STACK NUMBER, TRACK NUMBER, AND LAYERED PATHWIDTH

CÉLINE YELLE

A thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
Faculty of Engineering

University of Ottawa

ABSTRACT

In this thesis, we consider three parameters associated with graphs : stack number, track number, and layered pathwidth. Our first result is to show that the stack number of any graph is at most 4 times its layered pathwidth. This result complements an existing result of Dujmović et al. that showed that the queue number of a graph is at most 3 times its layered pathwidth minus one (Dujmović, Morin, and Wood [*SIAM J. Comput.*, 553–579, 2005]).

Our second result is to show that graphs of track number at most 3 have layered pathwidth at most 4. This answers an open question posed by Banister et al. (Bannister, Devanny, Dujmović, Eppstein, and Wood [*GD 2016*, 499–510, 2016, *Algorithmica*, 1–23, 2018]).

ACKNOWLEDGMENTS

I would like to thank my thesis advisors Vida Dujmović at the University of Ottawa and Pat Morin at Carleton University. Their guidance led me in the right direction and introduced me to the world of graph drawing. Pat's door was always open, for any questions and our weekly meetings were crucial to keeping myself on schedule. Pat and Vida were always there to respond to any of my questions and worries. Without their help, this thesis would never have been completed, and I am gratefully indebted to them for their very valuable comments on this thesis.

I would like to thank the University of Ottawa, NSERC, and OGS organizations, whose funding allowed my research to occur.

Finally, I must express my thanks to my parents and to my partner Cavin for providing me with unfailing support and continuous encouragement throughout my years of studies, and for their patience throughout the various stages of researching and writing this thesis. This accomplishment would not have been possible without them. My deepest gratitude for their never ending patience and support.

CONTENTS

LIST OF TABLES AND FIGURES

1

INTRODUCTION

This thesis addresses the relationship between three types of graph drawing layouts: layered path decompositions, stack layouts and track layouts. In this chapter, we introduce the types of graph layout by presenting an informal definition and discussing the relationships that exist between these graph layouts and popular classes of graphs. We also discuss the applications of various types of graph layouts as well as survey some of the related research. In Chapter 2, we present more formal definitions and provide concrete visual examples.

1.1 GRAPH DECOMPOSITIONS

Layered path decompositions are a type of graph drawing layout that were first introduced by Banister et al. [3, 4]. Layered path decompositions are a generalized version of path decompositions. A *path decomposition* is a graph layout where the vertices of a graph are placed in a sequence of bags such that if uv is an edge in the graph, then both vertices u and v must be placed in a bag together and such that all the bags that contain a particular vertex v_i must be consecutive. The *width* of a path decomposition is the size of its largest bag minus 1. The *pathwidth* is the minimum width of any path decomposition. In a *layered path decomposition*, the vertices are also partitioned into disjoint sets called *layers*. Edges in the graph can only connect vertices within one layer or within subsequent layers. The *width* of a layered path decomposition is the size of the largest intersection of a bag and a layer. The *layered pathwidth* is the minimum width over all layered path decompositions.

A tree decomposition is defined very similarly to a path decomposition. A *tree decomposition* is given by a tree whose nodes index a set of bags. Similarly to the path decomposition, if uv is an edge in the graph, then both vertices u and v must be placed in a bag and all bags that contain v_i must form a connected subtree. The *width* of a tree decomposition is equal to the size of the largest bag minus 1, and the *treewidth* is the minimum width over all tree decomposition. The concept of treewidth was first introduced under the name of *dimension* by Bertelé and Brioschi in 1972 [8]. The concept was then reintroduced by Halin in 1976 [34] and rediscovered again by Roberston and Seymour in 1984 [51]. The terms *treewidth* and *tree decomposition* were first introduced in Robertson and Seymour's work. Layered tree decomposition is a comparatively newer concept which was introduced by Dujmović, Morin and Wood in 2013 [24]. In a *layered tree decomposition*, the vertices of a tree decomposition are also partitioned into disjoint sets called *layers*. Edges in the graph can only connect vertices within one layer or within subsequent layers. The *width* of a layered tree decomposition is the size of the largest intersection of a bag and a layer. The *layered treewidth* is the minimum width over all layered tree decompositions.

Note that while layered pathwidth is at most pathwidth, pathwidth is not bounded by layered pathwidth. There are graphs—for example the $n \times n$ planar grid—that have unbounded pathwidth and bounded layered pathwidth. Thus upper bounds proved in terms of layered pathwidth are quantitatively stronger than those proved in terms of pathwidth. In addition, while having pathwidth at most k is a minor-closed property, having layered pathwidth at most k is not. For example, the $2 \times n \times n$ 3-dimensional grid graph has layered pathwidth at most 3, but it has K_n as a minor, and thus it has a minor of unbounded layered pathwidth. An analogous statement holds for layered treewidth as well.

In the table 1.1 we have summarized the best known bounds for pathwidth and layered pathwidth with regards to specific classes of graphs. In the table 1.2, we have also summarized the relationships known for pathwidth and layered pathwidth in regards to other graph layouts. If no bounds are known, we have simply omitted the results in order to keep the tables pertinent and brief. These tables also contain the bounds for treewidth, layered treewidth, stack number, queue number and track number.

Pathwidth and treewidth are central to the fields of structural and algorithmic graph theory [13, 36, 49, 30]. Notably, in the field of three-dimensional graph drawing, Dujmović et al. showed that graphs with bounded pathwidth could be drawn as a three-dimensional straight-line grid drawing with $O(n)$ volume [22]. Pathwidth can be applied to the domain of natural language processing [42]. Treewidth also has applications in complexity theory. For instance, it was shown by several authors that there are many NP-Hard problems that can be solved in polynomial time on inputs with bounded treewidth [2, 10, 30].

Layered treewidth and layered pathwidth have been shown to have applications in the domains of intersection [53] and structural graph theory [21] as well as nonrepetitive graph colouring [24]. Other applications of layered treewidth include the study of queue layouts, track layouts [18, 24] as well as stack layouts [19]. Similar applications hold for layered pathwidth, namely we can use bounded layered pathwidth to obtain bounds on stack number (the proof of which is presented in Section 3), queue number [39] and track number [4].

1.2 STACK AND QUEUE LAYOUTS

A *stack layout* of a graph G consists of a total order σ of $V(G)$ and a partition of $E(G)$ into sets, called *stacks*, such that no two edges in the same stack *cross*; that is, there are no edges vw and xy in a single stack with $v \prec_\sigma x \prec_\sigma w \prec_\sigma y$. The minimum number of stacks in a stack layout of G is the *stack number* of G and is denoted by $\mathrm{sn}(G)$. A stack layout is also called a *book embedding* and stack number is also referred to as *book thickness* or *page number*. An *s-stack graph* is a graph that has a stack layout with at most s stacks. Queue layouts are defined similarly however, $E(G)$ is partitioned into sets, called *queues*, such that no two edges in the same queue *nest*; which is to say that there are no edges vw and xy in a single queue with $v \prec_\sigma x \prec_\sigma y \prec_\sigma w$. The minimum number of queues in a queue layout is the *queue number* of G. Stack layouts were first introduced by Ollmann in 1973 [43]. Queue layouts were later defined by Heath and Rosenberg in 1992 [40].

Currently, it is an open problem whether stack number and queue number are bounded by one another. Heath et al. proved that every graph with a 1-queue layout, has a 2-stack layout and that every graph with a 1-stack layout has a 2-queue layout [39]. Yannakakis proved that planar graphs could be embedded in 4 stacks in 1989 [62]. Until recently, It remained an open problem, whether planar graphs had bounded queue number. In 1992, Heath, Leighton and Rosenberg conjectured that queue number was bounded for planar graphs [39]. In 2019, Dujmović et al. proved this conjecture, showing that planar graphs could be embedded in 49 queues [20]. However, this bound is not tight, since the best lower bound on the queue number of planar graphs is 4 which was shown by Alam et al. [1].

Stack layouts have applications in sorting permutations [28, 35, 44], complexity theory [31, 41], and graph drawing [9, 60, 61]. Applications of queue layouts include: scheduling parallel processors [40, 46], fault-tolerant processing [47], parallel matrix computations [45], and sorting networks [54]. Queue layouts have also contributed to the field of graph drawing, notably, graphs of bounded queue number have $O(1) \times O(1) \times O(n)$ volume [27].

1.3 TRACK LAYOUTS

Track layouts are a type of graph layout closely related to queue layouts. The terms track layout and track number were first defined by Dujmović et al. [23] in 2005. A *t-track layout* of a graph is a partition of the vertices into t sets $V_1 \ldots V_t$ called *tracks* such that each track is a linear order $\prec_1 \ldots \prec_t$ and such that there are no edges vw, xy such that $v, x \in V_i$ and $w, y \in V_j$ with $v \prec_i x$ and $y \prec_j w$. The track number is the minimum t for which G has a t-track layout and is denoted by $\text{tn}(G)$.

Track layouts are closely related to queue layouts since track number is bounded by queue number and queue number, in turn, is bounded by track number. Dujmović et al. showed that a graph with track-number k has queue number at most $k - 1$ [23]. Dujmović et al. also established that the track number of a graph G is at most $2^{O(\text{qn}(G)^2)}$ [25].

Since track number is closely related to queue number, we find that many applications of queue layouts also apply to track layouts.

1.4 BOUNDS AND RELATIONSHIPS

In this section, we summarize the known bounds for the graph layout measures discussed above. Table 1.1 reviews the best known bounds on the graph invariants for various classes of graphs. Table 1.2 summarizes the relationships known for the graph layouts discussed above in relation to the other graph invariants. The classes of graphs we chose to include are Halin graphs, trees, X-trees, planar, outer planar, series parallel, arched levelled planar, and bipartite graphs.

Below, we will define the less commonly known classes of graphs. Halin graphs are a special type of planar graphs. A Halin graph is constructed by connecting the leaves of a tree into a cycle. The tree must have at least four vertices where none of the vertices have degree equal to 2. X-trees are also a type of planar graphs. An X-tree is constructed by taking the complete binary tree on $2^{n+1} - 1$ vertices and adding paths connecting each consecutive vertices at level i, from left to right, for $1 \leq i \leq n$. Outerplanar graphs are planar graphs where the vertices of the graph can be placed on the outer face of the figure. A series parallel graph has a source vertex s and a target vertex t and can be transformed into the K_2 graph by repeating the following operations: (1) replace a pair of parallel edges with a single edge; and (2) collapse a pair of edges connected to a vertex of degree 2 other than s or t into a single edge. Finally, arched levelled planar graphs are levelled planar graphs which have been augmented with arches that connect vertices on the same level. Arched levelled planar graphs can be embedded in the plane in such a way that none of the arches cross each other.

Pathwidth		
Halin	$\mathrm{pw}(G) = O(\log(n))$	(Bounded by treewidth)
tree	$\mathrm{pw}(G) \leq \log_3(2n+1)$	[52, Theorem 5]
planar	$\mathrm{pw}(G) \leq \sqrt{n}$	[13, Corollary 23]
outerplanar	$\mathrm{pw}(G) = O(\log(n))$	(Bounded by treewidth)
series parallel	$\mathrm{pw}(G) = O(\log(n))$	(Bounded by treewidth)

Treewidth		
Halin	$\mathrm{tw}(G) \leq 3$	[11, Theorem 4.7]
tree	$\mathrm{tw}(G) = 1$	[12, Proposition 3.1]
planar	$\mathrm{tw}(G) \leq \sqrt{n}$	(Bounded by pathwidth)
outerplanar	$\mathrm{tw}(G) \leq 2$	[13, Lemma 78]
series parallel	$\mathrm{tw}(G) \leq 2$	[57]

Layered pathwidth		
Halin	$\mathrm{lpw}(G) \leq 2$	[4, Corollary 18]
tree	$\mathrm{lpw}(G) = 1$	[21, Theorem 28]
planar	$\mathrm{lpw}(G) \leq \sqrt{n}$	(Bounded by pathwidth)
outerplanar	$\mathrm{lpw}(G) \leq 2$	[4, Corollary 16]
arched levelled planar	$\mathrm{lpw}(G) \leq 2$	[4, Theorem 3]

Layered treewidth		
Halin	$\mathrm{ltw}(G) \leq 2$	(Bounded by layered pathwidth)
tree	$\mathrm{ltw}(G) = 1$	(Bounded by treewidth)
planar	$\mathrm{ltw}(G) \leq 3$	[24, Theorem 17]
outerplanar	$\mathrm{ltw}(G) \leq 2$	(Bounded by treewidth)
series parallel	$\mathrm{ltw}(G) \leq 2$	(Bounded by treewidth)

Stack number		
Halin	$\text{sn}(G) \leq 2$	[32, Theorem 1]
tree	$\text{sn}(G) = 1$	[7, Theorem 2.5]
2-trees	$\text{sn}(G) \leq 2$	[50, Theorem 3]
X-trees	$\text{sn}(G) \leq 2$	[14, Proposition 3.3]
planar	$\text{sn}(G) \leq 4$	[62, Theorem 2]
planar bipartite	$\text{sn}(G) \leq 2$	[15] [45, Corollary 2.2]
planar 3-tree	$\text{sn}(G) \leq 3$	[38]
outerplanar	$\text{sn}(G) = 1$	[7, Theorem 2.5]
series-parallel	$\text{sn}(G) \leq 2$	[50]
arched levelled planar	$\text{sn}(G) \leq 2$	[39, Theorem 4.2]

Queue number		
Halin	$\text{qn}(G) \geq 2$	[32, Theorem 2]
	$\text{qn}(G) \leq 3$	
tree	$\text{qn}(G) = 1$	[40]
2-trees	$\text{qn}(G) \leq 3$	[50, Theorem 9]
X-trees	$\text{qn}(G) \leq 2$	[40]
planar	$\text{qn}(G) \geq 4$	[1, Theorem 2]
	$\text{qn}(G) \leq 49$	[20, Theorem 1]
	$\text{qn}(G) = O(\log n)$	[18, Corollary 2]
planar bipartite	$\text{qn}(G) \leq 49$	[20, Theorem 1]
planar 3-tree	$\text{qn}(G) \geq 4$	[1, Theorem 2]
	$\text{qn}(G) \leq 5$	[1, Theorem 1]
outerplanar	$\text{qn}(G) \leq 2$	[39, Theorem 4.3]
	$\text{qn}(G) \geq 2$	[40]
outerplanar bipartite	$\text{qn}(G) = 1$	[4]
series-parallel	$\text{qn}(G) \leq 3$	[50, Theorem 9]
	$\text{qn}(G) \geq 3$	[58]
arched levelled planar	$\text{qn}(G) = 1$	[40, Theorem 3.2]

Track number		
Halin	$\text{tn}(G) \leq 6$	[4, Lemma 8, Corollary 15]
	$\text{tn}(G) \geq 5$	[48]
tree	$\text{tn}(G) = 3$	[29, Theorem 3]

2-trees	$\text{tn}(G) \geq 5$	[16, Theorem 5]
	$\text{tn}(G) \leq 18$	[16, Theorem 9]
planar	$\text{tn}(G) \geq 7$	[25, Theorem 13]
	$\text{tn}(G) \leq 461,184,080$	[20, Theorem 52]
planar bipartite	$\text{tn}(G) \leq 461,184,080$	[20, Theorem 52]
planar 3-tree	$\text{tn}(G) \geq 7$	[16, Theorem 5]
	$\text{tn}(G) \leq 4,000$	[1, Corollary 1]
outerplanar	$\text{tn}(G) = 5$	[25, Lemma 22]
outerplanar bipartite	$\text{tn}(G) = 3$	[4, Theorem 3]
series-parallel	$\text{tn}(G) \geq 6$	[23]
	$\text{tn}(G) \leq 15$	[16, Theorem 9]
arched levelled planar	$\text{tn}(G) \leq 5$	[17, Theorem 2]

Table 1.1: Bounds for common classes of graphs

Bounds on pathwidth		
treewidth	$\text{pw}(G) \leq (\text{tw}(G) + 1) \log n)$	[13, Corollary 24]
layered pathwidth	unbounded	(Banister et al. showed that several classes of graphs have bounded layered pathwidth but unbounded pathwidth [4])
layered treewidth	unbounded	(The complete binary tree has $\text{ltw}(G) = 3$, but unbounded pathwidth)
stack number	$\text{pw}(G) \leq \text{sn}(G) \log n$	[13, Corollary 24] [56]
queue number	$\text{pw}(G) \leq \text{qn}(G) \log n$	[13, Corollary 24] [58, Theorem 3]
track number	unbounded for $\text{tn}(G) \geq 3$	(The complete binary tree has $\text{tn}(G) = 3$, but unbounded pathwidth)

Bounds on treewidth		
pathwidth	$\text{tw}(G) \leq \text{pw}(G)$	[13, Lemma 3]
layered pathwidth	$\exists G : \text{lpw}(G) = 1, \text{tw}(G) = \Omega(n)$	(G is the $n \times n$ planar grid)

layered treewidth	unbounded	(Dujmović et al. showed that planar graphs have ltw ≤ 3 but unbounded treewidth) [24, Theorem 17]
stack number	$\mathrm{tw}(G) \leq \mathrm{sn}(G) - 1$	[56]
queue number	$\mathrm{tw}(G) \leq \mathrm{qn}(G) - 1$	[58, Theorem 3]
track number	$\exists G : \mathrm{tn}(G) \geq 3$, $\mathrm{tw}(G) = \Omega(n)$	(G is the $n \times n$ planar grid)

Bounds on layered pathwidth		
pathwidth	$\mathrm{lpw}(G) \leq \mathrm{pw}(G)$	(We can make a trivial layering where all vertices are placed in one layer)
treewidth	$\mathrm{lpw}(G) \leq (\mathrm{tw}(G) + 1) \log n$	[13, Corollary 23]
layered treewidth	$\mathrm{lpw}(G) \leq \mathrm{ltw}(G) \log_3(2n + 1)$	[4, Lemma 10]
stack number	$\mathrm{sn}(G) \leq 1 \Rightarrow \mathrm{lpw}(G) \leq 2$	[4, Corollary 16]
	$\exists G : \mathrm{sn}(G) = 2$, $\mathrm{lpw}(G) = \Omega(\log n)$	(G is a binary tree plus an apex vertex)
	$\exists G : \mathrm{sn}(G) = 3$, $\mathrm{lpw}(G) = \Omega(n / \log n)$	[26, Theorem 1.5]
queue number	$\mathrm{qn}(G) = 1 \Rightarrow \mathrm{lpw}(G) \leq 2$	[40, Theorem 3.2][4, Corollary 7]
	$\exists G : \mathrm{qn}(G) = 2$, $\mathrm{lpw}(G) = \Omega(n / \log n)$	(There exists 3-monotone expanders with qn $= 2$ but they have pw $\Omega(n)$ and diameter $O(\log n)$, so their lpw is $\Omega(n / \log n)$)
track number	$\mathrm{tn}(G) = 1 \Rightarrow \mathrm{lpw}(G) = 1$	(G has no edges)
	$\mathrm{tn}(G) \leq 2 \Rightarrow \mathrm{lpw}(G) \leq 2$	(G is a forest of caterpillars)
	$\mathrm{tn}(G) \leq 3 \Rightarrow \mathrm{lpw}(G) \leq 4$	**Theorem 4.1**
	$\exists G : \mathrm{tn}(G) = 4$, $\mathrm{lpw}(G) = \Omega((n / \log n)$	[26, Theorem 1.5]

Bounds on layered treewidth		
pathwidth	$\mathrm{ltw}(G) \leq \mathrm{pw}(G) + 1$	(By transitivity: $\mathrm{ltw}(G) \leq \mathrm{tw}(G) + 1$ and $\mathrm{tw}(G) \leq \mathrm{pw}(G)$)
treewidth	$\mathrm{ltw}(G) \leq \mathrm{tw}(G) + 1$	(We can make a trivial layering where all vertices are placed in one layer)
layered pathwidth	$\mathrm{ltw}(G) \leq \mathrm{lpw}(G)$	(A valid layered path decomposition is also a valid layered tree decomposition)
stack number	$\exists G : \mathrm{sn}(G) \geq 3, \mathrm{ltw}(G) = \Omega(\frac{n+2}{3})$	(G is the $n \times n$ grid with a dominant vertex)
queue number	$\exists G : \mathrm{qn}(G) = 2, \mathrm{ltw}(G) = \Omega(n/\log n)$	(There exists 3-monotone expanders with $\mathrm{qn} = 2$ but they have tw $\Omega(n)$ and diameter $O(\log n)$, so their ltw is $\Omega(n/\log n)$)
track number	$\exists G : \mathrm{tn}(G) = 4, \mathrm{ltw}(G) = \Omega(n/\log n)$	(There exists 3-monotone expanders with $\mathrm{tn} = 4$ but they have tw $\Omega(n)$ and diameter $O(\log n)$, so their ltw is $\Omega(n/\log n)$)

Bounds on stack number		
pathwidth	$\mathrm{sn}(G) \leq \mathrm{pw}(G) + 1$	By [33, Theorem 1] and [13, Lemma 3]
treewidth	$\mathrm{sn}(G) \leq \mathrm{tw}(G) + 1$	[33, Theorem 1]
	$\mathrm{sn}(G) \geq \mathrm{tw}(G) + 1$	[56, Theorem 1.1]
layered pathwidth	$\mathrm{sn}(G) \leq 4\,\mathrm{lpw}(G)$	**Theorem 3.1**
layered treewidth	$\mathrm{sn} \leq 4\,\mathrm{ltw}(G)(\log_3(2n+1))$	**Theorem 3.1** and [4, Lemma 10]
	Open Problem 1	
queue number	$\mathrm{qn}(G) = 1 \Rightarrow \mathrm{sn}(G) \leq 2$	[39, Theorem 4.2]

	Open Problem 2	(It is still an open problem whether stack number is bounded by queue number for qn > 1)
track number	**Open Problem 3**	

Bounds on queue number		
pathwidth	$\text{qn}(G) \leq \text{pw}(G)$	[59, Lemma 2]
treewidth	$\text{qn} \leq 36\,\text{tw}(G)\Delta(G)$	[59, Lemma 3]
	$\text{qn}(G) \leq 2^{\text{tw}(G)} - 1$	[58, Theorem 1]
layered pathwidth	$\text{qn}(G) \leq 3\,\text{lpw}(G) - 1$	[23, Theorem 2.6] [4, Lemma 9]
layered treewidth	**Open Problem 4**	
stack number	$\text{sn}(G) = 1 \Rightarrow \text{qn}(G) \leq 2$	[39, Theorem 4.3]
	Open Problem 5	(It is still an open problem whether queue number is bounded by stack number for sn > 1)
track number	$\text{qn}(G) \leq \text{tn}(G) - 1$	[23, Theorem 2.6]

Bounds on track number		
pathwidth	$\text{tn}(G) \leq \text{pw}(G) + 1$	[23, Lemma 3.2]
treewidth	$\text{tn}(G) \leq 1 + (\text{tw}(G) + 1) \log n$	[23, Theorem 2.4]
	$\text{tn}(G) \geq \frac{1}{2}(\text{tw}(G) + 1)(\text{tw}(G) + 2)$	[23, Theorem 7.4]
layered pathwidth	$\text{tn}(G) \leq 3\,\text{lpw}(G)$	[4, Lemma 9]
layered treewidth	$\text{tn}(G) \leq 3\,\text{ltw}(G) \log_3(2n + 1)$	[4, Theorem 11]
queue number	$\text{tn}(G) \leq 2^{O(\text{qn}(G)^2)}$	[25, Theorem 8]
stack number	**Open Problem 6**	

Table 1.2: Relationships between graph layouts

1.5 NEW RESULTS

As previously mentioned, this thesis delves into the relationship between layered pathwidth and two graph layouts: stack layouts and track layouts. In Chapter 3, we prove that the stack number of a graph is at most four times its layered pathwidth.

It was known that graphs that have bounded layered pathwidth have bounded queue number and since track number is bounded by queue number, graphs with bounded layered pathwidth also have bounded track number. It was shown by Bannister et al. that a graph's track number is at most $3\,\mathrm{lpw}(G) - 1$ [4]. In the other direction, graphs with track number at most 2 are forests of caterpillars and have layered pathwidth at most 2. However, the existence of 4-track graphs that are expanders implies that there are 4-track graphs whose layered pathwidth is $\Omega(n/\log n)$. These results leave a gap for 3-track graphs. In their paper, Bannister et al. conjectured that 3-track graphs do in fact have bounded layered pathwidth [4]. The second result of this thesis, is to confirm this conjecture by showing that 3-track graphs have layered pathwidth at most 4. The proofs presented in this thesis establish the relationship between these different types of graph drawing layouts.

2

PRELIMINARIES

In this section, we will present the formal definitions and notations and introduce key concepts that pertain to our proofs. We will first introduce path decompositions, and their general form: tree decomposition.

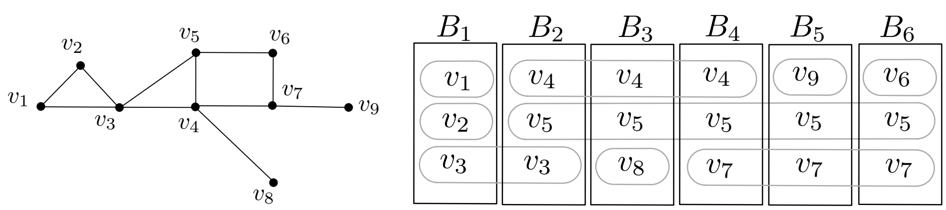
2.1 GRAPH DECOMPOSITIONS

A *path decomposition* of a graph G is a sequence of subsets of $V(G)$ called bags $(B_1, B_2, ...B_n)$ that satisfies the following properties:

1. $\bigcup_{i=1}^{n} B_i = V(G)$
2. For each edge $uv \in E(G)$ there exists a bag B_i such that $u \in B_i$ and $v \in B_i$
3. If $u \in B_i$ and $u \in B_k$ for $i < k$ then $u \in B_j$ for all $j \in \{i, \ldots, k\}$

The *width* of a path decomposition is the size of the largest bag. The *pathwidth* of a graph G, denoted by $\text{pw}(G)$ is defined as the minimum width of any path decomposition of G minus 1.

Figure 2.1 demonstrates an example of a graph and its path decomposition of pathwidth 2. We separate the graph's vertices into a sequence of bags such that if an edge uv exists in G_1 then both vertices u and v must appear in a bag together. Furthermore, if a vertex u is contained in a bag B_i and is contained in a bag B_k then u must be present in every bag between B_i and B_k in the path decomposition.



(a) Graph G_1

(b) Path decomposition of G_1 with pathwidth 2

Figure 2.1: An example graph $G_1 = (V_1, E_1)$ and a path decomposition of G_1 with $\text{pw}(G_1) = 2$

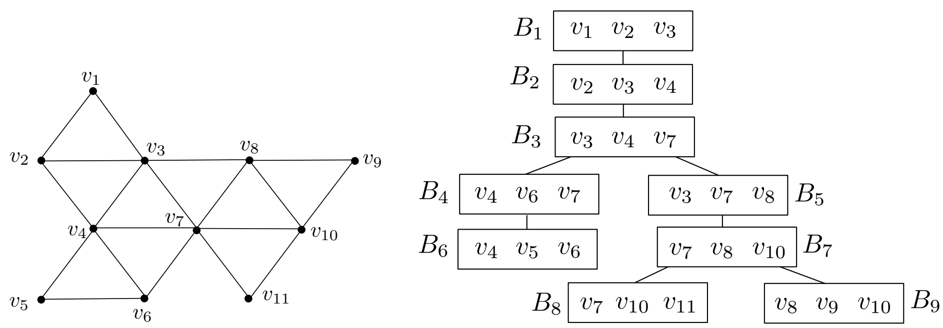
A *tree decomposition* of a graph G is given by a tree T whose nodes index a collection of subsets of $V(G)$ into bags $(B_1, B_2, ...B_n)$ that satisfies the following properties:

1. $\bigcup_{i=1}^{n} B_i = V(G)$

2. For each edge $uv \in E(G)$ there exists a bag B_i such that $u \in B_i$ and $v \in B_i$

3. For each $v \in V(G)$, the set $T(v)$ of bags that contain v induce a non-empty connected subtree in T

Similarly, the *treewidth* of a graph G, denoted by $\text{tw}(G)$ is defined as the minimum width of any tree decomposition of G minus 1. The Figure 2.2 demonstrates a graph and a tree decomposition with treewidth 2.



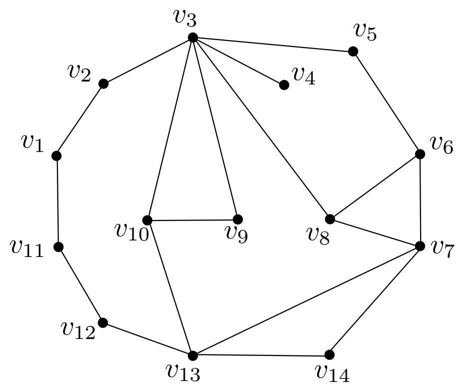
(a) Graph G_2

(b) Tree decomposition of G_2 with treewidth 2

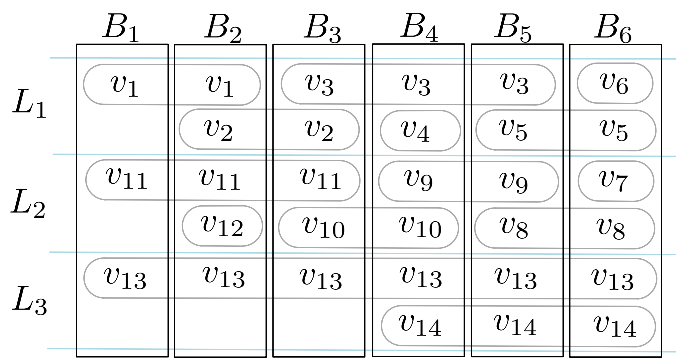
Figure 2.2: An example graph $G_2 = (V_2, E_2)$ and a tree decomposition of G_2 with $\text{tw}(G_2) = 2$

Next, we will establish the definitions of a layering as well as the concepts of layered path decompositions and layered tree decompositions. A *layering* of G is a mapping of G's vertices to a sequence of disjoint sets which we will call layers. This layering will be denoted by $\ell : V(G) \to \mathbb{Z}$ and has the property that if $vw \in E(G)$ then $|\ell(u) - \ell(v)| \leq 1$. We denote *layer* i as $L_i = \{v \in V(G) : \ell(v) = i\}$.

A *layered path decomposition* consist of a layering and a path decomposition. The *width* of a layered path decomposition is the size of the largest intersection of a bag and a layer. The *layered pathwidth* of G is defined as the minimum width of any layered path decomposition on G. We will use the notation $\text{lpw}(G)$ to denote the layered pathwidth of a graph G. An example of a graph and its layered path decomposition is presented in Figure 2.3.



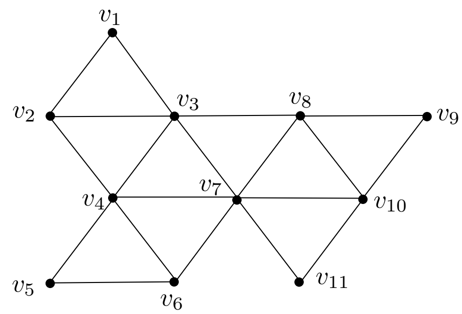
(a) Graph G_3



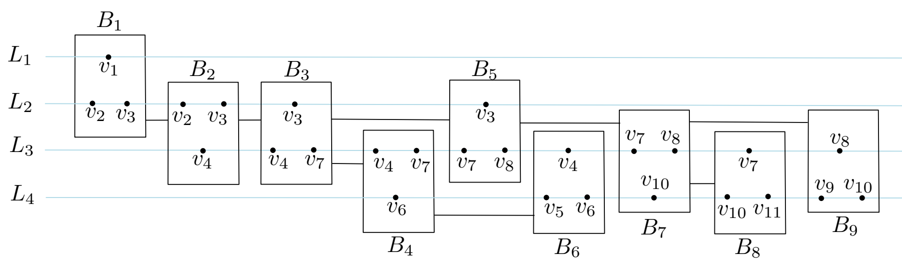
(b) Layered path decomposition of G_3 with layered pathwidth 2

Figure 2.3: An example graph $G_3 = (V_3, E_3)$ and a layered path decomposition of G_3 of width 2

Likewise, a *layered tree decomposition* consists of a layering and a tree decomposition. The *width* of a layered tree decomposition is the size of the largest intersection of a bag and a layer. The *layered treewidth* of G is defined as the minimum width of any layered tree decomposition on G and will be denoted by $ltw(G)$. An example of a graph and layered tree decomposition is presented in Figure 2.4.



(a) Graph G_2



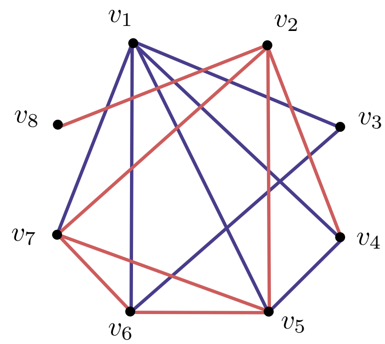
(b) Layered tree decomposition of G_2 with layered treewidth 2

Figure 2.4: An example graph $G_2 = (V_2, E_2)$ and a tree decomposition of G_2 of width 2

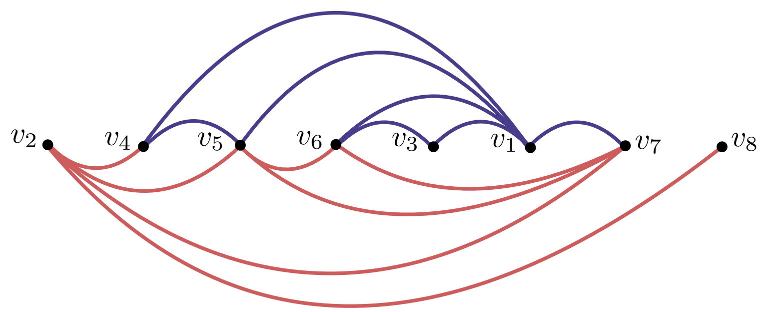
2.2 STACK AND QUEUE LAYOUTS

We will now introduce the formal definitions of stack layouts and queue layouts. A *stack layout* of a graph G is a total ordering σ of the vertices of G and a partitioning of the edges of G into sets called *stacks* such that there exists no crossings. Two edges uv and xy *cross* if both edges share a stack and $u \prec_\sigma x \prec_\sigma v \prec_\sigma y$. The *stack number* of G corresponds to the minimum number of stacks required to achieve a stack layout of G. We denote the stack number of G with $\text{sn}(G)$.

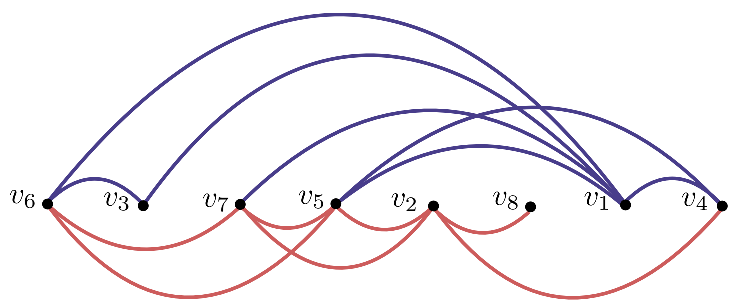
A *queue layout* of a graph G is a total ordering σ of the vertices of G and a partitioning of the edges of G into sets called *queues* such that there exists no nestings. Two edges uv and xy *nest* if both edges are on the same queue and $u \prec_\sigma x \prec_\sigma y \prec_\sigma v$. The *queue number* of G corresponds to the minimum number of queues required to achieve a queue layout of G. We use $\text{qn}(G)$ to denote the queue number of a graph G. Figure 2.5 illustrates an example of a graph, G_4, and a stack layout with $\text{sn}(G_4) = 2$. For this same graph, we were able to find a queue layout with $\text{qn}(G_4) = 2$ with a different ordering σ while retaining the same partition of the edges.



(a) Graph G_4



(b) Stack layout of G_4 with stack number 2



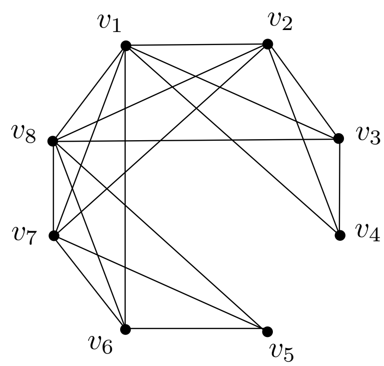
(c) Queue layout of G_4 with queue number 2

Figure 2.5: An example graph $G_4 = (V_4, E_4)$, a stack layout of G_4 with $\text{sn}(G_4) = 2$ and queue layout of G_4 with $\text{qn}(G_4) = 2$

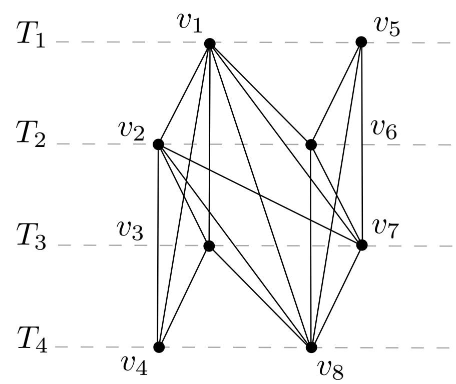
2.3 TRACK LAYOUTS

Finally, we will introduce the concept of t-track layouts. A *t-track layout* of a graph G is a partition of $V(G)$ into t ordered independent sets $T_1, ..., T_t$ (with a total order $\prec_i$ for each $T_i, i \in \{1, ..., t\}$) with no X-crossings. An *X-crossing* is a pair of edges vw and xy such that, for some $i, j \in \{1, ..., t\}$, $v, x \in T_i$ with $v \prec_i x$ and $w, y \in T_j$ with $y \prec_j w$. The minimum number of tracks in any t-track layout of G is called the track number of G and is denoted as $\text{tn}(G)$. A *t-track graph* is a graph that has a t-track layout.

Figure 2.6 demonstrates an example of a graph and a 4-track layout. We separate the graph's vertices into 4 tracks such that no pair of edges cross between any two tracks.



(a) Graph G_5



(b) t-track layout of G_5 with $\text{tn}(G_5) = 4$

Figure 2.6: Graph $G_5 = (V_5, E_5)$ and a 4-track layout of G_5

3 STACK NUMBER IS BOUNDED BY LAYERED PATHWIDTH

THEOREM 3.1. *Every graph with layered pathwidth k has stack number at most $4k$.*

Proof of Theorem 3.1. Let $B = B_1, \ldots, B_p$ be a path decomposition of G and $\ell : V(G) \to \mathbb{Z}$ be a layering such that B will have layered pathwidth $\text{lpw}(G) = k$ with respect to the layering ℓ.

The path decomposition B is *left-normal* if for every distinct pair $v, w \in V(G)$, $\min\{i \in \mathbb{Z} : v \in B_i\} \neq \min\{i \in \mathbb{Z} : w \in B_i\}$. In order to achieve this left-normal path decomposition, we can simply say that if a bag has two vertices $v, w \in V(G)$ such that $\min\{i \in \mathbb{Z} : v \in B_i\} = \min\{i \in \mathbb{Z} : w \in B_i\}$ we can prepend an identical bag to B_i and remove one of the vertices. It is straightforward to see that this transformation will not increase the layered pathwidth of B. We use the notation $v \prec_B w$ if $\min\{i \in \mathbb{Z} : v \in B_i\} < \min\{i \in \mathbb{Z} : w \in B_i\}$. When B is left-normal, $\prec_B$ is a total order.

For any edge vw with $v \prec_B w$ we call v the *left endpoint* of the edge and w the *right endpoint*.

First, we establish how to obtain the total vertex ordering σ that will be used for the stack layout. For every distinct pair $x, y \in V(G)$ where x and y are not necessarily joined by an edge and where $x \prec_B y$, we will assign their order in σ as follows.

1. If $\ell(x) = i$ and $\ell(y) = j$ with $i < j$ then $x \prec_\sigma y$.
2. If $\ell(x) = \ell(y) = i$ and i is even, then $x \prec_\sigma y$.
3. If $\ell(x) = \ell(y) = i$ and i is odd, then $y \prec_\sigma x$.

This completes the description of the total ordering σ on the vertices of G. With this ordering we can obtain a stack layout whose stack number is at most $4k$.

If an edge vw where v is the left endpoint and w is the right endpoint, has $\ell(w) = \ell(v) + 1$ then this edge has a *positive slope* and has a *nonpositive slope* otherwise.

Next, we define the colouring $\varphi : E(G) \to \{1, \ldots, 4k\}$ which determines the partition of $E(G)$ into stacks. We begin by establishing a greedy vertex colouring $\varphi(v) : V \to \{1, \ldots, k\}$ such that for every $i, j \in \mathbb{N}$, if two vertices are in $B_i \cap L_j$ then they will receive different colours. We can colour these vertices with k colours since the layered pathwidth is k and so no intersection of a bag and layer will have more than k vertices. We now colour the edge vw with the colour $\varphi(vw) = (a, b, c)$ where $a = \ell(v) \bmod 2$, b is a bit indicating whether vw has positive or non-positive slope, and c is the colour $\varphi(v)$ of the left endpoint v. This means that all edges can be coloured with $2 \times 2 \times k$ colours, which amounts to a stack number of at most $4k$. Figure Figure 3.1 shows a left-normal layered path decomposition and Figure 3.2 demonstrates the stack layout we would obtain by following the vertex ordering and edge partitioning we have described. In the example, since the original layered path decomposition had a width of 2, we can show that we are able to obtain a stack layout with fewer than 8 stacks. In this instance, we obtained a 5-stack layout. All that remains is to show that σ and the partition $\{\{vw \in E(G) : \varphi(vw) = (a, b, c)\} : (a, b, c) \in \{0, 1\} \times \{0, 1\} \times \{1, \ldots, k\}\}$ is indeed a stack layout.

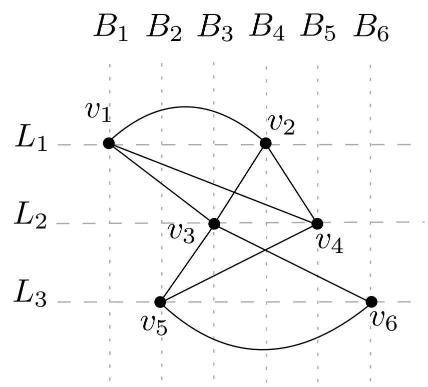
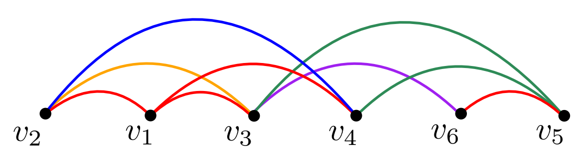


Figure 3.1: A layered path decomposition of width 2



$$\phi(v_1v_2) = (1, 1, 1) \quad \phi(v_3v_2) = (0, 0, 1)$$
$$\phi(v_1v_3) = (1, 1, 1) \quad \phi(v_3v_6) = (0, 1, 1)$$
$$\phi(v_1v_4) = (1, 1, 1) \quad \phi(v_5v_3) = (1, 0, 1)$$
$$\phi(v_5v_6) = (1, 1, 1) \quad \phi(v_5v_4) = (1, 0, 1)$$
$$\phi(v_2v_4) = (1, 1, 2)$$

Figure 3.2: The stack layout of the example graph in Figure 3.1

Consider any two distinct edges $vw, xy \in E(G)$ (whose left-endpoints are v and x, respectively). First observe that, if $\ell(v) \equiv \ell(x) \pmod 2$ then either $\ell(v) = \ell(x)$ or $|\ell(v) - \ell(x)| \geq 2$. In the latter case, the only way in which vw and xy can cross with respect to $\prec_\sigma$ is if $\ell(v) + b = \ell(y) = \ell(w) = \ell(x) - b$ for some $b \in \{-1, 1\}$. However, in this case, vw has positive slope and xy has non-positive slope, or vice-versa, so $\varphi(vw)$ and $\varphi(xy)$ differ in their second component.

Therefore, we only need to consider pairs of edges xy and vw where $\ell(v) = \ell(x) = i$. We assume, without loss of generality that i is even and that $v \prec_\sigma x$. With these assumptions, there are only three cases in which vw and xy can cross:

1. $v \prec_\sigma x \prec_\sigma w \prec_\sigma y$. Since $\ell(v) = \ell(x) = i$ is even and $v \prec_\sigma x$, we have $v \prec_B x$ and $\ell(w) \geq i$. If $\ell(w) = i$, then $v \prec_B x \prec_B w$, so v and x both appear in some bag B_j and $\varphi(v) \neq \varphi(x)$, so $\varphi(vw)$ and $\varphi(xy)$ differ in their third component. If $\ell(w) = i + 1$, then $w \prec_\sigma y$ implies that $\ell(y) \geq \ell(w)$, which implies $\ell(y) = \ell(w) = i + 1$, so $y \prec_B w$. We now have $v \prec_B x \prec_B y \prec_B w$ so v and x appear in a common bag B_j and $\varphi(vw)$ and $\varphi(xy)$ differ in their third component.

2. $v \prec_\sigma y \prec_\sigma w \prec_\sigma x$. Since $v \prec_\sigma y$, $\ell(y) \geq \ell(v) = i$. Similarly, since $y \prec_\sigma x$, $\ell(y) \leq \ell(x) = i$. Therefore, $\ell(y) = i$, so $y \prec_B x$. This is not possible since, by definition, x is the left endpoint of xy.

3. $y \prec_\sigma v \prec_\sigma x \prec_\sigma w$. Since $y \prec_\sigma x$, x is the left endpoint of xy, and $\ell(x) = i$ is even, we have $\ell(y) = \ell(x) - 1$, so xy has positive slope.

   Since $v \prec_\sigma w$ and $\ell(v) = i$ is even, we have $\ell(v) \leq \ell(w)$, so vw has non-positive slope. Therefore $\varphi(vw)$ and $\varphi(xy)$ differ in their second component.

Therefore, for any pair of edges $vw, xy \in E(G)$ that cross, $\varphi(vw) \neq \varphi(xy)$, so the partition P is a partition of $V(G)$ into $4k$ stacks with respect to $\prec_\sigma$, as required. □

4

3-TRACK LAYOUTS HAVE BOUNDED LAYERED PATHWIDTH

THEOREM 4.1. *Every graph G that has* $\text{tn}(G) \leq 3$, *has* $\text{lpw}(G) \leq 4$.

It will be easier to prove our result for a weaker notion of layering. An *s-weak layering* of G is a mapping $\ell_s : V(G) \to \mathbb{Z}$ with the property that, for every $vw \in E(G)$, $|\ell_s(v) - \ell_s(w)| \leq s$. The sets $L_{s,i} = \{v \in V(G) : \ell_s(v) = i\}$ are called s-weak layers. The terms *s-weak layered path decomposition* and *s-weak layered pathwidth* of G, denoted $\text{lpw}_s(G)$, are defined the same way as layered path decompositions and layered pathwidth, but with respect to s-weak layerings of G.

LEMMA 4.2. *For any* $s \in \mathbb{N}$, $\text{lpw}(G) \leq s \cdot \text{lpw}_s(G)$.

Proof. Given an s-weak layered decomposition of G with s-weak layering ℓ_s, we define $\ell(v) = \lfloor \ell_s(v)/s \rfloor$. Now ℓ is clearly a layering of G and, by definition, for any bag B_j and any s-weak layer $L_{s,i} = \{v \in V(G) : \ell_s(v) = i\}$, $|L_{s,i} \cap B_j| \leq \text{lpw}_s(G)$. Therefore, since any layer $L_i = \{v \in V(G) : \ell(v) = i\}$ is the union of at most s-weak layers, $|L_i \cap B_j| \leq s \cdot \text{lpw}_s(G)$ for all $i \in \mathbb{Z}$. □

Let G be an edge-maximal n-vertex graph with $\text{tn}(G) = 3$. Here edge-maximality implies that for any pair of vertices $x_i, y_j \in V(G)$, either $x_i y_j \in E(G)$; or there exists $x_{i'} y_{j'} \in E(G)$ for some $i' > i$ and $j' < j$; or there exists $x_{i'} y_{j'} \in E(G)$ for some $i' < i$ and $j' > j$. Similar comments hold for y_i, z_j and z_i, x_j. It is helpful to recall that G is a planar graph that has a straight-line crossing-free drawing with the vertices of T_1 placed on the positive x-axis, the vertices of T_2 placed on the positive y-axis and the vertices of T_3 placed on the ray $\{(a, a) : a < 0\}$. See Figure 4.1. The edges in Figure 4.1 are drawn as curves only to help readability. The drawing would be non-crossing even if all edges were drawn as straight line segments.

Let T_1, T_2, T_3 be a 3-track layout of G with $T_1 = \{x_1, \ldots, x_{n_1}\}$, $T_2 = \{y_1, \ldots, y_{n_2}\}$, and $T_3 = \{z_1, \ldots, z_{n_3}\}$ and the total orders $\prec_1, \prec_2, \prec_3$ are implicit so that, for example $z_i \prec_3 z_j$ if and only if $i < j$. In terms of Figure 4.1, this means that x_1, y_1, z_1 form the triangular face containing the origin and $x_{n_1}, y_{n_2}, z_{n_3}$ form the cycle on the boundary of the outer face. From this point onward, all track indices are implicitly taken "modulo 3" so that for any integer i, T_i refers to the track $T_{i'}$ with index $i' = ((i-1) \bmod 3) + 1$.

The following observation follows from the fact that G's 3-track layout is edge-maximal.

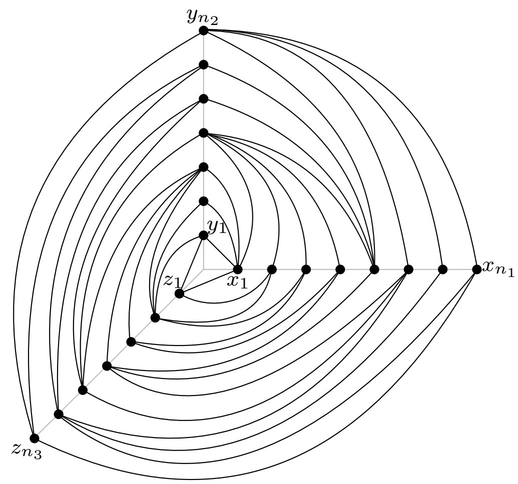


Figure 4.1: The standard planar embedding of a 3-track graph.

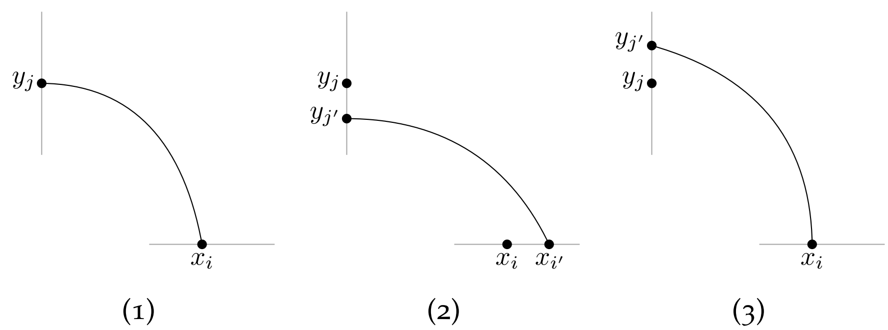


Figure 4.2: The three cases in Observation 4.3.

OBSERVATION 4.3. *For any two vertices of G on distinct tracks, say x_i and y_j, at least one of the following conditions is satisfied (see Figure 4.2):*

1. *$x_i y_j \in E(G)$; or*
2. *there exists $x_{i'} y_{j'} \in E(G)$ with $i' > i$ and $j' < j$; or*
3. *there exists $x_i y_{j'} \in E(G)$ with $j' > j$.*

Theorem 4.1 is a consequence of the following lemma.

LEMMA 4.4. *The edge-maximal n-vertex graph G with $\text{tn} = 3$ and $|V(G)| \geq 3$ has a 2-weak layered path decomposition, $B_1, \ldots, B_p$, with 2-weak layering ℓ_2 of width 2 in which*

1. *for each $i \in \{1,2,3\}$ and each $v \in T_i$, $\ell_2(v) \equiv i \pmod 3$;*
2. *$B_1 = \{x_1, y_1, z_1\}$;*
3. *$\ell_2(x_1) = 1$, $\ell_2(y_1) = 2$, and $\ell_2(z_1) = 3$;*
4. *$B_p = \{x_{n_1}, y_{n_2}, z_{n_3}\}$; and*

5. $x_{n_1}, y_{n_2}, z_{n_3}$ *appear in 3 distinct consecutive layers.*

Before proving Lemma 4.4, we show how it implies Theorem 4.1. Since layered pathwidth is monotone with respect to the addition of edges, it is safe to assume (as Lemma 4.4 does) that G's 3-track layout is edge-maximal. By Lemma 4.4, therefore G has $\mathrm{lpw}_2(G) \leq 2$ and therefore, by Lemma 4.2, $\mathrm{lpw}(G) \leq 4$.

Proof of Lemma 4.4. The proof is by induction on the number of vertices of G. If $|V(G)| \leq 4$, then the result is trivial. Next, suppose that G has a cut set $C = \{x_i, y_j, z_k\}$ having exactly one vertex in each track. Since G is an edge-maximal graph with $\mathrm{tn} = 3$, x_i, y_j, z_k form a cycle in G. Now, the subgraph G_1 of G induced by $\{x_1, \ldots, x_i, y_1, \ldots, y_j, z_1, \ldots, z_k\}$ is an edge-maximal graph with $\mathrm{tn}(G_1) = 3$ and we can inductively apply Lemma 4.4 to find a width-2 2-weak layered path decomposition of G_1 in which x_i, y_j, z_k are in the last bag and are assigned to three consecutive distinct layers $r+1$, $r+2$, and $r+3$. Note that there are three possible assignments of x_i, y_j, z_k to these three layers depending on the value of $r \bmod 3$. Suppose, without loss of generality, that $\ell_2(y_j) = r+1$ (so $\ell_2(z_k) = r+2$ and $\ell_2(x_i) = r+3$.)

Next, consider the graph G_2 induced by $\{x_i, \ldots, x_{n_1}, y_j, \ldots, y_{n_2}, z_k, \ldots, z_{n_3}\}$. We apply Lemma 4.4 inductively on G_2 relabelling tracks to ensure that in the resulting layered decomposition $\ell_2(y_j) = 1$, $\ell_2(z_k) = 2$ and $\ell_2(x_i) = 3$. We can now obtain a width-2 2-weak layered path decomposition of G by joining the two decompositions. In particular, concatenating the sequence of bags for G_1 with the sequence of bags for G_2 gives a path decomposition of G and adding r to the indices of all layers in the layering of G_2 gives a 2-weak layering of G.

Thus, all that remains is to study the case where G contains no cut set having exactly one vertex on each track. We claim that, in this case, G contains the edge x_1z_2 or it contains the edge z_1x_2. Since G is edge-maximal, this is obvious unless $n_1 = n_3 = 1$ so that neither z_2 nor x_2 exist. However, since $|V(G)| \geq 5$, this would imply that x_1, z_1, y_2 is a cut set with one vertex on each track, since its removal separates y_1 from $\{y_2, \ldots y_{n_2}\}$.

We will construct a path $P = v_1, \ldots, v_r$, an example of which is illustrated in Figure 4.3. The first vertex of P will be one of x_1, y_1, z_1 and the final three vertices are x_{n_1}, y_{n_2}, and z_{n_3}, though not necessarily in that order. The path P will *spiral* in the sense that $v_i \in T_i$ for all $i \in \{1, \ldots, r\}$. Observe that this spiralling implies that the subsequence of vertices of P on any track T_i is increasing (getting further from the origin).

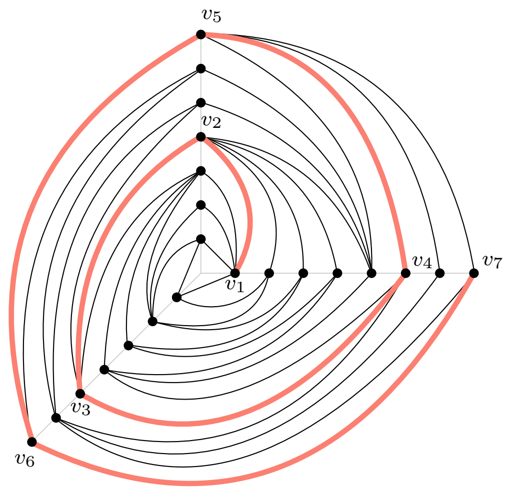


Figure 4.3: The path P used in the proof of Lemma 4.4.

P is constructed greedily: if P has reached vertex v_k, it continues to the neighbouring vertex v_{k+1} of v_k with the highest index on track T_{k+1} that is not yet in P. We will call this vertex v_{k+1} the *furthest neighbouring vertex* of v_k. To see why this is always possible, recall that P already contains an edge v_{k-3}, v_{k-2}. Now, without loss of generality we can apply Observation 4.3 with $x_i = v_k$ and $y_j = v_{k-2}$, so there are three cases (see Figure 4.4):

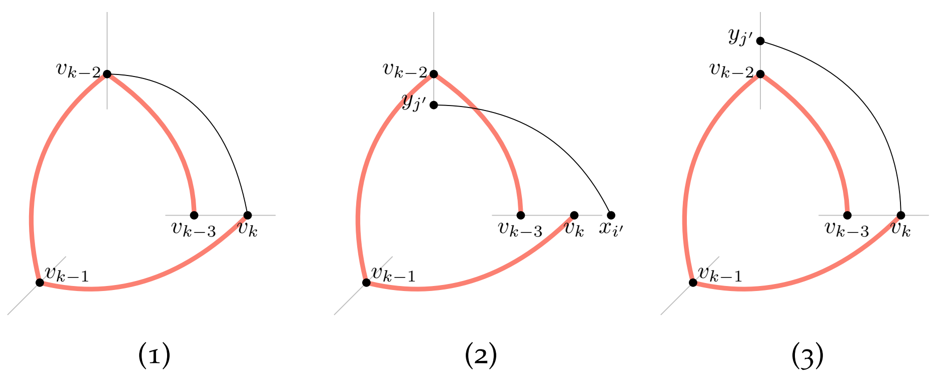


Figure 4.4: The path P can always be extended.

1. $v_k v_{k-2} \in E(G)$. In this case v_{k-2}, v_{k-1}, and v_k form a cycle in G. Then either $\{v_{k-2}, v_{k-1}, v_k\} = \{x_{n_1}, y_{n_2}, z_{n_3}\}$ or $\{v_{k-2}, v_{k-1}, v_k\}$ is a cut set with exactly one vertex in each track. In the former case, the path P is complete. The latter case is excluded by the assumption that G contains no such cut sets.

2. there exists an edge $x_{i'} y_{j'} \in E(G)$ with $i' > i$ (i.e. $i' > k$) and $j' < j$ (i.e. $j' < k-2$). This case is not possible, since this edge would cross $v_{k-3} v_{k-2}$.

3. there exists an edge $v_k y_{j'} \in E(G)$ with $j' > j$ (i.e. $j' > k-2$). In this case, P is extended by adding $v_{k+1} = y_{j'}$.

Thus we have constructed the furthest vertex path $P = v_1, \ldots, v_r$, whose first vertex is one of x_1, y_1, z_1 and whose last three vertices are x_{n_1}, y_{n_2} and z_{n_3} (not necessarily in order). We assign layers to the vertices of P as follows: For each vertex v_i on P, we set $\ell_2(v_i) = i$. Note that this satisfies Conditions 3 and 5 of the lemma and also satisfies Condition 1 for the vertices of P. For each $t \in \{1, 2, 3\}$, any vertex $v \in T_t$ that is not in P is assigned to the same layer as v's immediate successor in $P \cap T_t$. This assignment satisfies Condition 1 for vertices not in P. Finally, we will prove that this gives a 2-weak layering of G. In other words, in the worst case, a vertex v with $\ell_2(v) = i$ can only share an edge with vertex u where $i - 2 \leq \ell_2(u) \leq i + 2$.

Any edge between v and w where neither v nor w is in P will only span one layer. Any edge between any two vertices v_i and v_j where $v_i, v_j \in P$, will span only one layer if $j = i \pm 1$. This would mean that $v_i v_j$ is an edge in the graph G and that this edge was used to construct our furthest vertex path P. If $j \neq i \pm 1$, then there are two cases:

1. $j = i \pm 2$ This edge is possible and would create a cut set. This edges will only span 2 layers since $\ell_2(v_i) = i$ and $\ell_2(v_j) = i + 2$.

2. $j = i \pm 4$ This edge cannot exist since it would contradict our greedy path constructing algorithm. If the edge $v_i v_{i+4}$ (or the edge $v_{i-4} v_i$) existed then the edge $v_i v_{i+1}$ ($v_{i-4} v_{i-3}$) would not have been added to P.

Whenever we have an edge between a vertex on the furthest vertex path and a vertex not located on the vertex path then we have 8 cases (see Figure 4.5). Without loss of generality, assume the spiral is travelling from T_1 to T_2 to T_3. Let x_i be a vertex on the constructed path. First, we look at the possible cases for an edge between x_i with $\ell_2(x_i) = m$ and y_j where $y_j \notin P$.

1. $\ell_2(y_j) = m + 4 + 3n$ where $n \geq 1$. This edge cannot exist, since it would create a crossing.

2. $\ell_2(y_j) = m + 4$. This edge cannot exists since it would once more contradict our greedy path algorithm.

3. $\ell_2(y_j) = m + 1$. This edge will only span one layer.

4. $\ell_2(y_j) = m + 1 - 3n$ where $n \geq 1$. This edge cannot exist, since it would create a crossing.

Second, we look at the possible cases for an edge between x_i with $\ell_2(x_i) = m$ and z_k where $z_k \notin P$.

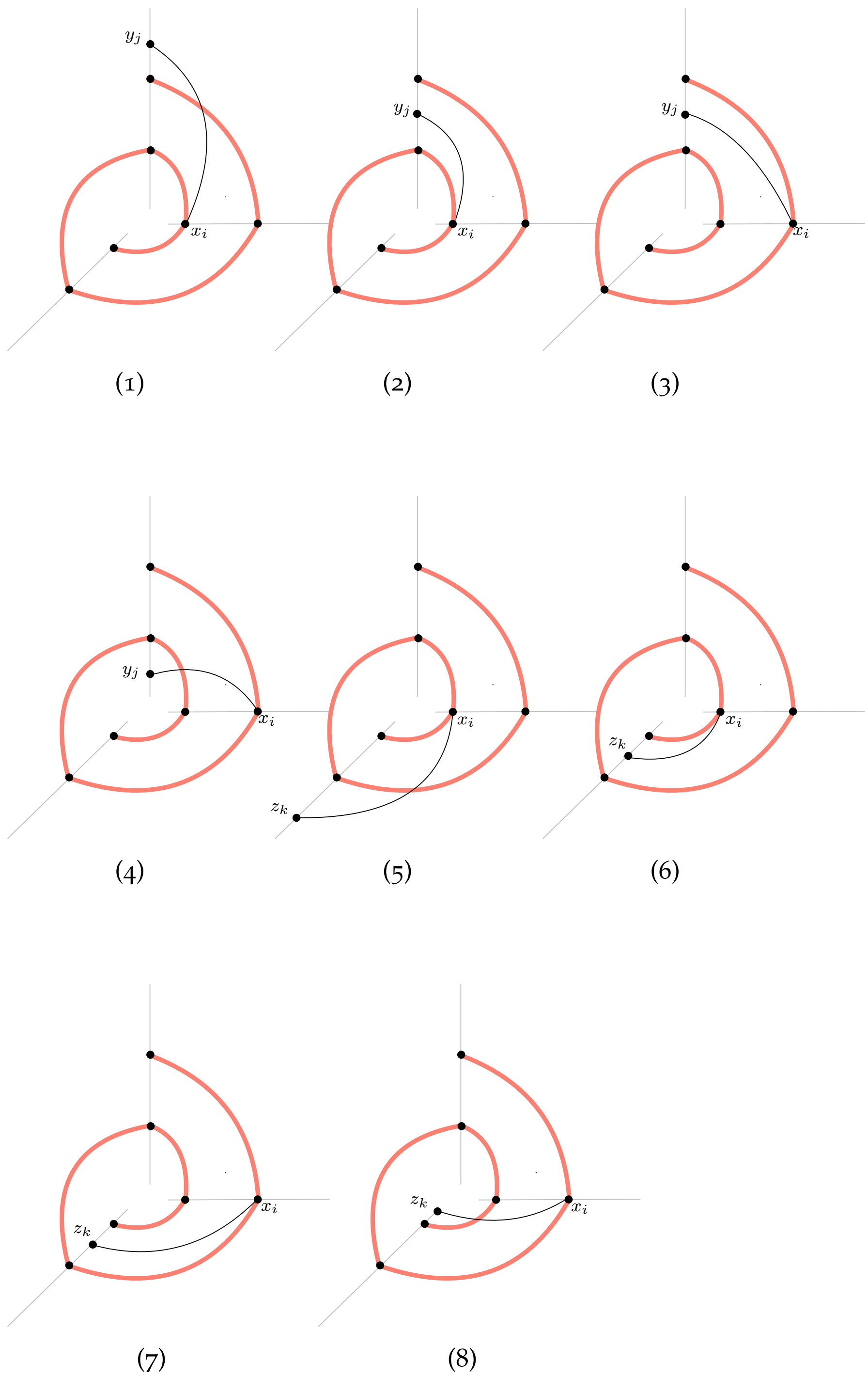


Figure 4.5: The edge between a vertex x_i and a vertex y_j or z_k cannot span more than 2 layers

5. $\ell_2(z_k) = m + 2 + 3n$ where $n \geq 1$. This edge cannot exist, since it would create a crossing.

6. $\ell_2(z_k) = m + 2$. This edge spans exactly two layers.

7. $\ell_2(z_k) = m - 1$. This edge will only span one layer.

8. $\ell_2(z_k) = m - 1 - 3n$ where $n \geq 1$. This edge cannot exist, since it would create a crossing.

Next, we will need the notion of levelled planar graphs. The class of levelled planar graphs was introduced in 1992 by Heath and Rosenberg [40] in their study of queue layouts of graphs. A levelled planar drawing of a graph is a straight-line crossing-free drawing in the plane, such that the vertices are placed on a sequence of parallel lines (called levels), where each edge joins vertices in two consecutive levels. A graph is levelled planar if it has a levelled planar drawing. (This is a well studied model for planar graph drawing, so called Sugiyama-style [55, 5, 37, 6].)

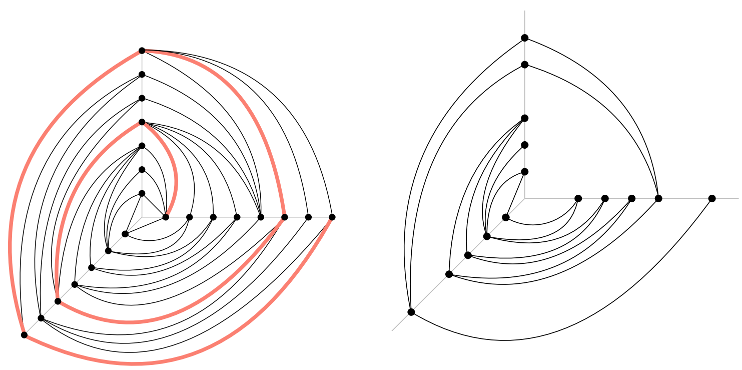


Figure 4.6: The graph $G - P$ is a levelled planar graph.

Now, consider the graph $G - P$ obtained by removing the vertices of P from G (see Figure 4.6). We claim that this graph is a levelled planar graph in which the levels of the vertices are given by the layering ℓ_2 defined above. Refer to Figure 4.7. One way to see this is to imagine G as being drawn with its vertices on the three vertical edges of a hollow triangular prism so that x_1, y_1, z_1 are the vertices of one triangular face and $x_{n_1}, y_{n_2}, z_{n_3}$ are the vertices of the other triangular face. Now, if we remove the triangular faces of this prism, cut it along the embedding of P, and unfold the resulting surface so that it lies in the plane, then we obtain a drawing of $G - P$ in which the vertices lie on a set of parallel lines and in which the edges only join vertices on two consecutive lines. This gives the desired levelled planar drawing of $G - P$.

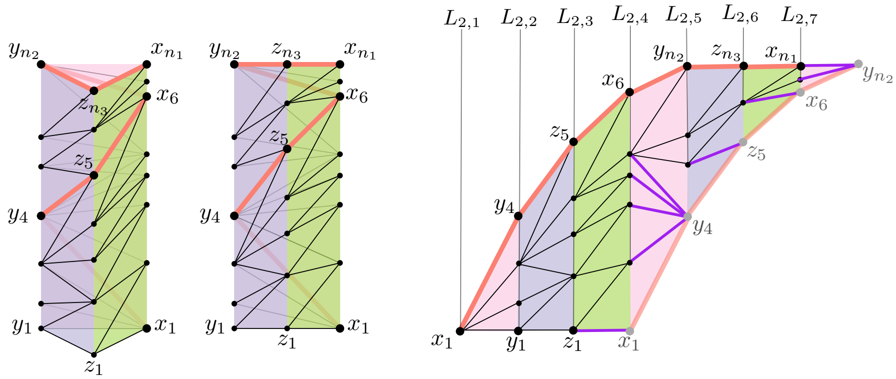


Figure 4.7: Cutting a prism along P to obtain a levelled planar drawing of $G - P$. Edges that span 2 layers are drawn in purple.

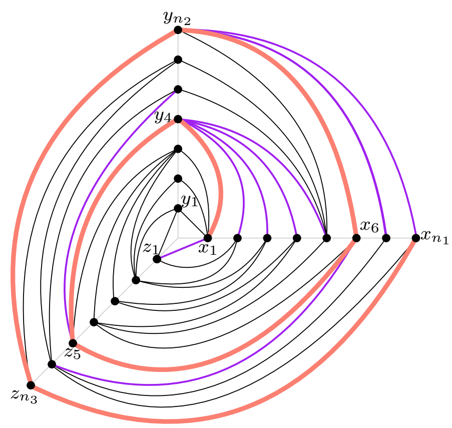


Figure 4.8: The graph G with the path P and edges that span 2 layers drawn in purple

By a result of Bannister et al. [4, Proof of Theorem 5], $G - P$ has a layered path decomposition $B_1, \ldots, B_p$ of width 1 using the layering ℓ_2 defined above. If we add the vertices of P to every bag of this decomposition we obtain a width-2 2-weak layered path decomposition of G. Finally, to satisfy Conditions 2 and 4 of the lemma, we prepend a bag $B_0 = \{x_1, y_1, z_1\}$ and append a bag $B_{p+1} = \{x_{n_1}, y_{n_2}, z_{n_3}\}$. □

We note that it is possible to unravel the induction in the proof of Lemma 4.4. In particular, even if G has cut sets with exactly one vertex on each track, we can still construct the path P starting from one of x_1, y_1, z_1 until it is not possible to extend P. This happens either because P includes $x_{n_1}, y_{n_2}, z_{n_3}$ or because P includes some cut set x_i, y_j, z_k. In the latter case, we continue extending P by spiralling in the other direction. Again, cutting the prism along P produces a layering of G that is compatible with a levelled planar drawing of $G - P$. The fact that this layering is also a 2-weak layering of G follows from the fact that P only changes directions after crossing a cut set.

5 CONCLUSION

The main contributions of this thesis are:

1. In Chapter 1, we provide a summary of bounds on pathwidth, treewidth, layered pathwidth, layered treewidth, stack number, queue number and track number for common graph classes.
2. Also in Chapter 1, we summarize the relationships that are known between these different graph layouts.
3. In Chapter 3, we provide a new bound on stack number for graphs of bounded layered pathwidth. We prove that graphs with layered pathwidth k have stack number at most $4k$.
4. In Chapter 4, we provide a new bound on layered pathwidth for 3-track graphs. This result closes an open conjecture of Bannister et al. [4]. To close this conjecture, we show that graphs with 3-track layouts have layered pathwidth at most 4.

Finally, we would like to discuss a few open problems related to this area of research. In Chapter 1 we presented a table with the known relationships between the graph layouts. We would now like to present the following open problems that discuss the relationships that are still unknown.

OPEN PROBLEM 1. *Is stack number bounded by layered treewidth?*

We know that $\operatorname{lpw}(G) \leq \operatorname{ltw}(G)(\log_3(2n+1))$ [4, Lemma 10] and in this thesis we've proved that $\operatorname{sn}(G) \leq 4\operatorname{lpw}(G)$. By transitive property, we can say that $\operatorname{sn}(G) \leq 4\operatorname{ltw}(G)(\log_3(2n+1))$. However, we pose the question, whether a function using only layered treewidth can express a bound on stack number.

OPEN PROBLEM 2. *Is stack number bounded by queue number?*

OPEN PROBLEM 3. *Is stack number bounded by track number?*

OPEN PROBLEM 4. *Is queue number bounded by layered treewidth?*

OPEN PROBLEM 5. *Is queue number bounded by stack number?*

OPEN PROBLEM 6. *Is track number bounded by stack number?*

We remark that the bounds we have presented in this thesis were not shown to be tight. In Chapter 3, we showed that any graph with layered pathwidth k, has stack number at most $4k$.

OPEN PROBLEM 7. *Can we reduce the bound on stack number for layered pathwidth k?*

In Chapter 4, we showed that any graph with a 3-track layout has layered pathwidth 4. The best lower bound on 3-track layouts is $\text{lpw}(G) \geq 2$ (a triangle will give this result).

OPEN PROBLEM 8. *Can we reduce the bound for layered pathwidth of graphs with* $\text{tn}(G) = 3$*?*

BIBLIOGRAPHY

[1] Jawaherul Alam, Michael Bekos, Martin Gronemann, and Sergey Pupyrev. Queue layouts of planar 3-trees. *Algorithmica*, 03 2020.

[2] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1):11 – 24, 1989.

[3] Michael J. Bannister, William E. Devanny, Vida Dujmović, David Eppstein, and David R. Wood. Track layout is hard. In *Graph Drawing and Network Visualization - 24th Int. Symposium, GD 2016*, volume 9801 of *LNCS*, pages 499–510, 2016.

[4] Michael J. Bannister, William E. Devanny, Vida Dujmović, David Eppstein, and David R. Wood. Track layouts, layered path decompositions, and leveled planarity. *Algorithmica*, pages 1–23, July 2018.

[5] Oliver Bastert and Christian Matuszewski. Layered drawings of digraphs. *Drawing Graphs, Methods and Models*, 2025:87–120, 2001.

[6] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Layered drawings of digraphs. *Graph Drawing: Algorithms for the Visualization of Graphs*, pages 265–302, 1999.

[7] Frank Bernhart and Paul C Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320 – 331, 1979.

[8] Umberto Bertelè and Francesco Brioschi. Nonserial dynamic programming. *SERBIULA (sistema Librum 2.0)*, 01 1974.

[9] Therese C. Biedl, Thomas C. Shermer, Sue Whitesides, and Stephen K. Wismath. Bounds for orthogonal 3-D graph drawing. *J. Graph Algorithms Appl.*, 3:63–79, 1999.

[10] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In Timo Lepistö and Arto Salomaa, editors, *Automata, Languages and Programming*, pages 105–118, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[11] Hans L. Bodlaender. *Planar graphs with bounded treewidth*. Technical Report RUU-CS-88-14. Department of Information and Computing Sciences, Utrecht University, 1988.

[12] Hans L. Bodlaender. Some classes of graphs with bounded treewidth. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 36, 01 1988.

[13] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1 – 45, 1998.

[14] F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg. Graph theory with applications to algorithms and computer science. chapter Embedding Graphs in Books: A Layout Problem with Applications to VLSI Design, pages 175–188. John Wiley & Sons, Inc., New York, NY, USA, 1985.

[15] H. de Fraysseix, P. O. de Mendez, and J. Pach. A left-first search algorithm for planar graphs. *Discrete & Computational Geometry*, 13(3):459–468, Jun 1995.

[16] Emilio Di Giacomo, Giuseppe Liotta, and Henk Meijer. *3D straight-line drawings of k-trees*. Technical Report TR-2003-473. Queen's University, School of Computing, 2003.

[17] Emilio Di Giacomo and Henk Meijer. Track drawings of graphs with constant queue number. In Giuseppe Liotta, editor, *Graph Drawing*, pages 214–225, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[18] Vida Dujmović. Graph layouts via layered separators. *J. Comb. Theory, Ser. B*, 110:79–89, 2015.

[19] Vida Dujmovic and Fabrizio Frati. Stack and queue layouts via layered separators. *J. Graph Algorithms Appl.*, 22(1):89–99, 2018.

[20] Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 862–875, 2019.

[21] Vida Dujmović, Gwenaël Joret, Pat Morin, Sergey Norin, and David R. Wood. Orthogonal tree decompositions of graphs. *SIAM J. Discrete Math.*, 32:839–863, 2017.

[22] Vida Dujmović, Pat Morin, and David R. Wood. Path-width and three-dimensional straight-line grid drawings of graphs. In Michael T. Goodrich and Stephen G. Kobourov, editors, *Graph*

Drawing, pages 42–53, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[23] Vida Dujmović, Pat Morin, and David R. Wood. Layout of graphs with bounded tree-width. *SIAM J. Comput.*, 34(3):553–579, 2005.

[24] Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-closed graph classes with applications. *J. Comb. Theory, Ser. B*, 127:111–147, 2017.

[25] Vida Dujmović, Attila Pór, and David R. Wood. Track layouts of graphs. *Discrete Mathematics & Theoretical Computer Science*, 6(2):497–522, 2004.

[26] Vida Dujmović, Anastasios Sidiropoulos, and David R. Wood. Layouts of expander graphs. *Chicago J. Theor. Comput. Sci.*, 2016, 2016.

[27] Vida Dujmović and David R. Wood. Tree-partitions of k-trees with applications in graph layout. In *WG*, 2003.

[28] S. Even and A. Itai. Queues, stacks and graphs. In Zvi Kohavi and Azaria Paz, editors, *Theory of Machines and Computations*, pages 71 – 86. Academic Press, 1971.

[29] Stefan Felsner, Giussepe Liotta, and Stephen K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In *Proc. 9th International Symp. on Graph Drawing (GD '01)*, volume 2265 of *LNCS*, pages 328–342. Springer, 2002.

[30] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, pages 261–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[31] Zvi Galil, Ravi Kannan, and Endre Szemerédi. On nontrivial separators for k-page graphs and simulations by nondeterministic one-tape turing machines. *Journal of Computer and System Sciences*, 38(1):134 – 149, 1989.

[32] Joseph L. Ganley. Stack and queue layouts of Halin graphs, 1995. (manuscript).

[33] Joseph L. Ganley and Lenwood S. Heath. The pagenumber of k-trees is $O(k)$. *Discrete Applied Mathematics*, 109:215–221, 2001.

[34] Rudolf Halin. S-functions for graphs. *Journal of Geometry*, 8(1):171–186, Mar 1976.

[35] Tero Harju and Lucian Ilie. *Forbidden Subsequences and Permutations Sortable on Two Parallel Stacks*, pages 267–275. Springer Netherlands, Dordrecht, 2001.

[36] Daniel J. Harvey and David R. Wood. Parameters tied to treewidth. *Journal of Graph Theory*, 84(4):364–385, 2016.

[37] Patrick Healy and Nikola S. Nikolov. Hierarchical drawing algorithms. *Handbook on Graph Drawing and Visualization*, pages 409–453, 2013.

[38] Lenwood S. Heath. Embedding planar graphs in seven pages. In *25th Annual Symposium on Foundations of Computer Science, 1984.*, pages 74–83, Oct 1984.

[39] Lenwood S. Heath, Frank Thomson Leighton, and Arnold L. Rosenberg. Comparing queues and stacks as mechanisms for laying out graphs. *SIAM J. Discrete Math.*, 5:398–412, 1992.

[40] Lenwood S. Heath and Arnold L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.*, 21(5):927–958, 1992.

[41] Nidhi Kapoor, Mark Russell, Ivan Stojmenović, and Albert Y. Zomaya. A genetic algorithm for finding the pagenumber of interconnection networks. *Journal of Parallel and Distributed Computing*, 62(2):267 – 283, 2002.

[42] András Kornai and Zsolt Tuza. Narrowness, pathwidth, and their application in natural language processing. *Discrete Appl. Math.*, 36(1):87–92, March 1992.

[43] L Taylor Ollmann. On the book thicknesses of various graphs. In *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume 8, page 459, 1973.

[44] Edward Ordman and William Schmitt. Permutations using stacks and queues. *Congressus Numerantium*, 01 1993.

[45] Shannon Overbay. *Generalized Book Embeddings*. PhD thesis, Fort Collins, CO, USA, 1998. UML Order No. GAX99-11135.

[46] Christos Papadimitriou and Mihalis Yannakakis. Towards an architecture-independent analysis of parallel algorithms. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 510–513, New York, NY, USA, 1988. ACM.

[47] Sriram Venkata Pemmaraju. *Exploring the Powers of Stacks and Queues via Graph Layouts*. PhD thesis, Blacksburg, VA, USA, 1992. UMI Order No. GAX93-10509.

[48] Sergey Pupyrev. Improved bounds for track numbers of planar graphs. arxiv:1910.14153, October 2019.

[49] B. A. Reed. *Tree Width and Tangles: A New Connectivity Measure and Some Applications*, pages 87–162. London Mathematical Society Lecture Note Series. Cambridge University Press, 1997.

[50] S. Rengarajan and C. E. Veni Madhavan. Stack and queue number of 2-trees. In Ding-Zhu Du and Ming Li, editors, *Computing and Combinatorics*, pages 203–212, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[51] Neil Robertson and P.D Seymour. Graph minors. III. planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49 – 64, 1984.

[52] P. Scheffler. *A Linear Algorithm for the Pathwidth of Trees*, pages 613–620. Physica-Verlag HD, Heidelberg, 1990.

[53] Farhad Shahrokhi. New representation results for planar graphs. arxiv:1502.06175, February 2015.

[54] Farhad Shahrokhi and Weiping Shi. On crossing sets, disjoint sets, and pagenumber. *Journal of Algorithms*, 34(1):40 – 53, 2000.

[55] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Systems Man Cybernet.*, 11(2):109–125, 1981.

[56] Jennifer Vandenbussche, Douglas West, and Gexin Yu. On the pagenumber of *k*-trees. *SIAM Journal on Discrete Mathematics*, 23:1455–1464, 07 2009.

[57] Joseph A. Wald and Charles J. Colbourn. Steiner trees, partial 2-trees, and minimum IFI networks. *Networks*, 13(2):159–167, 1983.

[58] Veit Wiechert. On the queue-number of graphs with bounded tree-width. *Electr. J. Comb.*, 24, 2016.

[59] David Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In *FST TCS 2002 Foundations of Software Technology and Theoretical Computer Science*, pages 348–359, 12 2002.

[60] David R. Wood. Bounded degree book embeddings and three-dimensional orthogonal graph drawing. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing*, pages 312–327, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[61] David R. Wood. Degree constrained book embeddings. *Journal of Algorithms*, 45(2):144 – 154, 2002.

[62] Mihalis Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36 – 67, 1989.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of April 8, 2020 (`classicthesis` version 1.0).