v_1) = 3$ is adjacent to a vertex $v_2 \in V(B)$ such that $N(v_2) = 3$. If there exists a vertex $v \in V(B) - \{v_1, v_2\}$ such that $N(v) \geq 2$, then $P(3202) \in \Omega(\mathcal{P}_2)$ is a proper minor of G, otherwise $ppw(G) = 2$ since G is a minor of $P(3311)$. Thus v_1 is not adjacent to v_2 if $N(v_1) = N(v_2) = 3$. If there exists a vertex $v \in V(B)$ such that $N(v) = 0$, then $ppw(G) = 2$ since G is either a minor of $P(3230)$ or a proper minor of $P(3202) \in \Omega(\mathcal{P}_2)$. Thus there exists no vertex $v \in V(B)$ such that $N(v) = 0$. If there exist at least three vertices v such that $N(v) \geq 2$, then $P(2221) \in \Omega(\mathcal{P}_2)$ is a proper minor of G, otherwise $ppw(G) = 2$ since G is a minor of either $P(3311)$ or $P(3131)$.

3. $|V(B)| = 5$.

   Assume that a vertex $v_1 \in V(B)$ such that $N(v_1) \geq 2$ is adjacent to a vertex $v_2 \in V(B)$ such that $N(v_2) \geq 2$. Let v be a vertex in $V(B)$ such that v is adjacent to neither v_1 nor v_2. If $N(v) \geq 1$, then $P(22010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus $N(v) = 0$, and $N(s) \geq 1$ and $N(t) \geq 1$ for the other two vertices s and t of B by Theorem 3.4. However, if $N(s) = N(t) = 1$ then $ppw(G) = 2$ since G is a minor of $P(33101)$, otherwise $P(2221) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus v_1 is not adjacent to v_2 if $N(v_1) \geq 2$ and $N(v_2) \geq 2$. If $N(v) \geq 1$ for any vertex $v \in V(B)$, $P(11111) \in \Omega(\mathcal{P}_2)$ is a proper minor of G, otherwise $ppw(G) = 2$ since G is a minor of either $P(31310)$ or $P(31130)$.

4. $|V(B)| = 6$.

   If there exists at most one vertex v in $V(B)$ such that $N(v) = 0$, then $P(11111) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus we assume that there exist at least two vertices v such that $N(v) = 0$. Let v_1 and v_2 be vertices in $V(B)$ such that $N(v_1) = N(v_2) = 0$. Notice that v_1 is not adjacent to v_2 by Theorem 3.4. If there exists a vertex adjacent to both v_1 and v_2, then $P(101010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G by Theorem 3.4. Thus there exists no vertex adjacent to both v_1 and v_2. Hence $N(v) \geq 1$ for each vertex in $V(B) - \{v_1, v_2\}$. If a vertex $v_3 \in V(B)$ such that $N(v_3) \geq 2$ is adjacent to a vertex $v_4 \in V(B)$ such that $N(v_4) \geq 2$, then $P(22010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G, otherwise $ppw(G) = 2$ since G is a minor of either $P(310310)$ or $P(310130)$.

5. $|V(B)| > 6$.

   Assume that $|V(B)| = n$ $(n \geq 7)$. There exists at least $\lceil \frac{n}{2} \rceil$ vertices v such that $N(v) \geq 1$ by Theorem 3.4. If there exist at least five vertices v such that $N(v) \geq 1$, then $P(11111) \in \Omega(\mathcal{P}_2)$ is a minor of G. Thus $n < 9$. In case when $n = 7, 8$, there exist at least four vertices v such that $N(v) \geq 1$ and $P(101010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G.

Thus $G \notin \Omega(\mathcal{P}_2)$ and we complete the proof. □

Next, we consider the case that there exist more than one region in outer planar cyclic block.

Lemma 3.14 *Let G be a simple connected outer planar graph such that G has an outer planar cyclic block B with more than one region and $N(v) \leq 3$ for any vertex $v \in V(B)$. If $G \in \Omega(\mathcal{P}_2)$ then G is isomorphic to either $P(222x)$, $P(220x0)$, $P(x1111)$, $P(x1x11)$, $P(x01010)$, $P(x0x010)$, or $P(x0x0x0)$.*

Proof: Assume that G is not isomorphic to either $P(222x)$, $P(220x0)$, $P(x1111)$, $P(x1x11)$, $P(x01010)$, $P(x0x010)$, or $P(x0x0x0)$.

Distinct regions F and F' are said to be adjacent if there exists an edge $e \in E(F) \cap E(F')$. Suppose that some region in G is adjacent to more than two regions in G. Since $P(x0x0x0) \in \Omega(\mathcal{P}_2)$ is a proper minor of G, $G \notin \Omega(\mathcal{P}_2)$. Thus we assume that each region in G is adjacent to at most two regions in G. Let $F_1, F_2, \ldots, F_f$ $(f \geq 2)$ be regions in G such that F_i and F_{i+1} are adjacent to each other $(1 \leq i \leq f-1)$. Notice that $|E(F_i) \cap E(F_{i+1})| = 1$ for any i $(1 \leq i \leq f-1)$ since G is outer planar. Let $(u, u') \in E(F_1) \cap E(F_2)$, and $u = u_0, u_1, u_2, \ldots, u_m, u_{m+1} = u'$ be vertices in F_1 such that $(u_i, u_{i+1}) \in E(B)$ $(0 \leq i \leq m, m \geq 1)$. Similarly, let $(v, v') \in E(F_{f-1}) \cap E(F_f)$, and $v = v_0, v_1, v_2, \ldots, v_{m'}, v_{m'+1} = v'$ be vertices in F_f such that $(v_i, v_{i+1}) \in E(B)$ $(0 \leq i \leq m', m' \geq 1)$.

Let $V_1 = V(F_1)$ if $u' \neq v'$, $V_1 = V(F_1) - \{u'\}$ otherwise. Let G_1 be the induced subgraph of G on $V(F_1) \cup \bigcup_{w \in V_1} V(L(w))$.

Claim 3.1 *If $u \neq v$ or $N(u) = 0$ then G_1 is a minor of $P(310000)$.*

Proof: Let $a = 1$ if $N(u) = 0$, $a = 0$ otherwise. Let $b = m+1$ if $u' \neq v'$, $b = m$ otherwise. Notice that G_1 is the induced subgraph of G on $V(F_1) \cup \bigcup_{a \leq i \leq b} V(L(u_i))$.

Assume that $N(u_i) \geq 1$ and $N(u_{i+j}) \geq 1$ for some i and j $(a \leq i \leq b-2, 2 \leq j \leq b-i)$. Then $P(x01010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G such that (v, v') corresponds to an inner edge of $P(x01010)$. If $N(u_i) \geq 2$ and $N(u_j) \geq 2$ for some i and j $(a \leq i < j \leq b)$ then $P(220x0) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus G_1 is a minor of $P(310000)$, since no vertices w with $N(w) = 0$ are adjacent to each other by Theorem 3.4. □

Let $V_2 = V(F_f)$ if $u \neq v$, $V_2 = V(F_f) - \{v\}$ otherwise. Let G_2 be the induced subgraph of G on $V(F_f) \cup \bigcup_{w \in V_2} V(L(w))$. By a similar argument as above, if $u' \neq v'$ or $N(v') = 0$ then G_2 is a minor of $P(310000)$.

Let $V_3 = V(F_2) \cup V(F_3) \cup \cdots \cup V(F_{f-1})$. If there exists a vertex $w \in V_3 - \{u, u', v, v'\}$ such that $N(w) \geq 1$, then $G \notin \Omega(\mathcal{P}_2)$ since $P(x0x010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus we assume that $N(w) = 0$ for each vertex $w \in V_3 - \{u, u', v, v'\}$. Let G_3 be the induced subgraph of G on V_3. Notice that $V(G) = V(G_1) \cup V(G_2) \cup V(G_3)$. We claim the following.

Claim 3.2 *If both G_1 and G_2 are minors of $P(310000)$, then $G \notin \Omega(\mathcal{P}_2)$.*

Proof: Since G_1 is a minor of $P(310000)$, there exists a 2-proper-path-decomposition $\mathcal{X}_1 = (X_1^1, X_2^1, \ldots, X_{r_1}^1)$ of G_1 such that $u, u' \in X_{r_1}^1$ by Lemma 3.12. Similarly, there exists a 2-proper-path-decomposition $\mathcal{X}_2 = (X_1^2, X_2^2, \ldots, X_{r_2}^2)$ such that $v, v' \in X_1^2$. Since $N(w) = 0$ for each vertex $w \in V_3 - \{u, u', v, v'\}$, it is not difficult to see that there exists a 2-proper-path-decomposition $\mathcal{X}_3 = (X_1^3, X_2^3, \ldots, X_{r3}^3)$ of G_3 such that $u, u' \in X_1^3$ and $v, v' \in X_{r3}^3$. Then $(\mathcal{X}_1, \mathcal{X}_3, \mathcal{X}_2)$ is a 2-proper-path-decomposition of G, and $G \notin \Omega(\mathcal{P}_2)$. □

Thus by Claims 3.1 and 3.2, without loss of generality, we assume that $u = v$ and $N(u) > 0$. In the following, we consider two cases.

Case 1. Assume that $u' \neq v'$ or $N(u') = 0$.

If $N(u_i) \geq 1$ and $N(v_j) \geq 1$ for some i and j $(2 \leq i \leq m+1, 2 \leq j \leq m'+1)$ then either $P(x1x11) \in \Omega(\mathcal{P}_2)$ or $P(101010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, we have either $N(u_i) = 0$ for any i $(2 \leq i \leq m+1)$ or $N(v_j) = 0$ for any j $(2 \leq j \leq m'+1)$. Without loss of generality, we assume that $N(u_i) = 0$ for any i $(2 \leq i \leq m+1)$. Assume that $N(v_j) \geq 1$ for some j $(2 \leq j \leq m'+1)$. If $N(u) \geq 2$ and $N(u_1) \geq 2$, $P(22010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, either $N(u) = 1$ or $N(u_1) \leq 1$, and G_1 is a minor of $P(310000)$. Since G_2 is a minor of $P(310000)$ by Claim 3.1, $G \notin \Omega(\mathcal{P}_2)$ by Claim 3.2. Thus, $N(v_j) = 0$ for any j $(2 \leq j \leq m'+1)$. There exist at most three vertices w such that $N(w) > 0$ in $V_1 \cup V_2$.

Assume that $N(u) \geq 3$. If $N(u_1) \geq 2$ and $N(v_1) \geq 2$ then $P(3202) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, without loss of generality, we assume that $N(u_1) \leq 1$ and G_1 is a minor of $P(310000)$. Since G_2 is a minor of $P(310000)$ by Claim 3.1, $G \notin \Omega(\mathcal{P}_2)$ by Claim 3.2. Assume that $N(u) \leq 2$. Let x and y be the vertices in $L(u)\backslash\{u\}$. Let G_a be the induced subgraph of G on $V(F_1) \cup \{x\}$, and G_b be the induced subgraph of G on $V(F_f) \cup \{y\}$. Since both G_a and G_b are minors of $P(310000)$, there exist a 2-proper-path-decomposition $\mathcal{X}_a = (X_1^a, X_2^a, \ldots, X_{ra}^a)$ of G_a such that $u, u' \in X_{ra}^a$, and a 2-proper-path-decomposition $\mathcal{X}_b = (X_1^b, X_2^b, \ldots, X_{rb}^b)$ of G_b such that $v, v' \in X_1^b$. Then $(\mathcal{X}_a, \mathcal{X}_3, \mathcal{X}_b)$ is a 2-proper-path-decomposition of G, and $G \notin \Omega(\mathcal{P}_2)$.

Case 2. Assume that $u' = v'$ and $N(u') > 0$.

If there exist at least two vertices u_i $(1 \leq i \leq m)$ such that $N(u_i) \geq 1$, then $P(x1111) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus there exists at most one vertex u_i such that $N(u_i) \geq 1$ for some i $(1 \leq i \leq m)$. We have $m \leq 3$ by Theorem 3.4. However, if $m = 3$ then $P(101010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G since $N(u_2) \geq 1$. Thus we have $m \leq 2$. Similarly, there exists at most one vertex v_i such that $N(v_i) \geq 1$

for some i $(1 \leq i \leq m')$, and $m' \leq 2$. Without loss of generality, we assume that $m' \leq m \leq 2$.

Assume that $m = 2$. Without loss of generality, we assume that $N(u_1) \geq 1$. If $N(u) \geq 2$ and $N(u_1) \geq 2$, then $P(22010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, G_1 is a minor of $P(310000)$. If $m' = 2$ and $N(v_1) \geq 1$ then $P(101010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, we have either $m' = 2$ and $N(v_2) \geq 1$, or $m' = 1$. If $N(v') \geq 2$ and $N(v_{m'}) \geq 2$ then $P(22010) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus G_2 is a minor of $P(310000)$ and $G \notin \Omega(\mathcal{P}_2)$ by Claim 3.2.

Assume that $m = m' = 1$. Without loss of generality, we assume that $N(u) \geq N(u')$. If $N(u) = 1$ then both G_1 and G_2 are minors of $P(310000)$, since $N(u') = 1$. Thus we assume that $N(u) \geq 2$. If $N(u_1) \geq 2$ and $N(v_1) \geq 2$ then $P(2221) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus, without loss of generality, we assume that $N(u_1) \leq 1$. Then G_1 is a minor of $P(310000)$. If $N(u') \geq 2$ and $N(v_1) \geq 2$ then $P(222x) \in \Omega(\mathcal{P}_2)$ is a proper minor of G. Thus G_2 is a minor of $P(310000)$ and $G \notin \Omega(\mathcal{P}_2)$ by Claim 3.2.

Thus $G \notin \Omega(\mathcal{P}_2)$ and we complete the proof. □

Theorem 3.10 *A graph G is contained in $\Omega(\mathcal{P}_2)$ if and only if G is isomorphic to one of 36 graphs shown in Fig. 3.13.*

Proof: Since $\Omega(\mathcal{P}_1) = \{K_3, K_{1,3}\}$, the number of graphs in $\Omega(\mathcal{P}_2)$ obtained by star-compositions of (not necessarily distinct) three graphs in $\Omega(\mathcal{P}_1)$ is ten. A graph in $\Omega(\mathcal{P}_2)$ with a cyclic block B and $N(v) \leq 3$ for any vertex $v \in V(B)$ is isomorphic to one of the fifteen graphs shown in Lemmas 3.11, 3.13, and 3.14. The number of graphs 1-equivalent to these graphs are eleven. Thus by Lemma 3.10, the number of graphs in $\Omega(\mathcal{P}_2)$ is 36, and the graphs shown in Fig. 3.13 are the minimal forbidden minors for $\mathcal{P}_2$. □

3.4 Remarks

We conclude with the following remarks.

Theorem 3.1 obtained independently by Kinnersley [48]. A special case of Theorem 3.2 when $k = 2$ was proved by Takeuchi, Soejima, and Kishimoto [93] and independently by Fukuhara [39].

Robertson and Seymour [83] proved that there exists an $\mathcal{O}(n^3)$ time algorithm to decide if a given n-vertex graph is in a minor-closed family of graphs. The time complexity is reduced to $\mathcal{O}(n \log^2 n)$ if the family does not contain all planar graphs. This is obtained by combining the results in [83] and [53]. Moreover, the time complexity is reduced to $\mathcal{O}(n)$ if the family avoids a $2 \times k$ grid graph and a circus-graph [13]. It follows from our results that we have an $\mathcal{O}(n \log^2 n)$ time membership test algorithm for $\mathcal{P}_2$, since we have listed all the minimal forbidden minors for $\mathcal{P}_2$ and $\mathcal{P}_2$ does not contain all planar graphs. Notice that $\mathcal{P}_2$ contains a $2 \times k$ grid graph for any k. It should be noticed that our algorithm is the first explicit membership test algorithm for $\mathcal{P}_2$, although it is believed that there exists a more efficient and practical algorithm for $\mathcal{P}_2$ which does not rely on a minor test algorithm.

For applications in linguistics, it would be of definite interest to know the structure of graphs with path-width at most 6 as mentioned by Kornai and Tuza in [52]. Although it is known that the number of minimal forbidden minors for $\mathcal{P}_k$ ($k \geq 3$) is stupendous, we might be able to characterize all the minimal forbidden minors by simple compositions like as the minimal acyclic forbidden minors. In this sense, some general properties of minimal forbidden minors for $\mathcal{P}_k$ shown in this chapter give us a clue to characterize not only the minimal forbidden minors for $\mathcal{P}_k$ but also the minimal forbidden minors for $\mathcal{F}_k$ ($k \geq 3$).

Chapter 4

Path-Width and Search Games

4.1 Introduction of Search Games

This chapter considers a new version of search game, called mixed-searching, which is a natural generalization of edge-searching and node-searching extensively studied so far. We establish a relationship between the mixed-search number of a simple graph G and the proper-path-width of G. We also prove complexity results.

Search games were first introduced by Breisch [17] and Parsons [66]. An undirected graph G is thought of as a system of tunnels. Initially, all edges of G are contaminated by a gas. An edge is *cleared* by some operations on G. A cleared edge is *recontaminated* if there is a path from an uncleared edge to the cleared edge without any searchers on its vertices or edges.

In *edge-searching*, the original search game variant, an edge is cleared by sliding a searcher along the edge. A search is a sequence of operations of placing a searcher on a vertex, deleting a searcher from a vertex, or sliding a searcher along an edge. The object of such an *edge-search* is to clear all edges by a search. An edge-search is *optimal* if the maximum number of searchers on G at any operation is minimum over all edge-searches of G. This number is called the *edge-search number* of G, and is denoted by $es(G)$. LaPaugh [54] proved that there exists an optimal edge-search without recontamination of cleared edges. This means that the problem of deciding whether $es(G) \leq k$ is in NP. Megiddo, Hakimi, Garey, Johnson, and Papadimitriou [61] showed that the problem of computing $es(G)$ is NP-hard for general graphs but can be solved in linear time for trees.

Another variant called *node-searching* was introduced by Kirousis and Papadimitriou [51]. In node-searching, an edge is cleared by placing searchers at both its ends simultaneously. A *node-search* is a sequence of operations of placing a searcher on a vertex or deleting a searcher from a vertex so that all edges of G are simultaneously clear after the last stage. A node-search is optimal if the maximum number of searchers on G at any operation is minimum over all node-searches of G. This number is called the *node-search number* of G, and is denoted by $ns(G)$. Kirousis and Papadimitriou [51] proved the following results: (1) There exists an optimal node-search without recontamination of cleared edges; (2) The problem of computing $ns(G)$ is NP-hard for general graphs; (3) $ns(G) - 1 \leq es(G) \leq ns(G) + 1$. The unexpected equality $ns(G) = pw(G) + 1$ was mentioned by Möhring [62], and implied by Kirousis and Papadimitriou [50]. This was also shown in [11]. This provides a linear time algorithm to compute $ns(G)$ for trees [62, 91].

Mixed-searching is a natural generalization of edge-searching and node-searching. In mixed-searching, an edge is cleared by placing searchers at both its ends simultaneously or by sliding a searcher along the edge. A *mixed-search* is a sequence of operations of placing a searcher on a vertex, deleting a searcher from a vertex, or sliding a searcher along an edge so that all edges of G are simultaneously clear after the last stage. A mixed-search is optimal if the maximum number of searchers on G at any operation is minimum over all mixed-searches of G. This number is called the *mixed-search number* of G, and is denoted by $ms(G)$.

In the following, we first show the inequalities $es(G) - 1 \leq ms(G) \leq es(G)$ and $ns(G) - 1 \leq ms(G) \leq ns(G)$. We next show that there exists an optimal mixed-search without recontamination of cleared edges. This implies that the problem of deciding, given a graph G and an integer k, whether $ms(G) \leq k$ is in NP. Finally, we characterize the mixed-search number of a simple graph by means of the proper-path-width. That is, we establish the equality $ms(G) = ppw(G)$, so the problem of computing $ms(G)$ is also NP-hard for general graphs but can be solved in linear time for trees.

4.2 Mixed-Searching

In *mixed-search game*, a graph G is considered as a system of tunnels. Initially, all edges are contaminated by a gas. An edge is *cleared* by placing searchers at both its ends simultaneously or by sliding a searcher along the edge. A cleared edge is *recontaminated* if there is a path from an uncleared edge to the cleared edge without any searchers on its vertices or edges.

Definition 4.1 *A search is a sequence of the following operations:*

(a) *placing a new searcher on a vertex;*

(b) *deleting a searcher from a vertex;*

(c) *sliding a searcher on a vertex along an incident edge and placing the searcher on the other end;*

(d) *sliding a searcher on a vertex along an incident edge;*

(e) *sliding a new searcher along an edge and placing the searcher on its end;*

(f) *sliding a new searcher along an edge.*

The object of such a *mixed-search* is to clear all edges by a search. A mixed-search is *optimal* if the maximum number of searchers on G at any operation is minimum over all mixed-searches of G. This number is called the *mixed-search number* of G, and is denoted by $ms(G)$. Mixed-searching was introduced independently by Bienstock and Seymour [12].

We first show a relation to edge-searching and node-searching.

Theorem 4.1 *For any graph G, $es(G)-1 \leq ms(G) \leq es(G)$ and $ns(G)-1 \leq ms(G) \leq ns(G)$.*

Proof: The edge-search and node-search are special cases of the mixed-search by definition. Thus we have $ms(G) \leq es(G)$ and $ms(G) \leq ns(G)$. Using at most one more searcher to traverse an edge that is cleared by placing searchers at both its ends, we can convert any mixed-search to an edge-search. Thus $es(G) \leq ms(G)+1$. Similarly, using at most one more searcher to clear an edge that is cleared by sliding a searcher along the edge, we can convert any mixed-search to a node-search. Thus $ns(G) \leq ms(G)+1$. □

All four cases are possible as shown in Fig. 4.1.

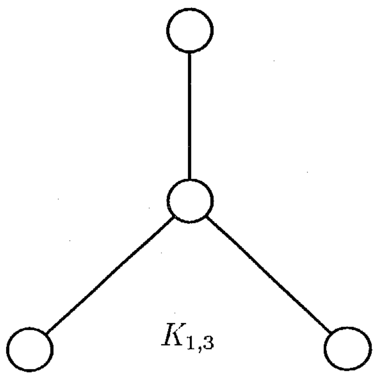
4.3 Monotonicity in Mixed-Searching

Kirousis and Papadimitriou proved that recontamination does not help in node-searching.

Theorem C ([51]) *For any graph G, there exists an optimal node-search without recontamination of cleared edges.*

Corollary A ([51]) *For any graph G, there exists an optimal node-search without recontamination of cleared edges satisfying the following two conditions:*

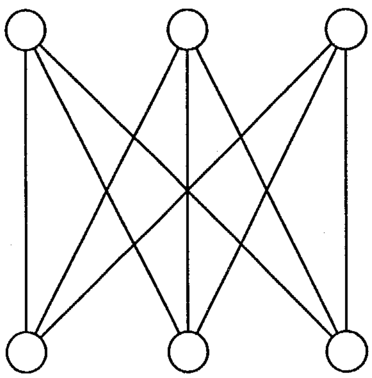
(i) every vertex is visited exactly once by a searcher,



(a) $ms(K_{1,3}) = 2$, $es(K_{1,3}) = 2$, $ns(K_{1,3}) = 2$

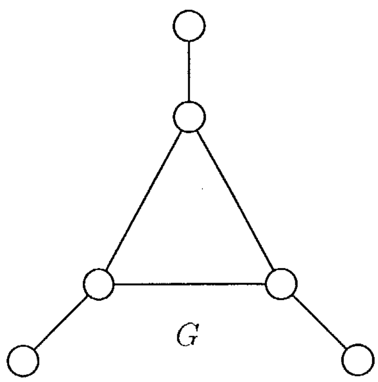
K_2

(b) $ms(K_2) = 1$, $es(K_2) = 1$, $ns(K_2) = 2$



$K_{3,3}$

(c) $ms(K_{3,3}) = 4$, $es(K_{3,3}) = 5$, $ns(K_{3,3}) = 4$



(d) $ms(G) = 2$, $es(G) = 3$, $ns(G) = 3$

Figure 4.1: Search numbers of graphs.

(ii) every searcher is deleted immediately after all the edges incident to it have been cleared (ties are broken arbitrarily).

We shall prove now that recontamination does not help even in mixed-searching.

Theorem 4.2 *For any graph G, there exists an optimal mixed-search without recontamination of cleared edges.*

Proof: Let G be a graph, and G_m be the graph obtained from G by subdividing every edge of G. We call the vertices of $V(G) \subseteq V(G_m)$ original vertices of G_m, and the vertices of $V(G_m) - V(G)$ middle vertices of G_m. We shall prove that $ms(G) = ns(G_m) - 1$ and an optimal mixed-search of G without recontamination of cleared edges can be obtained from an optimal node-search of G_m of the form described in Corollary A.

It is almost obvious that $ns(G_m) \leq ms(G)+1$ since by one extra searcher we can carry out node-search of G_m, simulating a mixed-search of G.

Conversely, we can carry out a mixed-search of G, simulating an optimal node-search of G_m of the form described in Corollary A as follows. We can assume that a searcher is placed on a middle vertex of G_m after a searcher is placed on one of its neighbors. The rules for the simulation are the following:

- When a searcher is placed on an original vertex v of G_m, a searcher is placed on v of G if v has no searcher.
- When a searcher is deleted from an original vertex v of G_m, delete the searcher from v of G if v has a searcher.
- When a searcher is placed on a middle vertex of G_m, clear the corresponding edge (u, v) of G, if it is contaminated, as follows: We can assume that u has a searcher and v does not have a searcher in G. If no recontamination is caused, clear $(u, v) \in E(G)$ by sliding a searcher on u along (u, v), and place it on v. Otherwise, clear $(u, v) \in E(G)$ by placing a new searcher on v.
- Do nothing in any other case.

It is not difficult to see that the simulation based on the rules above defines a mixed-search of G without recontamination of cleared edges, and the number of searchers used on G is at most $ns(G_m)$. We will show that $ns(G_m) - 1$ searchers are enough. Suppose that the number of searchers on G_m raises to $ns(G_m)$ when a searcher is placed on v of

G_m. The next operation on G_m must be deleting a searcher from a vertex. A searcher on v or a vertex adjacent to v must be deleted in the next operation by the assumption that the node-search is of the form described in Corollary A. There are the following four cases to be considered:

(1) v is an original vertex of G_m and the searcher on v is deleted in the next operation.

(2) v is an original vertex of G_m and a searcher on a vertex adjacent to v is deleted in the next operation.

(3) v is a middle vertex of G_m and the searcher on v is deleted in the next operation.

(4) v is a middle vertex of G_m and a searcher on a vertex adjacent to v is deleted in the next operation.

In the case of (1), all edges of G incident to v have been cleared before placing a searcher on v of G_m since all middle vertices of G_m adjacent to v have accepted searchers. Thus placing a new searcher on v of G is redundant. Similarly, we can show that no new searcher on v or a vertex adjacent to v is necessary for the other three cases. Thus we have $ms(G) \leq ns(G_m) - 1$. □

It should be noted that Theorem 4.2 implies that the problem of deciding, given a graph G and an integer k, whether $ms(G) \leq k$ is in NP.

A *crusade* in a graph G, introduced by Bienstock and Seymour [12], is a sequence $(C_1, C_2, \ldots, C_r)$ of subsets of $E(G)$, such that $C_1 = \emptyset, C_r = E(G)$, and $|C_i - C_{i-1}| \leq 1$ for $1 \leq i \leq r$. The crusade uses at most k searchers if the number of vertices which are ends of an edge in C_i and also of an edge in $E(G) - C_i$ is at most k for $1 \leq i \leq r$. Bienstock and Seymour characterized the mixed-search number of a graph with minimum degree at least two by means of the concept of crusade.

Theorem D ([12]) *For any graph G with minimum degree at least two, $ms(G) \leq k$ if and only if there exists a crusade in G using at most k searchers.*

Moreover, they independently proved Theorem 4.2 by using the crusade.

We obtain the following corollary from Theorem 4.2.

Corollary 4.1 *For any graph G, there exists an optimal mixed-search without recontamination of cleared edges such that it is a sequence of operations (a), (b), or (c) of Definition 4.1, and satisfying the following two conditions:*

(i) every vertex is visited exactly once by a searcher,

(ii) every edge is visited at most once by a searcher.

A mixed-search described above is said to be *simple.*

4.4 Proper-Path-Width and Mixed-Searching

Bienstock and Seymour characterized the mixed-search number of a graph with minimum degree at least two by the concept of crusade as shown in Theorem D. In the following, we characterize the mixed-search number of a simple graph by the proper-path-width.

Theorem 4.3 *For any simple graph G, $ms(G) = ppw(G)$.*

Proof: Suppose that $ppw(G) = k$ and $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ is a full k-proper-path-decomposition of G. If $r = 1$ then let v_1 and u_1 be distinct vertices in X_1 and place k searcher on the vertices of $X_1 - \{v_1\}$. If $(u_1, v_1) \in E(G)$, slide a searcher on u_1 to v_1 and place it on v_1. Otherwise, delete a searcher from u_1 and place a searcher on v_1. This defines a mixed-search with k searchers. Thus we assume $r \geq 2$. We can obtain a mixed-search with k searchers as follows:

1. Let v_1 be a vertex in $X_1 \cap X_2$. Place the k searchers on the vertices of $X_1 - \{v_1\}$.

2. Let u_1 be a vertex in $X_1 - X_2$. If $(u_1, v_1) \in E(G)$, slide a searcher on u_1 toward v_1 and place it on v_1. Otherwise, delete a searcher from u_1 and place a searcher on v_1. Let $i = 1$.

3. Repeat Step 3 while $i \leq r - 2$. Let $i = i + 1$. Let u_i be a vertex in $X_i - X_{i+1}$ and v_i be a vertex in $X_i - X_{i-1}$. If $(u_i, v_i) \in E(G)$, slide a searcher on u_i toward v_i and place it on v_i. Otherwise, delete a searcher from u_i and place a searcher on v_i.

4. Let u_r be a vertex in $X_{r-1} \cap X_r$ and v_r be a vertex in $X_r - X_{r-1}$. If $(u_r, v_r) \in E(G)$, slide a searcher on u_r toward v_r and place it on v_r. Otherwise, delete a searcher from u_r and place a searcher on v_r.

From the definition of full k-proper-path-decomposition, both u_i $(1 \leq i \leq r-1)$ and v_i $(2 \leq i \leq r)$ are uniquely determined. It should be noted that $((X_i - \{v_i\}) - \{u_i\}) \cup \{v_i\} = X_i \cap X_{i+1} = X_{i+1} - \{v_{i+1}\}$ and $u_{i+1} \in X_{i+1} - \{v_{i+1}\}$ for $1 \leq i \leq r-1$. An edge with both its ends in $X_i - \{v_i\}$ $(1 \leq i < r)$ is cleared since the vertices in $X_i - \{v_i\}$ have searchers simultaneously in 1, 2, or 3. Also, an edge with both its ends in $X_r - \{u_r\}$ is cleared since the vertices in $X_r - \{u_r\}$ have searchers simultaneously in 4. Since G is simple, there exists at most one edge connecting u_i and v_i $(1 \leq i \leq r)$, and each edge (u_i, v_i), if exists, is cleared by sliding a searcher along the edge. Thus all edges are cleared at least once. Suppose that all edges connecting the vertices in $\bigcup_{1\leq j\leq i-1} X_j$ are clear and k searchers are placed on the vertices in $X_i - \{v_i\}$. Since $u_i \notin \bigcup_{i+1\leq j\leq r} X_j$, all edges incident to u_i except for (u_i, v_i), if exists, are clear when a searcher on u_i is deleted or slid from u_i. Thus, when the searcher is placed on v_i, all edges in $\bigcup_{1\leq j\leq i} X_j$ are clear and k searchers are placed on the vertices in $X_{i+1} - \{v_{i+1}\}$. Thus by induction no edge is recontaminated. Thus the search above is indeed a mixed-search with at most $ppw(G)$ searchers, and we have $ms(G) \leq ppw(G)$.

Conversely, suppose that we have a simple mixed-search $\mathcal{S}$ with k searchers. For the i-th operation of $\mathcal{S}$, we define X_i as follows:

(1) When a searcher is placed on (deleted from) a vertex, we define X_i as the set of vertices having searchers.

(2) When a searcher is slid from u to v, we define X_i as the set consisting of u, v, and the vertices having searchers.

Let $\mathcal{X} = (X_1, X_2, \ldots, X_s)$ be the resulting sequence of sets of vertices. Since both ends of an edge which is cleared in the i-th operation are contained in X_i, all edges are contained in some X_i. Since $\mathcal{S}$ is simple, $\bigcup_{1\leq i\leq s} X_i = V(G)$ and each vertex of G appears in consecutive X_i's. Thus $\mathcal{X}$ satisfies conditions (ii), (iii), and (iv) in Definition 2.2. Notice that $|X_i| \leq k$ if X_i is defined by (1), $|X_i| \leq k+1$ otherwise. Let X_a, X_b, and X_c be elements in $\mathcal{X}$ such that each one is not a subset of the others $(1 \leq a < b < c \leq r)$. If X_b is defined by (1), then $|X_a \cap X_b| \leq k-1$ since $X_b \not\subseteq X_a$. Thus $|X_a \cap X_b \cap X_c| \leq k-1$. If X_b is defined by (2), then there exist distinct u and v in X_b such that $u \notin X_a$, and $v \notin X_c$. Therefore $|X_a \cap X_b \cap X_c| \leq k-1$. Hence, for any X_a, X_b, and X_c such that each one is not a subset of the others $(1 \leq a < b < c \leq r)$, $|X_a \cap X_b \cap X_c| \leq k-1$. Thus by Lemma 2.15, $ppw(G) \leq k$ and $ppw(G) \leq ms(G)$. □

It should be noted that Theorems 3.2 and 4.3 provide a structural characterization of trees T with $ms(T) \leq k$.

From Theorems 2.7, 2.8, and 4.3, we have the following complexity results on $ms(G)$.

Theorem 4.4 *The problem of computing $ms(G)$ is NP-hard for general graphs but can be solved in linear time for trees.*

4.5 Remarks

We conclude this chapter with the following remarks:

1. Notice that Theorem 4.3 does not hold for multiple graphs. If G is the graph consisting of two parallel edges, $ppw(G) = 1$, and $ms(G) = 2$. However we can prove that $ppw(G) \leq ms(G) \leq ppw(G) + 1$ for any multiple graph G.

2. Bodlaender and Kloks [15] showed an $O(n \log^2 n)$ time algorithm to decide whether $pw(G) \leq k$ for any graph G and a fixed integer k. We can modify their algorithm to decide whether $ppw(G) \leq k$ for any graph G and a fixed integer k.

3. We should mention the relation of mixed-searching with virus-searching [14, 92]. In virus-searching, initially, all vertices are contaminated by a virus. A vertex is cleared by placing a searcher on it. A cleared vertex is recontaminated if there is a path from an uncleared vertex to the cleared vertex without any searchers on its vertices or edges. A search is a sequence of operations of placing a searcher on a vertex, deleting a searcher from a vertex, or sliding a searcher along an edge. The object of such a virus-search is to clear all vertices by a search. A virus-search is optimal if the maximum number of searchers on G at any operation is minimum over all virus-searches of G. This number is called the *virus-search number* of G. Any virus-search S can be considered as a mixed-search, and vice versa. It is easy to see that an edge (u, v) is cleared by S as a mixed-search if and only if both its ends u and v are cleared by S as a virus-search. Thus the virus-search number is equal to the mixed-search number for any non empty graph G.

Chapter 5

Path-Width and Universal Graphs

5.1 Introduction of Universal Graphs

Given a family $\mathcal{F}$ of graphs, a graph G is said to be *universal* for $\mathcal{F}$ if G contains every graph in $\mathcal{F}$ as a subgraph. A *minimum universal graph* for $\mathcal{F}$ is a universal graph for $\mathcal{F}$ with the minimum number of edges. We denote the number of edges in a minimum universal graph for $\mathcal{F}$ by $f(\mathcal{F})$. $f(\mathcal{F})$ is $O(n^2)$ for any family $\mathcal{F}$ of graphs on n vertices, since a complete graph on n vertices is trivially a universal graph for $\mathcal{F}$. Determining $f(\mathcal{F})$ has been known to have applications to the circuit design, data representation, and parallel computing [8, 9, 24, 89, 97]. Bhatt, Chung, Leighton, and Rosenberg [9] showed a general upper bound for $f(\mathcal{F})$ for a family $\mathcal{F}$ of bounded-degree graphs by means of the size of separators.

For general families of (unbounded-degree) graphs, the following three results have been known:

(5.1) If $\mathcal{F}$ is the family of all planar graphs on n vertices, $f(\mathcal{F})$ is $\Omega(n \log n)$ and $O(n\sqrt{n})$ [7];

(5.2) If $\mathcal{F}$ is the family of all trees on n vertices, $f(\mathcal{F})$ is $\Theta(n \log n)$ [20];

(5.3) If $\mathcal{F}$ is the family of all 2-paths on n vertices, $f(\mathcal{F})$ is $\Theta(n \log n)$ [96]. (A 2-path is a special kind of outer planar graph.)

In this chapter, we show a generalization of (5.3). We consider finite undirected graphs without loops or multiple edges. We denote the family of all graphs on n vertices with path-width at most k by $\mathcal{F}_n^k$.

The purpose of this chapter is to prove the following:

Theorem 5.1 *For any integer k ($k \geq 1$) and n ($n \geq 12k$), $f(\mathcal{F}_n^k)$ is $\Theta(kn\log(n/k))$.* □

It follows from Theorem 5.1 that if $\mathcal{F}$ is the family of all planar graphs on n vertices with bounded path-width then $f(\mathcal{F})$ is $\Theta(n \log n)$.

Many related results can be found in the literature [7, 8, 9, 18, 19, 20, 21, 22, 23, 24, 89, 96, 97].

5.2 Universal Graphs for Graphs with Bounded Path-Width

5.2.1 Lower Bound

Let $d_G(v)$ be the degree of a vertex v in G. Let $D(G) = (\delta_G^1, \delta_G^2, \ldots, \delta_G^n)$ be the degree sequence for a graph G with n vertices, where $\delta_G^1 \geq \delta_G^2 \geq \cdots \geq \delta_G^n$. For graphs G and H with m and n vertices, respectively, we define $D(G) \geq D(H)$ if and only if $m \geq n$ and $\delta_G^i \geq \delta_H^i$ for any i ($1 \leq i \leq n$).

Lemma 5.1 *If a graph G is a universal graph for a family $\mathcal{F}$ of graphs, $D(G) \geq D(H)$ for any graph H in $\mathcal{F}$.*

Proof: For otherwise, G cannot contain H as a subgraph. □

Lemma 5.2 *For any integer k ($k \geq 1$) and s ($1 \leq s \leq \lfloor (n-2k)/k \rfloor$), there exists a k-intercat $R(k,s)$ on n vertices such that $\delta_{R(k,s)}^{ks} \geq \lfloor (n-2k)/s \rfloor + k$.*

Proof: Let $r = \lfloor (n-2k)/s \rfloor$. $R(k,s)$ can be constructed as follows:

1. Define that $Q(k,k)$ is the complete graph on the vertices $C_k = \{v_1, v_2, \ldots, v_k\}$;

2. Given $Q(k,i)$ and C_i ($k \leq i < 2k$), define that $Q(k,i+1)$ is the k-intercat obtained from $Q(k,i)$ by adding vertex v_{i+1} adjacent to the vertices in C_i, and let $C_{i+1} = (C_i \cup \{v_{i+1}\}) - \{v_{i+1-k}\}$;

3. Given $Q(k,i)$ and C_i ($2k + jr \leq i < r + k + jr, 0 \leq j \leq s-2$), define that $Q(k,i+1)$ is the k-intercat obtained from $Q(k,i)$ by adding vertex v_{i+1} adjacent to the vertices in C_i, and let $C_{i+1} = C_i$;

4. Given $Q(k,i)$ and C_i $(r+k+jr \leq i < r+2k+jr, 0 \leq j \leq s-2)$, define that $Q(k,i+1)$ is the k-intercat obtained from $Q(k,i)$ by adding vertex v_{i+1} adjacent to the vertices in C_i, and let $C_{i+1} = (C_i \cup \{v_{i+1}\}) - \{v_{i+1-r}\}$;

5. Given $Q(k,i)$ and C_i $(2k+(s-1)r \leq i \leq n-1)$, define that $Q(k,i+1)$ is the k-intercat obtained from $Q(k,i)$ by adding vertex v_{i+1} adjacent to the vertices in C_i, and let $C_{i+1} = C_i$;

6. Define $R(k,s) = Q(k,n)$.

It is easy to see that $|C_i| = k$ and $Q(k,i)$ is a k-intercat for any i $(k \leq i \leq n)$. It is also easy to see that $d_{R(k,s)}(v_{k+i+jr}) = r+k$ $(1 \leq i \leq k, 0 \leq j \leq s-2)$, and $d_{R(k,s)}(v_{k+i+(s-1)r}) \geq r+k$ $(1 \leq i \leq k)$. Thus we have $\delta^{ks}_{R(k,s)} \geq r+k$. □

For example, k-intercat $R(2,1)$ and $R(2,2)$ on 20 vertices such that $\delta^2_{R(2,1)} \geq 18$ and $\delta^4_{R(2,2)} \geq 10$ are shown in Figs. 5.1 and 5.2.

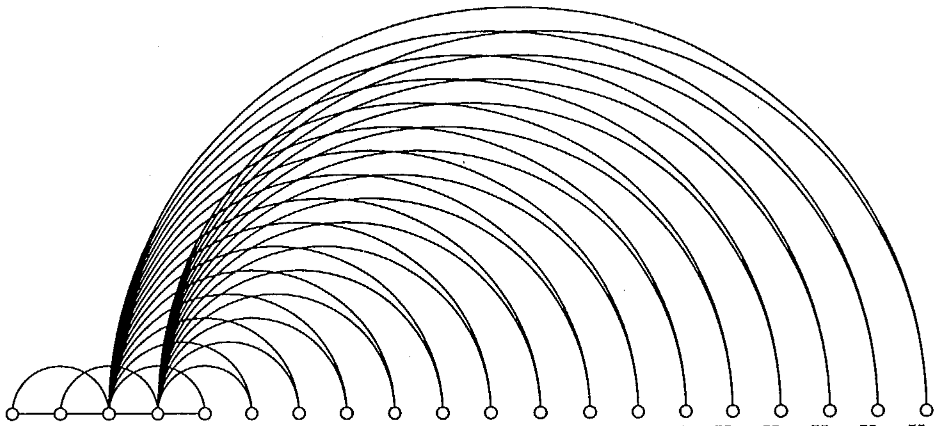


Figure 5.1: $R(2,1)$ on 20 vertices.

Theorem 5.2 *For any integer k $(k \geq 1)$ and n $(n \geq 3k)$, $f(\mathcal{F}^k_n)$ is $\Omega(kn\log(n/k))$.*

Proof: Let G be a universal graph for $\mathcal{F}^k_n$ and $t = \lfloor (n-2k)/k \rfloor$. Notice that $2|E(G)| = \sum_{v\in V(G)} d_G(v) \geq \sum_{i=1}^{n} \delta^i_G > \sum_{i=1}^{tk} \delta^i_G \geq k\sum_{i=1}^{t} \delta^{ki}_G$. By Lemmas 5.1, 5.2, and Theorem 2.4,

$$k\sum_{i=1}^{t} \delta^{ki}_G = k\sum_{i=1}^{t}\left(\left\lfloor \frac{n-2k}{i} \right\rfloor + k\right)$$

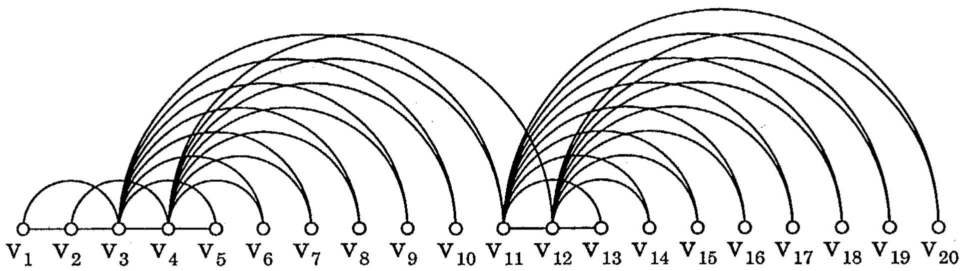


Figure 5.2: $R(2,2)$ on 20 vertices.

$$\begin{aligned} &\geq k\sum_{i=1}^{t}\left(\frac{n-2k}{i}+k-1\right) \\ &> k(n-2k)\log_e\left(\frac{n-2k}{k}\right)+(k-1)(n-3k). \end{aligned}$$

Thus $|E(G)|$ is $\Omega\left(kn\log(n/k)\right)$. □

5.2.2 Upper Bound

We show an upper bound by constructing the graph G_n^k with n vertices and $O\left(kn\log(n/k)\right)$ edges, and proving that G_n^k is a universal graph for $\mathcal{F}_n^k$.

Let $k^* = 2^{\lceil \log k \rceil}$, b_i be the maximum power of 2 such that $b_i | i$, and $b_{i,j} = \max(b_i, b_j)$. Notice that $k \leq k^* < 2k$. Let G_n^k $(k \geq 1, n \geq 1)$ be the graph obtained by the following construction procedure:

(1) Let $u_1, u_2, \ldots, u_n$ be n vertices;

(2) For any distinct i and j, join u_i and u_j by an edge if $|j-i| \leq 3k^* b_{i,j} + k - 1$.

For example, G_{20}^2 is shown in Fig. 5.3.

Theorem 5.3 *For any integer k $(k \geq 1)$ and n $(n \geq 12k)$, $|E(G_n^k)| = O\left(kn\log(n/k)\right)$.*

Proof: Let E_i $(1 \leq i \leq n)$ be the set of edges $(u_i, u_j) \in G_n^k$ such that $|j-i| \leq 3k^* b_i + k - 1$. It is easy to see that $|E_i| \leq \min(2(3k^* b_i + k - 1), n-1)$ for any i $(1 \leq i \leq n)$, and $\bigcup_{i=1}^{n} E_i = E(G_n^k)$. Notice that $|\{i \mid b_i = 2^h, 1 \leq i \leq n\}| = \left\lfloor (n+2^h)/2^{h+1} \right\rfloor$ and $|\{i \mid b_i \geq 2^h, 1 \leq i \leq n\}| = \left\lfloor n/2^h \right\rfloor$ for any integer h $(h \geq 0)$. Since $2(3k^* 2^{\log(n/(6k^*))} + k - 1) \geq n$,

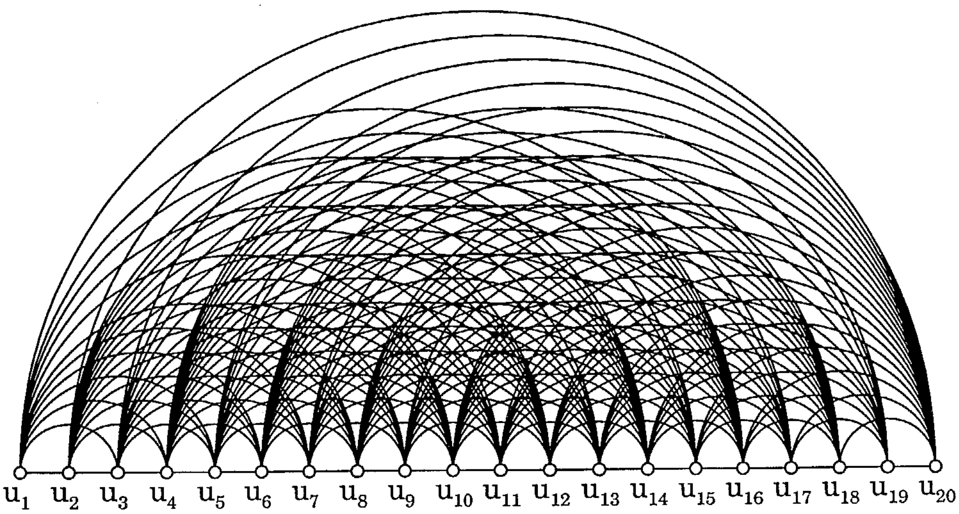


Figure 5.3: G^2_{20}.

the total number of edges added in (2) is at most

$$\begin{aligned}\sum_{i=1}^{n}|E_i| &< \sum_{h=0}^{\lfloor \log \frac{n}{6k^*}\rfloor} 2(3k^*2^h+k-1)\left\lfloor\frac{n+2^h}{2^{h+1}}\right\rfloor + (n-1)\left\lfloor\frac{n}{2^{\lfloor \log \frac{n}{6k^*}\rfloor+1}}\right\rfloor \\ &< \sum_{h=0}^{\lfloor \log \frac{n}{6k^*}\rfloor} (3k^*2^h+k-1)\left(\frac{n}{2^h}+1\right) + 6k^*(n-1) \\ &< (6kn+k-1)\log\frac{n}{6k} + (20k-1)n - (6k^2+8k+1).\end{aligned}$$

Thus $|E(G^k_n)| = O\left(kn\log(n/k)\right)$. □

Theorem 5.4 *For any integer k ($k \geq 1$) and n ($n \geq 1$), G^k_n is a universal graph for $\mathcal{F}^k_n$.*

Proof: By Theorem 2.4, it is sufficient to show that any k-intercat is a subgraph of G^k_n. Let R be a k-intercat in $\mathcal{F}^k_n$. We shall show that R is a subgraph of G^k_n. If $n \leq 4k$, R is a subgraph of G^k_n since G^k_n is the complete graph on n vertices. Thus we assume that $n \geq 4k+1$. As we mentioned before, we can assume that R can be obtained as follows:

1. Define that Q_k is the complete graph on the vertices $C_k = \{v_1, v_2, \ldots, v_k\}$;

2. Given Q_i and C_i ($k \leq i \leq n-1$), define that Q_{i+1} is the k-intercat obtained from Q_i by adding vertex $v_{i+1} \notin V(Q_i)$ adjacent to the vertices in C_i, and let $C_{i+1} = (C_i \cup \{v_{i+1}\}) - \{w_i\}$ where $w_i \in C_i \cup \{v_{i+1}\}$;

3. Define $R = Q_n$.

For the construction above, we have the following two lemmas.

Lemma 5.3 *If* $(v_a, v_c) \in E(R)$ *then* $(v_a, v_b) \in E(R)$ *for any distinct* a, b, *and* c $(1 \leq a < b < c \leq n)$.

Proof: Assume contrary that $(v_a, v_b) \notin E(R)$. Since $v_a \notin C_{b-1}$ and $v_a \in C_{c-1}$, $v_a = v_{i+1}$ for some i $(b-1 \leq i \leq c-2)$, contradicting that $v_{i+1} \notin V(Q_i)$. □

Define $l_i = \max(d \mid (v_i, v_{i+d}) \in E(R) \vee d = 0)$ for any i $(1 \leq i \leq n)$.

Lemma 5.4 *For any integer* i $(1 \leq i \leq n-1)$, $l_i = 0$ *if and only if* $|\{v_j \mid (v_j, v_{i+1}) \in E(R), j < i\}| = k$.

Proof: First, assume that $1 \leq i \leq k$. Since $(v_i, v_{k+1}) \in E(R)$, $l_i > 0$. Since Q_k is the complete graph on the vertices $v_1, v_2, \ldots,$ and v_k, $|\{v_j \mid (v_j, v_{i+1}) \in E(R), j < i\}| = i - 1 < k$.

Next, assume that $k+1 \leq i \leq n-1$. Notice that v_{i+1} is adjacent to the vertices in C_i in Q_{i+1}, $\{v_j \mid (v_j, v_{i+1}) \in E(R), j < i\} = C_i - \{v_i\}$, and $|C_i| = k$. Suppose that $l_i = 0$. By the definition of l_i, we have $(v_i, v_{i+1}) \notin E(R)$, and $v_i \notin C_i$. Thus $|\{v_j \mid (v_j, v_{i+1}) \in E(R), j < i\}| = |C_i| = k$. Conversely, suppose that $|\{v_j \mid (v_j, v_{i+1}) \in E(R), j < i\}| = k$. Since $|C_i| = k$, we have $v_i \notin C_i$, and $(v_i, v_{i+1}) \notin E(R)$. Thus $l_i = 0$ by Lemma 5.3. □

Let $l_i^* = 2^{\lceil \log l_i \rceil}$ if $l_i \geq 1$, $l_i^* = 1$ otherwise. Let $m_i = \lceil l_i^*/(2k^*) \rceil$. Now we define mapping ϕ:$\{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ as shown in Fig. 5.4.

A mapping ϕ:$\{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$

1. Let $D_0 = \emptyset$, $U_0 = \{1, 2, \ldots, n\}$, and $i = 1$.
2. Define that $\phi(i)$ is the minimum integer such that $\phi(i) \in U_{i-1}$ and $m_i | \phi(i)$.
3. Let $D_i = D_{i-1} \cup \{\phi(i)\}$ and $U_i = U_{i-1} - \{\phi(i)\}$.
4. If $i = n$, halt. Otherwise, set $i = i + 1$, and return to Step 2.

Figure 5.4: A mapping from k-intercat into $G_n^{\prime k}$.

Notice that $m_i \leq b_{\phi(i)}$ for any i $(1 \leq i \leq n)$ since both m_i and $b_{\phi(i)}$ are power of 2 that divide $\phi(i)$. Notice also that $l_i \leq l_i^* < 2l_i$ if $l_i \geq 1$.

Lemma 5.5 *ϕ is a one-to-one mapping satisfying that $-k \leq \phi(i) - i \leq \lceil l_i^*/2 \rceil - 1$ for any i $(1 \leq i \leq n)$.*

Proof: By induction on i, we show that

$$-k \leq \phi(i) - i \leq \left\lceil \frac{l_i^*}{2} \right\rceil - 1, \tag{5.4}$$

and

$$\phi(i) - i \leq l_i - k - 1 \text{ if } m_i \geq 2. \tag{5.5}$$

Assume that the algorithm have determined $\phi(1), \phi(2), \ldots, \phi(i-1)$ satisfying the inequalities (5.4) and (5.5), and $\{1, 2, \ldots, i-h-1\} \subseteq D_{i-1}$ and $i - h \in U_{i-1}$ $(0 \leq h \leq k, h < i)$. Notice h depends on i and that these assumptions are trivially true if $i = 1$, since $D_0 = \emptyset$ and $1 \in U_0$. We show that the inequalities (5.4) and (5.5) hold also for $\phi(i)$ $(i \geq 1)$, and there exists h' $(0 \leq h' \leq k, h' < i+1)$ such that $\{1, 2, \ldots, i-h'\} \subseteq D_i$ and $i - h' + 1 \in U_i$.

First, suppose that $0 \leq h \leq k-1$. It is easy to see that

$$\begin{aligned} -h \leq \phi(i) - i &\leq -h + (h+1)m_i - 1 \\ &= (h+1)(m_i - 1) \\ &< (h+1)\frac{l_i^*}{2k} \leq \frac{l_i^*}{2} \leq \left\lceil \frac{l_i^*}{2} \right\rceil. \end{aligned}$$

Notice that $\phi(i) \leq i + \lceil l_i^*/2 \rceil - 1 \leq i + l_i - 1 < n$ if $l_i \geq 1$, and $\phi(i) \leq i + \lceil l_i^*/2 \rceil - 1 = i$ otherwise. Thus $\phi(i)$ is uniquely determined in 2 in the algorithm. If $m_i \geq 2$ then

$$\begin{aligned} \phi(i) - i &\leq (h+1)\left(\frac{l_i^*}{2k^*} - 1\right) \\ &\leq (h+1)\frac{l_i - k - 1}{k} \\ &\leq l_i - k - 1. \end{aligned}$$

Thus the inequalities (5.4) and (5.5) hold also for $\phi(i)$. Since $h \leq k-1$, there exists h' $(0 \leq h' \leq h+1 \leq k, h' < i+1)$ such that $\{1, 2, \ldots, i-h'\} \subseteq D_i$ and $i - h' + 1 \in U_i$.

Next, suppose that $h = k$. We will show that $m_i = 1$ and $\phi(i) - i = -k$. Let $W = \{j \mid \phi(j) \geq i - k + 1, j < i\}$. Since $i - k \in U_{i-1}$, $|W| = k$ and $m_j \geq 2$ for any $j \in W$. Notice that $j < i + 1 < \phi(j) + k + 1 \leq j + l_j$ for any $j \in W$ by the definition of W and the inequality (5.5). Since $(v_j, v_{j+l_j}) \in E(R)$ for any $j \in W$ by the definition of l_j, $(v_j, v_{i+1}) \in E(R)$ by Lemma 5.3. Thus $l_i = 0$ by Lemma 5.4, and we have $m_i = 1$

and $\phi(i) - i = -k$. Therefore the inequalities (5.4) and (5.5) hold also for $\phi(i)$. Since $\phi(i) = i - k$, there exists h' ($0 \leq h' \leq k, h' < i+1$) such that $\{1, 2, \ldots, i-h'\} \subseteq D_i$ and $i - h' + 1 \in U_i$.

Thus ϕ is a one-to-one mapping satisfying the inequality (5.4) for any $\phi(i)$. □

Lemma 5.6 *If* $(v_i, v_j) \in E(R)$ *then* $(u_{\phi(i)}, u_{\phi(j)}) \in E(G_n^k)$.

Proof: Without loss of generality, we assume that $i < j$. Notice that $1 \leq j - i \leq l_i \leq l_i^*$. Since ϕ is a one-to-one mapping, $\phi(i) \neq \phi(j)$. From Lemma 5.5, we have $-k \leq \phi(i) - i \leq \lceil l_i^*/2 \rceil - 1$ and $-k \leq \phi(j) - j \leq \lceil l_j^*/2 \rceil - 1$. Thus $-(\lceil l_i^*/2 \rceil + k - 2) \leq \phi(j) - \phi(i) \leq l_i^* + \lceil l_j^*/2 \rceil + k - 1$ and $|\phi(j) - \phi(i)| \leq l_i^* + \lceil l_j^*/2 \rceil + k - 1$.

If $l_j^* > l_i^*$ then $|\phi(j) - \phi(i)| < \lceil 3l_j^*/2 \rceil + k - 1 \leq 3k^* m_j + k - 1 \leq 3k^* b_{\phi(i),\phi(j)} + k - 1$. Notice that $m_j \leq b_{\phi(j)} \leq b_{\phi(i),\phi(j)}$. Thus $(u_{\phi(i)}, u_{\phi(j)}) \in E(G_n^k)$ by the definition of G_n^k. The same type of argument applies when $l_j^* \leq l_i^*$. □

By Lemmas 5.5 and 5.6, we conclude that R is a subgraph of G_n^k. This completes the proof of Theorem 5.4. □

Theorem 5.1 follows from Theorems 5.2, 5.3, and 5.4.

5.3 Remarks

Notice that an embedding of a given graph with a path-decomposition can be obtained by the mapping shown in Fig. 5.4. An embedding of the graph $R(2,1)$ on 20 vertices and an embedding of the graph $R(2,2)$ on 20 vertices are shown in Figs. 5.5 and 5.6, respectively.

We conclude with the following open problems.

1. Close up the gap between upper and lower bounds in (5.1).

2. Generalize (5.2) to k-trees ($k \geq 2$).

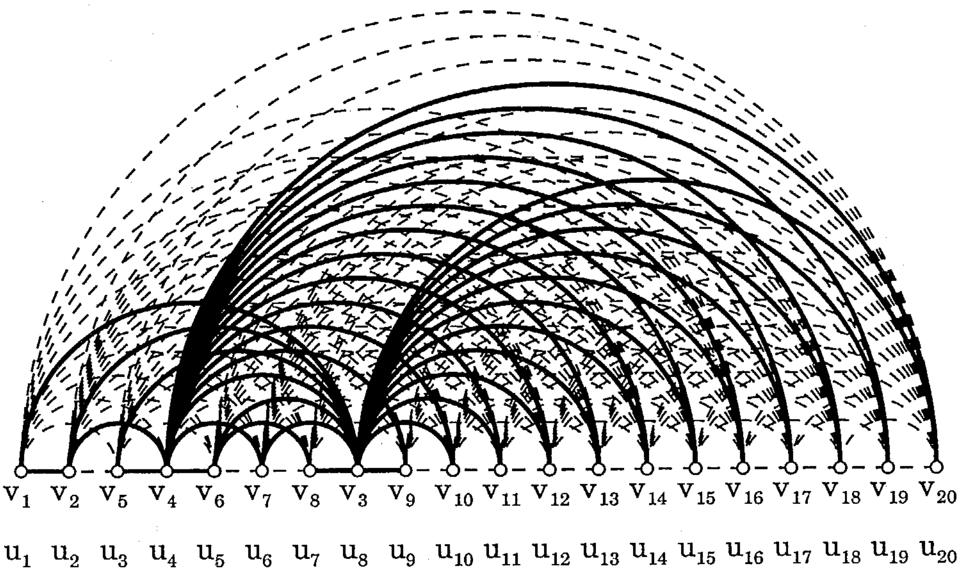


Figure 5.5: An embedding of $R(2,1)$ on 20 vertices into G_{20}^2.

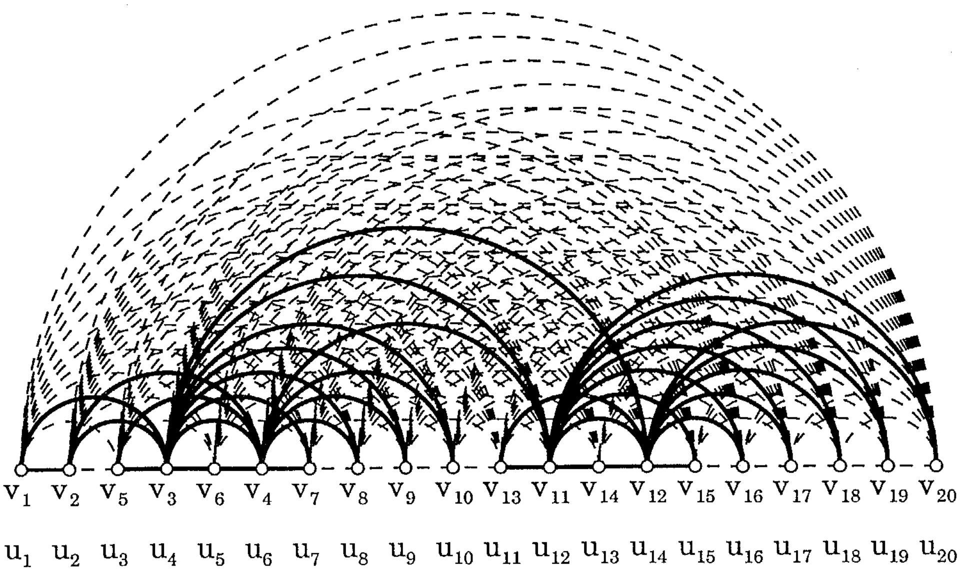


Figure 5.6: An embedding of $R(2,2)$ on 20 vertices into G_{20}^2.

Chapter 6

Conclusion

In this thesis, we investigate the path-width and proper-path-width of graphs.

In Chapter 2, we define the path-width and proper-path-width of graphs, and discuss properties of them. We show that a simple graph is a partial k-intercat if and only if its path-width is at most k, and a simple graph is a partial k-path if and only if its proper-path-width is at most k. Using these results, we show that the problem of computing the path-width and the problem of computing the proper-path-width are NP-hard. However, we show that the problem of computing the proper-path-width can be solved in linear time for trees. Moreover, similar to path-width, we show that a proper-path-decomposition of a tree can be obtained in linear time.

In Chapter 3, we list the minimal acyclic forbidden minors for the family of graphs with bounded path-width or proper-path-width. Moreover we list all 36 minimal forbidden minors for the family of graphs with proper-path-width at most two. This gives us the first explicit membership test algorithm for the family of graphs with proper-path-width at most two, although it is believed that there exists a more efficient and practical algorithm which does not rely on a minor test algorithm. Our proof contains many general methods that give us a clue to characterize the minimal forbidden minors for the family of graphs with path-width or proper-path-width at most k ($k \geq 3$).

In Chapter 4, we introduce a new version of search game, called mixed-searching, which is a natural generalization of edge-searching and node-searching. We show that there exists an optimal mixed-search without recontamination of cleared edges, and the mixed-search number of a simple graph G is equal to the proper-path-width of G. This also shows that the problem of computing mixed-search number is NP-hard for general graphs, but can be solved in linear time for trees. The optimal mixed-search strategy for a tree is obtained from a proper-path-decomposition of the tree.

In Chapter 5, we give a universal graph for the family of n vertex graphs with path-width at most k, and show that the number of edges in a minimum universal graph is $\Theta(kn\log(n/k))$. It follows that the number of edges in a minimum universal graph for the family of all planar graphs on n vertices with bounded path-width is $\Theta(n\log n)$. We also give an embedding algorithm of a graph with a path-decomposition on the universal graph.

The *tree-width* of graphs [72], which is a generalization of path-width, has been studied extensively so far. It is well known that a simple graph G is a partial k-tree if and only if its tree-width is at most k, and many problems become also solvable in polynomial for the family of graphs with bounded tree-width [5]. It is still open whether there exists a problem which becomes solvable in polynomial time for the family of graphs with bounded path-width, but still NP-complete for the family of graphs with bounded tree-width.

Acknowledgement

I would like to express my gratitude to Professor Yoji Kajitani and Associate Professor Shuichi Ueno for many valuable comments and encouragement. I would like to thank all the members in Kajitani and Ueno laboratory at Tokyo Institute of Technology for their cooperations and discussions.

Bibliography

[1] D. Archdeacon, "A Kuratowski theorem for the projective plane," *J. Graph Theory*, vol. 7, pp. 325–334, 1983.

[2] S. Arnborg, "Reduced state enumeration — another algorithm for reliability evaluation," *IEEE Trans. Reliability*, vol. R-27, pp. 101–105, 1978.

[3] S. Arnborg, "Efficient algorithms for combinatorial problems on graphs with bounded decomposability — a survey," *BIT*, vol. 25, pp. 2–33, 1985.

[4] S. Arnborg, D. G. Corneil, and A. Proskurowski, "Complexity of finding embeddings in a k-tree," *SIAM J. Algebraic and Discrete Methods*, vol. 8, pp. 277–284, April 1987.

[5] S. Arnborg and A. Proskurowski, "Linear time algorithms for NP-hard problems restricted to partial k-trees," *Discrete Applied Mathematics*, vol. 23, pp. 11–24, 1989.

[6] S. Arnborg, A. Proskurowski, and D. G. Corneil, "Forbidden minors characterization of partial 3-trees," *Discrete Mathematics*, vol. 80, pp. 1–19, 1990.

[7] L. Babai, F. R. K. Chung, P. Erdös, R. L. Graham, and J. Spencer, "On graphs which contain all sparse graphs," in *Theory and Practice of Combinatorics* (A. Rosa, G. Sabidussi, and J. Turgeon, eds.), vol. 12 of *Annals of Discrete Mathematics*, pp. 21–26, North-Holland, 1982.

[8] S. N. Bhatt, F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg, "Optimal simulations of tree machines," in *Proc. 27th Annual Symposium on Foundations of Computer Science*, pp. 274–282, 1986.

[9] S. N. Bhatt, F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg, "Universal graphs for bounded-degree trees and planar graphs," *SIAM J. Discrete Mathematics*, vol. 2, pp. 145–155, 1989.

[10] D. Bienstock, "Graph searching, path-width, tree-width and related problems (a survey)," in *Reliability of Computer and Communication Networks* (F. Hwang, C. Monma, and F. Roberts, eds.), vol. 5 of *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, pp. 33–49, American Mathematical Society, 1991.

[11] D. Bienstock, N. Robertson, P. Seymour, and R. Thomas, "Quickly excluding a forest," *J. Combinatorial Theory (B)*, vol. 52, pp. 274–283, 1991.

[12] D. Bienstock and P. Seymour, "Monotonicity in graph searching," *J. Algorithms*, vol. 12, no. 2, pp. 239–245, 1991.

[13] H. L. Bodlaender, "On linear time minor tests with depth-first search," *J. Algorithms*, vol. 14, pp. 1–23, 1993.

[14] H. L. Bodlaender, "Improved self-reduction algorithms for graphs with bounded treewidth," *Discrete Applied Mathematics*, vol. 54, pp. 101–115, 1994.

[15] H. L. Bodlaender and T. Kloks, "Better algorithms for the pathwidth and treewidth of graphs," in *Proc. 18th International Colloquium on Automata, Languages and Programming* (J. L. Albert, B. Monien, and M. R. Artalejo, eds.), vol. 510 of *Lecture Notes in Computer Science*, pp. 544–555, Berlin, New York: Springer-Verlag, 1991.

[16] H. L. Bodlaender and R. H. Möhring, "The pathwidth and treewidth of cographs," *SIAM J. Discrete Mathematics*, vol. 6, no. 2, pp. 181–188, 1993.

[17] R. L. Breisch, "An intuitive approach to speleotopology," *Southwestern Cavers*, vol. 6, no. 5, pp. 72–78, 1967.

[18] F. R. K. Chung, "Universal graphs and induced-universal graphs," *J. Graph Theory*, vol. 14, pp. 443–454, 1990.

[19] F. R. K. Chung and R. L. Graham, "On graphs which contain all small trees," *J. Combinatorial Theory (B)*, vol. 24, pp. 14–23, 1978.

[20] F. R. K. Chung and R. L. Graham, "On universal graphs for spanning trees," *J. London Math. Soc.*, vol. 27, pp. 203–211, 1983.

[21] F. R. K. Chung, R. L. Graham, and D. Coppersmith, "On trees which contain all small trees," in *The Theory and Applications of Graphs* (G. Chartrand, Y. Alavi, D. L. Goldsmith, L. Lesniak-Foster, and D. R. Lick, eds.), pp. 265–272, New York: John Wiley & Sons, 1981.

[22] F. R. K. Chung, R. L. Graham, and N. Pippenger, "On graphs which contain all small trees II," in *Proc. 1976 Hungarian Colloquium on Combinatorics*, pp. 213–223, 1978.

[23] F. R. K. Chung, R. L. Graham, and J. Shearer, "Universal caterpillars," *J. Combinatorial Theory (B)*, vol. 31, pp. 348–355, 1981.

[24] F. R. K. Chung, A. L. Rosenberg, and L. Snyder, "Perfect storage representations for families of data structures," *SIAM J. Algebraic and Discrete Methods*, vol. 4, pp. 548–565, 1983.

[25] F. R. K. Chung and P. D. Seymour, "Graphs with small bandwidth and cutwidth," *Discrete Mathematics*, vol. 75, pp. 113–119, 1989.

[26] M. J. Chung, F. Makedon, I. H. Sudborough, and J. Turner, "Polynomial time algorithms for the min cut problem on degree restricted trees," *SIAM J. Computing*, vol. 14, no. 1, pp. 158–177, 1985.

[27] C. J. Colbourn and A. Proskurowski, "Concurrent transmissions in broadcast networks," in *Proc. 11th International Colloquium on Automata, Languages and Programming*, vol. 172 of *Lecture Notes in Computer Science*, pp. 128–136, Springer-Verlag, 1984.

[28] W. W. Dai and M. Sato, "Minimal forbidden minor characterization of planar partial 3-trees and application to circuit layout," in *Proc. International Symposium on Circuits and Systems*, pp. 2677–2681, 1990.

[29] N. Deo, M. S. Krishnamoorthy, and M. A. Langston, "Exact and approximate solutions for the gate matrix layout problem," *IEEE Trans. Computer-Aided Design*, vol. 6, pp. 79–84, 1987.

[30] G. A. Dirac, "A property of 4-chromatic graphs and some remarks on critical graphs," *J. London Math. Soc.*, vol. 27, pp. 85–92, 1952.

[31] E. S. El-Mallah and C. J. Colbourn, "On two dual classes of planar graphs," *Discrete Mathematics*, vol. 80, pp. 21–40, 1990.

[32] J. A. Ellis, I. H. Sudborough, and J. S. Turner, "The vertex separation and search number of a graph," *Information and Computation*, vol. 113, pp. 50–79, 1994.

[33] A. M. Farley, "Networks immune to isolated failures," *Networks*, vol. 11, pp. 255–268, 1981.

[34] A. M. Farley and A. Proskurowski, "Networks immune to isolated line failures," *Networks*, vol. 12, pp. 393–403, 1982.

[35] M. R. Fellows, "The Robertson-Seymour theorems: A survey of applications," in *Graphs and Algorithms* (R. B. Richter, ed.), vol. 89 of *AMS Contemporary Mathematics*, pp. 1–18, American Math. Soc., 1989.

[36] M. R. Fellows and M. A. Langston, "Nonconstructive advances in polynomial-time complexity," *Information Processing Letters*, vol. 26, pp. 157–162, 1987.

[37] M. R. Fellows and M. A. Langston, "On search, decision and the efficiency of polynomial-time algorithms," in *Proc. 21st Annual ACM Symposium on the Theory of Computing*, pp. 501–512, 1989.

[38] H. Friedman, N. Robertson, and P. D. Seymour, "The metamathematics of the graph minor theorem," in *Logic and Combinatorics* (S. G. Simpson, ed.), vol. 65 of *AMS Contemporary Mathematics*, pp. 229–261, American Math. Soc., 1987.

[39] T. Fukuhara, "On structure of planar graphs drawable in k layers," Master's thesis, Tokyo Institute of Technology, 1990. in Japanese.

[40] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness.* New York: Freeman and Co., 1979.

[41] H. H. Glover, P. Huneke, and C. Wang, "103 graphs that are irreducible for the projective plane," *J. Combinatorial Theory (B)*, vol. 27, pp. 332–370, 1979.

[42] M. C. Golumbic, "Interval graphs and related topics," *Discrete Mathematics*, vol. 55, pp. 113–121, 1985.

[43] D. Granot and D. Skorin-Kapov, "On some optimization problems on k-trees and partial k-trees," *Discrete Applied Mathematics*, vol. 48, pp. 129–145, 1994.

[44] J. Gustedt, "On the pathwidth of chordal graphs," *Discrete Applied Mathematics*, vol. 45, pp. 233–248, 1993.

[45] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Design Automation Conference*, pp. 155–169, 1971.

[46] D. S. Johnson, "The NP-completeness column: an ongoing guide," *J. Algorithms*, vol. 8, pp. 285–303, 1987.

[47] T. Kashiwabara and T. Fujisawa, "NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph," in *Proc. International Symposium on Circuits and Systems*, pp. 657–660, 1979.

[48] N. G. Kinnersley, "The vertex separation number of a graph equals its path-width," *Information Processing Letters*, vol. 42, pp. 345–350, 1992.

[49] N. G. Kinnersley and M. A. Langston, "Obstruction set isolation for the gate matrix layout problem," *Discrete Applied Mathematics*, vol. 54, pp. 169–213, 1994.

[50] L. M. Kirousis and C. H. Papadimitriou, "Interval graphs and searching," *Discrete Mathematics*, vol. 55, pp. 181–184, 1985.

[51] L. M. Kirousis and C. H. Papadimitriou, "Searching and pebbling," *Theoretical Computer Science*, vol. 47, pp. 205–218, 1986.

[52] A. Kornai and Z. Tuza, "Narrowness, pathwidth, and their application in natural language processing," *Discrete Applied Mathematics*, vol. 36, pp. 87–92, 1992.

[53] J. Lagergren, "Efficient parallel algorithms for tree-decomposition and related problems," in *Proc. 31st Annual Symposium on Foundations of Computer Science*, pp. 173–182, 1990.

[54] A. S. LaPaugh, "Recontamination does not help to search a graph," *Journal of the Association for Computing Machinery*, vol. 40, no. 2, pp. 224–245, 1993.

[55] C. E. Leiserson, "Area-efficient graph layouts (for VLSI)," in *Proc. 21st Annual Symposium on Foundations of Computer Science*, pp. 270–281, 1980.

[56] T. Lengauer, "Black-white pebbles and graph separation," *Acta Informatica*, vol. 16, pp. 465–475, 1981.

[57] T. Lengauer, "Upper and lower bounds on the complexity of the min-cut linear arrangement problem on trees," *SIAM J. Algebraic and Discrete Methods*, vol. 3, pp. 99–113, 1982.

[58] G. T. Lingner, T. Politof, and A. Satyanarayana, "A forbidden minor characterization and reliability of a class of partial 4-trees," *Networks*, vol. 25, pp. 139–146, 1995.

[59] R. J. Lipton and R. E. Tarjan, "Applications of a planar separator theorem," *SIAM J. Computing*, vol. 9, no. 3, pp. 615–627, 1980.

[60] F. Makedon and I. H. Sudborough, "On minimizing width in linear layouts," *Discrete Applied Mathematics*, vol. 23, pp. 243–265, 1989.

[61] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, "The complexity of searching a graph," *Journal of the Association for Computing Machinery*, vol. 35, pp. 18–44, January 1988.

[62] R. H. Möhring, "Graph problems related to gate matrix layout and PLA folding," in *Computational Graph Theory* (G. Tinhofer, E. Mayr, H. Noltemeier, and M. Syslo, eds.), vol. 7 of *Computing suppl.*, pp. 17–51, Wien, New York: Springer-Verlag, 1990.

[63] B. Monien and I. H. Sudborough, "Min cut is NP-complete for edge weighted trees," *Theoretical Computer Science*, vol. 58, pp. 209–229, 1988.

[64] E. M. Neufeld and C. J. Colbourn, "The most reliable series-parallel networks," *Networks*, vol. 15, pp. 27–32, 1985.

[65] T. Ohtsuki, H. Mori, E. S. Kuh, T. Kashiwabara, and T. Fujisawa, "One-dimensional logic gate assignment and interval graphs," *IEEE Trans. Circuits and Systems*, vol. 26, pp. 675–684, 1979.

[66] T. D. Parsons, "Pursuit-evasion in a graph," in *Theory and Applications of Graphs* (Y. Alavi and D. R. Lick, eds.), vol. 642 of *Lecture Notes in Mathematics*, pp. 426–441, Michigan: Springer-Verlag, 1976.

[67] N. Pippenger, "Pebbling," Tech. Rep. RC8275, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1980.

[68] A. Proskurowski, "Separating subgraphs in k-trees: cables and caterpillars," *Discrete Mathematics*, vol. 49, pp. 275–285, 1984.

[69] B. A. Reed, "Finding approximate separators and computing tree width quickly," in *Proc. 21st Annual ACM Symposium on the Theory of Computing*, pp. 221–228, 1992.

[70] N. Robertson, P. Seymour, and R. Thomas, "Quickly excluding a planar graph," *J. Combinatorial Theory (B)*, vol. 62, pp. 323–348, 1994.

[71] N. Robertson and P. D. Seymour, "Graph minors. I. Excluding a forest," *J. Combinatorial Theory (B)*, vol. 35, pp. 39–61, 1983.

[72] N. Robertson and P. D. Seymour, "Graph minors. II. Algorithmic aspects of tree-width," *J. Algorithms*, vol. 7, pp. 309–322, 1986.

[73] N. Robertson and P. D. Seymour, "Graph minors. III. Planar tree-width," *J. Combinatorial Theory (B)*, vol. 36, pp. 49–64, 1984.

[74] N. Robertson and P. D. Seymour, "Graph minors. IV. Tree-width and well-quasi-ordering," *J. Combinatorial Theory (B)*, vol. 48, pp. 227–254, 1990.

[75] N. Robertson and P. D. Seymour, "Graph minors. V. Excluding a planar graph," *J. Combinatorial Theory (B)*, vol. 41, pp. 92–114, 1986.

[76] N. Robertson and P. D. Seymour, "Graph minors. VI. Disjoint paths across a disc," *J. Combinatorial Theory (B)*, vol. 41, pp. 115–138, 1986.

[77] N. Robertson and P. D. Seymour, "Graph minors. VII. Disjoint paths on a surface," *J. Combinatorial Theory (B)*, vol. 45, pp. 212–254, 1988.

[78] N. Robertson and P. D. Seymour, "Graph minors. VIII. A Kuratowski theorem for general surfaces," *J. Combinatorial Theory (B)*, vol. 48, pp. 255–288, 1990.

[79] N. Robertson and P. D. Seymour, "Graph minors. IX. Disjoint crossed paths," *J. Combinatorial Theory (B)*, vol. 49, pp. 40–77, 1990.

[80] N. Robertson and P. D. Seymour, "Graph minors. X. Obstructions to tree-decomposition," *J. Combinatorial Theory (B)*, vol. 52, pp. 153–190, 1991.

[81] N. Robertson and P. D. Seymour, "Graph minors. XI. Circuits on a surface," *J. Combinatorial Theory (B)*, vol. 60, pp. 72–106, 1994.

[82] N. Robertson and P. D. Seymour, "Graph minors. XII. Distance on a surface," *J. Combinatorial Theory (B)*, vol. 64, pp. 240–272, 1995.

[83] N. Robertson and P. D. Seymour, "Graph minors. XIII. The disjoint paths problem," *J. Combinatorial Theory (B)*, vol. 63, pp. 65–110, 1995.

[84] N. Robertson and P. D. Seymour, "Graph minors. XII. Excluding a non-planar graph." manuscript, 1986.

[85] N. Robertson and P. D. Seymour, "Graph minors. XIV. Taming a vortex." manuscript, 1987.

[86] N. Robertson and P. D. Seymour, "Graph minors. XVI. Wagner's conjecture." manuscript, 1987.

[87] N. Robertson and P. D. Seymour, "Disjoint paths - a survey," *SIAM J. Algebraic and Discrete Methods*, vol. 6, pp. 300–305, 4 1985.

[88] N. Robertson and P. D. Seymour, "Graph minors - a survey," in *Surveys in Combinatorics* (I. Anderson, ed.), pp. 153–171, London: Cambridge University Press, 1985.

[89] A. R. Rosenberg, L. J. Stockmeyer, and L. Snyder, "Uniform data encodings," *Theoretical Computer Science*, vol. 2, pp. 145–165, 1980.

[90] A. Satyanarayana and L. Tung, "A characterization of partial 3-trees," *Networks*, vol. 20, pp. 299–322, 1990.

[91] P. Scheffler, "A linear algorithm for the pathwidth of trees," in *Topics in Combinatorics and Graph Theory* (R. Bodendiek and R. Henn, eds.), pp. 613–620, Heidelberg: Physica-Verlag, 1990.

[92] S. Shinoda, "On some problems of graphs — including Kajitani's conjecture and its solution —," in *Proc. 2nd Karuizawa Workshop on Circuits and Systems*, pp. 414–418, 1989. in Japanese.

[93] M. Takeuchi, S. Soejima, and W. Kishimoto, "A characterization of trees contained in a generalized fan in terms of forbidden subgraphs," *Trans. IEICE*, vol. J75-A, no. 7, pp. 1216–1219, 1992. in Japanese.

[94] L. Tesnière, *Eléments de Syntaxe Structurale*. Paris: Klincksieck, 1959.

[95] S. Ueno, "Path-width of graphs," in *Discrete Structures and Algorithms* (S. Fujishige, ed.), vol. 1, ch. 5, pp. 213–230, Kindai-Kagaku-Sha, 1992. in Japanese.

[96] S. Ueno, T. Kitakaku, and Y. Kajitani, "Universal graphs for 2-paths," in *Proc. International Symposium on Circuits and Systems*, pp. 1147–1151, 1991.

[97] L. G. Valiant, "Universality considerations in VLSI circuits," *IEEE Trans. Computers*, vol. C-30, pp. 135–140, 1981.

[98] K. Wagner, "Bemerkung zu hadwigers vermutung," *Math. Ann.*, vol. 141, pp. 433–451, 1960.

[99] J. A. Wald and C. J. Colbourn, "Steiner trees, partial 2-trees, and minimum IFI networks," *Networks*, vol. 13, pp. 159–167, 1983.

[100] M. Yannakakis, "A polynomial algorithm for the min-cut linear arrangement of trees," *Journal of the Association for Computing Machinery*, vol. 32, no. 4, pp. 950–988, 1985.

Publications

Papers

[1] A. Takahashi, S. Ueno, and Y. Kajitani, "Minimal acyclic forbidden minors for the family of graphs with bounded path-width," *Discrete Mathematics*, vol. 127, pp. 293–304, 1994.

[2] A. Takahashi, S. Ueno, and Y. Kajitani, "Mixed-searching and proper-path-width," *Theoretical Computer Science*, vol. 137, no. 2, pp. 253–268, 1995.

[3] A. Takahashi, S. Ueno, and Y. Kajitani, "On the proper-path-decomposition of trees," *IEICE Trans. Fundamentals*, vol. E78-A, no. 1, pp. 131–136, 1995.

[4] A. Takahashi, S. Ueno, and Y. Kajitani, "Universal graphs for graphs with bounded path-width," *IEICE Trans. Fundamentals*, vol. E78-A, no. 4, pp. 458–462, 1995.

[5] A. Takahashi, S. Ueno, and Y. Kajitani, "Minimal forbidden minors for the family of graphs with proper-path-width at most two," *IEICE Trans. Fundamentals*, vol. E78-A, no. 12, pp. 1828–1839, 1995.

Conference Papers

[1] A. Takahashi, S. Ueno, and Y. Kajitani, "Mixed-searching and proper-path-width," *Proc. 4th Karuizawa Workshop on Circuits and Systems*, pp. 215–220, April 1991.

[2] A. Takahashi, S. Ueno, and Y. Kajitani, "Mixed-searching and proper-path-width," *ISA '91 Algorithms: Proc. 2nd Annual International Symposium on Algorithms*, vol. 557 of *Lecture Notes in Computer Science*, pp. 61–71, Springer-Verlag, December 1991.

[3] A. Takahashi, S. Ueno, and Y. Kajitani, "Universal graphs for graphs with bounded path-width," *Proc. 5th Karuizawa Workshop on Circuits and Systems*, pp. 179–184, April 1992.

[4] A. Takahashi, S. Ueno, and Y. Kajitani, "Universal graphs for graphs with bounded path-width," *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 419–423, December 1992.

Presentations

[1] A. Takahashi, S. Ueno, and Y. Kajitani, "A characterization of the cycle-free k-path in terms of forbidden minors," *The 2nd Japan Conference on Graph Theory and Combinatorics*, August 1990.

[2] A. Takahashi, S. Ueno, and Y. Kajitani, "Minimal acyclic forbidden minors for the family of graphs with bounded path-width," *IPSJ SIG Notes*, vol. 91, no. 11, 91-AL-19-3, January 1991.

[3] A. Takahashi, S. Ueno, and Y. Kajitani, "Path-width and proper-path-width," *International Workshop on Graph and Graph Transformations: Tree-structured graphs, forbidden configurations and graph algorithms*, March 1991.

[4] A. Takahashi, S. Ueno, and Y. Kajitani, "Mixed-searching and proper-path-width," *IPSJ SIG Notes*, vol. 91, no. 69, 91-AL-22-7, July 1991.

[5] A. Takahashi, S. Ueno, and Y. Kajitani, "On the proper-path-decomposition of trees," *IEICE Technical Report*, vol. 91, no. 255, pp. 23–26, CAS91-74, September 1991.

[6] A. Takahashi, S. Ueno, and Y. Kajitani, "Universal graphs for graphs with bounded path-width," *IPSJ SIG Notes*, vol. 91, no. 102, 91-AL-24-3, November 1991.

[7] A. Takahashi, S. Ueno, and Y. Kajitani, "Minimal forbidden minors for the family of graphs with proper-path-width at most two," *IEICE Technical Report*, vol. 92, no. 236, pp. 69–76, CAS92-51, September 1992. in Japanese.