Visualizing Graphs: Optimization and Trade-offs

by

Debajyoti Mondal

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science
University of Manitoba
Winnipeg

Thesis advisor
Dr. Stephane Durocher

Author
Debajyoti Mondal

Abstract

Effective visualization of graphs is a powerful tool to help understand the relationships among the graph's underlying objects and to interact with them. Several styles for drawing graphs have emerged over the last three decades. Polyline drawing is a widely used style for drawing graphs, where each node is mapped to a distinct point in the plane and each edge is mapped to a polygonal chain between their corresponding nodes. Some common optimization criteria for such a drawing are defined in terms of area requirement, number of bends per edge, angular resolution, number of distinct line segments, edge crossings, and number of planar layers.

In this thesis we develop algorithms for drawing graphs that optimize different aesthetic qualities of the drawing. Our algorithms seek to simultaneously optimize multiple drawing aesthetics, reveal potential trade-offs among them, and improve many previous graph drawing algorithms.

We start by exploring probable trade-offs in the context of planar graphs. We prove that every n-vertex planar triangulation G with maximum degree Δ can be drawn with at most $2n + t - 3$ segments and $O(8^t \cdot \Delta^{2t})$ area, where t is the number of leaves in a Schnyder tree of G. We then show that one can improve the area by allowing the edges to have bends. Since compact drawings often suffer from bad angular resolution, we seek to compute polyline drawings with better angular resolution. We develop a polyline drawing algorithm that is simple and intuitive, yet implies significant improvement over known results.

At this point we move our attention to drawing nonplanar graphs. We prove that every thickness-t graph can be drawn on t planar layers with $\min\{O(2^{t/2} \cdot n^{1-1/\beta}), 2.25n + O(1)\}$ bends per edge, where $\beta = 2^{\lceil (t-2)/2 \rceil}$. Previously, the bend complexity, i.e., the number of bends per edge, was not known to be sublinear for $t > 2$. We then examine the case when the number of available layers is restricted. The layers may now contain edge crossings. We develop a technique to draw complete graphs on two layers, which improves previous upper bounds on the number of edge crossings in such drawings.

Contents

List of Figures

List of Tables

Acknowledgments

I am immensely blessed to have Dr. Stephane Durocher as my advisor. No words can express my heartfelt gratitude to Stephane for his continuous inspiration, persistent guidance, and endless support throughout my MSc and PhD career. He was deeply involved in every aspect of my academic life. We had countless research discussions in weekly meetings, lab seminars, and as of today, I see over a thousands of email threads with him.

Stephane is a man of ingenious wisdom and knowledge. His passion for theoretical computer science is truly impressive. But I think what influenced me the most are the moral values that I learned from him. He always encouraged me to choose my own research, get involved in graduate student committees, attend conferences and workshops. His kindness to my mistakes and endeavour to keep me free from any anxiety beyond research often took me by surprise. I will never forget Stephene's contribution to my academic, professional and personal development.

Besides the generous funding I received from Stephane, I must acknowledge the Guaranteed Funding Package (GFP) offered by the Department of Computer Science, the University of Manitoba Graduate Fellowship (UMGF), the Manitoba Graduate Scholarship (MGS), the Clarence Bogardus Sharpe Memorial Scholarship, and the Microsoft Intern Top Up Award, which have been highly beneficial for my research and collaboration.

I gratefully acknowledge the valuable feedback from the thesis reviewers Dr. Prosenjit Bose, Dr. Andriy Praymak, and Dr. David Wood. I thank all my collaborators, especially Dr. Therese Biedl, Dr. Stefan Felsner, Dr. Ellen Gethner, Dr. Stephen Kobourov, Dr. Anna Lubiw, Dr. Matthew Skala, and Dr. Sue Whitesides, for introducing me with many interesting research problems. I fondly remember the Summer of 2015, when I worked as a Microsoft Research Intern with Dr. Lev Nachmanson. He substantially motivated me to apply my theoretical knowledge to develop softwares that could potentially be used in practice. Lev's yoga lessons and hikes in the Pacific Northwest were some of the most exciting events in those internship days.

I am extremely thankful to my undergraduate thesis supervisor Dr. Md. Saidur Rahman, who first introduced me to graph drawing, and to all my friends, especially Md. Jawaherul Alam, Sudip Biswas, Khalad Hasan, Saeed Mehrabi, Rahnuma Nishat, Mrigank Rochan, Sharma Thankachan, and Md. Abdul Wahid, for cheering me up!

The last year of my PhD coincides with the first year of my marriage, and hence it was more colorful than ever before. I express my greatest appreciation and love to my wife, Purbasha Mistry, for her support, encouragement, and utmost patience with my thesis writing. Thanks Purbasha, for being so caring, for always being there for me, for reminding me that there is life outside my laptop, and for always surrounding me with your vibrant spirit!

Debajyoti Mondal

This thesis is dedicated to my beloved father, Dinesh Mondal; mother, Shova Mondal; and sister, Dipanwita Mondal, for their unconditional love and endless sacrifice throughout my life.

Chapter 1

Introduction

This chapter starts with an overview of the field of graph drawing, and some of the key previous results that motivate this thesis. The chapter then highlights the research questions and our main contributions. Finally, the chapter briefly describes the organization of the thesis.

1.1 Graphs and Their Drawings

A graph is a common tool to represent the interaction among different objects. The vertices of a graph model the objects (e.g., people, airports and webpages), while an edge between two nodes models the interaction (e.g., emails, routes, and hyperlinks) between the corresponding objects. Technically, a *graph* G is a tuple (V, E), where V is a set of $n = |V|$ vertices, and E is a set of pairs of vertices that belong to V. If E contains all possible pairs, then G is known as a *complete graph* K_n of n vertices. A *drawing* Γ of a graph G in the Euclidean plane $\mathbb{R}^2$ is a

mapping of the vertices to distinct points of $\mathbb{R}^2$, where each edge is drawn with a simple curve between its endpoints [113].

A rich body of graph drawing literature focuses on *planar graphs*, i.e., the graphs admitting drawings without any edge crossing in the Euclidean plane. Note that the graph of Figure 1.1(a) is planar since it has a planar drawing, as shown in Figure 1.1(b). A planar graph G is *maximal* or *triangulated* if the addition of any edge to G results in a nonplanar graph.

Theoretical interest and industrial needs have motivated the development of several styles of graph drawings [113], some of them can be categorized by the ways of drawing edges. The drawings with each edge drawn as a polygonal chain of at most t straight line segments is known as a *polyline drawing of bend complexity* $(t-1)$. Figure 1.1(a) illustrates a graph G, and Figure 1.1(b) is a polyline drawing of G. A drawing without any bend is called a *straight-line drawing*, as shown in Figures 1.1(c)–(d). If the edges are drawn with orthogonal polygonal chains, i.e., each segment is either horizontal or vertical, then the drawing is known as an *orthogonal drawing*. Figure 1.1(e) illustrates an orthogonal drawing. Some common optimization in such drawings are maximizing angular resolution, minimizing edge crossings, number of bends, area, segments and slopes, as defined below.

Let Γ be a drawing of some graph G. A point p on some edge $e = (a, b)$ in Γ is an *interior point* of e if $p \notin \{a, b\}$. A pair of edges creates an *edge crossing* in Γ if they intersect in Γ and the intersection point is an interior point for at least one of these edges. For example, the drawings Γ_2, Γ_3 and Γ_5 in Figure 1.1 are straight-line drawings of the same graph, but the number of edge crossings are

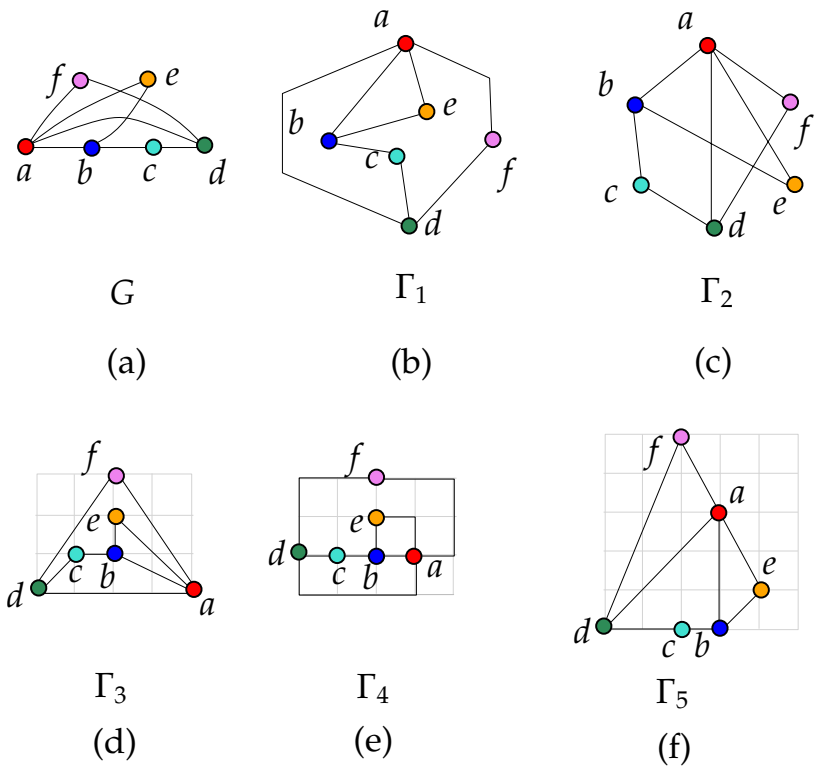


Figure 1.1: (a) A graph G. (b)–(f) Different drawings of G.

3, 0 and 0, respectively. A *segment* in a straight-line drawing is a maximal path P such that all the vertices on P are collinear in the drawing. The *number of segments* is the number of distinct straight line segments determined by the edges of the drawing. In Figure 1.1, the drawings Γ_3, Γ_4 and Γ_5 have 8, 8 and 6 segments, respectively. The number of *slopes* in a drawing is the number of distinct slopes for the segments in the drawing. For example, in Figure 1.1, the slope number of Γ_4 is two. The *angular resolution* of a drawing is the minimum over all the angles formed at the vertices in the drawing. In Figure 1.1, the angular resolution of the drawing Γ_4 is 90°, which is larger than any other drawing of Figure 1.1. The *area* of a drawing is the number of grid points in the smallest axis-aligned rectangle

that encloses the drawing. While measuring the area, we assume that the bends and vertices lie on integer grid points. In Figure 1.1, the area of both Γ_3 and Γ_4 is 20, which is smaller than the area of Γ_5.

1.2 Related Work

In this thesis we develop graph drawing algorithms, suggesting several potential trade-offs among different drawing aesthetics. Here we review some significant results that are related to our work.

1.2.1 Angular Resolution, Area and Bends

Every maximal planar graph (hence any planar graph) with n vertices admits a straight-line drawing in $O(n^2)$ area [32], where vertices are drawn at integer grid coordinates. The constant hidden in $O(.)$ notation has been improved several times [15, 17, 32, 108], and the best known bound is $8n^2/9$ [17]. On the other hand, the best known lower bound is $4n^2/9$ [39, 61]. Better area upper bounds have been achieved in polyline drawings. For example, every n-vertex planar graph admits a polyline drawing in $4n^2/9$ area with bend complexity one and at most $n-2$ bends in total [14]. Recently, Zhang [120] improved the bounds on the number of bends to $2n/3$. None of these results made any conscious effort to maximize the angular resolution. Therefore, although these drawings require small area, the angular resolution can be as small as $1/n^2$ in the worst case. The angular resolution of an n^2-area drawing is $\Omega(1/n^2)$. Kurowski [86] improved the bound on angular resolution by proving that every planar graph admits a drawing with

angular resolution $\Omega(1/n)$, and at most $9n^2/2$ area. Allowing bends helps to reduce area and to improve the angular resolution to $\Omega(1/\Delta)$, where Δ is the maximum degree of the graph. For example, every planar triangulation admits a drawing with $2/\Delta$ radians of resolution, 3 bend complexity and $3n^2$ area [71]. The angular resolution can be improved further to $1/d(v)$ radians of resolution, for each $v \in V$, with 2 bend complexity, but the area becomes $200n^2$ [69]. Duncan and Kobourov [43] proved that with $0.5/d(v)$ radians of resolution, for each $v \in V$, and 1 bend complexity, one can reduce the upper bound on area to $12.5n^2$.

Table 1.1 presents some evidence that when the bend complexity is fixed, the area and angular resolution are conflicting optimization criteria.

Graph Class	Area	Resolution (rad.)	Bend Complexity	Total Bends	Ref.
Planar graphs	$8n^2/9$	$\Omega(1/n^2)$	0	0	[17]
Planar graphs	$9n^2/2$	$\Omega(1/n)$	0	0	[86]
Planar graphs	$4n^2/9$	$\Omega(1/n^2)$	1	$2n/3$	[120]
Planar graphs	$12.5n^2$	$0.5/d(v)$	1	$3n$	[43]
Planar graphs	$200n^2$	$1/d(v)$	2	$6n$	[69]
Planar graphs	$3n^2$	$2/\Delta$	3	$5n-15$	[71]

Table 1.1: Trade-off among angular resolution, area and bends.

1.2.2 Segments, Area, and Bends

Straight-line drawings are preferable since the use of bends makes it difficult to follow the edges in the drawing. The notion of 'straightness' has been taken to its extreme by introducing the segment and slope number of a drawing. Several recent studies have examined drawings with few slopes and segments that use real coordinates for positioning vertices and bends. These drawings seem to need

exponential area, but lower bounds on their area requirements have not been formally explored. The problem of minimizing the number of segments is NP-hard [53]. However, polynomial-time algorithms to compute straight-line drawings with minimum number of segments have been achieved for trees [41], 2-trees with maximum degree three, and 3-connected cubic planar graphs [94]. Dujmović et al. [41] showed that every 3-connected planar graph admits a straight-line drawing with at most $5n/2$ segments. For planar 2 and 3-trees they could improve the bound to $2n$.

Table 1.2 summarizes the main results. Note that the '$\times$' marked cells denote that no non-trivial area upper bound is known for the corresponding cases.

Graph Class	Number of Segments	Area	Bend Complexity	Ref.
Trees	minimum	n^8	0	[46, 53]
3-Connected cubic planar	minimum	$\times$	0	[94]
2-Trees with $\Delta \leq 3$	minimum	$\times$	0	[106]
Planar 2- and 3-trees	$2n$	$\times$	0	[53]
3-Connected planar	$5n/2$	$\times$	0	[53]

Table 1.2: Results on optimizing segments.

1.2.3 Slopes, Area, and Bends

An upper bound on the number of segments is also an upper bound on the number of slopes in the drawing. Dujmović et al. [53] proved that every planar graph admits a drawing with at most $2n$ slopes. This bound has later been improved further in terms of the maximum degree Δ of the graphs. Using the result that every planar graph G admits a representation with vertices as non-intersecting circles [84], where two circles touch if and only if their corresponding vertices are

adjacent in G, Keszegh et al. [80] proved that every planar graph admits a planar straight-line drawing with at most c^{Δ} slopes, where c is a positive constant. They also showed that allowing one (respectively, two) bend complexity improves the upper bound to 2Δ (respectively, $\Delta/2$).

Further progress has been made for subclasses of planar graphs. For example, Jelínek et al. [77] showed that Δ^5 slopes suffice for planar 3-trees, and Knauer et al. [83] proved that $\Delta - 1$ slopes are sufficient for every outerplanar graph with $\Delta \geq 4$. Recently, Lenhart et al. [88] gave an algorithm to draw 2-trees such that slopes of the edges around each vertex is a subset of the set $S(\Delta)$=$\{i\pi/(2\Delta)$ radians, where $0 \leq i \leq 2\Delta - 1\}$. In other words, $S(\Delta)$ is a *universal slope set* for all planar 2-trees with maximum degree Δ.

Table 1.3 compiles the best known upper bounds on the number of slopes for different classes of graphs. Note that the '$\times$' marked cells denote that no non-trivial area upper bound is known for the corresponding cases.

Graph Class	Number of Slopes	Area	Bend Complexity	Ref.
Planar graphs	c^{Δ}	$\times$	0	[80]
Planar graphs	2Δ	$\times$	1	[77]
Planar graphs	$\Delta/2$	$\times$	2	[77]
Outerplanar graphs	$\Delta - 1$, where $\Delta \geq 4$	$\times$	0	[83]
2-Trees	2Δ	$\times$	0	[88]
Planar 3-trees	Δ^5	$\times$	0	[77]

Table 1.3: Results on optimizing slopes.

1.2.4 Planar Layers and Bends

While drawing nonplanar graphs, the idea of planarity has been extended through the concept of planar layers. The *thickness* of a graph G is the minimum number of planar subgraphs whose union is G. Let Γ be a polyline drawing of some graph G. The *layer complexity* of Γ is the minimum number of planar subgraphs of G such that each of these subgraphs induces a planar drawing in Γ. Every thickness-t graph G admits a polyline drawing Γ, where each of the t planar subgraphs of G corresponds to a planar drawing in Γ [102]. Therefore, we can define the *thickness* of a graph G to be the minimum integer $\theta(G)$ such that G admits a polyline drawing with layer complexity $\theta(G)$. The *geometric thickness* of a graph G is the minimum integer $\bar{\theta}(G)$ such that G admits a straight-line drawing with layer complexity $\bar{\theta}(G)$.

Duncan et al. [44] gave an algorithm for finding straight-line drawings of maximum-degree-4 graphs with layer complexity two. In the proof they showed that every maximum-degree-3 graph can be decomposed into two forests, where each connected component is a path. Thus each forest can be augmented to a path and then these two paths can be drawn simultaneously on an integer grid of n^2 area. Specifically, let $P_v = (v_1, v_2, \ldots, v_n)$ and $P_u = (u_1, u_2, \ldots, u_n)$ be the paths, where there is a bijection between the vertices of P_v and the vertices of P_u. Then the coordinate of each vertex v_i in the drawing would be (i, j), where u_j is mapped to v_i. Figure 1.2(a) illustrates a pair of paths P_v, P_u, and Figure 1.2(b) depicts such a drawing. For maximum-degree-4 graphs, Duncan et al. [44] decomposed the input graph into two subgraphs, each consists of some disjoint cycles and paths.

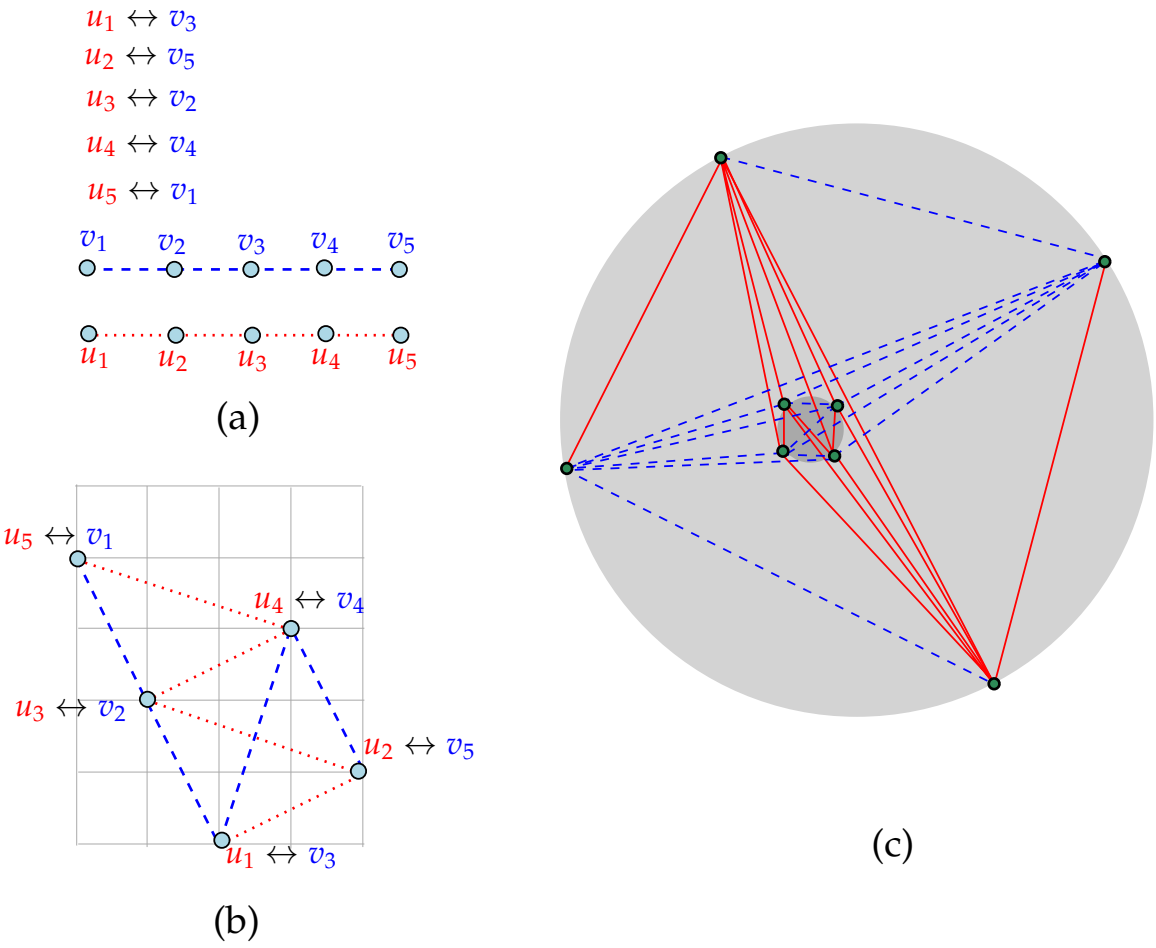


Figure 1.2: (a) A pair of paths P_u, P_v. (b) Simultaneous embedding of P_u, P_v. (c) A straight-line drawing of K_8 with layer complexity two.

Next they extended the idea of embedding pairs of paths to embed pairs of cycles.

Dillencourt et al. [37] proved that the geometric thickness of complete graphs is at least $\lceil n + 1/5.646 \rceil$ and at most $\lceil n/4 \rceil$. The proof for the upper bound is constructive, which arranges the vertices on a pair of circles in a regular interval, one nesting the other. Each circle contains roughly half of the vertices, and the inner circle is small enough to ensure necessary visibilities. An example is illustrated in Figure 1.2(c). Their construction takes $O(n^6)$ area. Wood [118] proved that every

n-vertex graph with m edges admits a polyline drawing with bend complexity one and layer complexity $O(\sqrt{m})$. He first proved the existence of such a drawing with probabilistic argument. Later he proposed an $O(m \log^3 n \log\log n)$-time algorithm for constructing such a drawing with high probability.

A graph G is called a *k-tree* if and only if it can be constructed by starting with a complete graph K_k, and then incrementally adding the other vertices such that the neighbors of each added vertex form a complete graph of k vertices. Besides bounded degree graphs, geometric thickness has been studied for *graphs of treewidth k*, i.e., the subgraphs of k-trees. Dujmović and Wood [40] proved that for graphs of treewidth k, the maximum thickness and maximum geometric thickness are both equal to $\lceil k/2 \rceil$. Duncan [42] proved that all graphs with 'arboricity' two or 'outerthickness' two have geometric thickness $O(\log n)$, and observed that the geometric thickness of every thickness-two graph is bounded by $O(\sqrt{n})$.

Graph Class	Layer Complexity	Area	Bend Complexity	Ref.
Maximum degree 4 graphs	2	$\times$	0	[44]
Graphs of treewidth k	k	$\times$	0	[40]
Complete graphs	$\lceil n/4 \rceil$	n^6	0	[37]
Graphs of thickness 2	$O(\sqrt{n})$	$\times$	0	[42]
Graphs of thickness 2	2	n^4	2	[66, 58]
General graphs	$O(\sqrt{m})$	$\times$	1	[118]

Table 1.4: Optimizing layer and bend complexities.

1.2.5 Layers and Crossing Number

Crossing number is an important parameter in nonplanar graph drawing. The *crossing number* $cr(G)$ of a graph G is the minimum number of edge crossings over

all drawings of G in the Euclidean plane. Determining the crossing number of a graph is a classic problem in graph theory [107]. In addition to theoretical interest, drawings with few edge crossings are important in many practical applications such as network visualization [113] and multilayer VLSI layout [87]. Determining the crossing numbers of complete and complete bipartite graphs is one of the most studied problems in this line of research (e.g., [2, 18, 33, 97, 105, 107]). The problem of determining $cr(K_n)$, i.e., the crossing number of a complete graph with n vertices, has been studied since the early 1960s [72, 73, 119]. From that time it was known [72] that $cr(K_n)$ is at most Z_n, where $Z_n = \frac{1}{4}\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$, which is still the best known upper bound on $cr(K_n)$.

The definition of crossing number naturally extends to an arbitrary number of planes. The *k-planar crossing number* $cr_k(G)$ of G is $\min\{cr(G_1) + cr(G_2) + \ldots + cr(G_k)\}$, where the minimum is taken over all possible decompositions of G into k subgraphs $G_1, G_2, \ldots, G_k$. In 1971, Owens [99] showed that $cr_2(K_n)$ is at most W_n, where

$$W_n = \begin{cases} Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{n^2(n-4)(n-8)}{384}, & \text{if } n = 4m. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{(n-1)(n-3)^2(n-5)}{384}, & \text{if } n = 4m+1. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{n(n-2)(n-4)(n-6)}{384}, & \text{if } n = 4m+2. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{(n+1)(n-3)^2(n-7)}{384}, & \text{if } n = 4m+3. \end{cases} \tag{1.1}$$

Several works on k-planar crossing number examine the asymptotic behaviour of $cr_k(G)$, where G is a complete or complete bipartite graph [9, 110]. There have also been significant efforts to determine tight bounds on *biplanar*, i.e., 2-planar, crossing numbers for these classes of graphs [29, 30, 65].

1.3 Motivation and Research Questions

In the previous section we saw that a graph drawing algorithm usually focuses on optimizing a particular aesthetic of the drawing while satisfying some given drawing constraints, e.g., minimizing area under some fixed number of bends per edge restriction, maximizing angular resolution satisfying the straight-line edge restriction, etc. Although these results suggest possible trade-offs among different drawing aesthetics, for several combinations of aesthetics the nature of interaction has not yet been examined. This gap in the previous research motivates us to examine the possible trade-offs while optimizing different drawing aesthetics.

We started by examining the number of segments in a drawing. Dujmović et al. [41] showed that every 3-connected planar graph admits a straight-line drawing with at most $5n/2$ segments, but the drawing uses real coordinates to place the vertices. In fact, all the known algorithms that optimize the number of segments, except the algorithms for trees [46, 53] and cubic graphs [94], use real coordinates. Therefore, we examined the following question.

> **Problem 1:** What is the trade-off between the segments and area in planar straight-line drawings? Can every planar graph be drawn on small area with few segments?

The next question that naturally arises is what happens when we allow edges to have bends? We have already seen that a rich body of literature deals with this question. Specifically, every n-vertex planar graph admits a straight-line drawing

with $8n^2/9$ area [17], and with one bend complexity this area upper bound can be reduced to $(4/9)n^2$ [14], which is the best possible even for higher bend complexities. All these algorithms are for *fixed embedding setting*, where the input is a combinatorial embedding of a planar graph, and the output drawing respects the input embedding. Frati and Patrignani [61] showed that the $(4/9)n^2$ lower bound does not hold in the *variable embedding setting*, i.e., when the output embedding may differ from the input embedding. The best known lower bound in the variable embedding setting is $(1/9)n^2$. Therefore, we asked can we achieve a better area upper bound by choosing a suitable embedding, possibly allowing a higher bend complexity?

> **Problem 2:** How much can we improve the area of a planar drawing if we allow the edges to have bends?

A major weakness of a compact drawing is its low angular resolution, which makes the drawing difficult to read. Every planar graph admits a straight-line drawing with angular resolution $\Omega(1/n)$ and at most $9n^2/2$ area [86]. Known drawing algorithms that achieve an angular resolution of $\Omega(1/d(v))$, require larger area and allow higher bend complexity [43]. To better understand the trade-off between angular resolution and area, we examined polyline drawings with a fixed bend complexity.

> **Problem 3:** Characterize the interaction between the angular resolution and area requirement in planar polyline drawings.

At this point we moved our focus on drawing non-planar graphs. From the known results [60, 108], we have observed that every planar graph (thickness one)

can be drawn on a single planar layer with 0 bend complexity. On the other hand, any graph with thickness t admits a polyline drawing on t planar layers with $O(n)$ bends per edge [102]. Therefore, we were motivated to investigate possible trade-offs between layer and bend complexities in polyline drawings.

> **Problem 4:** Characterize the trade-off between the layer and bend complexities while drawing non-planar graphs.

An interesting question here is what happens if we restrict the drawing to have only a few layers? If the input graph G is very dense, then the layers will no longer be planar, and a natural optimization here would be to minimize the number of crossings. For example, if we restrict the drawing to have one layer, then the optimization is equivalent to determining the crossing number $cr(G)$. Similarly, for k layers, the optimization would ask for k-planar crossing number $cr_k(G)$. Crossing number problem has long been examined for complete graphs, and the conjectured upper bound Z_n for arbitrary n matches the lower bound for all the complete graphs for which the crossing numbers are known [72, 104], i.e., when $n \leq 12$. On the other hand, the best known general upper bound W_n on biplanar crossing [99] is not known to be tight, e.g., $W_9 = 4$ and $cr_2(K_9) = 1$. Besides, very little is known about the k-planar crossing number, where $k \geq 2$, even when the number of vertices is small. While tight bounds for $cr(K_n)$ are known for $n \leq 12$ [104], the value of $cr_2(K_n)$ is known only when $n \leq 9$ [96]. These observations motivated us to consider the following questions.

> **Problem 5:** What are the exact values of $cr_2(K_n)$, where $n \leq 12$? Can we further improve the upper bound W_n on biplanar crossing number?

1.4 Contributions and Organization of Thesis

This section describes the organization of the thesis chapters, and highlights the major contributions.

1.4.1 Basics of Graph Drawing (Ch. 2)

In this chapter we introduce some preliminary definitions, and describe some known algorithmic tools and techniques that are commonly used in graph drawing.

1.4.2 Minimizing Segments and Area (Ch. 3)

Dujmović et al. [41] showed that every 3-connected planar graph G with n vertices admits a straight-line drawing with at most $2.5n - 3$ segments. In this chapter we prove that every planar triangulation admits a straight-line drawing with at most $7n/3$ segments. If the input triangulation is 4-connected, then our algorithm computes a drawing with at most $2.25n$ segments. For general planar graphs with n vertices and m edges, our algorithm requires at most $5.34n - m$ segments, which is less than $2.5n - 3$ for all $m \geq 2.84n$. If the vertices are restricted to have integer coordinates, then we show that every triangulation with maximum degree Δ can be drawn with at most $2n + t - 3$ segments and $O(8^t \cdot \Delta^{2t})$ area, where t is the minimum number of leaves over all the trees of the minimum realizer [121]. This is the first non-trivial attempt to simultaneously optimize the area and the number of segments while drawing triangulations. This partially

answers Question 1. Some of the results of this chapter appeared in preliminary form at the 26th Canadian Conference on Computational Geometry [48].

1.4.3 Compact Polyline Drawings of Planar Graphs (Ch. 4)

In this chapter we compute compact polyline drawings of planar graphs. Specifically, we prove that every planar graph with maximum degree $o(n)$ admits a polyline drawing with bend complexity two and at most $(3/8)n^2 + o(n^2)$ area, where the output combinatorial embedding can be freely chosen. The previously best known bound for this class of graphs was $(4/9)n^2$ area with bend complexity one. This partially answers Question 2. Some of the results of this chapter appeared in preliminary form at the 22nd International Symposium on Graph Drawing [51].

1.4.4 Polyline Drawings with Good Angular Resolution (Ch. 5)

In this chapter we develop a new technique to compute polyline drawings for planar triangulations. Our algorithm is simple and intuitive, yet implies significant improvement over the known results. For any given n-vertex triangulation, our algorithm computes a drawing with angular resolution $r/d(v)$ at each vertex v, and area $f(n,r)$, for any $r \in (0,1]$, where $d(v)$ denotes the degree at v. For $r < 0.389$ or $r > 0.5$, $f(n,r)$ is less than the drawing area required by previous algorithms; $f(n,r)$ ranges from $7.12n^2$ when $r \leq 0.3$ to $32.12n^2$ when $r = 1$. This partially answers Question 3. Some of the results of this chapter appeared in preliminary form at the 22nd International Symposium on Graph Drawing [49].

1.4.5 Drawing Graphs on Multiple Layers (Ch. 6)

By Fáry's theorem [60], planar graphs can be drawn on a single layer with bend complexity 0. In this chapter we present the first non-trivial extension of Fáry's theorem to draw graphs of thickness more than two. We prove that every thickness-t graph can be drawn on t layers with bend complexity $\min\{O(2^{t/2} \cdot n^{1-1/\beta}), 2.25n + O(1)\}$, where $\beta = 2^{\lceil (t-2)/2 \rceil}$. Previously, the bend complexity was not known to be sublinear for $t > 2$. We then show that every graph with linear arboricity k can be drawn on k layers with bend complexity $\frac{3(k-1)n}{(4k-2)}$. This partially answers Question 4. Some of the results of this chapter appeared in preliminary form at the 43rd International Colloquium on Automata, Languages and Programming [50].

1.4.6 Drawing Complete Graphs on Few Planes (Ch. 7)

In this chapter we develop an improved technique to construct biplanar drawings of K_n, which reduces Owens' upper bound on $cr_2(K_n)$. For small fixed n, it was known previously that $cr_2(K_n) = 0$ for $n \leq 8$ and $cr_2(K_9) = 1$; we show that $cr_2(K_{10}) = 2$, $cr_2(K_{11}) \in \{4, 5, 6\}$, and for $n \geq 12$, we improve previous upper and lower bounds on $cr_2(K_n)$. This partially answers Question 5. Some of the results of this chapter appeared in preliminary form at the 28th Canadian Conference on Computational Geometry [54].

1.4.7 Conclusion (Ch. 8)

This chapter summarizes the results of the thesis, poses new open problems, and suggests directions for future research.

Chapter 2

Basics of Graph Drawing

In this chapter we introduce some preliminary definitions and notation related to this thesis. We first describe different types of graph classes that have been used in this thesis. We then review different graph drawing styles and present some widely used combinatorial tools and techniques for graph drawing that are within the scope of this thesis. Finally, we discuss a few definitions from complexity theory. The readers familiar with the field of graph drawing may skip this chapter.

2.1 Graphs Classes

In this section we describe the graph classes that have appeared in the subsequent chapters of this thesis. For a detailed discussion on different types of graphs and their properties we refer the reader to [113, 117].

Let $G = (V, E)$ be a graph with vertex set V and edge set E, where $n = |V|$

and $m = |E|$ are the number of vertices and edges of G. Throughout the thesis we assume that G is *simple*, i.e., G does not contain any loop or multi-edge. G is called a *complete graph* if every vertex in V is incident to all other vertices in V, i.e., $m = \binom{n}{2}$, as shown in Figure 2.1(a). A complete graph with k vertices is also known as a *k-clique*. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. G' is called an *induced subgraph* if for every pair of vertices $\{a, b\} \subseteq V'$, $(a, b) \in E'$ if and only if $(a, b) \in E$.

c-Partite Graphs: A *c-coloring* of the vertices of G assigns every vertex of G a color from a set of c colors such that no two adjacent vertices receive the same color, e.g., see Figure 2.1(b). Two colorable graphs are known as *bipartite graphs*. Similarly, c-colorable graphs are referred to as c-partite graphs. G is called a *complete c-partite* graph if addition of any more edges to G makes it impossible to color G with c colors.

k-Trees: A *k-tree* is a graph G with $n \geq k$ vertices, which is recursively defined as follows.

- If $n = k$, then G is a complete graph of k vertices.
- If $n > k$, then G contains a vertex v such that the neighbors of v induces a k-clique in G and deletion of v from G gives another k-tree with $n - 1$ vertices.

Figure 2.1(c) illustrates a k-tree, where $k = 4$. Any subgraph of a k-tree is called a *partial k-tree*.

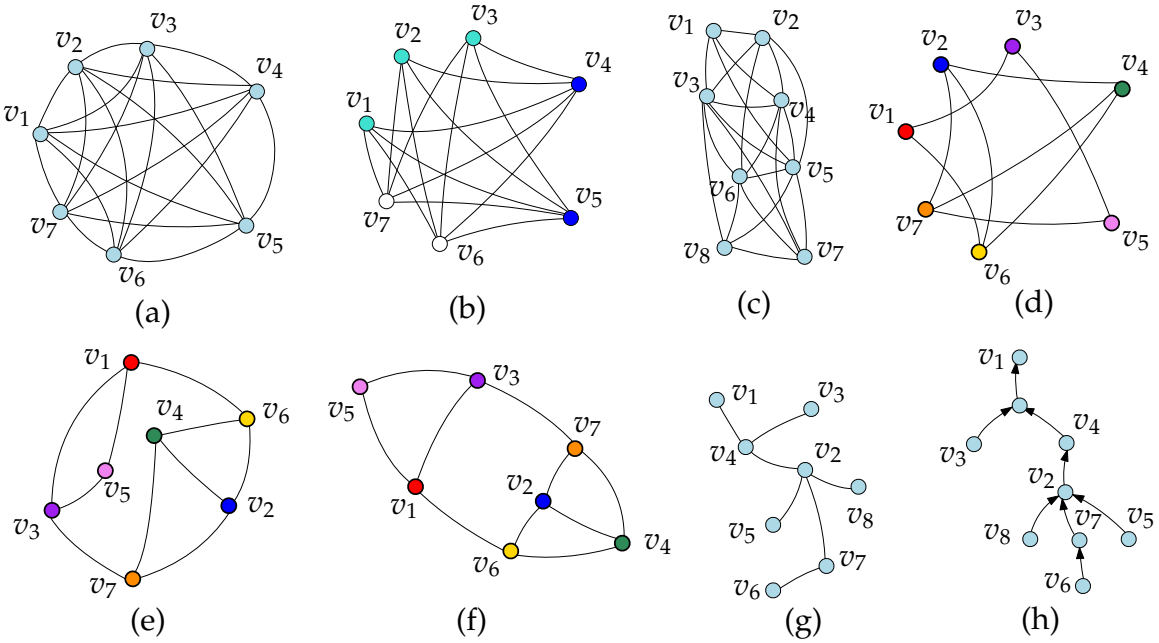


Figure 2.1: (a) A complete graph. (b) A 3-partite graph, where the partitions are shown in different colors. (c) A 4-tree. (d) A planar graph G. (e)–(f) Two planar drawings of G. (g) A tree. (h) A tree rooted at v_1, where each edge is directed from a child to its parent.

Planar Graphs: A *drawing* Γ of a graph G on the Euclidean plane maps every vertex of G to a distinct point, and each edge of G to a simple curve between its endpoints. Two edges of G create an *edge crossing* in Γ if their corresponding curves intersect except possibly at their common endpoints. A graph is called *planar* if it can be drawn on the Euclidean plane without any edge crossing. The corresponding drawing Γ is called a *planar drawing*. Figure 2.1(d) depicts a planar graph and Figure 2.1(e) illustrates a planar drawing of G. Every bounded region in Γ is known as an *inner face*, and the unbounded region is called the *outer face*. G is called an *outerplanar graph* if it admits a planar drawing with all the vertices incident to the outer face. Given a graph, there exist linear-time algorithms to decide whether it is a planar graph [113]. If the input is a planar graph, then

these algorithms produce a planar drawing also in linear time.

Planar Triangulations: A planar graph G is called a *maximal planar graph* if addition of any more edges to G violates planarity. Let Γ be a planar drawing of a maximal planar graph. Then every face in Γ (including the outer face) contains exactly three edges on its boundary. Hence a maximal planar graph is often referred to as a *planar triangulation*.

By Euler's formula [59], every planar graph G satisfies the equality $n - m + f = 2$, where f is the number of faces in the graph. As a consequence, the maximum number of edges and faces in any planar graph can be upper bounded by $3n - 6$ and $2n - 4$, respectively. These bounds are tight for planar triangulations with $n \geq 3$.

Rotation System and Plane Graphs: A *rotation system* of a planar drawing is the clockwise order of the incident edges around each vertex of the drawing. Two drawings with the same rotation system are considered to be equivalent, and the equivalence class is known as the *combinatorial embedding* of G. Figures 2.1(e)–(f) show two different combinatorial embeddings of a common planar graph. A *plane graph* is a fixed combinatorial embedding of a planar graph, i.e., it consists of a rotation system and a designated outer face.

Let G be a plane graph with n vertices, m edges and f faces. The *dual graph* G^* of G is a plane graph such that each vertex of G^* corresponds to a distinct face of G, and two vertices in G^* are adjacent if and only if the corresponding faces in G share an edge on their boundary.

Trees and Forests: A *tree* is a connected graph without any cycle. Every tree with n vertices contains exactly $n-1$ edges. A *forest* is a collection of trees. A *linear forest* is a collection of paths. Both trees and forests are outerplanar graphs.

A *rooted tree* T contains a designated vertex, which is called the *root* of the tree. Each vertex of degree one in T, which is not the root, is a *leaf* of T. Every vertex in T corresponds to a unique path from itself to the root. A vertex v is an *ancestor* of another vertex w, if the path from w to the root contains v, where w is called the *descendent* of v. Every immediate descendent of v is a *child* of v, where v plays the role of the *parent*. An *ordered tree* is a rooted tree with a rotation system. A *k-ary tree* is a tree, where each node can have at most k children. Figures 2.1(g) and (h) depict a tree and a 3-ary rooted tree, respectively.

2.2 Graph Properties and Parameters

In this section we give an overview of various graph parameters related to this thesis. We refer interested readers to [82, 117] for more details on graph properties.

Connectedness: A graph G is *connected* if every pair of vertices belong to some path in G. G is called a *k-connected* graph, where $k \geq 1$, if removal of fewer than k vertices leaves the graph connected. Consequently, a k-connected graph, where $k > 1$, is also $(k-1)$-connected. Every k-tree with $n > k$ is k-connected. Every triangulation with more than three vertices is 3-connected. A 3-connected planar graph can have at most $O(n)$ combinatorial embeddings [114]. Since every

planar graph has a vertex of degree at most five [117], planar graphs are at most 5-connected.

Treewidth and Pathwidth: : A *tree decomposition* of a graph $G = (V_G, E_G)$ corresponds to a tree $T = (V_T, E_T)$ that satisfies the following conditions.

(a) Each node $w \in V_T$ is associated to a set $S_w \subseteq V_G$, and $\bigcup_{w \in V_T} S_w = V_G$.

(b) For every edge $(u, v) \in E_G$, there exists a node w in T such that $\{u, v\} \subseteq S_w$.

(c) For every vertex $u \in V_G$, the nodes in T that correspond to u induce a connected subgraph of T.

The *width of the tree decomposition* is $\max_{w \in V_T}\{|S_w| - 1\}$. The *treewidth* of G is the minimum width over all the tree decompositions of G. If T is a path, then the above decomposition is called a *path decomposition* of G, and the width of the decomposition is called the *pathwidth* of G. Figure 2.2(a) illustrates a graph G of treewidth 3, and Figures 2.2(b) and (c) illustrate a tree decomposition and a path decomposition of G with width 3 and width 4, respectively.

The pathwidth of a graph with n vertices is bounded by $O(t \log n)$, where t is the treewidth of the graph [13]. The upper bound on the pathwidth, and hence also on the treewdth, of planar graphs is $O(\sqrt{n})$ [13], and there exist planar graphs with treewidth $\Omega(\sqrt{n})$, e.g., a $\sqrt{n} \times \sqrt{n}$ grid graph. We refer the reader to [12, 82] for more details on the treewidth and pathwidth parameters.

Thickness and Arboricity: : The *thickness* of a graph $G = (V, E)$ is the smallest integer $\theta(G)$ such that the edges of G can be partitioned into $\theta(G)$ subsets, where for each subset $E' \subseteq E$, the graph $G' = (V, E')$ is planar [56]. Different variants of

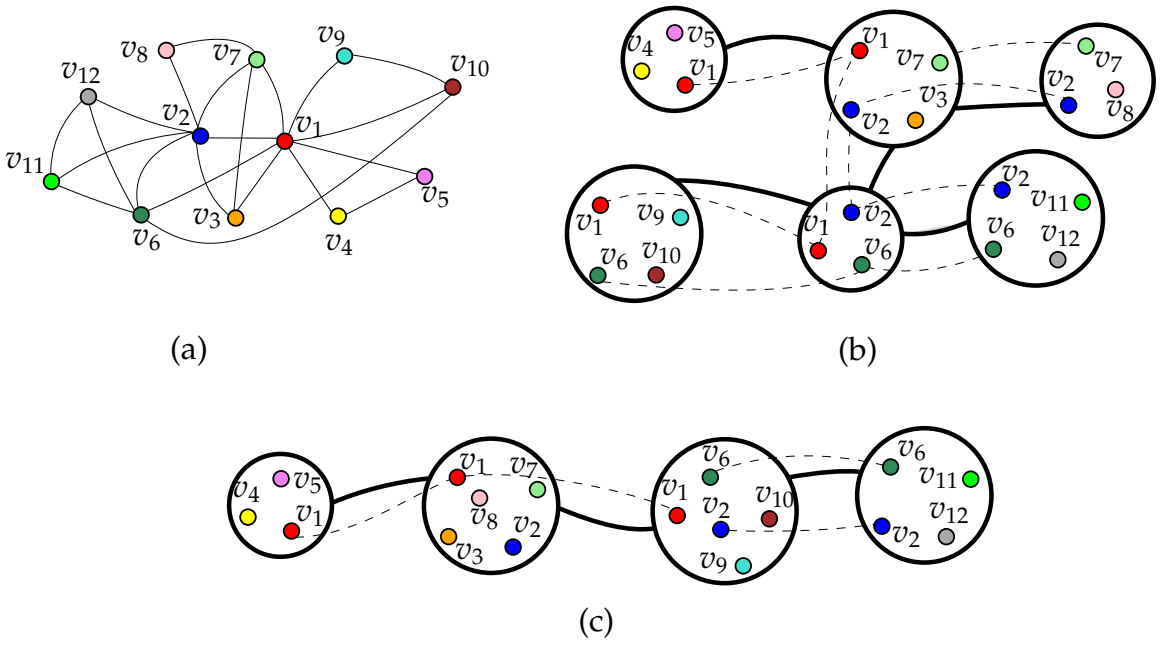
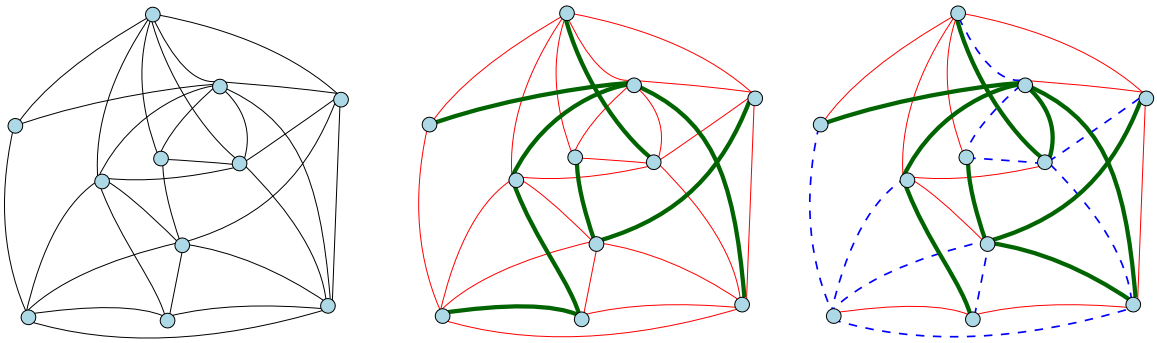


Figure 2.2: (a) A graph G. (b) A tree decomposition of G with width 3. (c) A path decomposition of G with width 4. The tree is shown in bold, and the Property (c) of the decomposition is illustrated in dashed lines.

thickness have been studied in the literature. For example, the *outerthickness* of G is the minimum integer t such that E can be partitioned into t subsets, each forming an outerplanar graph. Similarly, the *(linear) arboricity* of G is the minimum integer t such that E can be partitioned into t subsets, each forming a (linear) forest. Figure 2.3(a) illustrates a graph G of thickness 2, and Figure 2.3(b) illustrates a decomposition of the edges of G into two planar graphs. The arboricity of G is 3, as shown in Figure 2.3(c). We refer the reader to the survey [95] for more details on graph thickness.

Bandwidth and Diameter: The *bandwidth* of a graph $G = (V, E)$ is the minimum integer b such that the vertices can be assigned distinct integers satisfying for each edge $(u, v) \in E$, the condition $|f(u) - f(v)| \leq b$, where for every $w \in V$,



(a) (b) (c)

Figure 2.3: (a) A graph G with 11 vertices and 29 edges. (b) A decomposition of the edges of G into two planar graphs. (c) A decomposition of the edges of G into three forests.

$f(w)$ denotes the integer associated to w. The corresponding labelling is called a *bandwidth labelling*. Intuitively, a bandwidth labelling minimizes the maximum edge length, where an edge length is defined as the absolute difference of the integral labels associated to its endpoints. The *diameter* of G is the maximum shortest path distance over all pairs of vertices of G. The bandwidth of G is at least $\lceil (n-1)/\text{diameter}(G) \rceil$, and at most $n - \text{diameter}(G)$ [23].

2.3 Drawing Styles and Drawing Parameters

This section compiles a brief description of some popular graph drawing styles that are in the scope this thesis. We refer the interested readers to [98, 113].

2.3.1 Drawing Styles

Here we discuss three different forms of drawing graphs. The first one is the polyline drawing and its variants, the second one restricts the position of the

vertices to a set of given locations, and finally, the third one is a more general representation of graphs that allows the vertices to take different shapes.

Polyline Drawings: Let G be a graph and let Γ be a drawing of G on the Euclidean plane. Γ is called a *k-bend polyline drawing* if each edge in Γ is drawn as a polygonal chain of at most $k+1$ straight line segments. Each intermediate vertex of such a polygonal chain is called a *bend* on the corresponding edge. Consequently, each edge in a k-bend polyline drawing can have at most k bends. If the polygonal chains of the drawing consist of only horizontal and vertical line segments, then the drawing is called an orthogonal drawing. A polyline drawing is called a *grid drawing* if the vertices and bends in the drawing are located at integral coordinates. Figure 2.4(a) illustrates a 1-bend polyline grid drawing, and Figure 2.4(b) depicts a 2-bend orthogonal grid drawing.

Polyline drawings with no bend are known as *straight-line drawings*, i.e., every edge in such a drawing is mapped to a straight line segment. Let Γ be a straight-line drawing. Γ is called a *convex drawing* if the boundary of every face in the drawing forms a convex polygon.

Point-set Embeddings: Sometimes graphs are drawn on a given set of points. Such drawings are known as point-set embeddings. Formally, a *(planar) point-set embedding* of a graph G with n vertices on a set S of n points is a (planar) polyline drawing of G, where every vertex of G is mapped to a distinct point of S. In a k-bend point-set embedding, each edge is restricted to have at most k bends. Figure 2.4(c) and (d) illustrate a planar graph G with 6 vertices and a set of 6

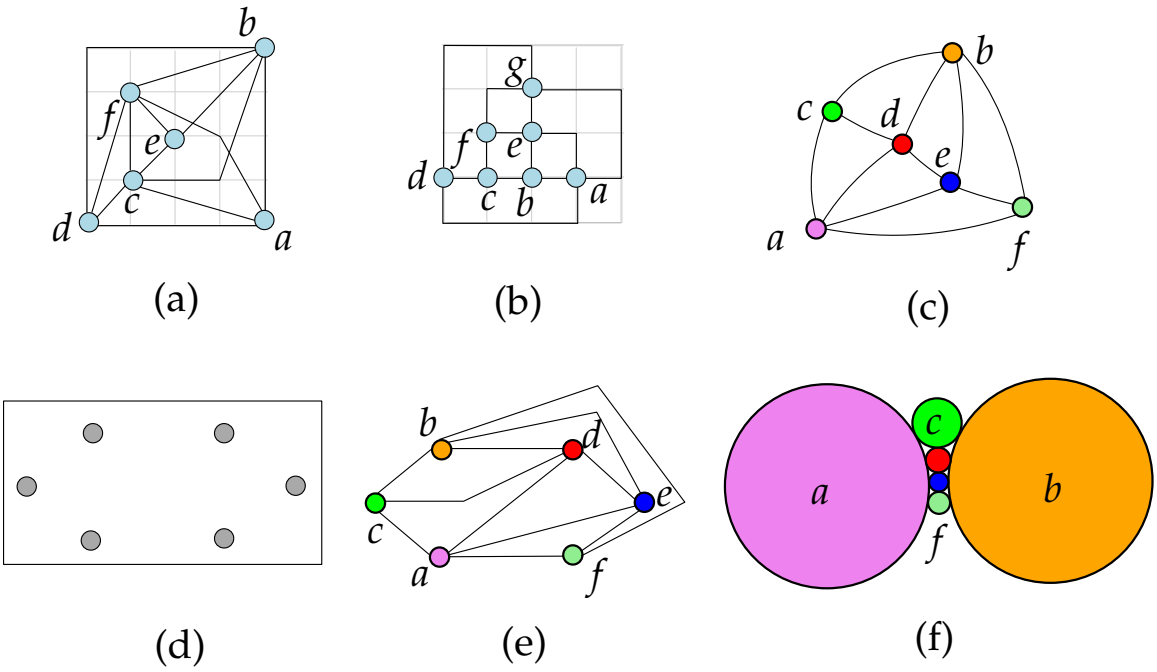


Figure 2.4: (a) A polyline grid drawing. (b) An orthogonal grid drawing. (c) A planar graph G of 6 vertices. (d) A point set S of 6 points. (e) A point-set embedding of G on S. (f) A circle-contact representation of G.

points, respectively. Figure 2.4(e) depicts a 2-bend planar point-set embedding of G on S.

Every planar graph with n vertices admits a 2-bend planar point-set embedding on any set of n points in general position [79], whereas outerplanar graphs do not require any bend [16]. Any reference to the term 'point-set embedding' in this thesis will denote a planar polyline drawing of the underlying graph.

Contact Representations: Let Ψ be a two dimensional geometric shape, e.g., disks, hexagons, triangles, etc. A Ψ-*contact representation* of a graph G on the Euclidean plane maps each vertex v of G to a two dimensional object O_v of shape Ψ such that two objects intersect if and only if the corresponding vertices are ad-

jacent in G. However, intersection is allowed only at the boundary, i.e., the open interiors of the objects do not intersect. For example, Figure 2.4(f) depicts a disk-contact representation of the planar graph of Figure 2.4(c). Disk-contact representations are also known as *circle-contact representations*. Every planar graph admits a circle-contact representation [84] and a contact representation with hexagons [45]. We refer the reader to [3, 100] for more details on the contact representations of planar graphs.

2.3.2 Drawing Parameters

Here we discuss some standard drawing parameters that are commonly used to measure the quality of a drawing.

Area: The *area* of a drawing Γ is the area of the smallest axis-aligned rectangle R that encloses Γ. The *width* and the *height* of Γ are denoted by the width and the height of R, respectively. To measure the area we often require Γ to be a grid drawing.

Bend Complexity: The *bend complexity* is a parameter of polyline drawings. The bend complexity is the maximum number of bends per edge in the drawing, i.e., a k-bend polyline drawing has bend complexity k.

Angular Resolution: An *angle* between a pair of straight line segments incident to a vertex v is the minimum amount of rotation required to place one segment on top of the other. The *angular resolution* of a drawing is the minimum angle

formed at any vertex in the drawing.

Segments: A *segment* in a drawing is a maximal path such that all the vertices on the path is collinear. The *(planar) segment number* of a graph G is the minimum integer t such that G admits a (planar) drawing with at most t segments.

Slopes: The *slopes* of a drawing are the slopes of its segments. The *(planar) slope number* of a graph G is the minimum integer t such that G admits a (planar) drawing, where the number of distinct slopes for the edges is at most t.

Layer Complexity and Geometric Thickness: Two edges are *non-crossing* in a polyline drawing if they do not intersect except possibly at their common endpoints. Otherwise, we say that the edges cross. The *layer complexity* of a polyline drawing Γ is the minimum integer t such that the polyline edges of Γ can be partitioned into t sets of non-crossing edges. The *geometric thickness* of a graph G is the minimum integer $\bar{\theta}(G)$ such that G admits a straight-line drawing with layer complexity $\bar{\theta}(G)$.

Figure 2.5(a) illustrates a polyline drawing of K_9 on 3 planar layers with bend complexity 1. At first glance the layer complexity of Γ may appear to be related to the thickness of G. However, the layer complexity is a property of the drawing Γ, while thickness is a graph property. The layer complexity of Γ can be arbitrarily large even when G is planar, e.g., consider the case when G is a matching and Γ is a straight-line drawing, where each edge crosses all the other edges; see Figure 2.5(b).

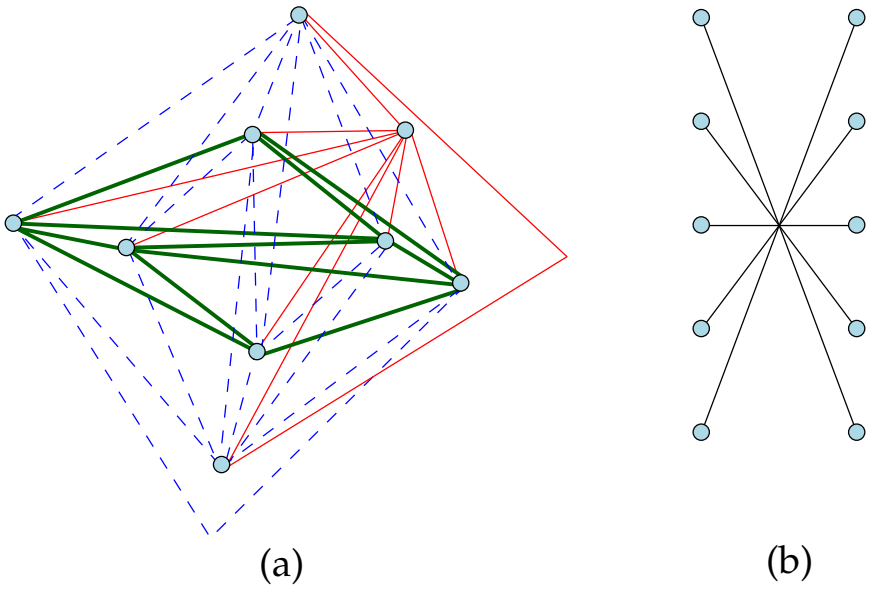


Figure 2.5: (a) A polyline drawing of K_9. (b) A drawing of a matching of size 5.

Crossings: The number of *crossings* in a drawing is the number of pairs of edges that create a proper intersection. The *crossing number* $cr(G)$ of a graph G is the minimum integer t such that G admits a drawing with at most t crossings.

2.4 Tools for Drawing Graphs

In this section we review a few important combinatorial properties of graphs, which are commonly used as the basic tools for drawing graphs.

2.4.1 Canonical Ordering

Let G be a triangulated plane graph with n vertices. A subgraph of G is *internally triangulated* if each of its inner faces is a cycle of length three.

Let v_1, v_2 and v_n be the outer vertices of G in clockwise order, and let $\sigma =$

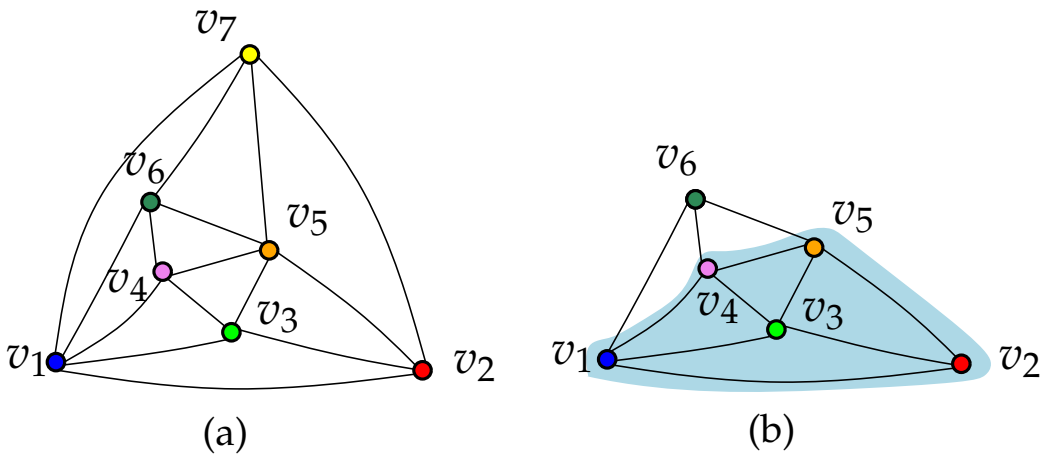


Figure 2.6: (a) A canonical ordering. (b) The graph G_k, where $k = 6$. The graph G_5 lies in the shaded region.

$(v_1, v_2, ..., v_n)$ be an ordering of all vertices of G. By G_k, $3 \leq k \leq n$, we denote the subgraph of G induced by $\{v_1, v_2, \ldots, v_k\}$. By P_k we denote the path (while walking clockwise) on the outer face of G_k that starts at v_1 and ends at v_2. We call σ a *canonical ordering* of G with respect to the outer edge (v_1, v_2) if for each k, $3 \leq k \leq n$, the following conditions are satisfied.

(a) G_k is 2-connected and internally triangulated.

(b) If $k + 1 \leq n$, then v_{k+1} is an outer vertex of G_{k+1} and the neighbors of v_{k+1} in G_k appear consecutively on P_k.

Every planar triangulation admits a canonical ordering [32]. Figure 2.6 illustrates a planar triangulation, where the vertices are labeled in a canonical order. We refer the reader to [113, 121] for different variants and generalization of canonical ordering to 3-connected planar graphs.

2.4.2 Schnyder Realizer

Let G be a plane triangulation with n vertices, and let $\sigma = (v_1, v_2, ..., v_n)$ be a canonical ordering of G, as illustrated in Figure 2.7(a). For some k, where $3 \leq k \leq n$, let P_k be the path $w_1(= v_1), \ldots, w_l, v_k(= w_{l+1}), w_r, \ldots, w_t(= v_2)$ on the subgraph G_k. We call the edges (w_l, v_k) and (v_k, w_r) the *l-edge* and the *r-edge* of v_k, respectively. The other edges incident to v_k in G_k are called the *m-edges* of v_k. For example, in Figure 2.7(c), the edges $(v_1, v_6), (v_6, v_5)$, and (v_4, v_6) are the l-, r- and m-edges of v_6, respectively. Let E_m be the set of all m-edges in G. Then the graph T_m induced by the edges in E_m is a tree with root v_n. The graph T_l induced by all l-edges except (v_1, v_n) is a tree rooted at v_1, e.g., see Figure 2.7(b). Similarly, the graph T_r induced by all r-edges except (v_2, v_n) is a tree rooted at v_2. These three trees form the *Schnyder realizer* [108] of G. Each of T_l, T_r and T_m corresponds to a canonical ordering of G, and hence is called a *canonical ordering tree* of G [31, 121]. Finally, each internal vertex v of G respects the following *edge-ordering properties*:

P_1. The vertex v is incident to exactly three outgoing edges, one from each Schnyder tree.

P_2. If we order the edges incident to v in clockwise order (Figure 2.7(d)), then starting from the outgoing m-edge, we observe 0 or more incoming l-edges, then the outgoing r-edge, then 0 or more incoming m-edges, then the outgoing l-edge, then 0-or more incoming r-edges, and then again the outgoing m-edge.

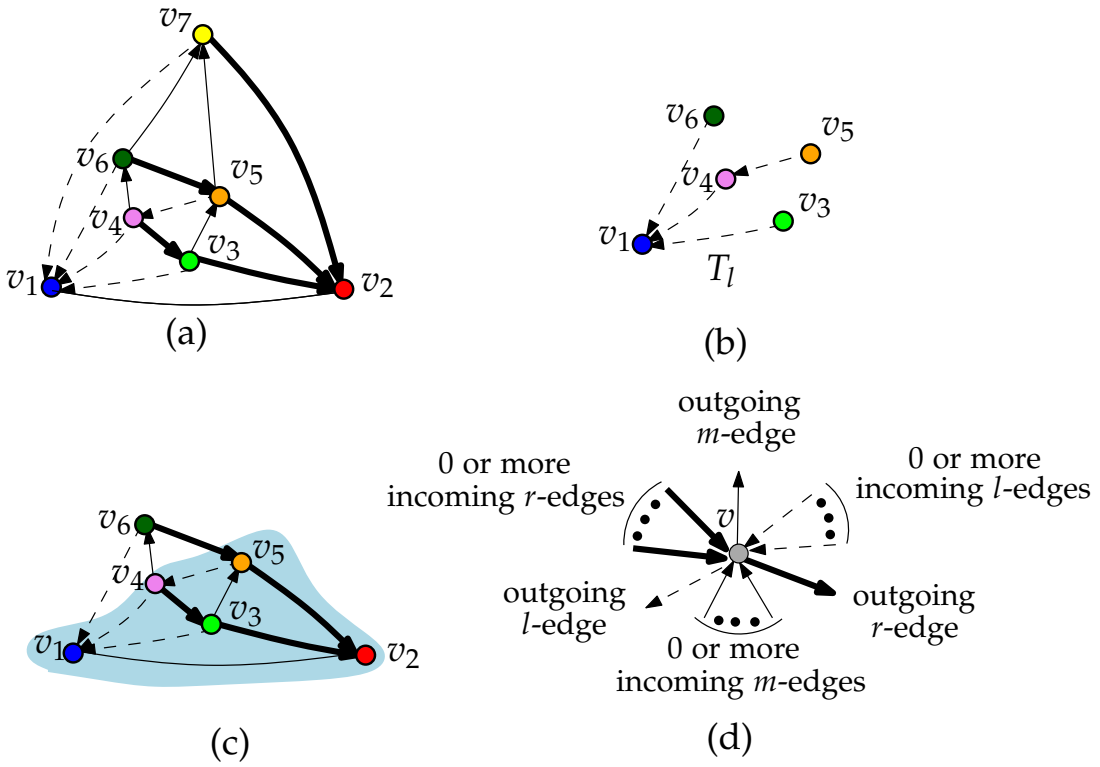


Figure 2.7: (a) A plane triangulation G with a canonical ordering of its vertices, and a corresponding Schnyder realizer, where the l-, r- and m- edges are shown in dashed, bold-solid, and thin-solid edges respectively. (b) T_l. (c) Illustration for canonical ordering, when $k+1=6$. (d) Edge-ordering property.

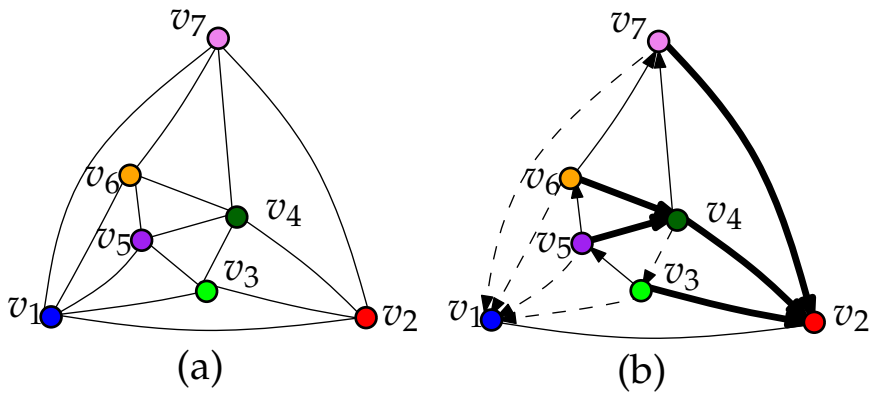


Figure 2.8: (a) Another canonical ordering of the graph of Figure 2.7(a). (b) The associated Schnyder realizer, which is a minimum realizer. The only cyclic face is v_4, v_3, v_5, v_4, which is oriented clockwise.

A planar graph may have many canonical orderings, and hence many different Schnyder realizers. The *minimum realizer* is a unique Schnyder realizer with all the cyclic inner faces oriented clockwise, as shown in Figure 2.8. The number of cyclic inner faces in the minimum realizer is denoted by δ_0 [121]. A minimum realizer can be found in linear time using the edge contraction method for computing Schnyder realizer [108], and by carefully choosing the edges to contract [20].

Like canonical orderings, Schnyder realizer has also been extended to 3-connected planar graphs. We refer the readers to [4, 10] for more details on Schnyder realizer.

2.4.3 Graph Separator

A *separator* of a graph $G = (V, E)$ is a set of vertices $S \subset V$ such that the set $V \setminus S$ can be partitioned into two non-empty sets A and B such that there does not exist any edge $(a, b) \in E$, where $a \in A$ and $b \in B$. The size of a separator is the number of vertices in S. A separator is called an *α-separator* if the $\max\{|A|, |B|\} \leq \alpha n$. Throughout the thesis, by a separator we will denote a $(2/3)$-separator. In this thesis we will only use separators of planar graphs. For more details on graph separators and its applications we refer the reader to [89].

In 1951, Ungar [115] proved that every planar graph G with n vertices contains a separator of size $O(\sqrt{n}\log n)$. Later, Lipton and Tarjan [90] improved the bound to $O(\sqrt{n})$. A simple cycle C in G is called a *cycle separator* if the interior and exterior of C each contains at most $2n/3$ vertices. Every planar graph has a simple cycle separator of size $O(\sqrt{n})$ [38]. An *edge separator* of $G = (V, E)$ is a subset of

edges M of G such that $G = (V, E \setminus M)$ consists of two induced subgraphs, each containing at most $2n/3$ vertices. Every planar graph admits an edge separator of size $2\sqrt{2\Delta n}$, where the corresponding edges in the dual graph form a simple cycle [36].

2.5 Algorithms and Complexity

In this thesis we will develop different algorithms to solve various graph drawing problems. We will measure the efficiency of these algorithms based on its running time or space requirements. Divide and conquer algorithm, dynamic programming and greedy algorithms are some of the common types of algorithmic techniques that are in the scope of this thesis. We refer interested readers to Kleinberg and Tardos [81] for the details of these algorithmic techniques.

By a *polynomial-time algorithm* we denote an algorithm whose running time is bounded by a polynomial function of the size of the input instance. For graph drawing problems, the size of an input is measured by the number of vertices and edges of the input graph. Therefore, for a graph G with n vertices and m edges, a polynomial-time drawing algorithm would take $O(n^{O(1)} + m^{O(1)})$ time to process G and to produce the output.

We will also require some background on the complexity classes of the problems. Let P be a *decision problem*, i.e., a problem that can be answered with either "yes" or "no", and let $I(P)$ be the set of all instances of the problem X. A *polynomial time reduction* of a decision problem P into a decision problem P' defines a function $f : I(P) \to I(P')$ such that f is computable in polynomial time, and

for every instance $I \in I(P)$, there exists an affirmative solution to I if and only if there exists an affirmative solution to $f(I) \in I(P')$.

A decision problem Q belongs to the *complexity class* P (respectively, NP), if a deterministic Turing machine (respectively, non-deterministic Turing machine) can solve the problem in polynomial time. Q is called *NP-hard*, if some problem in complexity class NP can be reduced to Q in polynomial time. Q is an *NP-complete* problem, if it is NP-hard, and any given solution to Q can be verified in polynomial-time. We refer interested readers to [62] for more details on complexity theory.

Chapter 3

Minimizing Segments and Area

This chapter presents our results on the interaction between segment and area in straight-line drawings of planar graphs. Dujmović et al. [41] showed that every 3-connected planar graph G with n vertices admits a straight-line drawing with at most $2.5n - 3$ segments, which is also the best known upper bound when restricted to planar triangulations. On the other hand, they showed that there exist triangulations requiring $2n - 6$ segments.

In this chapter we prove that every planar triangulation admits a straight-line drawing with at most $(7n - 2\delta_0 - 10)/3 \leq 2.34n$ segments, where δ_0 is the number of cyclic faces in the minimum realizer of G. For general planar graphs with n vertices and m edges, our algorithm requires at most $(16n - 3m - 28)/3 \leq 5.34n - m$ segments, which is less than $2.5n - 3$ for all $m \geq 2.84n$. In the context of grid drawings, where the vertices are restricted to have integer coordinates, we show that every triangulation with maximum degree Δ can be drawn with at most $2n + t - 3$ segments and $O(8^t \cdot \Delta^{2t})$ area, where t is the minimum number of leaves

over all the trees of the minimum realizer. This is the first non-trivial attempt to simultaneously optimize the area and the number of segments while drawing triangulations. These results extend to the case when the goal is to optimize the number of slopes in the drawing.

We first briefly review the motivation behind minimizing the segments and area while drawing planar graphs. We then discuss our contribution in the context of related research. The subsequent sections present the technical background necessary to describe the results (Section 3.2), the drawing algorithms (Sections 3.3–3.4), a trade-off between the number of segments and area (Section 3.5), limitations of our approach and directions to future research (Section 3.6).

Some of the results of this chapter appeared in preliminary form at the 26th Canadian Conference on Computational Geometry [48].

3.1 Minimizing Segment and Area

Straight-line drawings are preferable since the use of bends may make it more difficult to follow the edges in the drawing. Drawings with few segments further enhance this straightness aesthetic. Besides the theoretical appeal of the problem, segment minimization finds application in edge-disjoint path cover problems [25, 26], where the task is to decompose the edges of the input graph into a small number of edge-disjoint paths. A k-segment drawing corresponds to a path cover of the underlying graph using k paths. In addition to being edge disjoint, these paths satisfy a stronger condition, i.e., each of them is an induced path. Such a decomposition of graphs into edge-disjoint induced subgraphs has also been

examined in the literature [76, 103].

Since the area of a display is limited, minimizing the area of a drawing is a natural optimization for most visualization techniques [113]. Area minimization becomes particularly important for circuit layout construction in VLSI design, class diagram visualization of large software systems, and visualization on small display devices such as tablets or smartphones. Intuitively, minimizing segment and area are conflicting optimization goals. Since no formal study examining the interaction between segment and area is known, we were motivated to bound the drawing area while optimizing the number of segments.

3.1.1 Related Work

Dujmović et al. [41] gave a constructive proof that every 3-connected planar graph with n vertices admits a drawing with at most $5n/2$ segments and $2n$ slopes. However, their algorithm does not necessarily produce a grid drawing, i.e., the algorithm uses real coordinates for positioning the vertices. Dujmović et al. [41] did not formally explore the area requirement of the drawings; instead, one of their open questions was to examine the interaction between the area and segment parameters.

Besides 3-connected planar graphs, Dujmović et al. [41] proved tight upper and lower bounds on the number of segments for several other classes of planar graphs such as outerplanar graphs (n segments), and planar k-trees ($2n$ segments) with $k \in \{2, 3\}$. They also examined planar graphs, i.e., when the embedding of the input graph is not given. As a natural open problem they asked to determine

the minimum constant c such that every planar graph with n vertices admits a straight-line drawing with at most $cn + O(1)$ segments (in both fixed and variable embedding settings).

The optimization version of the problem, i.e., the problem of computing a drawing with minimum number of segments, is NP-hard even for arrangement graphs [53]. Besides, there exist arrangement graphs that require doubly exponential area in any of their minimum-segment drawings [55]. However, polynomial-time algorithms for computing drawings with minimum number of segments have been achieved for trees [41] and planar 2-trees with maximum degree three [106]. For trees, it is straightforward to arrange the edges around each vertex such that only the odd-degree vertices become the endpoints of the segments. Thus there are exactly $\lambda/2$ segments, where λ is the number of vertices with odd degree. Such a drawing can be computed by adding a leaf node to each odd degree vertex of the tree, and then drawing the tree with perfect angular resolution [46], which takes $O(n^8)$ area. Halupczok and Schulz [75] proved that every tree can be drawn with perfect angular resolution inside a disk of radius $n^{3.0367}$, but this drawing is not a grid drawing. The drawing algorithm for planar 2-trees with maximum degree three [106] uses real coordinates for the vertices, and no careful analysis on area is known.

Nearly tight bounds on the number of segments have been achieved for 3-connected cubic planar graphs [94]. The proof relies on the simple property that every vertex in an optimal drawing can have at most one 180° angle, and then on the construction where $n - 4$ vertices satisfy this property. Unlike the drawings

proposed in [53], this drawing for cubic planar graphs is a grid drawing with $O(n^2)$ area.

3.1.2 Our Contribution

We assume that the input is a plane graph, i.e., a combinatorial embedding of the graph is also given as an input, and the output drawing must respect the given embedding. Table 3.1 lists our results, as well as summarizes the best known upper and lower bounds on the number of segments for different classes of planar graphs. Note that the '$\times$' marked cells denote that no non-trivial area upper bound is known for the corresponding cases.

Graph Class	L.B.	U.B.	Area	Ref.
Trees	$\lambda/2$	$\lambda/2$	$O(n^8)$	[46, 53]
Maximal outerplanar graphs	n	n	$\times$	[53]
Planar 2-tree (max-degree 3)	$2n$	$2n$	$\times$	[106]
Planar 2- and 3-trees	$2n$	$2n$	$\times$	[41]
3-Connected cubic planar graphs	$n/2+3$	$n/2+4$	$O(n^2)$	[94]
3-Connected planar graphs	$2n$	$5n/2$	$\times$	[41]
Our Results				
Triangulations	$2n$	$7n/3$	$\times$	Theorem 1
Triangulations	$2n$	$2n+t-3$	$O(8^t \cdot \Delta^{2t})$	Theorem 4
4-Connected triangulations	$2n$	$9n/4$	$\times$	Theorem 2

Table 3.1: Upper and lower bounds on the number of segments, ignoring additive constants. Here λ is the number of vertices of odd degree, Δ is the maximum degree of the graph, and t is the minimum number of leaves over all the trees of the minimum realizer. All area bounds correspond to the upper bounds on the number of segments, i.e., no algorithm to compute minimum-segment drawings in polynomial area is known except for trees.

3.2 Technical Foundation

In this section we introduce some preliminary definitions and results.

Let p be a point in $\mathbb{R}^2$. We denote the x and y-coordinates of p by $x(p)$ and $y(p)$, respectively. Let $b_1, b_2, \ldots, b_k$ be a strictly x-monotone polygonal chain C. For each i, where $0 < i < k$, an edge (b_i, b_{i+1}) is called a left (respectively, right) edge if the edge has positive (respectively, negative) slope. Later, while describing the drawing algorithm, these left and right edges will correspond to the l-edges and r-edges of a Schnyder realizer.

Let Γ be a straight-line drawing of a planar graph G. A *segment* in Γ is a maximal path of G whose vertices are collinear in Γ. A segment is a *left or right segment* if it contains only left or right edges, respectively. The *tip of a left or right segment* s, denoted by $\texttt{tip}(s)$, is the vertex on s with the highest y-coordinate. The *tip of an edge* (v, w), denoted by $\texttt{tip}(v, w)$, is the tip of the segment that contains the edge (v, w). Two points p and q are *visible* to each other with respect to Γ if the straight line segment pq does not intersect Γ at any point except possibly at p and q. By l_{pq} we denote the line through p and q. We denote the slope of l_{pq} by $\texttt{slope}(p, q)$. A set of rays is *divergent* if no two rays in the set are parallel, and no two rays intersect (except possibly at their common origin).

We now prove two geometric lemmas, which will be use useful to describe our drawing algorithms.

Lemma 1 *Let $a(= b_0), b_1, b_2, \ldots, b_k, c(= b_{k+1})$ be a strictly x-monotone polygonal chain C that lies above the line l_{ac}. Let p be a point above C such that the segments*

ap and cp do not intersect C except at a and c. Assume that for each edge (b_i, b_{i+1}), where $0 \leq i \leq k$, the ray r_i with origin $\texttt{tip}(b_i, b_{i+1})$ and slope $\texttt{slope}(b_i b_{i+1})$ does not intersect C except at $\texttt{tip}(b_i, b_{i+1})$. If the slopes of the left edges of C are smaller than $\texttt{slope}(a, p)$, and the slope of the right edges of C are greater than $\texttt{slope}(p, c)$, then every vertex of C is visible from p (e.g., Figure 3.1(a)).

Proof: It is straightforward to observe that a and c are visible to p. Suppose for a contradiction that some vertex b_j, where $1 \leq j \leq k$ is not visible to p. We now consider three cases depending on the types of the edges (b_{j-1}, b_j) and (b_j, b_{j+1}).

Case 1 ((b_{j-1}, b_j) is a left edge and (b_j, b_{j+1}) is a right edge): Since $\texttt{slope}(a, p) > \texttt{slope}(b_{j-1} b_j)$, the point p lies above the line determined by ray r_{j-1}. Similarly, since $\texttt{slope}(p, c) < \texttt{slope}(b_j b_{j+1})$, the point p lies above the line determined by ray r_j. Such a scenario is illustrated in Figure 3.1(b). Since r_j and r_{j-1} do not intersect C except at b_j, and since C is strictly x-monotone, the straight line segment pb_j cannot intersect C except at b_j. Hence b_j is visible to p.

Case 2 ((b_{j-1}, b_j) is a right edge and (b_j, b_{j+1}) is a left edge): Since $\texttt{slope}(p, c) < \texttt{slope}(b_{j-1} b_j)$, the point p lies above the line determined by ray r_{j-1}. Similarly, since $\texttt{slope}(a, p) > \texttt{slope}(b_j b_{j+1})$, the point p lies above the line determined by ray r_j. Figure 3.1(c) illustrates such an example. Since C lies above l_{ac}, the vertices a and c must lie below r_{j-1} and r_j, respectively. Since r_j and r_{j-1} do not intersect C except at their origins, and since C is strictly x-monotone, the straight line segment pb_j cannot intersect C except at b_j. Hence b_j is visible to p.

Case 3 ((b_{j-1}, b_j) and (b_j, b_{j+1}) are of same type): If both are right edges, then we consider the ray r_{j-1}, and another ray r with origin at b_j. We choose the slope

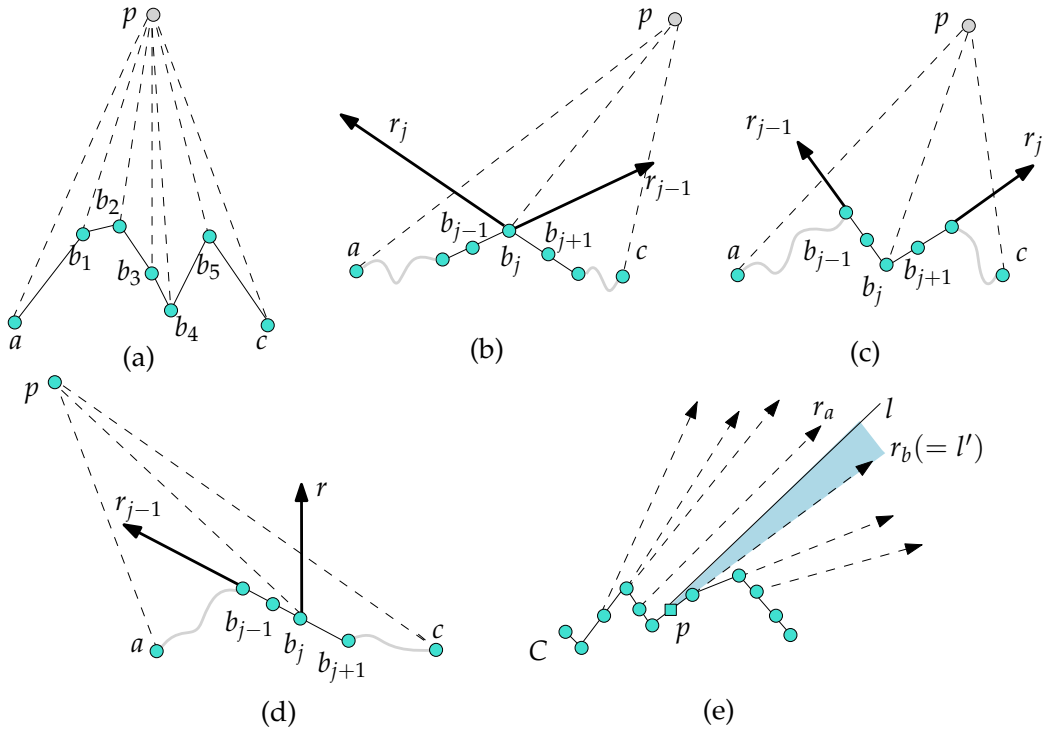


Figure 3.1: (a)–(d) Illustration for the proof of Lemma 1. (e) Illustration for the proof of Lemma 2, where the set R is shown in dashed lines, respectively.

of r to be either $\texttt{slope}(a, p)$ or ∞ depending on whether the slope of ap is negative or positive, e.g., see Figure 3.1(d). We now can use the analysis for Case 2 to show that b_j is visible to p. The case when (b_{j-1}, b_j) and (b_j, b_{j+1}) both are left edges, is handled symmetrically. ■

Lemma 2 *Let C be a strictly x-monotone polygonal chain. Let R be a set of divergent rays obtained by extending the left segments of C above C such that none of these rays intersect C except at their origin. Given a point p on C, one can find a ray r with origin p such that the rays in $R \cup \{r\}$ are divergent, and r does not intersect C except at p.*

Proof: We prove the lemma by constructing the ray r. Let r_a be the last ray that we encounter while walking on C from left to right before we visit p, as shown in Figure 3.1(e). If there are several candidates for r_a, i.e., all with the same origin on C, then we choose the last ray in R in the clockwise order around the origin. If we do not encounter any ray before visiting p, then there is no left edge before p, and we choose r_a as a vertical ray directed upward starting at the leftmost point on C. We now draw a line l parallel to r_a at p.

Similarly, find the last ray r_b that we encounter while walking on C from the right end point of C to the point p (here p itself can be the origin of r_b). If there are several candidates for r_b, i.e., all with the same origin on C, then we choose the first ray in R in the clockwise order around the origin. If we do not encounter any ray before visiting p, then choose r_b as a horizontal ray directed to the right starting at the rightmost point on C. We now draw a line l' parallel to r_b at p.

We now construct the ray r with origin p and slope $(\texttt{slope}(l') + \texttt{slope}(l))/2$. Since $R \cup \{l, l'\}$ is divergent, the set $R \cup \{r\}$ is divergent. Since C is strictly x-monotone and no ray in R intersects C except at its origin, all the points in the region bounded by l and l' above C is visible to p. Hence r cannot intersect C except at p. ∎

3.3 Drawing Planar Triangulations

In this section we prove that every n-vertex planar triangulation G admits a straight-line drawing with at most $(7n - 2\delta_0 - 10)/3$ segments.

Let $\sigma = (v_1, v_2, \ldots, v_n)$ be a canonical ordering of the vertices of G, which corresponds to the realizer T_l, T_r and T_m of G. We first construct a drawing of G using σ, and then bound the number of segments in the constructed drawing.

We now present an overview of the algorithm, and then describe the algorithmic details.

3.3.1 Algorithm Overview

We first draw the face v_1, v_2, v_3 as an isosceles triangle $v_1v_2v_3$, where the base v_1v_2 is aligned along the x-axis and v_1 is to the left of v_2, as shown in Figure 3.2(b). We then add the subsequent vertices according to the canonical order. Let G_i be the graph induced by the vertices $v_1, \ldots, v_i$, and let $T_l(i)$ and $T_r(i)$ be the subgraphs of T_l and T_r in G_i, respectively.

At each step the drawing of the graph G_i maintains some induction invariants. In brief, for each i, the path $P_i = (v_1, \ldots, v_2)$ on the outer face of G_i is drawn x-monotone. The l-edges and r-edges are drawn as left and right edges, i.e., with positive and negative slopes, respectively. Every vertex on P_i, except v_1, v_2, is a leaf of $T_l(i)$ or $T_r(i)$.

While adding the vertex v_{i+1}, we first connect v_{i+1} to P_i via its l-edge and r-edge, and then draw the m-edges. If the l-edge connects to a node that is currently a leaf in $T_l(i)$, then we extend the segment incident to the leaf, e.g., see Figure 3.2(d). Similarly, we do the same for the right edge and the leaves in $T_r(i)$, e.g., see Figure 3.2(e). Note that these are the scenarios where we add the l-edges and r-edges without creating new segments. We need to create new segments for

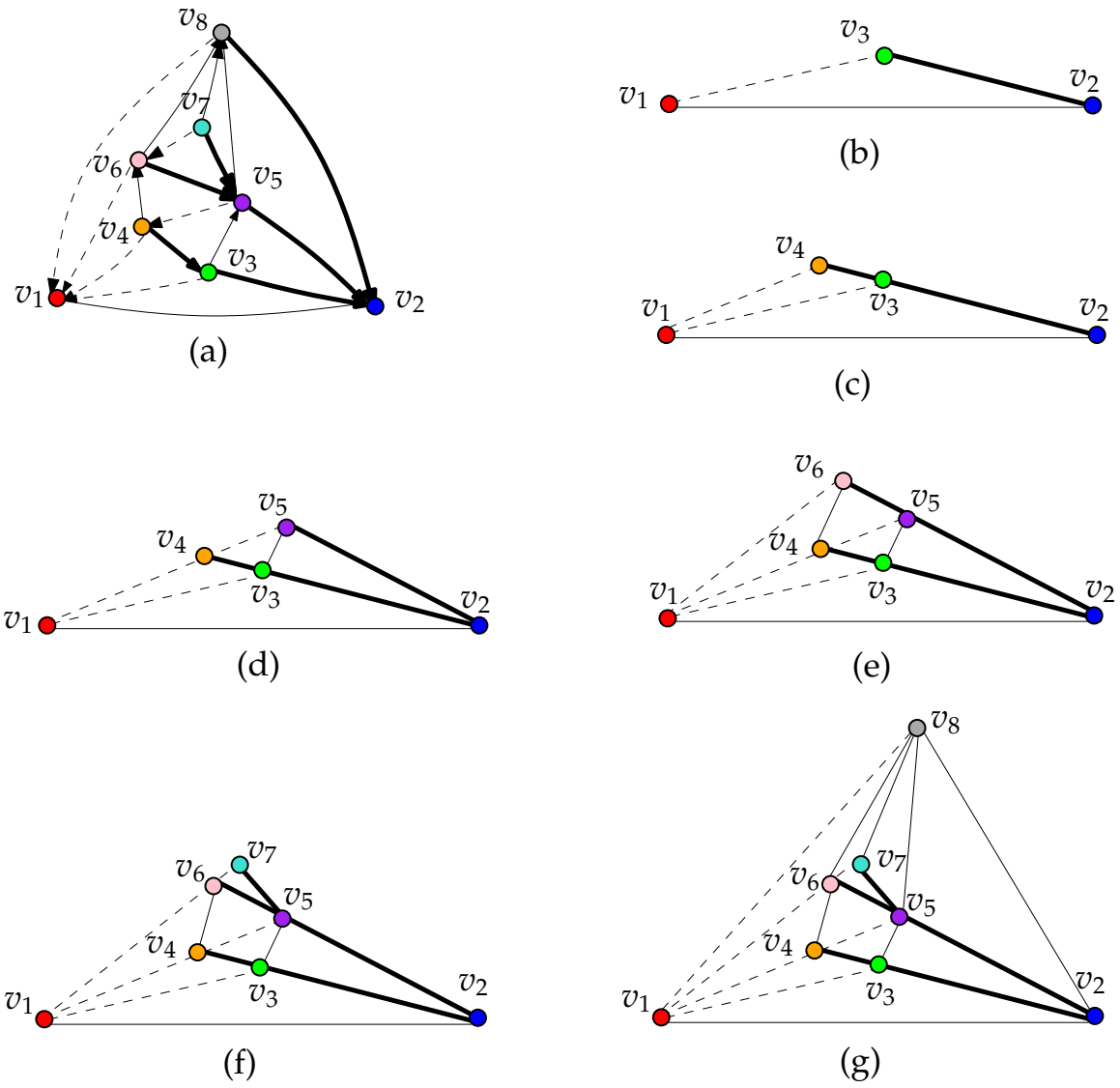


Figure 3.2: A planar graph and the incremental construction of its drawing.

the l-edges and r-edges only when the l-edge (respectively, r-edge) connects to a vertex that is currently not a leaf in T_l (respectively, T_r), e.g., see Figure 3.2(f).

All the m-edges may create new segments, and the leaves of T_l and T_r correspond to distinct segments. Therefore, the total number of segments can be at most $\text{leaf}(T_l) + \text{leaf}(T_r) + n$, which will give us an upper bound of $(7n - 2\delta_0 -$

10)/3 segments.

3.3.2 Algorithm FewSegDraw

We first draw the edge (v_1, v_2) using a horizontal straight line segment. We now complete the drawing of G by adding the vertices $v_3, v_4, \ldots, v_n$ incrementally. Let Γ_i be the drawing of G_i. At each addition, Γ_i will maintain the following invariants, as shown in Figure 3.3(a).

1. The drawing of P_i in Γ_i is strictly x-monotone, and every vertex on P_i, except v_1, v_2, is a leaf of $T_l(i)$ or $T_r(i)$.

2. Let (u, v) be an edge in G_i. If (u, v) is an l-edge, then (u, v) is drawn as a left edge in Γ_i. If (u, v) is an r-edge, then (u, v) is drawn as a right edge in Γ_i.

3. Let Q_l be the set of rays obtained by shooting for every left segment s that has an end point on the outer face of Γ_i, an upward ray with origin $tip(s)$ and slope $\texttt{slope}(s)$. Then Q_l is divergent. Analogously, we define a set of rays Q_r for the right segments, which must be divergent.

4. No ray in $Q_l \cup Q_r$ intersects Γ_i except at its origin. Any two rays $r \in Q_l$ and $r' \in Q_r$ intersect if and only if the origin of r precedes the origin of r' on P_i.

We now add v_3 such that Γ_i is an isosceles triangle with $\angle v_3 v_1 v_2 = \angle v_3 v_2 v_1$, as shown in Figure 3.3(c). Since (v_1, v_3) and (v_3, v_2) are the only l- and r-edges in Γ_3, Invariants 1–4 are straightforward to verify. Assume that the invariants hold for the additions of v_i, where $i < n$, and let Γ_i be the drawing of G_i that respects

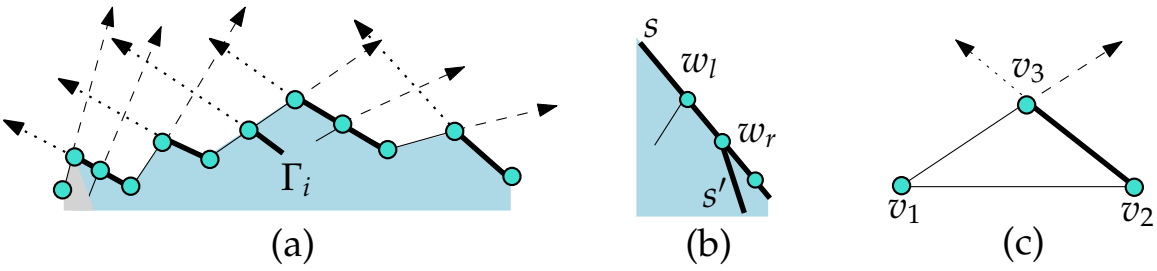


Figure 3.3: (a) Illustrating the invariants for some Γ_i. The path P_i is the upper envelope of the shaded region. The edges of T_l and T_r are shown in thin and bold solid lines, respectively. The set Q_l and Q_r are shown in dashed and dotted lines, respectively. (b) Illustration for Case 1. (c) Drawing of G_3.

Invariants 1–4. We now show how to add v_{i+1} to Γ_i such that the constructed drawing Γ_{i+1} respects all the invariants.

We call a vertex $w \in P_i$ a *peak vertex* if all of its neighbors have smaller y-coordinates than $y(w)$ in Γ_i. The distinction between 'tip' and 'peak' is important, i.e., a vertex w may be a tip of some left (respectively, right) segment, but w is not a peak unless it is also a tip of some right (respectively, left) segment.

Let $w_l, w_{l+1}, \ldots, w_r$ be the neighbors of v_{i+1} in G_i. Note that (w_l, v_{i+1}) and (v_{i+1}, w_r) are the l- and r-edges of v_{i+1}, respectively. We now consider the following three cases. For convenience we assume that v_1 and v_2 are the tips of some left and right segments, respectively, such that the cases when $v_1 (= w_l)$ or $v_2 (= w_r)$ are handled by Case 2.

Case 1 (w_l is a leaf of $T_l(i)$ and w_r is a leaf of $T_r(i)$): By Invariant 2, w_l is a tip of some left segment and w_r is a tip of some right segment. We claim that the segment containing w_l is different than the segment containing w_r.

Otherwise, without loss of generality assume that both lie on some right segment s. By definition, $y(w_l) > y(w_r)$. Hence w_r cannot be a tip of s. If w_r is a tip

of some right segment s' other than s, as shown in Figure 3.3(b), then Invariant 2 will imply that w_r is a child of two different vertices in T_r, which is a contradiction that T_r is a tree.

Note that we can use the above argument to claim that w_r cannot be an internal vertex of a right segment, and similarly, w_l cannot be an internal vertex of a left segment. Figures 3.4(a)–(d) depict the remaining four scenarios.

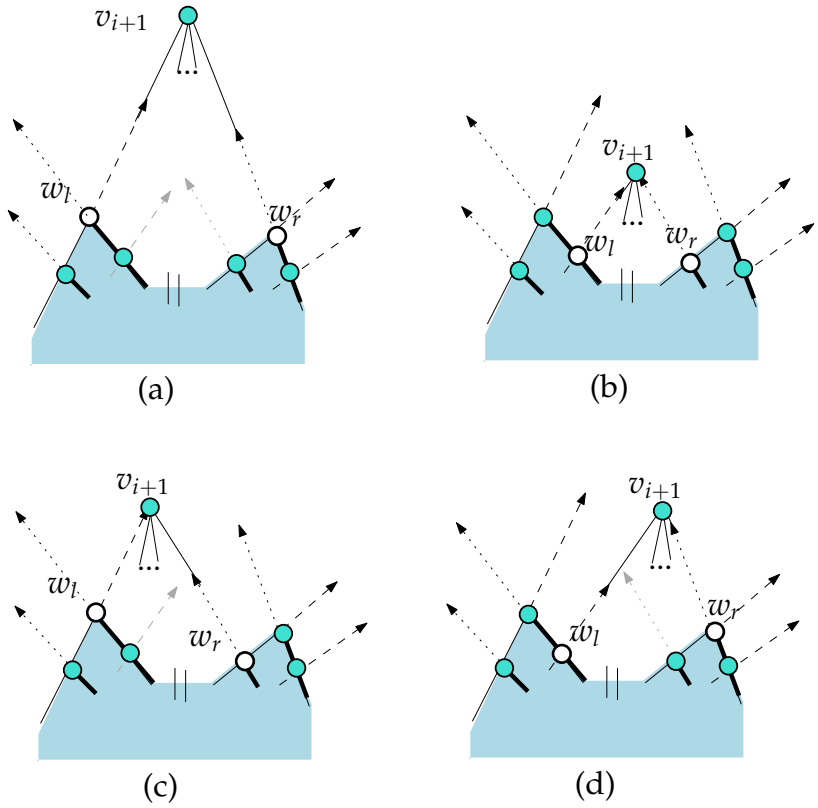


Figure 3.4: Illustrations for different drawings in Case 1, where w_l and w_r are shown in white circles. The left and right edges are shown in thin and bold lines, respectively. (a) w_l and w_r are leaves in both $T_l(i)$ and $T_r(i)$. (b) w_l is a leaf in $T_l(i)$ and w_r is a leaf in $T_r(i)$. (c) w_l is a leaf in both $T_l(i)$ and $T_r(i)$, and w_r is a leaf in $T_r(i)$. (d) w_l is a leaf in $T_l(i)$, and w_r is a leaf in both $T_l(i)$ and $T_r(i)$.

Observe now that by Invariants 3–4, the ray in Q_l emanating from w_l intersects the ray in Q_r emanating from w_r, and none of these rays intersect Γ_i. Let c be

the intersection point of these two rays. We place v_{i+1} at c and draw the edges (v_{i+1}, w), where $w \in \{w_l, w_{l+1}, \ldots, w_r\}$. We claim that the drawing of the m-edges does not create any edge crossing, as follows. By Invariant 3, all the right edges in the path $w_l, w_{l+1}, \ldots, w_r$ have slope larger than $\texttt{slope}(v_{i+1}, w_r)$. Similarly, all the left edges have slope smaller than $\texttt{slope}(v_{i+1}, w_l)$. Hence by Lemma 1, the drawing of the m-edges does not create any edge crossings.

Case 2 (w_l is a leaf of $T_r(i)$ and w_r is a leaf of $T_l(i)$): By Invariant 2, w_l is a tip of some right segment and w_r is a tip of some left segment. If Case 1 is also satisfied, i.e., if w_l and w_r both are peaks in Γ_i, then we add v_{i+1} as in Case 1. Otherwise, at most one of w_l and w_r is a peak.

Case 2A. If none of w_l and w_r are peaks, then we construct two rays r_1 and r_2 starting from w_l and w_r, respectively, such the slope of r_1 is $\texttt{slope}(w_l, w_{l+1}) + \epsilon_1$, and the slope of r_2 is $\texttt{slope}(w_{r-1}, w_r) - \epsilon_2$. Here ϵ_1 and ϵ_2 are two constants such that the sets Q_l and Q_r respect Invariant 3. Lemma 2 guarantees the existence of such constants. Figure 3.5(a) illustrates such a scenario. We then place the vertex v_{i+1} at the intersection point of r_1 and r_2, and draw its l-, r- and m-edges. By Lemma 1, the drawing of these edges does not create any edge crossing.

Case 2B. If exactly one of w_l and w_r is a peak, then without loss of generality assume that w_r is a peak vertex. We then construct a ray r starting from w_l with $\texttt{slope}(w_l, w_{l+1}) + \epsilon$ such that the rays of $Q_l \cup \{r\}$ are divergent and maintain Invariant 3. Lemma 2 guarantees the existence of such a constant ϵ. Figure 3.5(b) illustrates this scenario. We then place the vertex v_{i+1} at

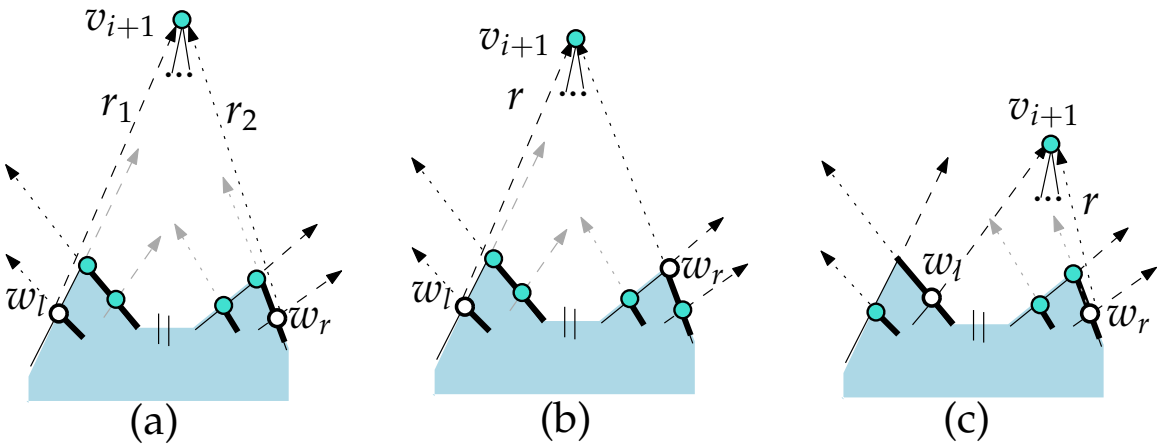


Figure 3.5: (a)–(b) Illustration for Case 2. (c) Illustration for Case 3.

the intersection point of r and the ray in Q_r emanating from w_r. Finally, we draw the l-, r- and m-edges of v_{i+1}. Lemma 1 ensures that the drawing of these edges does not create any edge crossing.

Case 3 (Either $T_l(i)$ or $T_r(i)$ contains both w_l and w_r as leaves): By Invariant 2, w_l and w_r both are tips of the same type of segments.

Consider first the case when at least one of w_l and w_r is a peak. If both are peaks, then we follow Case 1. Otherwise, exactly one of them is a peak. If w_l is a peak, then we insert v_{i+1} following either Case 1 or Case 2B depending on whether w_r is a tip of some right or left segment. Similarly, if w_r is a peak, then we insert v_{i+1} following either Case 1 or Case 2B depending on whether w_l is a tip of some left or right segment.

If none of w_l and w_r is a peak, without loss of generality assume that both w_l and w_r are tips of some left segments. In such a scenario we construct a ray r starting from w_r with $\texttt{slope}(w_r, w_{r-1}) - \epsilon$ such that the rays of $Q_r \cup \{r\}$ are

divergent and maintain Invariant 3. Lemma 2 guarantees the existence of such a constant ϵ. Figure 3.5(c) illustrates this scenario. We then place the vertex v_{i+1} at the intersection point of r and the ray in Q_l emanating from w_l. Finally, we draw the l-, r- and m-edges of v_{i+1}. Lemma 1 ensures that the drawing of these edges does not create any edge crossing.

This completes the description of our drawing algorithm.

Γ_{i+1} respects Invariants 1–4: According to our construction, v_{i+1} is a peak in Γ_{i+1} such that $x(w_l) < x(v_{i+1}) < x(w_r)$. Hence the l-edges and r-edges are drawn as left and right edges, respectively. Consequently, Invariants 1–2 hold for Γ_{i+1} in all the three cases. We now consider Invariants 3–4. Since Case 1 does not increase the number of rays, it is straightforward to verify that Γ_{i+1} respects these invariants. On the other hand, Cases 2–3 create new rays. Note that these new rays have been constructed according to Lemma 2, which ensures that for any new ray $r \in Q_l$ (respectively, $r' \in Q_r$), the set $r \cup Q_l$ (respectively, $r' \cup Q_r$) is divergent. Since all the rays have origin on P_{i+1}, it is straightforward to observe that the ray r intersects all the other rays that belong to Q_r and appear after r while visiting P_{i+1} from left to right. The rays emanating from $w_{l+1}, \ldots, w_{r-1}$ in Γ_i disappear in Γ_{i+1}. Hence no ray in Q_l and Q_r in Γ_{i+1} intersects Γ_{i+1} except at its origin.

3.3.3 Computing the Upper Bound

Let $\Gamma = \Gamma_n$ be the drawing of G computed using the above drawing algorithm. Let T_l, T_r, T_m be the Schnyder realizer that corresponds to σ. We now claim that

the drawing has at most $\text{leaf}(T_l) + \text{leaf}(T_r) + n$ segments.

Lemma 3 *Let G be a planar triangulation. Then Algorithm* FEWSEGDRAW *computes a drawing Γ of G with at most $\text{leaf}(T_l) + \text{leaf}(T_r) + n$ segments, where T_l and T_r are a pair of trees in a Schnyder realizer of G.*

Proof: The idea is to show that the drawings of T_l and T_r have at most $\text{leaf}(T_l)$ and $\text{leaf}(T_r)$ segments in Γ, respectively. Since $G \setminus (T_l \cup T_r)$ has n edges, the claim follows.

Let Γ'_i, where $3 \leq i \leq n$, be the drawing obtained from Γ_i by deleting the edges of T_m. While adding v_i, the algorithm adds one edge of T_l (i.e., the l-edge of v_{i+1}), and one edge of T_r (i.e., the r-edge of v_{i+1}) to Γ'_{i-1}. Case 1 does not create any new segment. A new segment in the drawing of T_l and T_r may appear only in Cases 2–3. Whenever the algorithm creates a new segment above P_{i-1}, it ensures that the corresponding vertex w on P_{i-1} is an internal vertex of some left or right segment in Γ'_{i-1}. For example, see Figure 3.5. If $w = w_l$ (respectively, $w = w_r$), then such a new segment eventually ends at a new leaf of T_l (respectively, T_r). Therefore, the drawings of T_l and T_r have exactly $\text{leaf}(T_l)$ and $\text{leaf}(T_r)$ segments, respectively. ■

In the minimum Schnyder realizer T_l, T_r, T_m of G, we have $\text{leaf}(T_l) + \text{leaf}(T_r) + \text{leaf}(T_m) = 2n - 5 - \delta_0$ [14], where $0 \leq \delta_0 \leq \lfloor (n-1)/2 \rfloor$. Note that the tree with the largest number of leaves must have at least $(2n - 5 - \delta_0)/3$ leaves. Hence the remaining two trees have at most $2(2n - 5 - \delta_0)/3 \leq (4n - 2\delta_0 - 10)/3$ leaves. Using Lemma 3 we obtain the following theorem.

Theorem 1 *Let G be an n-vertex planar triangulation. Then G admits a drawing with at most $(7n - 2\delta_0 - 10)/3$ segments.*

3.3.4 Constraints and Generalizations

We can improve the upper bound of $7n/3 - O(1)$ segments for triangulations to $9n/4 - O(1)$ segments under 4-connectivity constraint, as follows.

Zhang and He [121] showed that for 4-connected triangulations, there exists a canonical ordering tree with at most $(n+1)/2$ leaves. For the corresponding Schnyder realizer, we have $\text{leaf}(T_l) + \text{leaf}(T_r) + \text{leaf}(T_m) = 2n - 5 - \delta$ [14], where δ is the number of cyclic faces. Without loss of generality assume that $\text{leaf}(T_l) \leq (n+1)/2$. Then $\text{leaf}(T_r) + \text{leaf}(T_m) \leq 2n - 5 - \text{leaf}(T_l)$. Hence either T_r or T_m has at most $(2n - 5 - \text{leaf}(T_l))/2$ leaves. Without loss of generality assume that $\text{leaf}(T_m) \leq (2n - 5 - \text{leaf}(T_l))/2$. Therefore, $\text{leaf}(T_m) + \text{leaf}(T_l) \leq (2n - 5)/2 - \text{leaf}(T_l)/2 + \text{leaf}(T_l) = (2n-5)/2 + \text{leaf}(T_l)/2$. Since $\text{leaf}(T_l) \leq (n+1)/2$, we have $\text{leaf}(T_m) + \text{leaf}(T_l) \leq (5n-9)/4$. In summary, there exists a Schnyder realizer such that two of its trees have at most $(5n-9)/4$ leaves. Using Lemma 3 we obtain the following theorem.

Theorem 2 *Let G be an n-vertex 4-connected planar triangulation. Then G admits a drawing with at most $(9n-9)/4$ segments.*

It is straightforward to use our algorithm to draw general planar graphs: Given a planar graph G, we first triangulate the graph, then draw the triangulation with $(7n-10)/3$ segments using Theorem 1, and finally remove the added edges. Note that removal of edges may increase the number of segments in the

drawing. Since removal of one edge from any segment of some straight-line drawing can increase the number of segments by at most one, the over all increase in the number of segments is at most the total number of edges removed. Since an n-vertex triangulation has exactly $m = 3n - 6$ edges, the drawing we obtain can have at most $(7n - 10)/3 + (3n - 6 - m) = (16n - 3m - 28)/3$ segments.

Theorem 3 *Let G be a planar graph with n vertices and m edges. Then G admits a straight-line drawing with at most $(16n - 3m - 28)/3$ segments.*

Dujmović et al. [41] gave an algorithm to draw n-vertex m-edge 3-connected planar graphs with at most $\min\{m - n/2 + \alpha - 3, m - \alpha\}$ segments, where the parameter α lies in the interval $[0, 3n - 6 - m]$, giving an upper bound of $2.5n$ segments. Theorem 3 gives a better upper bound when the graph contains at least $m \geq 2.84n$ edges.

3.4 An Algorithm for Grid Drawing

In this section we compute grid drawings of planar triangulations. We first introduce some technical details, and then describe the drawing algorithm.

Let Γ be a straight-line grid drawing with k segments. The following fact states that scaling Γ does not change the number of segments, which is straightforward to verify from the properties of affine transformations.

Fact 1 *Let Γ be a k-segment grid drawing of G. Let Γ' be a drawing of Γ scaled horizontally, i.e., the x-coordinate of every vertex in Γ is multiplied by some positive integer λ.*

Then Γ' is also a k-segment grid drawing of G. Besides, for every segment l in Γ, there is a corresponding segment l' in Γ' that contains all the vertices of l, and vice versa.

In our drawing algorithm, we will use some particular variant of canonical ordering, as introduced in the following lemma.

Lemma 4 (de Fraysseix and de Mendez [31, Lemma 3.5]) *Let T_l, T_r, T_m be a Schnyder realizer of G, where v_1, v_n, v_2 are the outer vertices of G in clockwise order and the roots of T_l, T_m, T_r, respectively. Let $z_1(= v_1), z_2, \ldots, z_{n-2}$ be the vertices listed according to a preorder traversal of T_l, where the children are visited in anticlockwise order. Then $z_1(= v_1), v_2(= z_0), z_2, \ldots, z_{n-2}, z_{n-1}(= v_n)$ is a canonical ordering of G.*

We refer to the canonical ordering of G as described in Lemma 4 as *T_l-ordering*. Figure 3.6(a) illustrates such a T_l-ordering. Let $l_1, l_2, \ldots, l_t$ be the set of leaves that are ordered according to their appearance in the preorder traversal of T_l. We assign *pseudo-segments* to the leaves of T_l that decompose T_l into some paths, as follows. The *pseudo-segment assigned to l_1* is the unique path from z_1 to l_1. The *pseudo-segment assigned to l_j*, where $1 < j \leq t$, is the shortest path in T_l that starts at l_j and ends at a vertex of some previous pseudo-segment. Figure 3.6(b) illustrates such a pseudo-segment decomposition of T_l. We now present an overview of the algorithm, and then describe the algorithmic details.

3.4.1 Algorithm Overview

In the following we present an algorithm that constructs a grid drawing of G, where each pseudo-segment of T_l corresponds to a single segment in the drawing.

Recall that the previous algorithm, i.e., Algorithm FEWSEGDRAW, constructed the drawing by inserting one vertex at a time, and in the final output, T_l and T_r are drawn with leaf(T_l) and leaf(T_r) segments, respectively. Consequently, the total number of segments were leaf(T_l) + leaf(T_r) + n.

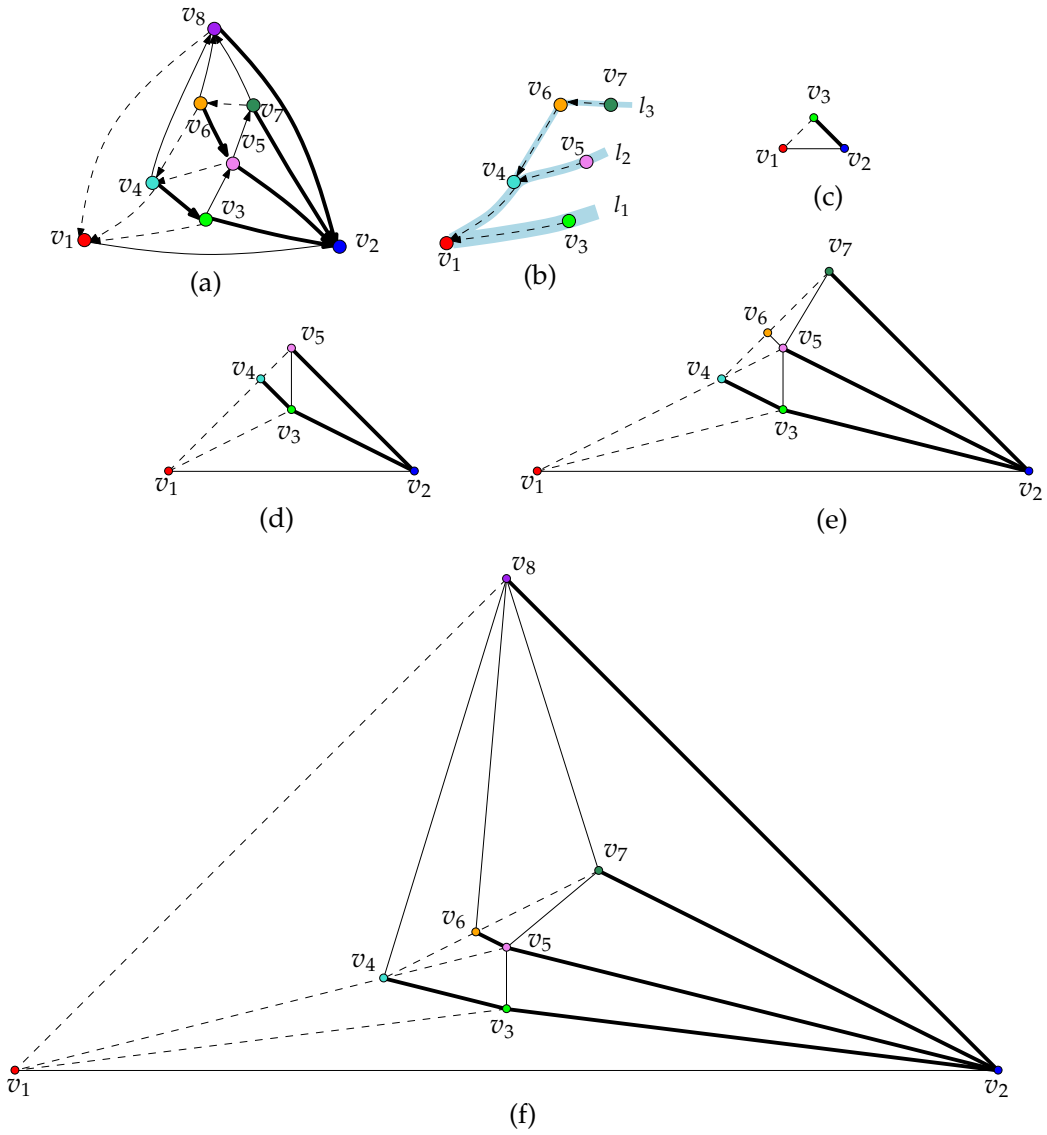


Figure 3.6: (a) A T_l ordering. (b) Pseudo-segment decomposition of T_l. (c)–(e) State of the output after the addition of each pseudo-segment. (f) Final output of the algorithm.

The grid drawing algorithm we present here is also based on an incremental construction; but to keep the area small, we allow the drawing of T_r to have more than leaf(T_r) segments. We only focused on minimizing the segments of T_l. In particular, while adding the vertices, we maintained some drawing invariants so that every pseudo-segment of T_l becomes straight. Figures 3.6(c)–(f) illustrate the state of the output after the drawing of each pseudo-segment, and the final output of the algorithm, respectively.

Since the number of pseudo-segments is equal to leaf(T_l), the final drawing consists of at most leaf$(T_l) + (3n-6) - E(T_l) =$ leaf$(T_l) + (3n-6) - (n-3) =$ leaf$(T_l) + 2n - 3$ segments.

3.4.2 Algorithm Grid-Draw

We now describe the details of the drawing algorithm. The algorithm computes the drawing incrementally by adding the vertices one after another. However, to keep the number of segments small, sometimes we need to scale the drawing. We first draw the edge (z_1, z_0) as a unit horizontal segment, where the coordinates of z_1 and z_0 are $(0,0)$ and $(1,0)$, respectively. We denote this drawing by Γ_0. For each leaf l_i of T_l, let s_i be the pseudo-segment associated with l_i. Let Γ_i, where $1 \leq i \leq t$, be the drawing obtained at step i, where we added the vertices lying on s_i to Γ_{i-1}. For each i, where $0 \leq i \leq t$, we maintain the following drawing invariants.

A. Γ_i is a grid drawing. The path P_i, while walking clockwise from z_1 to z_0 on the boundary of Γ_i, is x-monotone. For each l-edge e on P_i, we have

$\texttt{slope}(e) \in (0, +1)$. Similarly, for every r-edge e', we have $\texttt{slope}(e') \in (-1, 0)$.

B. For every leaf l_i of T_l, the pseudo-segment s_i belongs to Γ_i, and forms a segment in Γ_i.

To prove these invariants, we employ a induction on i. The case $i = 0$ is straightforward to verify. Assume now that the drawings $\Gamma_0, \Gamma_1, \ldots, \Gamma_{i-1}$ respect Invariants A–B. We now add the vertices of pseudo-segment s_i to Γ_{i-1} to construct a drawing Γ_i which, as we now show, also maintains Invariants A–B.

Let $a_i, \ldots, l_i$ be the pseudo-segment s_i. If $i > 1$, then by definition of pseudo-segment decomposition, the vertex a_i belongs to some segment s_j, where $1 \leq j < i$, and thus belongs to Γ_{i-1}. Otherwise, $i = 1$, and in this case $a_i = z_1$, which belongs to Γ_0. Consequently, for each i, a_i belongs to Γ_i.

Starting at a_i, we first construct an upward ray Q with slope $+1$. For each vertex $v \in (s_i \setminus \{a_i\})$, we denote the parent of v in T_r by $P_r(v)$, and construct an upward ray Q_v with slope -1 that starts at $P_r(v)$. Since the slope of each l- and r-edge of Γ_{i-1} is in $(0, +1)$ and $(-1, 0)$, respectively, none of the above rays intersects Γ_{i-1} except at its starting vertex. We now place the vertex v at the intersection point of Q and Q_v, and draw the outgoing l-edge and the outgoing r-edge incident to v. Let the drawing be Γ'_{i-1}. Since the intersection points of the rays do not necessarily lie on grid points, Γ'_{i-1} may not be a grid drawing. Furthermore, we show that Γ'_{i-1} may contain overlapping edges. Observe that the vertices $P_r(v)$ are not necessarily disjoint, i.e., there may exist two or more vertices on s_i with same parent on T_r. However, all these vertices must be consecutive on

s_i, otherwise, let $w, x, \ldots, y$ be a subpath of s_i with $P_r(w) = P_r(y) (\neq P_r(x))$. By the property of canonical ordering, $P_r(x)$ would lie after $P_r(w)$ on P_{i-1}. Consequently, the edge $(y, P_r(y))$ would intersect the edge $(x, P_r(x))$, as illustrated in Figure 3.7, which contradicts that T_l-ordering is a canonical ordering.

Since two or more vertices of s_i may have the same parent on T_r, the rays corresponding to these vertices overlap in Γ'_{i-1}, and the vertices lie at a single point in Γ'_{i-1}. Figure 3.8(a) illustrates such an example.

Among the vertices that with the same parent on T_r, only the vertex that appears first in the T_l-order may have some incoming m-edges. Otherwise, let $x, \ldots, x', y', \ldots, y$ be a maximal set of consecutive vertices in the T_l order that have the same parent in T_r, and suppose for a contradiction that y' is incident to some incoming m-edges. Then all other end points of these m-edges must appear before x' in the T_l order, and hence $P_r(x') \neq P_r(y')$, a contradiction.

Hence for each vertex on s_i, we now can draw the incoming m-edges, and by Lemma 1, this does not create any edge crossing. We now transform Γ'_{i-1} to a grid drawing and remove degeneracies. To transform Γ'_{i-1} to a grid drawing, we scale Γ'_{i-1} with a factor of 2 both horizontally and vertically, as illustrated in

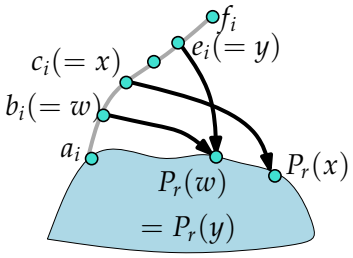


Figure 3.7: The vertices of s_i that have the same parent on T_r must be consecutive on s_i, here $s_i = (a_i, \ldots, f_i)$.

Figures 3.8(a)–(b). Since the slope of Q is $+1$ and the slope of each Q_v is -1, such a scaling will ensure integer grid points for all the vertices of Γ'_{i-1}. Let κ be the largest number of vertices that have the same parent in T_r. Then at most κ vertices may lie on the same point on Q. To remove the degeneracies, we scale Γ'_{i-1} with a factor of κ both horizontally and vertically. This scaling ensures at least $\kappa - 1$ different integer grid points on Q between each pair of consecutive intersection points. Therefore, we can distribute the vertices lying at the same grid point (x, y) into the grid points $\{(x, y), (x + 1, y + 1), \ldots, (x + \kappa - 1, y + \kappa - 1)\}$ according to the T_l-ordering. Let the resulting drawing be Γ''_{i-1}. Figure 3.8(c) depicts Γ''_{i-1}, where $\kappa = 3$.

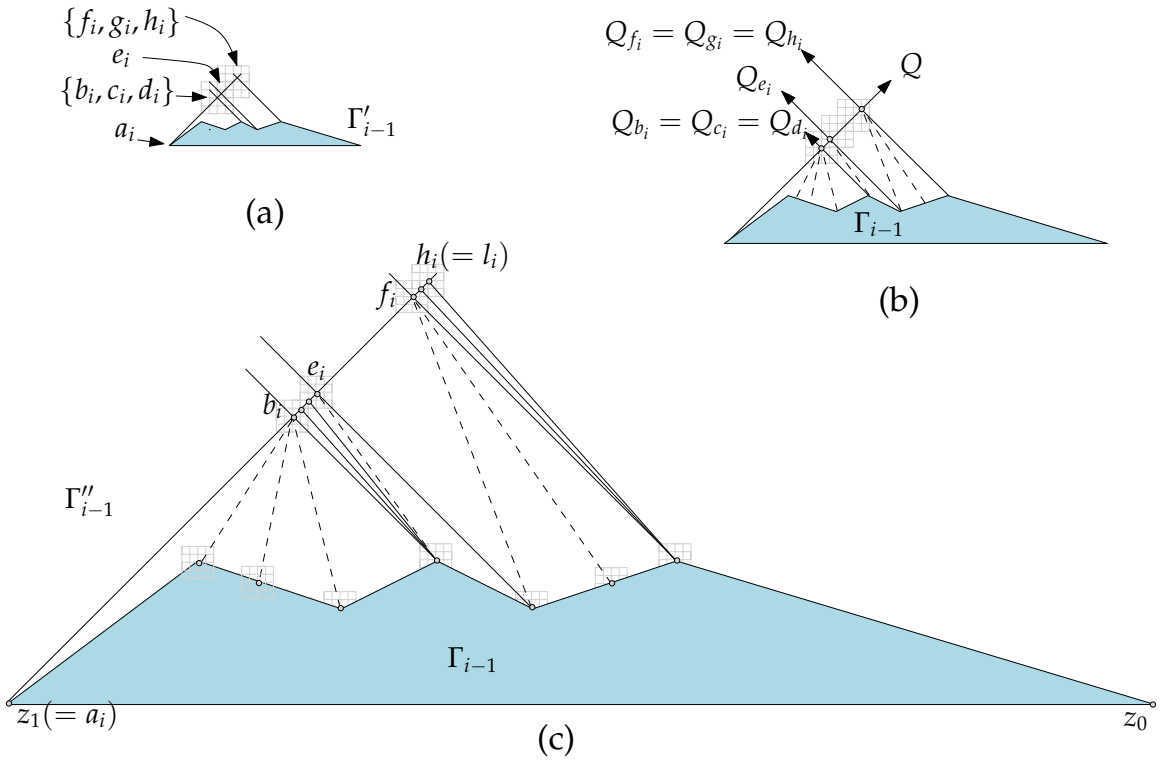


Figure 3.8: (a) Illustration for the degeneracies. (b)–(c) Transforming Γ'_{i-1} to a grid drawing, and the drawing Γ''_{i-1} computed after removing the degeneracies.

Note that Γ''_{i-1} respects Invariant B. However, since the l-edges on s_i have slope $+1$ and some r-edges incident to s_i have slope smaller than or equal to -1, Γ''_{i-1} violates Invariant A. Hence we again scale Γ''_{i-1} with a factor of 2 horizontally. The resulting drawing Γ_i satisfies Invariant A, and by Fact 1, Γ_i also respects Invariant B.

The above technique gives us a grid drawing Γ_t. We now complete the drawing of G by adding the vertex z_{n-1} to Γ_t. Let Q_l be the upward ray with slope $+1$ that starts at the vertex z_1. Similarly, let Q_r be the upward ray with slope -1 that starts at the vertex z_0. Since the slopes of the l-edges and r-edges of P_t are strictly smaller than $+1$ and greater than -1, respectively, none of Q_l and Q_r intersects Γ_t except at its starting vertex. Let c be the point of intersection of Q_l and Q_r. If c is not a grid point, then we scale the drawing with a factor of 2 both horizontally and vertically, which does not increase the number of segments in Γ_t. Finally, we place the vertex w at c and draw its incident edges, which do not create any edge crossing by Lemma 1.

Theorem 4 *Let G be an n-vertex planar triangulation and let T_l, T_r, T_m be the minimum realizer of G. Then Algorithm* GRID-DRAW *constructs a grid drawing of G with at most $(t + 2n - 3)$ segments and on an $O((5.36n - 4t)^t) \times O((2.68n - 2t)^t)$ grid, where $t = \text{leaf}(T_l)$.*

Proof: Recall that each time we add a pseudo-segment, we scale the previous drawing with a factor of at most 4κ horizontally, and with a factor of at most 2κ vertically, where κ is bounded by the maximum number of leaves in T_r.

Since T_l, T_r, T_m is the minimum realizer, we have $\text{leaf}(T_l) + \text{leaf}(T_r) + \text{leaf}(T_m) = 2n - 5 - \delta_0$ [14]. Without loss of generality we can assume that $\text{leaf}(T_l) \leq \text{leaf}(T_r)$ and $\text{leaf}(T_l) + \text{leaf}(T_r) \leq 4n/3$. Therefore, the value of κ is bounded by $(4n/3 - leaf(T_l)) \leq (1.34n - t)$. Since there are t pseudo-segments, the size of the drawing is $O((5.36n - 4t)^t) \times O((2.68n - 2t)^t)$ grid.

Since every pseudo-segment forms a segment in the final drawing, the number of segments is bounded by $t + (3n - 6) - (n - 3) = (t + 2n - 3)$. ■

For an n-vertex planar triangulation with maximum degree Δ, the value of κ is bounded by Δ. Hence we obtain the following corollary.

Corollary 1 *Every n-vertex planar triangulation with maximum degree Δ admits a drawing with at most $(2n + t - 3)$ segments and $O(8^t \cdot \Delta^{2t})$ area, where t is the minimum number of leaves over all the trees of the minimum realizer.*

3.5 Trade-off Between Segment and Area

The best known upper bound on the area of straight-line planar drawings is $8n^2/9$ [17]. Therefore, every n-vertex planar triangulation admits a drawing with $3n - 6$ segments and $8n^2/9$ area. Corollary 1 states that the number of segments can be improved to $(2n + t - 3)$ if we allow the area to be $O(8^t \cdot \Delta^{2t})$. Since the parameters n and t are fixed for a given planar graph, these algorithms do not provide any control for changing the area smoothly from $O(8^t \cdot \Delta^{2t})$ to $8n^2/9$, which may gradually increase the number of segments from $(2n + t - 3)$ to $3n - 6$.

However, we can find a trade-off among the area and the number of segments

by slightly modifying the Algorithm GRID-DRAW, as follows. Instead of stretching all the segments of T_l, we can stretch the largest β pseudo-segments among the t pseudo-segments of T_l. For adding each of these β pseudo-segments, we scale the previous drawing by a factor of at most 4κ horizontally and 2κ vertically.

For each vertex v that does not lie on these β pseudo-segments, we add v at the intersection point of the two rays with slopes $+1$ and -1 that start at $P_l(v)$ and $P_r(v)$, respectively. We may need to scale the drawing by a factor of two (both horizontally and vertically) to ensure integer coordinates. We then again scale the drawing with a factor of 2 horizontally to maintain the slope invariants. Let β' be the number of vertices added in this way. Then the area of the drawing becomes $O(8^{\beta} \cdot \Delta^{2\beta} \cdot 16^{\beta'})$. On the other hand, the number of segments is at most $\beta + \beta' + 2n$.

3.6 Summary and Open Questions

In this chapter we have given an algorithm to draw any n-vertex planar triangulation with at most $7n/3$ segments, which improves to $9n/4$ when the input triangulation is 4-connected. Since the realizers we use can be computed in linear time [121], our algorithm runs in linear time.

In the fixed embedding setting, the lower bound on the number of segments to draw planar triangulations is $2n - 2$, as shown in Figure 3.9(a). Similar lower bound holds also in the variable embedding setting, e.g., Dujmović et al. [41] showed that there exist planar triangulations requiring $2n - 6$ segments for every possible choice of the outer face. A natural open question is to reduce the gap

between the lower and upper bounds.

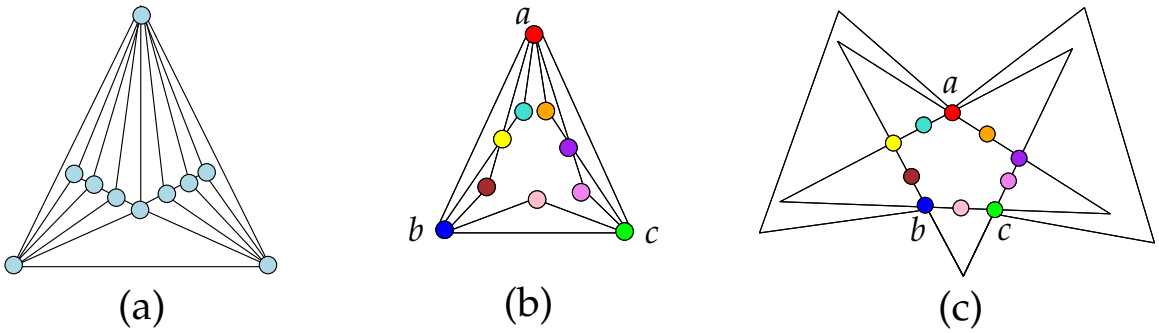


Figure 3.9: (a) Illustration for lower bounds for general planar triangulations. (b) A straight-line drawing with 13 segments. (c) A polyline drawing with 11 segments.

Open Problem 3.1. Does every planar triangulation admit a straight-line drawing with $2n + O(1)$ segments?

In addition to computing drawings with few segments, it would be interesting to further explore the area requirements of these drawings. An intriguing research direction would be to examine the feasibility of computing straight-line drawings with few segments and in polynomial area.

Open Problem 3.2. Does there exist a constant $c \in [2, 3)$ such that every n-vertex triangulation can be drawn with at most $cn + O(1)$ segments on a polynomial size grid?

In this work we attempted to characterize a trade-off between the area and number of segments in straight-line drawings. It would be interesting to examine the scenario when we allow the edges to have bends. Figure 3.9(b) illustrates a graph G such that any straight-line drawing of G with a, b, c as the outer face

would require at least 13 segments, whereas there exists a polyline drawing with only 11 segments, as shown in Figure 3.9(c).

Open Problem 3.3. Does a polyline drawing allow us to achieve a better upper bound on the number of segments while drawing planar triangulations?

There exist some interesting lower bounds on the slope number of a graph in terms of its maximum degree. Dujmović et al. [41] asked whether the slope number of a graph is bounded by its maximum degree. Although the answer is affirmative for planar graphs [80], there exist graphs with maximum degree 5 whose slope number is at least $n^{1/6-o(1)}$ [101]. Since the slope number of a graph is a lower bound on its segment number, the segment number of bounded degree graphs can also be unbounded. A natural question is whether we can prove similar negative results in the presence of edge bends.

Chapter 4

Compact Polyline Drawings of Planar Graphs

This chapter concentrates on computing polyline drawings of planar graphs with small area and few bends. Minimum-segment planar straight-line drawings may sometimes require doubly exponential area [55]. The area upper bound may be reduced using a higher number of segments, e.g., every n-vertex triangulation admits a straight-line drawing with $8n^2/9$ area [17], where the number of segments could be as large as $3n - 6$. Allowing bend complexity one, i.e., when every edge is either straight or has one bend, further reduces the area to $(4/9)n^2$ [14, 120].

In the fixed embedding setting, where the output drawing is restricted to have the same combinatorial embedding as the one given in the input, the bound of $(4/9)n^2$ on area is tight, even when the input is restricted to graphs with $\Delta \in O(1)$. Observe that the $(4/9)n^2$-area upper bound in the fixed embedding setting

applies also to the variable embedding setting, where the output combinatorial embedding can be chosen freely. The best known lower bound in the variable embedding setting is $n^2/9 + \Omega(n)$ [61].

In this chapter we improve the area upper bound when the maximum degree $\Delta \in o(n)$ by allowing higher bend complexity. Specifically, we prove that in the variable embedding setting, every planar graph with n vertices and maximum degree $o(n)$ admits a polyline drawing with bend complexity two and at most $(3/8)n^2 + o(n^2)$ area.

We first present the motivation behind minimizing area and bend complexity while drawing planar graphs. We then discuss our contribution in the context of related research. The subsequent sections present the technical background (Section 4.2), the drawing algorithm (Sections 4.3), limitations of our approach and directions to future research (Section 4.4). For simplicity we often omit the floor and ceiling functions while defining width, height or area parameters, but this has no effect on the asymptotic results discussed in this chapter.

Some of the results of this chapter appeared in preliminary form at the 22nd International Symposium on Graph Drawing [51].

4.1 Minimizing Area and Bends

Drawing planar graphs on a small integer grid is an active research area in graph drawing [32, 108, 14, 61], which is motivated by the need for compact layout in VLSI circuits and visualization of software architecture. In visualization applications, the constraint on area is imposed naturally by the size of the display screen.

For VLSI circuit layout, compact drawings reduce the microchip area. Minimizing area often requires the edges to have bends. However, allowing bends may decrease the readability of visualizations, and reliability of circuits.

Straight-line planar drawings with the best known area upper bound have bend complexity one. However, these drawings maintain the input combinatorial embedding. Hence it is natural to ask whether one can improve the area bound allowing the output embedding to be freely chosen, which motivated us to examine compact polyline drawings in the variable embedding setting.

4.1.1 Related Work

We first review results in the fixed embedding setting, and then results in the variable embedding setting.

Fixed Embedding Setting: In 1990, de Fraysseix et al. [32] and Schnyder [108] independently proved that every n-vertex planar graph can be drawn with at most $2n^2$ and n^2 area, respectively. Later, Brandenburg [17] improved the area bound to $8n^2/9$. The best known lower bound on area in the fixed embedding setting is $(4/9)n^2$ [39, 93].

Allowing higher bend complexity reduces the area upper bound. For example, Bonichon et al. [14] gave an algorithm to draw every planar graph with $(4/9)n^2$ area with bend complexity one and at most $(n-2)$ bends in total. Zhang [120] improved the bound on the number of total bends to $2n/3$. The bound of $(4/9)n^2$ area is tight for polyline drawings, even when $\Delta \in O(1)$, which is determined by a class of graphs called nested triangles graphs. Specifically, an n-vertex *nested*

triangles graph is a plane graph formed by a sequence of $n/3$ vertex disjoint cycles, $\mathtt{C}_1, \mathtt{C}_2, \ldots, \mathtt{C}_{n/3}$, where for each $i \in \{2, \ldots, n/3\}$, cycle $\mathtt{C}_i$ contains the cycles $\mathtt{C}_1, \ldots, \mathtt{C}_{i-1}$ in its interior, and a set of edges that connect each vertex of $\mathtt{C}_i$ to a distinct vertex in $\mathtt{C}_{i-1}$.

Variable Embedding Setting: Observe that the upper bound of $(4/9)n^2$ area [14, 120] with bend complexity one applies also to the variable embedding setting, and this is the best known upper bound even if the input is restricted to graphs with $\Delta \in O(1)$. Like the fixed embedding setting, the best known lower bound in the variable embedding setting is determined by nested planar graphs. Frati and Patrignani [61] showed that every n-vertex nested triangles graph requires at least $(1/9)n^2$ area.

4.1.2 Our Contribution

We assume that the input is an arbitrary planar graph, and the output combinatorial embedding may differ from the input embedding. Table 4.1 lists our results, as well as summarizes the best known upper bounds on area and bend complexity.

4.2 Technical Foundation

In this section we review the polyline drawing algorithm of Bonichon et al. [14], which will be useful to describe our algorithms.

Let G be an n-vertex planar triangulation and let T be a Schnyder tree of

Graph Class	Area	Bend Complexity	Total Bends	Embedding	Ref.
Planar graphs	$8n^2/9$	0	0	fixed	[17]
Planar graphs	$4n^2/9$	1	$n-2$	fixed	[14]
Planar graphs	$4n^2/9$	1	$2n/3$	fixed	[120]
Our Results					
Planar, $\Delta \in o(n)$	$3n^2/8 + o(n^2)$	2	$3n-4$	variable	Theorem 5

Table 4.1: Results on minimizing area and bends.

G with p leaves. Bonichon et al. [14] gave an algorithm to compute a polyline drawing Γ of G that satisfies the following properties.

(B_1) Γ is a polyline drawing with bend complexity 1 and at most $(n-2)$ bends in total.

(B_2) The width and height of Γ are at most $p+1$ and $n - \lfloor p/2 \rfloor - 1$, respectively.

(B_3) Let r be the root of T and let $N(r)$ be the set of neighbors of r. Then the outer edge in Γ, which is not incident to r, is drawn as a straight line segment.

(B_4) Let Γ' be the drawing obtained from Γ by deleting r and its incident edges. Then the vertices of $N(r)$ have distinct x-coordinates in Γ', and for each vertex $v \in N(r)$, the upward ray starting at v does not intersect Γ' except at v.

Figure 4.1(a) presents a drawing computed using the algorithm of Bonichon et al. [14], and Figure 4.1(b) depicts a schematic representation of Γ'. Observe that the y-coordinates of the bottommost vertices differ by 1, but the schematic representation places them along the same horizontal line. Only the x-coordinates of $N(r)$

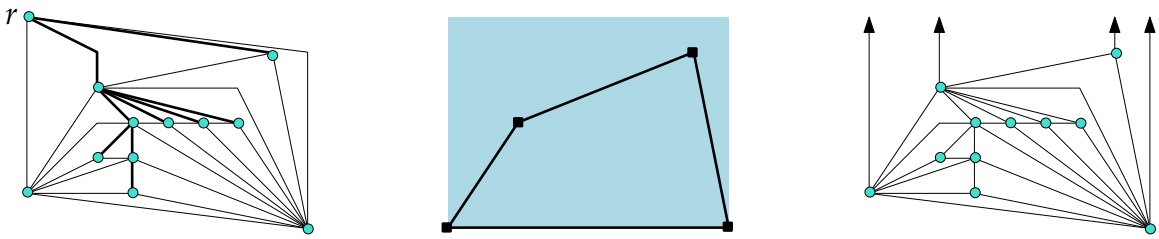


Figure 4.1: (a) Γ, where T is shown in bold. (b) A schematic representation of Γ'. (c) Illustration for Property (B_4).

will be important in our algorithm, therefore, we ignore such finer details in the schematic representation. Figure 4.1(c) illustrates Property (B_4).

4.3 Drawing Triangulations with Small Height

We first present an overview of our algorithm, and then describe the algorithmic details.

4.3.1 Algorithm Overview

Let $G = (V, E)$ be an n-vertex planar graph, where $n \geq 9$, and let Γ be a planar drawing of G on the Euclidean plane. Without loss of generality assume that G is a planar triangulation. Let $M \subseteq E$ be an edge separator of G such that the corresponding edges in the dual graph G^* form a simple cycle C^*. Let $V_o \subseteq V$ (respectively, $V_i \subseteq V$) be the vertices that lie outside (respectively, inside) of C^*. Diks et al. [36] proved that there always exists such an edge separator M such that $M \leq 2\sqrt{2\Delta n}$ and $\max\{|V_i|, |V_o|\} \leq 2n/3$. Figures 4.2(a)–(b) illustrate a planar triangulation G and an edge separator of G.

Let $G_i = (V_i, E_i)$ and $G_o = (V_o, E_o)$ be the subgraphs of G induced by the vertices of V_i and V_o, respectively. Since $n \geq 9$, each of G_i and G_o contains at least 3 vertices. Therefore, it is straightforward to find an embedding of G, where none of the edges of M are outer edges, as shown in Figure 4.2(b).

Since G is a planar triangulation, there must be an outer vertex q on G_i or G_o such that q is incident to two or more edges of M. Without loss of generality assume that q lies on G_i, e.g., see vertex v_5 in Figure 4.2(c). Let a, b, c be three consecutive neighbors of q in G in counter clockwise order such that $a \in V_i$ and $\{b, c\} \subseteq V_o$. We take an embedding G' of G with q, b, c as the outer face, as shown in Figure 4.2(d) with $q = v_5$, $a = v_3$, $b = v_2$, and $c = v_{11}$. Consequently, G_o and G_i lie on the outer face of each other, as illustrated in Figures 4.2(d)–(e).

We first draw G_o and G_i separately with small area, and then merge these drawings to compute the final output. The drawings of G_o and G_i are placed side by side. Consequently, the height of the final output can be expressed in terms of the maximum height of the drawings of G_o and G_i, and hence the area of the final drawing becomes small.

4.3.2 Algorithm Details

Let G' be the embedding obtained from G by choosing q, b, c as the outer face. We first construct a graph G'_o from G_o by adding a vertex w_o on the outer face of G_o, and making w_o adjacent to all the outer vertices of G_o such that the edge (b, c) remains as an outer edge. We remove any resulting multi-edges by adding dummy vertices to the corresponding inner edges and then by triangulating the

resulting graph. Note that we do not need to add dummy vertices on the outer edges. Figure 4.3(a) illustrates an example of G'_o, where d is a dummy vertex. Since there are $O(\sqrt{\Delta n})$ edges in M, the number of vertices in G'_o is at most $2n/3 + O(\sqrt{\Delta n})$.

We now construct a graph G'_i from G_i, as follows. Observe that the vertex a is an outer vertex of G_i, which appears immediately after q while walking on the outer face of G_i. We add a vertex w_d on the outer face of G_i, and make it adjacent to q and a. We now add another vertex w_i on the outer face, and make it adjacent to w_d and q such that the cycle w_i, q, w_d becomes the boundary of the outer face, e.g., see Figure 4.3(b).

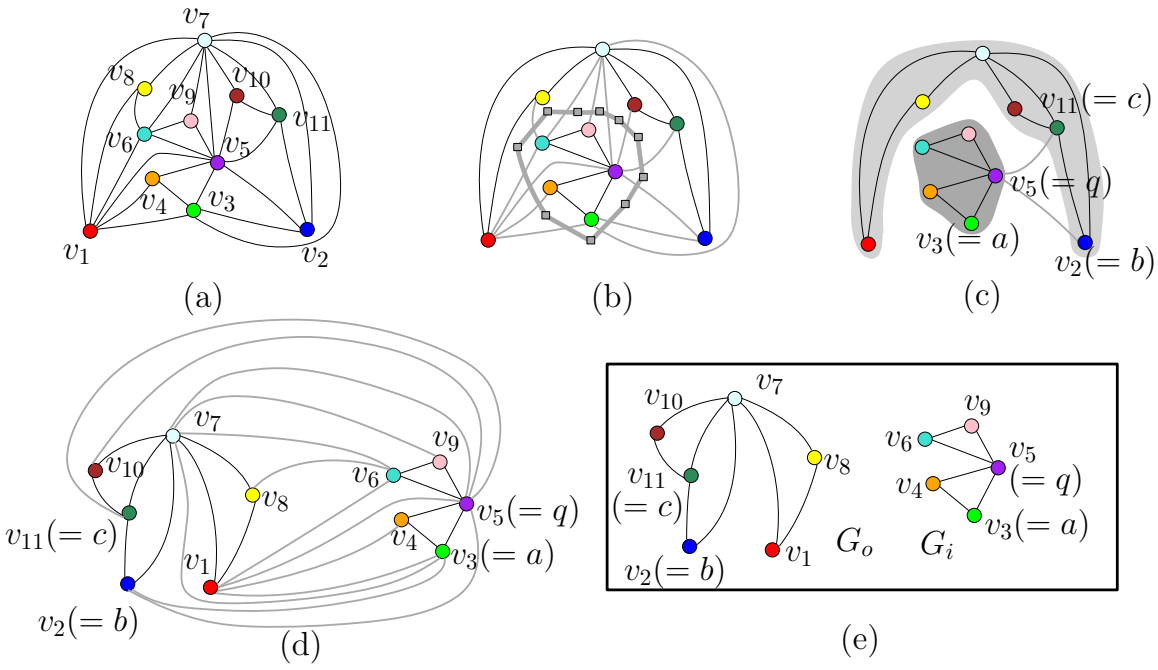


Figure 4.2: (a) A planar triangulation. (b) An edge separator M of G, and the corresponding simple cycle in the dual graph. The edges of M and C^* are shown in thin and thick gray, respectively. (c) G_o and G_i are shaded in light-gray and dark-gray, respectively. (d)–(e) Choosing a suitable embedding G'.

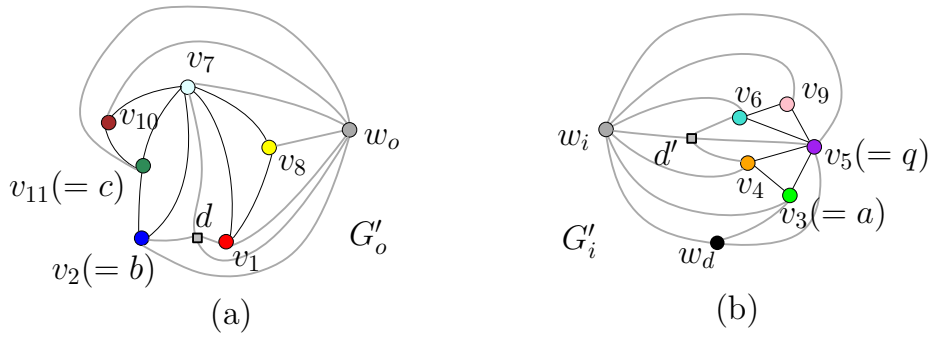


Figure 4.3: Construction of (a) G'_o and (b) G'_i.

To construct a drawing of G', we first draw G'_o and G'_i separately, and then merge these drawings. Let T_{w_i} (respectively, T_{w_o}) be the Schnyder tree rooted at w_i in G'_i (respectively, w_o in G'_o). Let L_i and L_o be the number of leaves in T_{w_i} and T_{w_o}, respectively, i.e., $L_i = \text{leaf}(T_{w_i})$ and $L_o = \text{leaf}(T_{w_o})$.

Assume that the maximum degree $\Delta \in o(n)$, and hence we have $|M| \leq 2\sqrt{2\Delta n} \in o(n)$. We draw both G'_o and G'_i using the algorithm of Bonichon et al. [14]. Let Γ'_o and Γ'_i be the drawings of G'_o and G'_i, respectively, and let Γ_o and Γ_i be the drawings obtained by deleting w_o and w_i, respectively. Without loss of generality assume that $|V_o| \geq |V_i|$. We now merge these drawings considering the following two cases.

Case 1 ($L_o \geq n/4$ and $L_i \leq n/4$): In this case we place Γ_o and Γ_i side by side with bottom aligned along the x-axis and two empty columns in between, as shown in Figure 4.4(a). By Property (B_2), the width W and height H of the combined drawing are as follows.

$$W = L_o + L_i + 4. \tag{4.1}$$

$$H = \max\left\{|V_o| - \frac{L_o}{2}, |V_i| - \frac{L_i}{2}\right\} \leq \max\left\{\frac{2n}{3} - \frac{L_o}{2}, \frac{n}{2} - \frac{L_i}{2}\right\} + o(n). \tag{4.2}$$

We now draw the edges of the edge separator M. Let $N(w_i)$ and $N(w_o)$ be the neighbors of w_i and w_o. For each vertex $v \in N(w_o)$ in the order of decreasing x-coordinates, we draw the edges of M incident to v by taking consecutive rows above the current drawing. Specifically, for the jth neighbor u_j of v on the outer face of G_i in clockwise order, we draw a polyline $(x(v), y(v)), (x(v)+1, H_c+1), (x(u_j)-1, H_c+1), (x(u_j), y(u_j))$, where H_c denotes the current height of the drawing. Figures 4.4(b)–(c) illustrate the drawing of the edges in M.

Using Property (B_4) we can observe that each new edge is added above the current drawing without introducing any edge crossing. Hence the final output is planar. Since we increase the height of the drawing by M, the area of the final drawing is at most

$$(L_o + L_i + 4) \times \left(|M| + \max\left\{\frac{2n}{3} - \frac{L_o}{2}, \frac{n}{2} - \frac{L_i}{2}\right\}\right) + o(n^2). \tag{4.3}$$

Case 2 (Otherwise): In this case we rotate Γ_o and Γ_i by clockwise 90° and counter clockwise 90°, respectively. We then place these rotated drawings side by side with bottom aligned along the x-axis and $|M|$ empty columns in between, as shown in Figure 4.5(a). By Property (B_2), the width W and height H of the com-

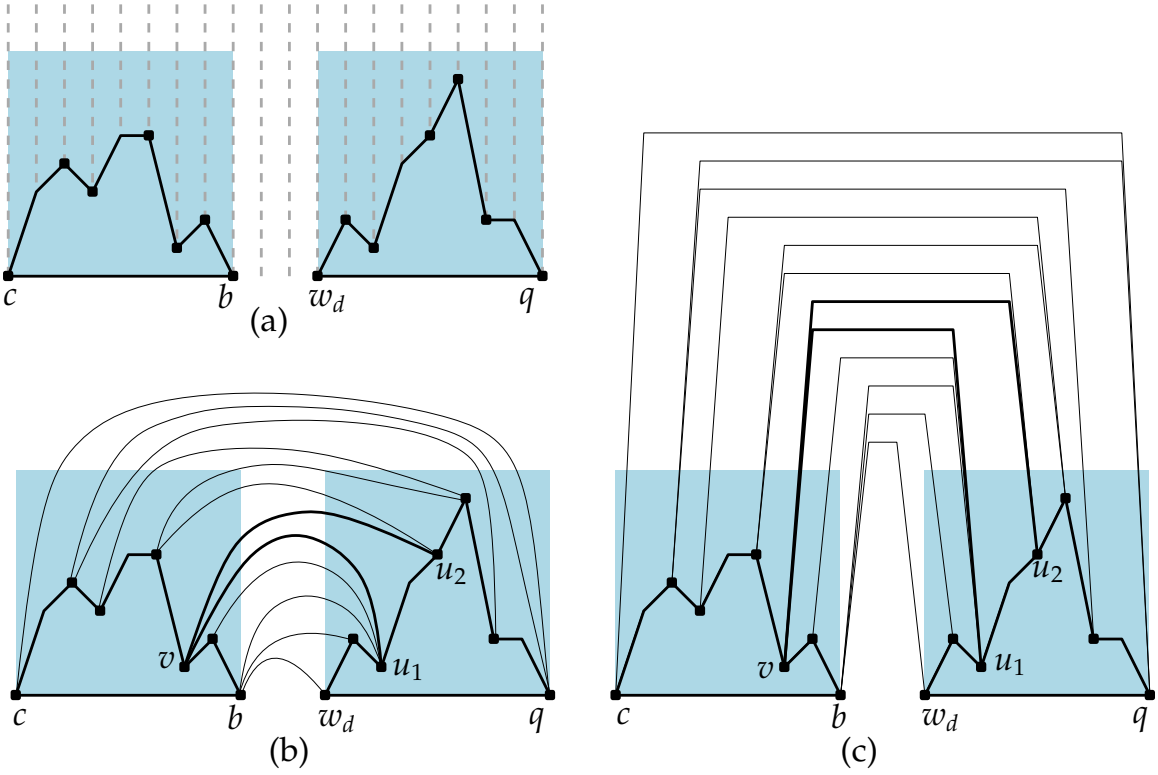


Figure 4.4: (a)–(c) Illustration for Case 1.

bined drawing are as follows.

$$W = |V_o| - \frac{L_o}{2} + |M| + |V_i| - \frac{L_i}{2} \tag{4.4}$$

$$\leq n - \frac{L_o + L_i}{2} + o(n). \tag{4.5}$$

$$H = \max\{L_o + 1, L_i + 1\}. \tag{4.6}$$

We now draw the edges of the edge separator M, as shown in Figures 4.5(b)–(c). Let $N(w_i)$ and $N(w_o)$ be the neighbors of w_i and w_o. For each vertex $v \in N(w_o)$ in the order of increasing y-coordinates, we draw the edges of M incident to v by taking consecutive columns from the empty space between Γ_o and Γ_i, as

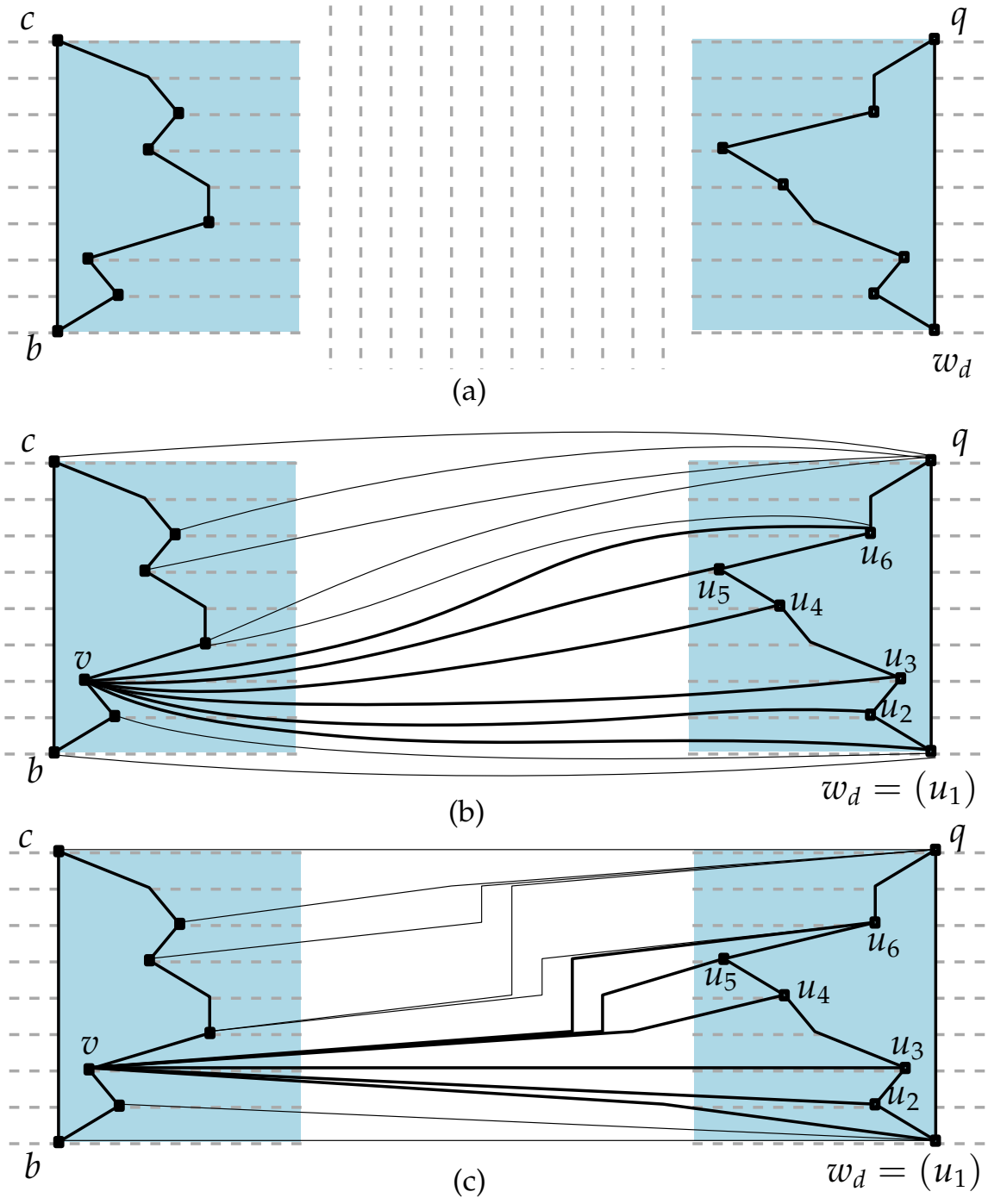


Figure 4.5: (a)–(c) Illustration for Case 2.

follows. Let R be the set of empty columns between Γ_o and Γ_i, and let u_j be the jth neighbor of v on the outer face of G_i in clockwise order.

- If $|y(v) - y(v_j)| \leq 1$, then we draw the edge (v, u_j) using a single straight line segment, e.g., the edges (v, u_2) and (v, u_3) in Figure 4.5(c).

- Otherwise, if $y(v) < y(v_j)$, then we draw (v, u_j) using a polyline $(x(v), y(v))$, $(\ell, y(v) - 1)$, $(\ell, y(u_j) + 1)$, $(x(u_j), y(u_j))$, where ℓ is the rightmost column of R that has not yet been used to place bends. For example, see the edges $(v, u_4), (v, u_5)$ and (v, u_6) in Figure 4.5(c). If $y(v) > y(v_j)$, then we draw (v, u_j) symmetrically.

We can use property (B_4) to observe that each new edge is added above the current drawing without introducing any edge crossing. Hence the final output is planar. Observe that the addition of the edges of M does not increase the width of the drawing. Therefore, the area of the final drawing is at most

$$(n - \frac{L_o + L_i}{2}) \times \max\{L_o + 1, L_i + 1\} + o(n^2). \tag{4.7}$$

Area Computation for Graphs with $\Delta \in o(n)$: If the maximum degree $\Delta \in o(n)$, then we have $|M| \leq 2\sqrt{2\Delta n} \in o(n)$. Therefore, the area upper bound of Case 1 (Equation (4.3)) can be expressed as

$$(L_o + L_i) \times \left(\max \left\{ \frac{2n}{3} - \frac{L_o}{2}, \frac{n}{2} - \frac{L_i}{2} \right\} \right) + o(n^2),$$

which is at most $0.35n^2 + o(n^2)$. Figure 4.6(a) illustrates the corresponding plot. In all other scenarios, the area upper bound is computed from Case 2 (Equa-

tion (4.7)), which is at most

$$(n - \frac{L_o + L_i}{2}) \times \max\{L_o, L_i\} + o(n^2).$$

The maximum value attained in this case is $0.375n^2 + o(n^2)$, as illustrated in Figures 4.6(b)–(d).

By Property (B_1), each edge of G_o and G_i is drawn using one bend per edge. The edges of $M \setminus (q, b)$ are drawn using at most two bends per edge. The edge (q, b) passes through a dummy vertex w_d. By Property (B_3), the edge (q, w_d) is drawn using a straight line segment. Therefore, the polyline corresponding to edge (q, b) may have at most three bends. In such a scenario, since b and w_d are two bottommost vertices in the drawing, it is straightforward to reduce a bend on the edge (q, b) without increasing the area. Hence we obtain a drawing with bend complexity two.

By Property (B_1), the total number of bends in Γ_o and Γ_i are at most $|V_i| - 2$ and $|V_o| - 2$, respectively. Since an n-vertex outerplanar graph can have at most $2n - 3$ edges, $|M| \leq 3n - 6 - (2|V_i| - 3) - (2|V_o| - 3) = n$. Therefore, the total number of bends in our drawing is at most $2|M| + |V_i| - 2 + |V_o| - 2 = 3n - 4$.

The following theorem summarizes the result of this section.

Theorem 5 *Every planar graph with n vertices and maximum degree $o(n)$ admits a polyline drawing with at most $3n^2/8 + o(n^2)$ area, bend complexity two, and $3n - 4$ bends in total.*

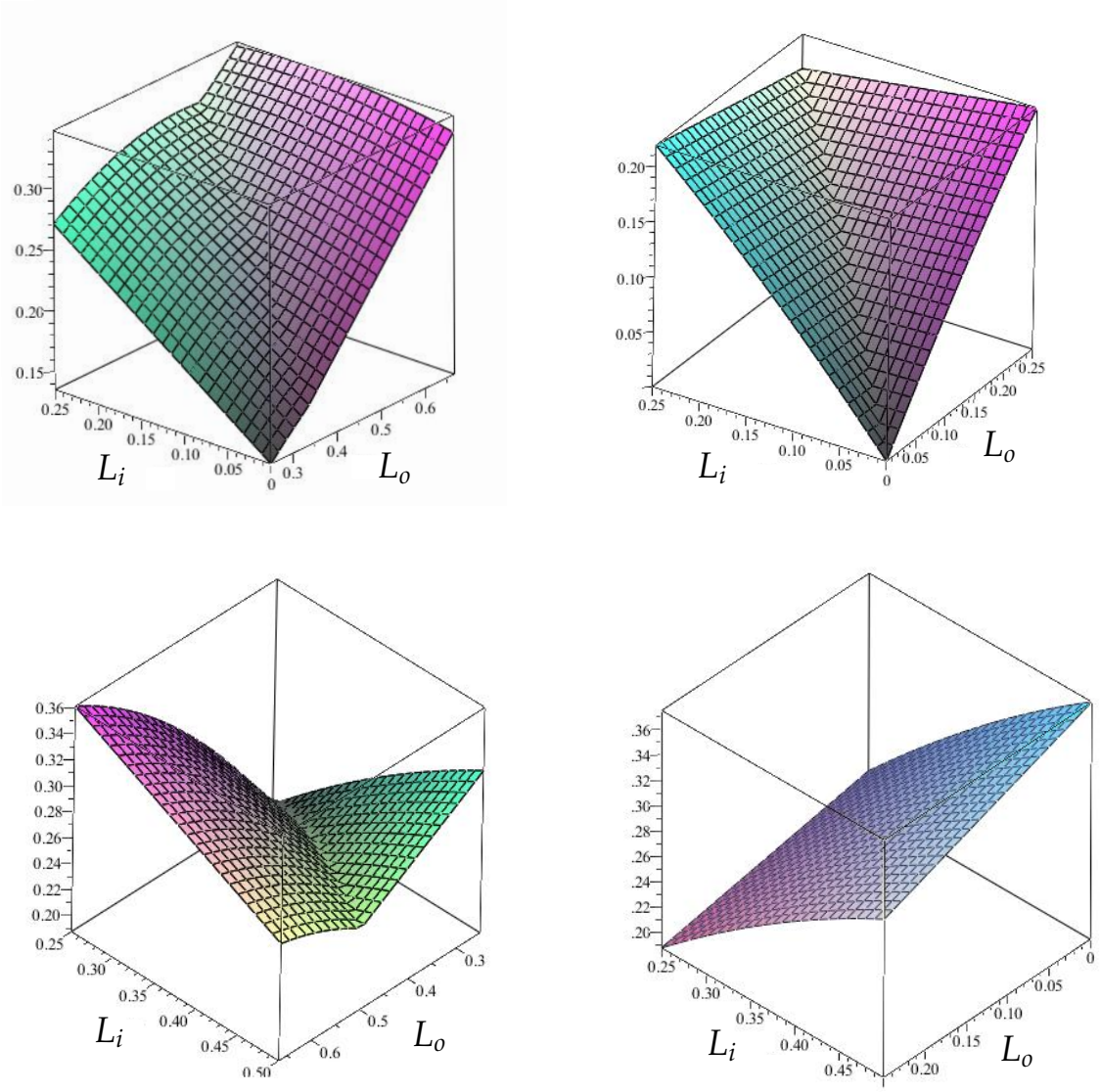


Figure 4.6: Maple [92] plot for the coefficient of n^2 while varying L_o and L_i under the constraint that $L_o \leq |V_o| \leq 2n/3$ and $L_i \leq |V_i| \leq n/2$. (a) $L_o \geq n/4$ and $L_i \leq n/4$, area coefficient is at most 0.348. (b) $L_o \leq n/4$ and $L_i \leq n/4$, area coefficient is at most 0.219. (c) $L_o \geq n/4$ and $L_i \geq n/4$, area coefficient is at most 0.362. (d) $L_o \leq n/4$ and $L_i \geq n/4$, area coefficient is at most 0.375.

4.4 Summary and Open Questions

In this chapter we have shown that every n-vertex planar graph with maximum degree $o(n)$, admits a polyline drawing with $(3/8)n^2 + o(n^2)$ area and bend com-

plexity 2. Our result is an initial step towards compact drawings of planar triangulations via choosing a suitable embedding of the input graph. Note that $(4/9)n^2$ is still the best known upper bound on the area of arbitrary planar graphs [14].

We envision two directions to improve the area bound for arbitrary planar graphs based on our drawing technique. The first is to attempt a more careful adaptation of Bonichon et al.'s algorithm [14] along with an involved area analysis while merging the smaller drawings. The second is to use stronger edge separators. Recall that in our algorithm we use edge separators of size $2\sqrt{2\Delta n}$ that partitions the vertices into two sets, each containing at most $2n/3$ vertices [36]. Instead of edge separators, one may use edge bisectors of size $(6\sqrt{2}+4\sqrt{3})\sqrt{\Delta n}$ that partitions the vertices into two sets, each containing at most $\lceil n/2 \rceil$ vertices.

Open Problem 4.1. Does every n-vertex planar graph admit a polyline drawing with $(3/8)n^2$ area and bend complexity two?

In the fixed embedding setting, a tight bound on area can be achieved using bend complexity 1, but it is not known whether 1 bend per edge is necessary. Our drawings in the variable embedding setting have bend complexity two. Therefore, it would be interesting to examine whether allowing higher bend complexity can reduce the drawing area.

Open Problem 4.2. Determine the minimum integer b (both in fixed and variable embedding settings) such that the area upper bound cannot be improved by allowing a bend complexity higher than b.

Chapter 5

Polyline Drawings with Good Angular Resolution

This chapter examines the interaction between area and angular resolution in planar polyline drawings of triangulations. Small area drawings often suffer from low angular resolution, e.g., well known straight-line drawing algorithms [17, 32, 108] that bounds the drawing area within $2n^2$ have angular resolution $\Omega(1/n^2)$. Better angular resolution has been achieved in polyline drawings. For example, one can construct polyline drawings with bend complexity two such that the minimum angle at each vertex v is at least $1/d(v)$ and the area is bounded by $O(n^2)$ [43, 69].

In this chapter we give a new technique to compute polyline drawings for planar triangulations. Our algorithm is simple and intuitive, yet implies significant improvement over known results. We present a smooth interaction between the area and angular resolution of polyline drawings with bend complexity 2.

Specifically, for any given n-vertex triangulation, our algorithm computes a drawing with angular resolution $r/d(v)$ at each vertex v, and area $f(n,r)$, for any $r \in (0,1]$, where $d(v)$ denotes the degree at v. For $r < 0.389$ or $r > 0.5$, $f(n,r)$ is less than the drawing area required by previous algorithms; $f(n,r)$ ranges from $7.12n^2$ when $r \leq 0.3$ to $32.12n^2$ when $r = 1$.

We first discuss the motivation behind maximizing angular resolution while drawing graphs. We then briefly review the related research and discuss our contribution. The subsequent sections present the technical background (Section 5.2), the drawing algorithm (Sections 5.3), a probable trade-off between angular resolution and area (Sections 5.4), limitations of our approach and directions to future research (Section 5.5).

Some of the results of this chapter appeared in preliminary form at the 22nd International Symposium on Graph Drawing [49].

5.1 Maximizing Angular Resolution

Polyline drawing has a wide range of applications in the area of software visualization [28, 111] and layout of circuit diagrams [27]. In previous chapters we have seen how the use of bends helps minimize the area of the drawing. Besides minimizing area and bends, it is also important to maximize the angular resolution, specially for visualization applications, where a drawing avoiding thin angles may significantly improve the readability of the drawing [113].

The interaction among angular resolution, area and bend complexity have been examined over the past decades. There exist drawing algorithms that can

compute polyline drawings with constant bend complexity such that the angular resolution is $\Omega(1/d(v)$, and the area is within $O(n^2)$. However, the trade-off between the constants hidden in the asymptotic notations are not yet known to be optimal. This motivated us to examine the interplay between angular resolution and area in polyline drawings.

5.1.1 Related Work

Every planar triangulation with n vertices admits a straight-line drawing in $O(n^2)$ area [32]. Several improvements on the constant hidden in $O(.)$ notation have been achieved [17, 32, 108], and the best known bound is $8n^2/9$ [17]. Better upper bounds, i.e., $4n^2/9$, can be attained in polyline drawings with bend complexity 1 [15, 120]. However, none of these algorithms try to optimize angular resolution, and hence the angular resolution of these drawings may be as low as $1/n^2$.

Kurowski [86] showed that allowing $9n^2/2$ area helps improve the angular resolution of straight-line drawings to $O(1/n)$. On the other hand, Garg and Tamassia [64] showed that there exists an n-vertex planar graph such that any of its straight-line drawings with angular resolution $\Omega(1/\rho)$ requires $\Omega(c^{\rho n})$ area, where $c > 1$, which suggests that drawings with angular resolution $\Omega(1/\Delta)$ and polynomial area may exist only if we allow the edges to have bends.

Allowing bends helps both to reduce area and to improve angular resolution, e.g., one can construct a polyline drawing with bend complexity 3 and angular resolution at least $2/\Delta$ radians, where the area is bounded by $3n^2$ [71]. The angular resolution can be improved to $\Omega(1/d(v))$ radians (for each vertex v) with an

expense of higher area [43, 69].

Early polyline drawing algorithms with good angular resolution and $O(n^2)$ area were based on the idea of assigning an empty square surrounding each vertex. These empty squares helped to draw the incident edges without creating sharp angles, but forced the area to be very large. For example, Goodrich and Wagner [69] gave such an algorithm that computes polyline drawings with bend complexity two and $200n^2$ area, and guarantees at least $1/d(v)$ radians of angular resolution. Later, Duncan and Kobourov [43] developed an algorithm to produce polyline drawings with smaller area, but these drawings do not have the square-emptiness property around the vertices, as well the angular resolution decreases by a factor of 2.

5.1.2 Our Contribution

We assume that the input is a plane graph, i.e., a combinatorial embedding of the graph is also given as an input, and the output drawing must respect the given embedding. Table 5.1.2 presents a brief summary of the best known upper and lower bounds on area and angular resolution for different bend complexities.

No result listed in Table 5.1.2 completely dominates another result. For example, although the drawing of [43] has smaller area than that of [22], it is not an improvement over [22] because of its lower angular resolution.

Figure 5.1 illustrates the solution space dominated by our algorithm in gray. Our algorithm dominates all the previous polyline drawing algorithms either in area or in angular resolution, except Duncan and Kobourov's algorithm [43],

Graph Class	Area	Resolution	Bend Complexity	Total Bends	Ref.
Triangulations	$8n^2/9$	$\Omega(1/n^2)$	0	0	[17]
Triangulations	$9n^2/2$	$\Omega(1/n)$	0	0	[86]
Triangulations	$12.5n^2$	$0.5/d(v)$	1	$3n$	[43]
Triangulations	$450n^2$	$1/d(v)$	1	$3n$	[22]
Triangulations	$4n^2/9$	$\Omega(1/n^2)$	1	$2n/3$	[120]
Triangulations	$200n^2$	$1/d(v)$	2	$6n$	[69]
Planar Graphs	$3n^2$	$2/\Delta$	3	$5n-15$	[71]
Our Results					
Triangulations	$(6\alpha+8/3)^2n^2$	$\frac{\alpha}{\alpha+d(v)(\alpha^2+1/4)}$	2	$5.5n$	Theorem 7
Triangulations	$(6\beta+2/3)^2n^2$	$\frac{\beta}{\beta+d(v)(\beta^2+1)}$	2	$5.5n$	Theorem 8

Table 5.1: Angular resolution, area, bend complexity and total bends in polyline drawings, where $\alpha \in [1/4, 1/2]$, and $\beta \in [1/3, 1]$.

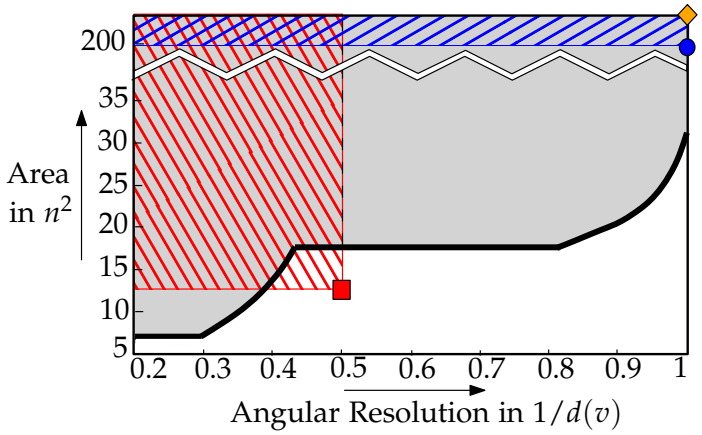


Figure 5.1: Trade-off between angular resolution and area for polyline drawings with bend complexity at most 2. The bold line denotes the trade-off established in this chapter. The square, circle and diamond denote the references [43], [69] and [22], respectively.

which dominates our algorithm along a small interval of the X-axis, corresponding to angular resolution in the interval $[0.38/d(v), 0.5/d(v)]$.

5.2 Technical Foundation

Let G be a planar triangulation with n vertices, and let $\{T_l, T_r, T_m\}$ be the minimum Schnyder realizer of G. Bonichon et al. [14] showed that $\text{leaf}(T_l) + \text{leaf}(T_r) + \text{leaf}(T_m) = 2n - 5 - \delta_0$, where δ_0 is the number of cyclic inner faces in the realizer. The value of δ_0 can be at most $\lfloor (n-1)/2 \rfloor$. Hence we can observe the following property.

Fact 2 *Let $\{T_l, T_r, T_m\}$ be a minimum Schnyder realizer of an n-vertex triangulation. Then* $\min\{\text{leaf}(T_l) + \text{leaf}(T_r), \text{leaf}(T_l) + \text{leaf}(T_m), \text{leaf}(T_r) + \text{leaf}(T_m)\} \leq (4n - 2\delta_0 - 10)/3$, where $0 \leq \delta_0 \leq \lfloor (n-1)/2 \rfloor$.

A non-root vertex in T_l is called a *primary vertex of T_l* if it is the first child of its parent in the clockwise order. Similarly, a non-root vertex in T_r is a *primary vertex of T_r* if it is the first child of its parent in the anticlockwise order.

Lemma 5 *Let n_l and n_r be the number of nonprimary vertices in T_l and T_r, respectively. Then* $n_l + n_r \leq \text{leaf}(T_l) + \text{leaf}(T_r)$.

Proof: We first prove the following claim: For every rooted tree T (where a vertex is defined as a primary vertex if it is the first child of its parent in clockwise order), the number of primary vertices in T is at most the number of leaves in T. We prove the claim using an induction on the number of vertices in T. If T consists of a single vertex, then the only nonprimary vertex of T is the root, which is also the leaf of T. We now assume that the claim holds for every rooted tree T having less than k vertices, where $k \geq 1$. We show that the claim holds when T has

k vertices. Let u be the primary vertex incident to the root of T and let P be a maximal path of primary vertices in T that includes u. The vertex of P with the deepest level is a leaf of T, otherwise, it will contradict the maximality of P. Now consider the connected components $C_1, C_2, \ldots, C_t$ obtained by deleting the path P from T. Each of these components is a rooted tree, and by induction hypothesis, the number of nonprimary vertices in each of these trees is at most the number of leaves in that tree. Let $L_1, L_2, \ldots, L_t$ be the leaves of $C_1, C_2, \ldots, C_t$, respectively. Then the number of nonprimary vertices in T is at most $1 + L_1 + L_2 + \ldots + L_t$, which is the number of leaves in T.

The above claim ensures that $n_l \leq \text{leaf}(T_l)$. The proof that $n_r \leq \text{leaf}(T_r)$ is symmetric. ■

In a *plus-contact representation* of G, each vertex of G is represented as an axis-aligned plus shape (i.e., a shape consisting of two intersecting line segments) such that two plus shapes touch if and only if their corresponding vertices are adjacent in G [47]. Let Γ be a plus contact representation, and let v be any vertex in Γ. Then by $\texttt{P}(v)$ we denote the plus-shape that corresponds to v in Γ. By the *center* $\texttt{C}(v)$ of $\texttt{P}(v)$, we denote the intersection point of the vertical and horizontal straight line segments of $\texttt{P}(v)$. The four straight line segments that start at $\texttt{C}(v)$ and extend to the left, right, above and below $\texttt{C}(v)$ are the *left, right, up and down hands* of v, which we denote by $\texttt{L}(v), \texttt{R}(v), \texttt{U}(v)$ and $\texttt{D}(v)$, respectively. A *j-shift operation* on Γ with respect to an infinite horizontal line (respectively, vertical line) ℓ is performed as follows:

Step 1. Remove all the edges that are lying completely above (respectively, to the

right of) ℓ.

Step 2. Increase the y-coordinate (respectively, x-coordinate) of every vertex lying above (respectively, to the right of) ℓ by j units.

Step 3. Draw the edges that were removed using the new vertex positions.

Step 4. Extend the edges intersected by ℓ upwards (respectively, to the right) until they reach to their other endpoint.

5.3 Polyline Drawing

We first present an overview of our algorithm, and then describe the algorithmic details.

5.3.1 Algorithm Overview

Given a planar triangulation G with n vertices, we first compute a Schnyder realizer $\{T_l, T_r, T_m\}$ of G, as shown in Figure 5.2(a). Let G' be the graph obtained by removing the m-edges from G.

We compute a plus-contact representation of G', as illustrated in Figure 5.2(b). Observe that every edge of G' is realized as a contact between two hands of the corresponding plus shapes. A natural attempt is to route the edges along these hands. However, a hand may contain two or more contact points, e.g., the down hand of v_1, and hence such an edge routing may create edge overlaps in the drawing.

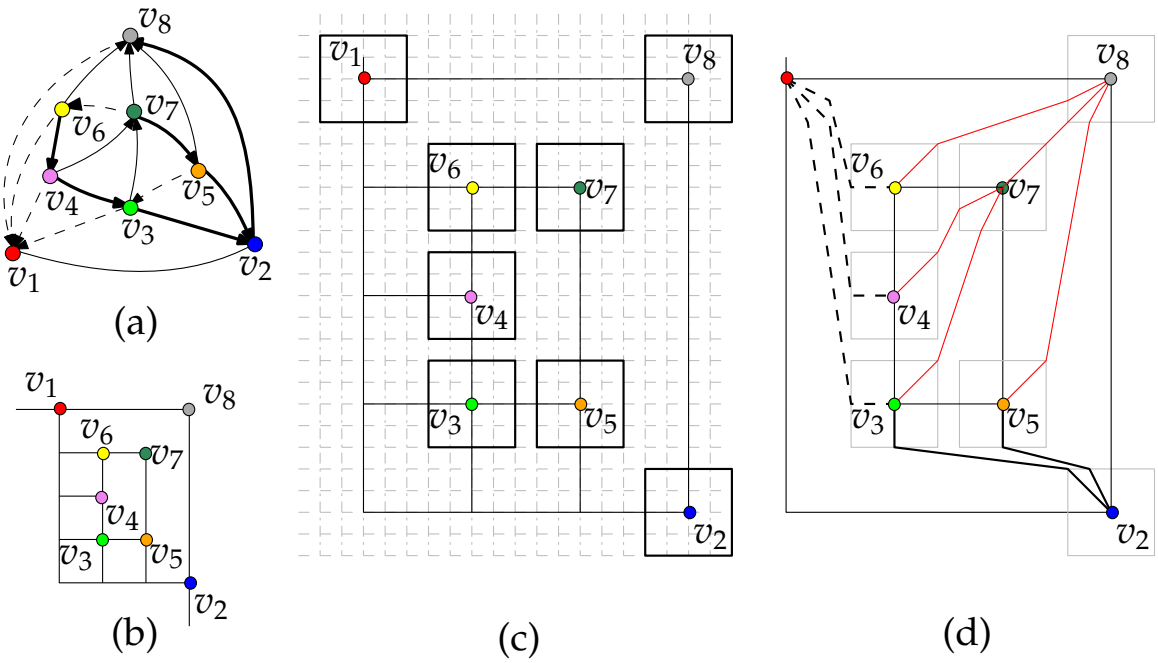


Figure 5.2: (a) A planar graph G. (b) A plus-contact representation of G'. (c) Insertion of dummy grid lines to create intermediate gaps. (d) Drawing of the m-edges of G.

To avoid edge overlaps, we create some intermediate gaps between adjacent rows and columns, which helps route the edges without introducing edge overlapping. Figure 5.2(c) illustrates such a scenario. Finally, we draw the m-edges.

While creating the intermediate gaps between adjacent rows and columns, we ensure that each vertex gets surrounded by an empty rectangle. For each vertex, we route its incident edges using the boundary points on the corresponding surrounding rectangle, as shown in Figure 5.2(d). Later, we show how to vary the size of these empty rectangles to control the area and angular resolution of the drawing simultaneously.

5.3.2 Algorithm Details

Let G be an n-vertex planar triangulation. We construct the drawing of G in three phases. In the first phase we construct a plus-contact representation of $G \setminus T_m$ on a rectangular grid. In the next phase we expand the drawing by inserting dummy grid lines, and in the third phase we use these grid lines to draw the edges of T_m, and route the l- and r-edges avoiding degeneracy.

5.3.2.1 Phase 1 (Plus-Contact)

Let $\sigma = (v_1, v_2, \ldots, v_n)$ be a canonical ordering of G and let $\{T_l, T_r, T_m\}$ be the corresponding Schnyder realizer. By $d_l(v), d_r(v)$ and $d_m(v)$ we denote the number of $l-$, $r-$ and m-edges that are incoming to v, respectively. For example, in Figure 5.2, we have $d_l(v_3) = 1, d_r(v_3) = 1$ and $d_m(v_3) = 0$. Let G_k, where $2 \leq k \leq n$, be the subgraph of G induced by the vertices $v_1, \ldots, v_k$, and let P_k be the clockwise path from v_1 to v_2 on the outer face of G_k. Let Γ_k, where $2 \leq k \leq n$, be the drawing of all the edges of G_k except the m-edges.

We first construct the drawing Γ_2 for G_2, as follows. Place $\texttt{C}(v_1)$ and $\texttt{C}(v_2)$ at coordinates $(1, 2)$ and $(2, 1)$, respectively. Then the horizontal and vertical unit-segments to the left and below $(1, 2)$ correspond to the hands $\texttt{L}(v_1)$ and $\texttt{D}(v_1)$, respectively. Similarly, the horizontal and vertical unit-segments to the left and below $(2, 1)$ correspond to the hands $\texttt{L}(v_2)$ and $\texttt{D}(v_2)$, respectively, as illustrated in Figure 5.3(b). We now insert the vertices in the canonical ordering maintaining the following invariants. While inserting a new vertex v_i, where $3 \leq i \leq n$, we only draw the $l-$ and r-edges.

$\mathcal{I}_1$. The upper envelope of Γ_i is both x- and y-monotone, where the upper envelope is determined by the left and down hands of the vertices in P_i.

$\mathcal{I}_2$. The ray with slope $+1$ starting at any outer vertex of Γ_i can be extended towards infinity avoiding any edge crossing.

$\mathcal{I}_3$. Every l-edge starts as a left hand of some plus shape and ends either at a center or at a down hand of some other plus shape.

$\mathcal{I}_4$. Every r-edge starts as a down hand of some plus shape and ends either at a center or at a left hand of some other plus shape.

Since the upper envelope of G_2 forms a staircase, and does not contain any l- or r-edge, it is straightforward to verify the invariants for Γ_2. We now assume that invariants $\mathcal{I}_1$–$\mathcal{I}_4$ hold for $G_2, G_3, \ldots, G_{k-1}$, where $k-1 < n$, and consider the insertion of vertex v_k.

Let $w_l, w_{l+1}, \ldots, w_{r-1}, w_r$ be the neighbors of v_k on P_{k-1}. Consider an infinite horizontal line ℓ_h that lies in between the horizontal grid line determined by $\mathtt{L}(w_l)$ and the horizontal grid line immediately below $\mathtt{L}(w_l)$. Similarly, let ℓ_v be an infinite vertical line that lies in between the vertical grid line determined by $\mathtt{D}(w_r)$ and the vertical grid line immediately to the left of $\mathtt{D}(w_r)$. We now add v_k considering the following cases. The case when $k = n$ is special, which is handled by Case 4.

Case 1 (v_k is a nonprimary vertex in both T_l and T_r): We first perform a 1-shift with respect to ℓ_h. This increases the number of horizontal lines by 1 and ensures

that $\mathtt{D}(w_l)$ contains at least 1 grid point p that does not contain any vertex or contact point. Similarly, we perform a 1-shift with respect to ℓ_v, which increases the number of vertical lines by 1 and ensures that $\mathtt{L}(w_r)$ contains at least 1 grid point q that does not contain any vertex or contact point. We now consider the horizontal ray r_p that starts at p. Since the upper envelope of Γ_{k-1} is x monotone and p does not contain any vertex or contact point, r_p does not intersect Γ_{k-1} except at p. Similarly, we define a vertical ray r_q that starts at q, which does not intersect Γ_{k-1} except at q. We now place v_k at the intersection point of r_p and r_q,

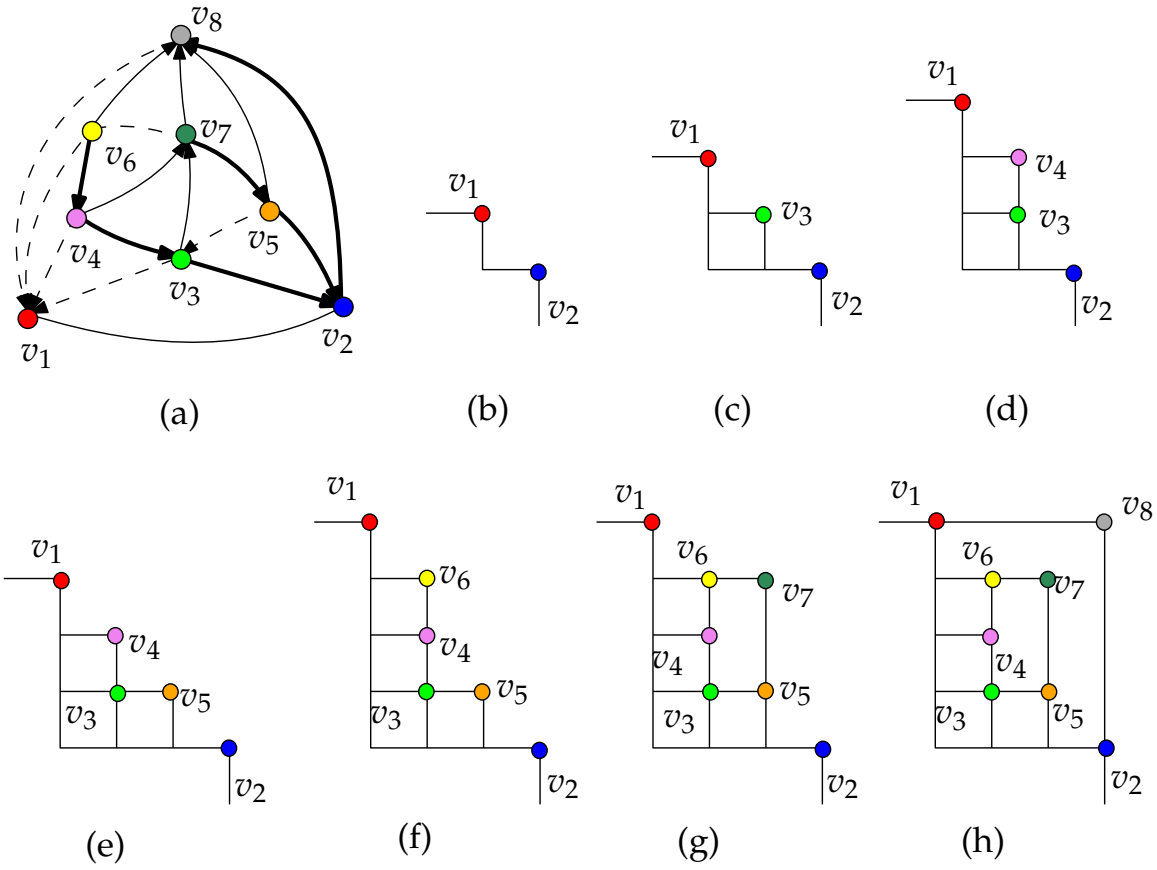


Figure 5.3: (a) A plane graph G and a minimum Schnyder realizer of G. (b)–(h) Illustration for the drawing of $G \setminus T_m$.

and draw the edges (v_k, w_l) and (v_k, w_r). Since r_p and r_q do not intersect Γ_{k-1} except at p and q, respectively, drawing of these edges does not introduce any crossing. Figure 5.3(c) illustrates such a scenario.

We now show that Γ_k respects the invariants $\mathcal{I}_1$–$\mathcal{I}_4$. By induction hypothesis, the subpaths of P_{k-1} from v_1 to w_l and from w_r to v_2 are x-monotone in Γ_{k-1}, which remain the same after any shift operation. The subpath $w_l, \ldots, w_r$, is covered by a horizontal and a vertical line segment in Γ_k. Thus the upper envelope of Γ_k is both x- and y-monotone, which satisfies $\mathcal{I}_1$. The ray r with slope $+1$ starting at v_k lies in between the parallel rays starting at w_l and w_r. Consequently, r can be extended towards infinity avoiding any edge crossing, which satisfies $\mathcal{I}_2$. Shift operations do not change the horizontal and vertical alignment of the edges. Hence by the construction of r_p and r_q, the invariants $\mathcal{I}_3$ and $\mathcal{I}_4$ also hold for Γ_k.

Case 2 (v_k is a primary vertex in T_l but a nonprimary vertex in T_r): In this case we perform a 1-shift with respect to ℓ_v, which increases the number of vertical lines by 1 and ensures that $\mathtt{L}(w_r)$ contains at least 1 grid point q that does not contain any vertex or contact point. Assume that $p = \mathtt{C}(w_l)$. We now consider the horizontal ray r_p that starts at p. Since the upper envelope of Γ_{k-1} is x monotone and p does not contain any vertex or contact point, r_p does not intersect Γ_{k-1} except at p. Similarly, we define a vertical ray r_q starting at q, which does not intersect Γ_{k-1} except at q. We now place v_k at the intersection point of r_p and r_q, and draw the edges (v_k, w_l) and (v_k, w_r). Figure 5.3(e) illustrates such a scenario.

In the similar way as in Case 1, one can observe that Γ_k respects invariants $\mathcal{I}_1$–$\mathcal{I}_4$.

Case 3 (v_k is a nonprimary vertex in T_l but a primary vertex in T_r): This case is symmetric to Case 2, i.e., we perform a 1-shift with respect to ℓ_h to obtain a new grid point p on $\mathtt{D}(w_l)$ and assume that $q = \mathtt{C}(w_r)$.

Case 4 (v_k is a primary vertex in both T_l and T_r): In this case we do not perform any shift, and assume that $p = \mathtt{C}(w_l)$ and $q = \mathtt{C}(w_r)$.

The following lemma bounds the area of Γ_n using Fact 2 and Lemma 5.

Lemma 6 *Γ_n is a drawing on a $(W+2) \times (H+2)$ grid, where $W + H \leq \text{leaf}(T_l) + \text{leaf}(T_r)$.*

Proof: Observe that Γ_2 is a drawing on a 2×2 grid (ignore the left and down hands of v_1 and v_2, respectively). While inserting a vertex v_k, where $2 \leq k \leq n-1$, we increase the width by one unit only when v_k is a nonprimary vertex in T_r. Similarly, we increase the height by one unit only when v_k is a nonprimary vertex in T_l. Insertion of v_n does not increase the size of the grid. Hence the total increase in the width and height (i.e., W and H) is bounded by the number of nonprimary vertices in T_r and T_l, respectively. By Lemma 5, the number of nonprimary vertices in T_r and T_l is at most $\text{leaf}(T_l) + \text{leaf}(T_r)$, which completes the proof. ■

5.3.2.2 Phase 2 (Expansion)

For any plus-contact representation on an integer grid, we define a *free grid line* as a grid line that does not contain any vertex-center or contact points. We refer the

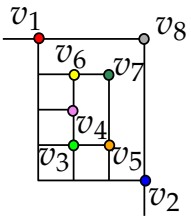
reader to Figure 5.4.

Consider the horizontal grid lines from top to bottom. For every horizontal grid line ℓ containing at least one vertex of Γ, we now perform two $\lfloor d(v)/2 \rfloor$-shifts, where v is the vertex with the largest degree over all the vertices on ℓ. Let ℓ_h (respectively, ℓ'_h) be an infinite horizontal line that lies in between the horizontal grid line ℓ and the horizontal grid line immediately below (respectively, above) ℓ. Perform a $\lfloor d(v)/2 \rfloor$-shift with respect to ℓ_h, and then a $\lfloor d(v)/2 \rfloor$-shift with respect to ℓ'_h. Observe that for each vertex w on ℓ, we now have a set of $\lfloor d(v)/2 \rfloor$ free grid lines above w and a set of $\lfloor d(v)/2 \rfloor$ free grid lines below w. We consider a corresponding set S_w that consists of these $2\lfloor d(v)/2 \rfloor$ free grid lines along with the line ℓ. Furthermore, we assume that the grid lines of S_w are ordered in the increasing order of y-coordinates. Figure 5.4(b) illustrates S_{v_4}.

Similarly, we consider the vertical grid lines from right to left, and for every vertical grid line ℓ' containing at least one vertex of Γ, we perform two $\lfloor d(v)/2 \rfloor$-shifts to the left and right side of ℓ', where v is the vertex with the largest degree over all the vertices on ℓ'. We consider a corresponding set S'_w that contains these $2\lfloor d(v)/2 \rfloor$ free vertical grid lines along with the line ℓ', where the lines are ordered in the decreasing order of x-coordinates. Let the resulting drawing be Γ'_n, as shown in Figure 5.4(c). The following property is a straightforward consequence of the Expansion phase.

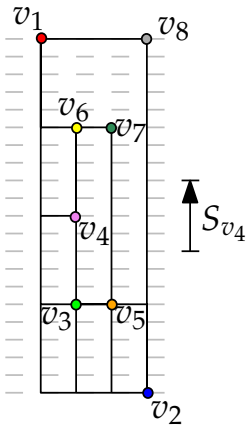
Fact 3 *For every vertex v in Γ'_n, the point $c(v)$ lies at the center of an integer grid A_v of size $(2\lfloor d(v)/2 \rfloor + 1) \times (2\lfloor d(v)/2 \rfloor + 1)$. The grid A_v does not contain any vertex, contact point, or edge of Γ' except the four hands of v. Furthermore, for any other vertex*

$u(\neq v)$, the grids A_u and A_v are disjoint, i.e., they do not share any common grid point.

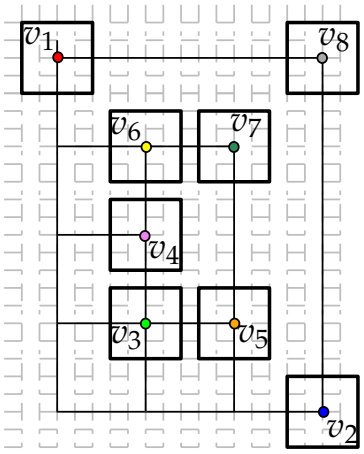


$d(v_1) = 5, \quad d(v_2) = 4,$
$d(v_3) = 5, \quad d(v_4) = 4,$
$d(v_5) = 4, \quad d(v_6) = 4,$
$d(v_7) = 5, \quad d(v_8) = 5.$

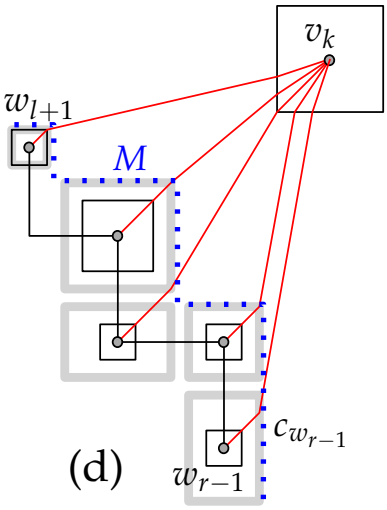
(a)



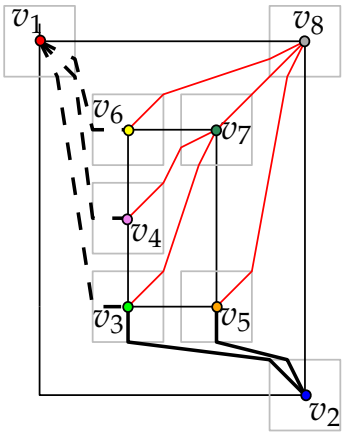
(b)



(c)



(d)



(e)

Figure 5.4: Illustration for (a) Γ_n, (b) S_{v_k}, and (c) Γ'_n, where the grid A_v, for each vertex v, is shown in black squares. (d) Illustration for M. Note that A_ws are bounded by gray rectangles determined by S_w and S'_w. (e) Γ''.

5.3.2.3 Phase 3 (Edge Routing)

For each vertex in canonical order, we first route the incoming m-edges incident to v_k, as follows. Recall that the m-edges start at the vertices $w_{l+1}, \ldots, w_{r-1}$ and ends at v_k.

By the construction of Γ'_n, the vertices $w_{l+1}, \ldots, w_{r-1}$ lie below S_{v_k} and to the left of S'_{v_k}. Let M be the monotone chain determined by the last line of S_w and first line of S'_w, where $w \in \{w_{l+1}, \ldots w_{r-1}\}$. Figure 5.4(d) illustrates M with a dotted line. For each j, where $l+1 \leq j \leq r-1$, let c_{w_j} be the top-right corner of A_{w_j}, and assume that $z = \lceil d_m(v)/2 \rceil$.

For each $w \in \{w_{l+1}, \ldots w_z\}$, we now route the m-edge incident to w through the top-right corner c_w up to M, and then to a distinct grid point on the leftmost boundary of A_{v_k} below $\mathtt{L}(v_k)$. Observe that $\lceil d_m(v_k)/2 \rceil \leq d_m(v_k)/2 + 1 \leq (d(v_k) - 3)/2 + 1 \leq (d(v_k) - 1)/2$. Since $(d(v_k) - 1)/2$ is at most $\lfloor d(v_k)/2 \rfloor$ (irrespective of the parity of $d(v_k)$), the grid points on the leftmost boundary of A_{v_k} below $\mathtt{L}(v_k)$ are sufficient to route all the m-edges incident to $\{w_{l+1}, \ldots w_z\}$. Similarly, for each $w \in \{w_{z+1}, \ldots w_{r-1}\}$, we now route the m-edge incident to w through the top-right corner c_w up to M, and then to a distinct grid point on the bottommost boundary of A_{v_k} to the left of $\mathtt{D}(v_k)$. Since $\lfloor d_m(v_k)/2 \rfloor \leq \lfloor d(v_k)/2 \rfloor - 1$ (irrespective of the parity of $d(v_k)$), we have sufficient number of boundary points to route all the m-edges incident to $\{w_{z+1}, \ldots w_{r-1}\}$.

The l- and r-edges of Γ'_n contain edge overlapping on the left and down hands. From the Expansion phase it is straightforward to observe that the l-edges that are incoming to some vertex v in Γ'_n, are incident to $\mathtt{D}(v)$, and properly intersects

the first half of the S'_v. Let ℓ be the nearest vertical grid line to the right of S'_v. Remove the parts of these l-edges that lie in between $\texttt{D}(v)$ and ℓ (except for the l-edge incident to $\texttt{C}(v)$). Since all these l-edges lie below S_v, the points where these l-edges are incident to ℓ can see all the grid points on the rightmost boundary of A_v and on the right-half of the bottommost boundary of A_v. Consequently, we can route the l-edges to $\texttt{C}(v)$ through these boundary grid points, which removes the edge overlaps on $\texttt{D}(v)$. Figure 5.4(e) illustrates such a scenario.

Symmetrically, we can remove the degeneracy of r-edges on $\texttt{L}(v)$. Fact 3 and the property that the lines in S_v and S'_v do not contain any vertex except v ensure that the above modifications do not introduce any edge crossing. The resulting drawing Γ'' is a planar polyline drawing of G, e.g., see Figure 5.4(e).

5.3.3 Bounding the Aesthetics

Here we bound the area, bend complexity and angular resolution of the drawing.

Area: By Lemma 6, the area before the Expansion phase was $(W+2) \times (H+2)$. For each i, where $1 \leq i \leq W+2$, the Expansion phase increases the width of the drawing by $2\lfloor d(u_i)/2 \rfloor$, where u_i is the vertex with the largest degree on the ith column. Hence the total increase is at most $(\sum_{i=1}^{W+2} d(u_i)) - 3(n-W-2) \leq (6n-12) - 3(n-W-2) = 3n+3W-6$. Similarly, the increase in height is at most $3n+3H-6$. Hence Γ'' is a drawing on an integer grid of size $(3n+4W-4) \times (3n+4H-4)$. Since $W+H \leq (4n-2\delta_0-10)/3$ (see Fact 2), the area can be at most $(3n+4(2n-\delta_0-5)/3)^2 = ((17n-4\delta_0-20)/3)^2 \leq 32.12n^2$.

Bends: If (v, v') is an l-edge or r-edge in Γ_G, which starts at v and ends at v', then the edge has at most 2 bends: one before entering $A_{v'}$, and another at the boundary of $A_{v'}$. If (v, v') is an m-edge, then it contains one bend on M, and another bend on the boundary of $A_{v'}$. The l-and r-edges that connect a primary vertex to its parent, do not contain any bend. Since $\delta_0 < n/2$ [14] and $\text{leaf}(T_m) < n$, the drawing has at most $6n - \text{leaf}(T_l) - \text{leaf}(T_r) \leq 11n/2$ bends.

Angular Resolution: To compute the angular resolution, observe that the smallest possible angle θ at v is realized by a pair of consecutive integer grid points on the boundary of A_v where one of them is the corner of A_v, e.g., see Figure 5.5(a). Since A_v is a grid of size $(2\lfloor d(v)/2 \rfloor + 1) \times (2\lfloor d(v)/2 \rfloor + 1)$, the length of the line segment l connecting the center to any corner is $\sqrt{2}\lfloor d(v)/2 \rfloor$. Hence we have

$$\theta = \arctan\left(\frac{1/\sqrt{2}}{(\sqrt{2}\lfloor d(v)/2 \rfloor - 1/\sqrt{2}}\right)$$
$$\Rightarrow \theta \geq \arctan\left(\frac{1}{d(v)-1}\right).$$

We simplify the expression using the MacLaurin Series Expansion of arctan in the same way as used in [78]. By the MacLaurin series expansion of arctan, if $|x| < 1$, then $\arctan(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \ldots \geq x - \frac{1}{3}x^3$. Since $\frac{1}{d(v)-1} < 1$, we have $\theta \geq \frac{1}{d(v)-1} - \frac{1}{3}\left(\frac{1}{d(v)-1}\right)^3 \geq \frac{1}{d(v)}$.

Theorem 6 *Every n-vertex planar triangulation admits a polyline drawing Γ with bend complexity 2, where the angular resolution is at least $1/d(v)$ for each vertex v, and area is at most $(3n + 4W - 4) \times (3n + 4H - 4)$, where $W + H \leq (4n - 2\delta_0 - 10)/3$.*

5.4 Trade-offs between Angular Resolution and Area

In this section we show that one can significantly improve the area with an small expense of angular resolution. We consider the following two scenarios.

5.4.1 Angular Resolution $\gamma/d(v)$, where $\gamma \in [0.8, 1]$

Observe that the bottom-left quadrants of A_v (with respect to the center $\mathtt{C}(v)$) has at most $2\lfloor d(v)/2 \rfloor - 1 \geq d_m(v)$ boundary points, which are sufficient to route the m-edges, and sometimes necessary. However, the boundary points that are available to route the l-edges (similarly, r-edges) are significantly more than necessary, e.g., the number of boundary points to route the l-edges is $3\lfloor d(v)/2 \rfloor - 2$ (lying on the bottom-right quadrants and on the right-boundary of A_v). Hence assigning a grid of size $(\lfloor d(v)/2 \rfloor + 1 + \lceil d(v)/4 \rceil) \times (\lfloor d(v)/2 \rfloor + 1 + \lceil d(v)/4 \rceil)$ to each vertex v would be sufficient for routing the edges.

Observe that for each vertex v, the increase in width is at most $(\lfloor d(v)/2 \rfloor + \lceil d(v)/4 \rceil) \leq (3d(v)/4 + 1)$. Since one column may contain multiple vertices, and the degree of each vertex is at least three, we are overcounting the increase for $(n - W - 2)$ vertices. The amount of over computation for each such vertex v' is at least $\lfloor 3d(v')/4 \rfloor + 1 \geq 3$. Consequently, the total increase in the width in the Expansion phase is now bounded by $(\sum_{i=1}^{W+2}(3d(v_i)/4 + 1)) - 3(n - W - 2) \leq 3n/2 + 4W - 1$. Similarly, the increase in height is at most $3n/2 + 4H + 1$. Since $W + H \leq (4n - 2\delta_0 - 10)/3$, the area can be at most $(3n/2 + 5(2n - \delta_0 - 5)/3 + 5)^2 \leq 23.37n^2$. The number of bends remains the same, but the minimum angle

θ is now at least $0.8/d(v)$, which is now determined by two consecutive points along the bottom-right corner, as shown in Figure 5.5(b).

We can parametrize the grid size with a parameter α, i.e., consider the grid assigned to v as $(\lfloor d(v)/2 \rfloor + 1 + \alpha d(v)) \times (\lfloor d(v)/2 \rfloor + 1 + \alpha d(v))$, where $\alpha \geq 1/4$. Then the increase in width is at most $(\sum_{i=1}^{W+2}((\alpha + 1/2)d(v_i) + 1)) - 3(n - W - 2) \leq (6(\alpha + 1/2)n - 3n + 4W + 8) \leq (6\alpha n + 4W + 8)$. Similarly, the increase in height is at most $(6\alpha n + 4H + 8)$, respectively. Hence the area is at most $(6\alpha n + 4(W + H)/2 + 10)^2 \leq (6\alpha n + 8n/3 + 10)^2 \approx (6\alpha + 8/3)^2 n^2$. The angular resolution is at least

$$\theta \geq \arctan\left(\frac{\alpha/\sqrt{\alpha^2 + 1/4}}{d(v)\sqrt{\alpha^2 + 1/4}}\right) = \arctan\left(\frac{\alpha}{d(v)(\alpha^2 + 1/4)}\right)$$

$$\Rightarrow \theta \geq \frac{\alpha}{\alpha + d(v)(\alpha^2 + 1/4)}, \text{ by the MacLaurin Series Expansion [78].}$$

The following theorem summarizes the trade-offs.

Theorem 7 *Every n-vertex planar triangulation admits a polyline drawing with bend complexity 2, where the angular resolution is at least $\frac{\alpha}{\alpha + d(v)(\alpha^2 + 1/4)}$ for each vertex v, and area is at most $(6\alpha n + 4W + 10) \times (6\alpha n + 4H + 10)$. Here $\alpha \in [1/4, 1/2]$, and $W + H \leq (4n - 2\delta_0 - 10)/3$.*

5.4.2 Angular Resolution $\gamma/d(v)$, where $\gamma \in [0.3, 0.5]$

Insert the new grid lines in the Expansion phase such that each vertex v has $h = \beta_v d(v)$ free grid lines, where $\beta_v \geq 1/d(v)$, in each of the four sides (above,

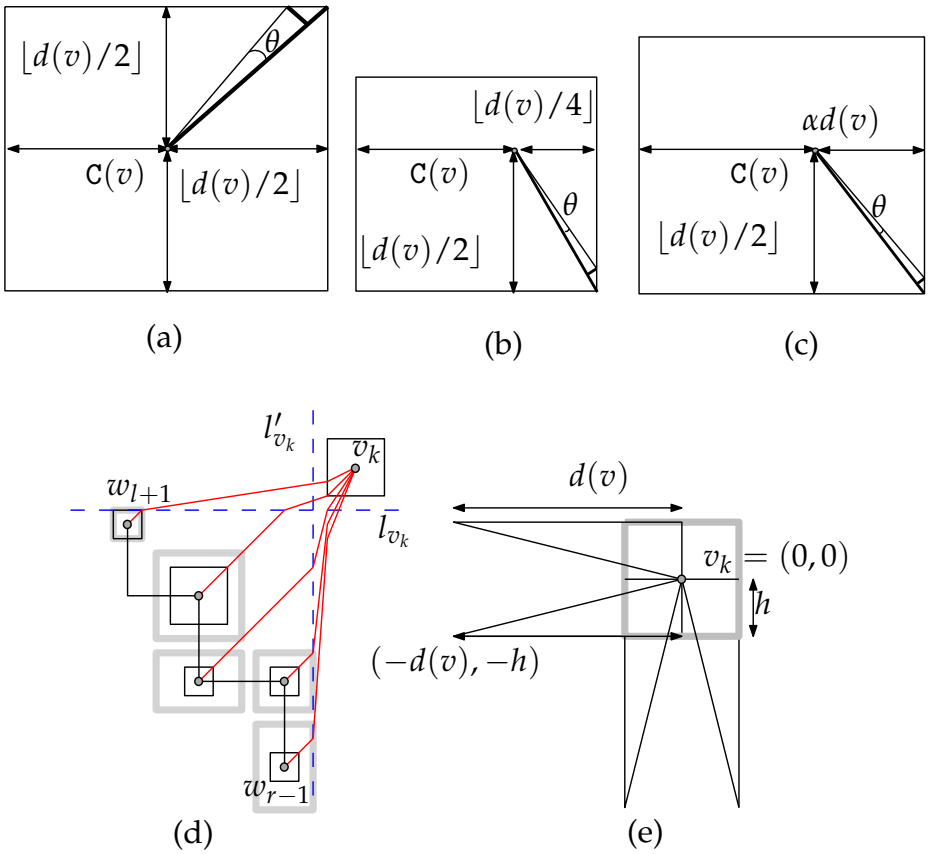


Figure 5.5: Illustration for angular resolution.

below, left, right) around v, i.e., $c(v)$ is at the center of a free integer grid A_v of size $h \times h$. Recall that S_v is an ordered set of horizontal free grid lines associated to v, and S'_v is a set of vertical free grid lines associated to v, where the lines of S_v and S'_v are ordered in increasing order of y-coordinate, and decreasing order of x-coordinate, respectively. We now show that these free grid lines are sufficient for routing the l-, r- and m-edges.

Routing m-edges: Let l_{v_k} and l'_{v_k} be the grid lines that are immediately below and to the left of S_{v_k} and S'_{v_k}, respectively. For each $w \in \{w_{l+1}, \ldots, w_{r-1}\}$, we now

extend a line segment with slope $+1$ from $\mathtt{C}(w)$ until we hit either l_{v_k} or l'_{v_k}. Let $B = \{b(w_{l+1}), \ldots, b(w_{r-1})\}$ be the set of points on l_{v_k} and l'_{v_k} reached by these extensions. We now extend these extensions further to reach $\mathtt{C}(v_k)$, as follows:

- If the number of points of B that lie on l_{v_k} is z', where $z' \leq h$, then we route the extensions of l_{v_k} through z' consecutive grid points lying on the left side of A_{v_k} immediately below $\mathtt{L}(v_k)$. We then route the extensions on l'_{v_k} through the next consecutive grid points along the same vertical line. Since there are at most $d_m(v_k)$ m-edges, we need at most $d(v_k)$ consecutive grid points below $\mathtt{L}(v_k)$. Figure 5.5(d) illustrates such a scenario, where $h = 1$.

- If the number of points of B that lie on l'_{v_k} is at most z', where $z' \leq h$, then the drawing is symmetric to the previous case.

- Otherwise, both l_{v_k} and l'_{v_k} contains more than h extensions. In this case $\min\{z, z'\} > h$, and hence $\max\{z, z'\} \leq d_m(v_k) - h$. We first extend the extensions on l_{v_k} to the grid points that lie consecutively to the left of A_v (on the first line of S_{v_k}). We then extend the extensions on l'_{v_k} to the grid points that lie consecutively below of A_v (on the last line of S'_{v_k}). Finally, we connect all these new extensions directly to $\mathtt{C}(v_k)$. Note that the maximum horizontal (respectively, vertical) distance between $\mathtt{C}(v)$ and a bend point on l_{v_k} (respectively, l'_{v_k}) is at most $(d_m(v_k) - h) + h \leq d(v_k)$.

Routing *l*-edges and *r*-edges: Let $u_1, u_2, \ldots, u_q$ be the vertices in top-to-bottom order that are incident to $\mathtt{D}(v_k)$ by incoming l-edges. Let ℓ be the nearest vertical grid line to the right of S'_v, and remove the parts of these l-edges that lie in

between $\mathtt{D}(v_k)$ and ℓ (except for the l-edge incident to $\mathtt{C}(v_k)$). We then connect these extensions to the q consecutive grid points on the first line of S'_{v_k} that lie immediately below the top-right corner of A_v. Finally, we connect all these new extensions directly to $\mathtt{C}(v_k)$. We route r-edges symmetrically.

Angular Resolution and Area: In all the cases, the smallest angle θ at any vertex v is equal to the angle determined by the points $(-d(v), -h)$ and $(-d(v)+1, -h)$ at $\mathtt{C}(v) = (0,0)$, as illustrated in Figure 5.5(e). Since the perpendicular distance from the point $(-d(v)+1, -h)$ to the line through $(0,0)$ and $(-d(v), -h)$ is $h/\sqrt{d(v)^2+h^2}$, we have

$$\theta \geq \arctan\left(\frac{h/\sqrt{d(v)^2+h^2}}{\sqrt{d(v)^2+h^2}}\right)$$
$$\Rightarrow \theta \geq \frac{h}{h+d(v)^2+h^2}, \text{ by the MacLaurin Series Expansion [78].}$$

Since $h = \beta_v d(v)$, where $\beta_v \geq 1/d(v)$, the angle $\theta \geq \frac{\beta_v}{\beta_v + d(v)(1+\beta_v^2)}$. For large values of β_v, the smallest angle could be determined by two consecutive boundary points of A_v, which can be as small as $\arctan(\frac{1/\sqrt{2}}{h\sqrt{2}}) > \frac{1}{1+2\beta_v d(v)}$. Hence the angular resolution is $\min\{\frac{\beta_v}{\beta_v+d(v)(1+\beta_v^2)}, \frac{1}{1+2\beta_v d(v)}\} = \frac{\beta_v}{\beta_v+d(v)(1+\beta_v^2)}$ when $1/d(v) \leq \beta_v \leq 1$.

The increase of width in the Expansion phase is bounded by $\sum_{i=1}^{i=W+2} \beta_v d(v) - 3(n-W-2)$. The term $-3(n-W-2)$ is only meaningful when $\beta_v d(v) \geq 3$. Hence we use a relaxed upper bound of $\sum_{i=1}^{i=W+2} \beta_v d(v)$ and for convenience of juxtaposition, we assume that for all v, $\beta_v = \beta \geq 1/3$ implying that $\beta_v \geq 1/d(v)$. We thus obtain an upper bound of $\sum_{i=1}^{i=W+2} \beta d(v) < 6n\beta$. Similarly, the height is also bounded by $6n\beta$. Since $W + H < 4n/3$, the area is at most $(6n\beta + (W+H)/2 + 2)^2 \leq (6n\beta + 2n/3 + 2)^2 \leq (6\beta + 2/3)^2 n^2$.

For example, if $\beta_v = \beta = 1/3$, then the angular resolution is at least $0.3/d(v)$, and the area is at most $7.12n^2$.

The following theorem summarizes the trade-offs.

Theorem 8 *Every n-vertex planar triangulation admits a polyline drawing with bend complexity 2, where the angular resolution is at least $\frac{\beta}{\beta+d(v)(1+\beta^2)}$ for each vertex v, and area is at most $(6n\beta + W + 2) \times (6n\beta + H + 2)$. Here $\beta \in [1/3, 1]$, and $W + H \leq (4n - 2\delta_0 - 10)/3$.*

5.5 Summary and Open Questions

In this chapter we have given a smooth trade-off between the area and angular resolution for polyline drawings with bend complexity 2. Similar to the previously known polyline drawing algorithms, one can implement our algorithm using standard techniques [24] such that the drawings are computed in linear time.

Our algorithm dominates all the previous polyline drawing algorithms either in angular resolution or in area, except Duncan and Kobourov's algorithm [43], which dominates our algorithm when the angular resolution is in the interval $[0.38/d(v), 0.5/d(v)]$. A natural open question is to find better trade-offs between area and angular resolution.

Open Problem 5.1. Characterize the interaction among the constants a, b and c such that any planar graph admits a polyline drawing with bend complexity a, angular resolution at least $b/d(v)$, and area at most cn^2.

Another challenging problem is to find tight lower bounds. The case for straight-line drawings is particularly interesting. Every planar graph admits a straight-line drawing with $O(n^2)$ area and $\Omega(1/n)$ angular resolution [86]. On the other hand, there exists an n-vertex planar graph such that any of its straight-line drawings with angular resolution $\Omega(1/\rho)$ requires at least $\Omega(c^{\rho n})$ area, where $c > 1$ [64]. Therefore, one may seek for straight-line drawings with $O(n^{2+\epsilon})$ area and $\Omega(1/n^{\epsilon})$ angular resolution, where ϵ is a constant in the interval $(0, 1)$.

Open Problem 5.2. Compute straight-line drawings of planar triangulations with $O(n^{2+\epsilon})$ area and $\Omega(1/n^{\epsilon})$ angular resolution, where ϵ is a constant in the interval $(0, 1)$?

Chapter 6

Drawing Graphs on Multiple Layers

This chapter focuses on polyline drawings of general graphs. Since the input graphs can be non-planar, the aesthetics of the output polyline drawings include number of crossings, number of planar layers (layer complexity), and bends per edge (bend complexity). Here we examine the interaction between the bend and layer complexities.

By Fáry's theorem [60], every graph with thickness one can be drawn on a single layer with bend complexity 0. A few extensions to higher thickness are known. For example, graphs with thickness two can be drawn on two layers with bend complexity two. In general, every n-vertex graph with thickness t admits a drawing on t layers with bend complexity $3n + O(1)$. Allowing a higher number of layers may reduce the bend complexity, e.g., complete graphs require $\Theta(n)$ layers to be drawn using 0 bends per edge.

In this chapter we present an elegant extension of Fáry's theorem [60] to draw graphs of thickness $t > 2$. We first prove that thickness-t graphs can be drawn on

t layers with $2.25n + O(1)$ bends per edge. We then develop another technique to draw thickness-t graphs on t layers with bend complexity, i.e., $\min\{O(2^{t/2} \cdot n^{1-1/\beta}), 2.25n + O(1)\}$, where $\beta = 2^{\lceil (t-2)/2 \rceil}$. Previously, the bend complexity was not known to be sublinear for $t > 2$. Finally, we show that graphs with linear arboricity k can be drawn on k layers with bend complexity $\frac{3(k-1)n}{(4k-2)}$.

We first discuss the motivation for minimizing bend and layer complexities in polyline drawings. We then briefly review the related research, and describe our contribution. The subsequent sections introduce the definitions and notation necessary to describe our results (Section 6.2), the drawing algorithms (Section 6.3), a trade-off between the number of segments and area (Section 6.4), limitations of our approach and directions to future research (Section 6.5).

Some of the results of this chapter appeared in preliminary form at the 43rd International Colloquium on Automata, Languages and Programming [50].

6.1 Minimizing Layer and Bend Complexities

Many problems in VLSI layout and software visualization are tackled using algorithms that produce polyline drawings [113]. For a variety of practical purposes [27, 28, 111], these algorithms often seek to produce drawings that optimize several drawing aesthetics, e.g., minimizing the number of planar layers, minimizing the number of bends, minimizing the number of crossings, etc.

We examine the parameters bend complexity and layer complexity. In a VLSI layout, minimizing the number of planar layers saves microchip area, and minimizing the number of bends improves reliability of the circuit. In visualization

applications, minimizing these parameters enhances the readability of the visualization.

Previous results on drawing graphs on few layers or with small number of bends per edge, suggest that the bend and layer complexities are conflicting optimization goals. However, no formal study on the interaction between bend and layer complexities has been done. Consequently, we were motivated to extend the understanding of the interplay between the layer complexity and bend complexity in polyline drawings.

6.1.1 Related Work

The layer complexity of every drawing of a thickness-t graph G is at least t, and every n-vertex thickness-t graph admits a drawing on t layers with bend complexity $O(n)$ [102]. The problem of drawing thickness-t graphs on t planar layers is closely related to the *simultaneous embedding* problem, where given a set of planar graphs $G_1, \ldots, G_t$ on a common set of vertices, the task is to compute their planar drawings $D_1, \ldots, D_t$ such that each vertex is mapped to the same point in the plane in each of these drawings. Figure 6.1(a)–(c) illustrate three planar graphs on a common set of vertices, and Figure 6.1(d) depicts a simultaneous embedding with bend complexity 1.

Graphs with low thickness admit polyline drawings on few layers with low bend complexity. If $t = 1$, then by Fáry's theorem [60], G admits a drawing on a single layer with bend complexity 0. Every pair of planar graphs can be simultaneously embedded using two bends per edge [58, 66]. Therefore, if $t = 2$,

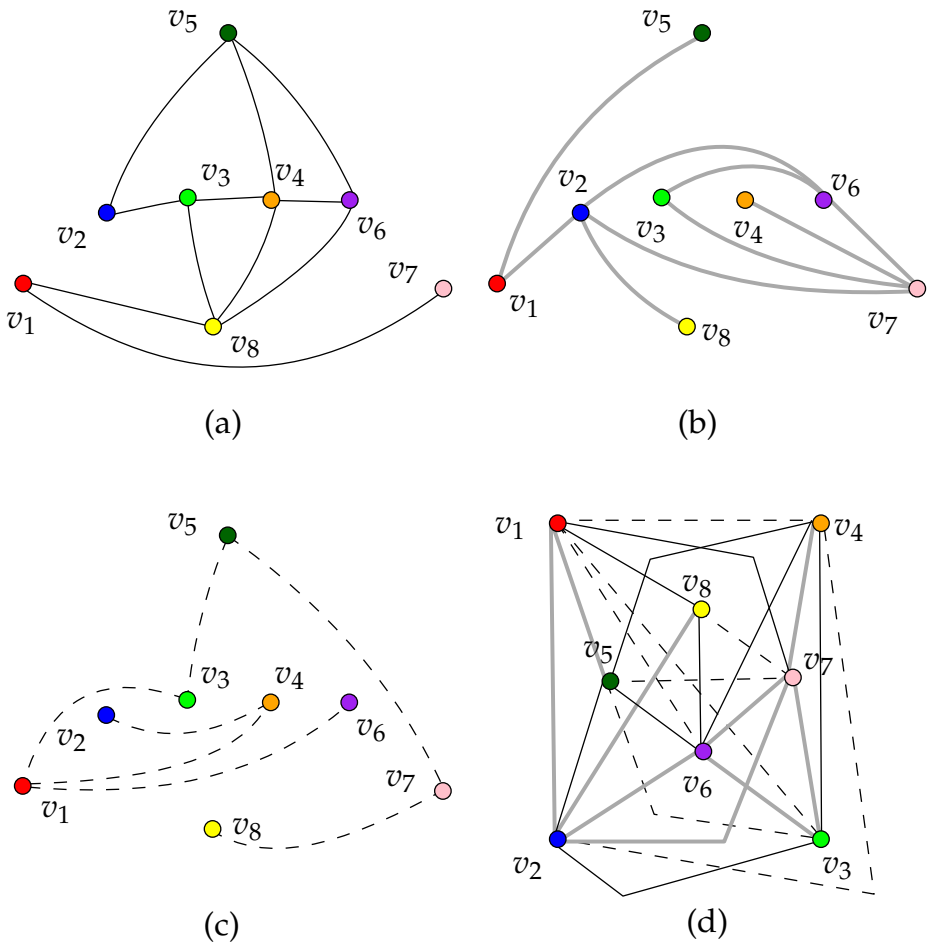


Figure 6.1: Simultaneous embedding of three planar graphs.

then G admits a drawing on two layers with bend complexity 2. The best known lower bound on the bend complexity of thickness-2 graphs is one [52]. Duncan et al. [44] showed that graphs with maximum degree four can be drawn on two layers with bend complexity 0. Wood [118] showed how to construct drawings on $O(\sqrt{m})$ layers with bend complexity 1, where m is the number of edges in G.

Given an n-vertex planar graph G and a point location for each vertex in $\mathbb{R}^2$, Pach and Wenger [102] showed that G admits a planar polyline drawing with the

given vertex locations, where each edge has at most $120n$ bends. They also showed that $\Omega(n)$ bends are sometimes necessary. Badent et al. [6] and Gordon [70] independently improved the bend complexity to $3n + O(1)$. Consequently, for $t \geq 3$, these constructions can be used to draw G on t layers with at most $3n + O(1)$ bends per edge.

A rich body of literature [8, 11, 56] examines *geometric thickness*, i.e., the number of planar layers that is necessary and sufficient to achieve 0 bend complexity. Dujmović and Wood [40] proved that $\lceil k/2 \rceil$ layers suffice for graphs of treewidth k. Duncan [42] proved that $O(\log n)$ layers suffice for graphs with arboricity two or outerthickness two, and $O(\sqrt{n})$ layers suffice for thickness-2 graphs. Dillencourt et al. [37] proved that complete graphs with n vertices require at least $\lceil (n/5.646) + 0.342 \rceil$ and at most $\lceil n/4 \rceil$ layers.

6.1.2 Our Contribution

We assume that the input is an arbitrary graph. Table 6.1 lists our results, as well as summarizes the best known upper bounds on layer and bend complexities for different classes of graphs.

6.2 Technical Foundation

In this section we describe some preliminary definitions, and review some known results.

Let $G = (V, E)$ be a planar graph. A *monotone topological book embedding* of G is a planar drawing Γ of G that satisfies the following properties.

Graph Class	Planar Layers	Bends per Edge	Ref.
Maximum degree 4 graphs	2	0	[44]
Graphs of treewidth k	k	0	[40]
Complete graphs	$\lceil n/4 \rceil$	0	[37]
Graphs of thickness 2	$O(\sqrt{n})$	0	[66, 58]
Graphs of thickness 2	2	2	[66, 58]
General graphs	$O(\sqrt{m})$	1	[118]
Graphs of thickness t	t	$3n + O(1)$	[6, 70]
Our Results			
Graphs of thickness t	t	$2.25n + O(1)$	Theorem 9
Graphs of thickness t	t	$O(2^{t/2} \cdot n^{1-1/\beta})$	Theorem 10
Graphs of linear arboricity k	k	$\frac{3(k-1)n}{(4k-2)}$	Theorem 11

Table 6.1: Results on minimizing layer and bend complexities, where $\beta = 2^{\lceil (t-2)/2 \rceil}$.

P_1: The vertices of G lie along a horizontal line ℓ in Γ. We refer to ℓ as the *spine* of Γ.

P_2: Each edge $(u, v) \in E$ is an x-monotone polyline in Γ, where (u, v) either lies on one side of ℓ, or crosses ℓ at most once.

P_3: Let (u, v) be an edge that crosses ℓ at point d, where u appears before v on ℓ. Let $u, \ldots, d, \ldots, v$ be the corresponding polyline. Then the polyline $u, \ldots, d$ lies above ℓ, and the polyline $d, \ldots, v$ lies below ℓ.

Figure 6.2(a) illustrates a monotone topological book embedding of a planar graph. Edges that are crossing the spine are shown in bold.

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs on a common set of vertices. A *simultaneous embedding* Γ of G_1 and G_2 consists of their planar drawings D_1 and D_2, where each vertex is mapped to the same point in the plane in both D_1 and D_2. Erten and Kobourov [58] showed that every pair of planar graphs admit

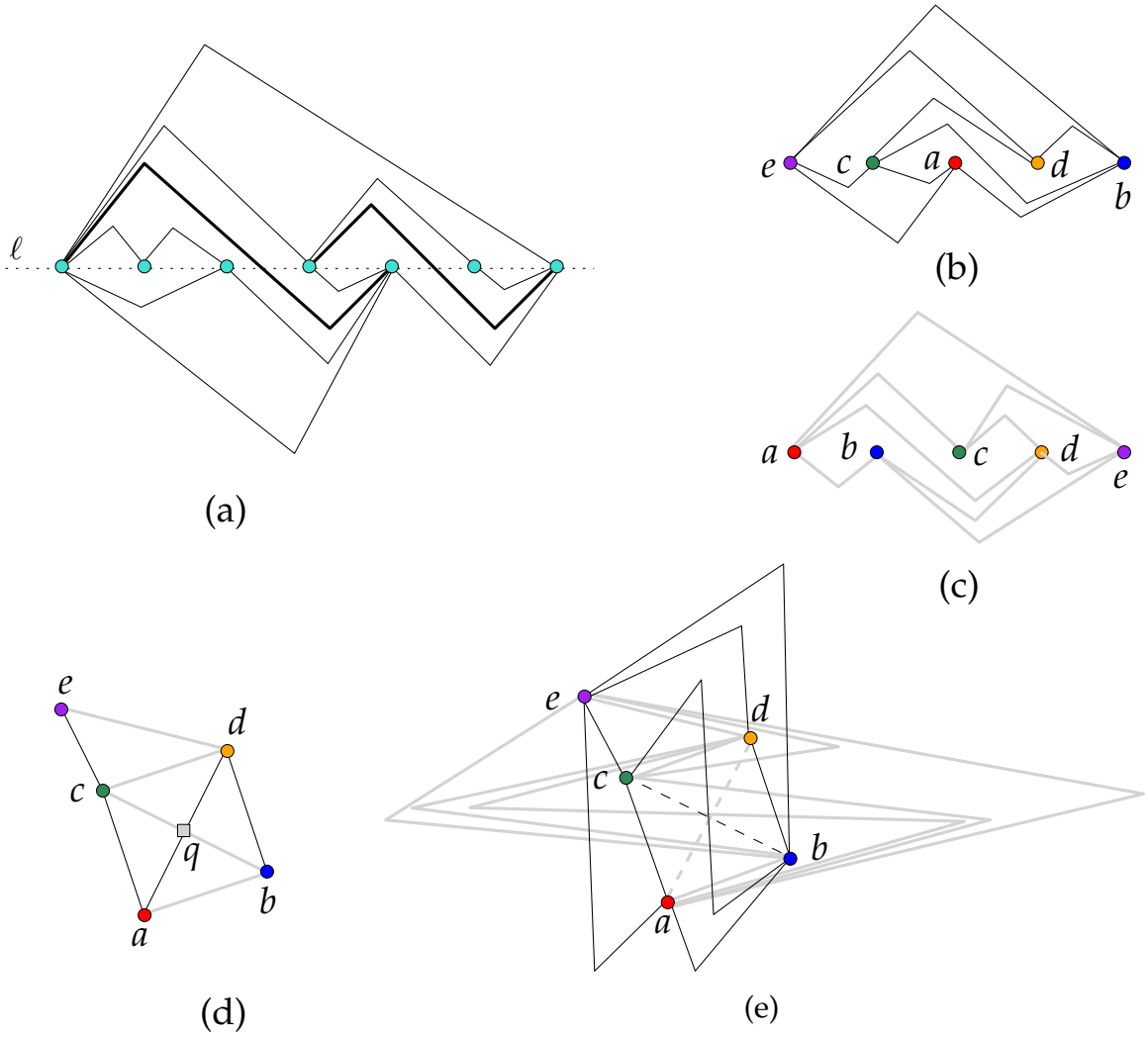


Figure 6.2: (a) A monotone topological book embedding of a planar graph. (b)–(c) Monotone topological book embeddings of G_1 and G_2. (d)–(e) Simultaneous embedding of G_1 and G_2, where the deleted edges are shown in dashed lines.

a simultaneous embedding with at most three bends per edge. Giacomo and Liotta [66] observed that using monotone topological book embedding Erten and Kobourov's [58] construction can achieve a drawing with two bends per edge.

Here we briefly recall this drawing algorithm. Without loss of generality assume that both G_1 and G_2 are triangulations. Let π_i, where $1 \leq i \leq 2$, be a vertex

ordering that corresponds to a monotone topological book embedding of G_i. Let P_i be the corresponding *spinal path*, i.e., a path that corresponds to π_i. Note that some of the edges of P_i may not exist in G_i, e.g., edges (a, d) and (b, c) in Figures 6.2(b) and (c), respectively, and these edges of P_i create edge crossings in G_i. Add a dummy vertex at each such edge crossing. Let $\delta_i(v)$ be the position of vertex v in π_i. Then P_1 and P_2 can be drawn simultaneously on an $O(n) \times O(n)$ grid [19] by placing each vertex at the grid point $(\delta_1(v), \delta_2(v))$; see Figure 6.2(d). The mapping between the dummy vertices of P_1 and P_2 can be arbitrary, here we map the dummy vertex on (a, d) to the dummy vertex on (b, c). Finally, the edges of G_i that do not belong to P_i are drawn. Let e be such an edge in G_i. If e does not cross the spine, then it is drawn using one bend on one side of P_i according to the book embedding of G_i. Otherwise, let q be a dummy vertex on the edge $e = (u, v)$, which corresponds to the intersection point of e and the spine. The edges (u, q) and (v, q) are drawn on opposite sides of P_i such that the polyline from u to v do not create any bend at q. Since each of (u, q) and (v, q) contains a bend, e contains only two bends. Finally, the edges of P_i that do not belong to G_i are removed from the drawing; see Figure 6.2(e).

Let Γ be a planar polyline drawing of a path $P = \{v_1, v_2, \ldots, v_n\}$. We call Γ an *uphill* drawing if for any point q on Γ, the upward ray from q does not intersect the path $v_1, \ldots, q$. Note that q may be a vertex location or an interior point of some edge in Γ. Let a and b be two points in $\mathbb{R}^2$. Then a and b are *r-visible* to each other if and only if their exists a polygonal chain of length r with end points a, b that does not intersect Γ at any point except possibly at a, b. A point p *lies between two*

other points a, b, if either the inequality $x(a) < x(p) < x(b)$ or $x(b) < x(p) < x(a)$ holds.

A set of points is *monotone* if the line connecting them from left to right is monotone with respect to y-axis. Let S be a set of n points in general position. By the Erdös-Szekeres theorem [57], S can be partitioned into $2\lfloor\sqrt{n}\rfloor$ disjoint monotone subsets, and such a partition can be computed in $O(n^{1.5})$ time [7].

6.3 Drawing Thickness-t Graphs on t Layers

In this section we give two separate drawing techniques to draw thickness-t graphs on t layers. We first present a simple construction achieving $2.25n + O(1)$ upper bound (Section 6.3.1). Although the technique is simple, the idea of the construction will be used frequently in the rest of the chapter. Therefore, we explained the construction in reasonable details.

Later, we present a second construction (Section 6.3.2), which is more involved, and relies on a deep understanding of the geometry of point sets. In this case, the upper bound on the bend complexity will depend on some generalization of Erdös-Szekeres theorem [57], e.g., partitioning a point set into monotone subsequences in higher dimensions (Section 6.3.2.4).

6.3.1 A Construction with Bend Complexity $2.25n + O(1)$

In this section we give a simple technique to draw thickness-t graphs on t layers and with bend complexity $2.25n + O(1)$. We first give an overview of the algorithm, and then describe the algorithmic details.

6.3.1.1 Algorithm Overview

Let G be the input graph with thickness t. We assume that a partition of the input graph into t planar subgraphs is given. Although decomposing thickness-t graphs into t planar subgraphs is an NP-hard problem [91], one can decompose into at most $3t$ planar subgraphs in polynomial time [68].

Let $G_1, \ldots, G_t$ be the planar subgraphs of G, and let $P_1, \ldots, P_t$ be the corresponding spinal paths. Our algorithm constructs the drawing by placing the vertices of G on the boundary of a semicircle. The algorithm first computes the drawings of the spinal paths. Each spinal path is drawn by adding the edges one after another from one end of the path, as shown in Figure 6.3(a). Informally, each new edge is drawn as an x-monotone polyline, which either lies on the top the current stack of the drawn edges, as illustrated in Figure 6.3(b). The algorithm then adds the other edges that do not lie on the spinal path, as illustrated in Figure 6.3(c) using dashed lines. The bend complexity of this drawing may be large. In subsequent steps the algorithm modifies this drawing to improve the bend complexity.

6.3.1.2 Algorithm Details

Let $G_1, \ldots, G_t$ be the planar subgraphs of the input graph G, and let S be an ordered set of n points on a semicircular arc. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of G. We show that each G_i, where $1 \leq i \leq t$, admits a polyline drawing with bend complexity $2.25n + O(1)$ such that vertex v_j is mapped to the jth point of S. To draw G_i, we will use the vertex ordering of its monotone topological book

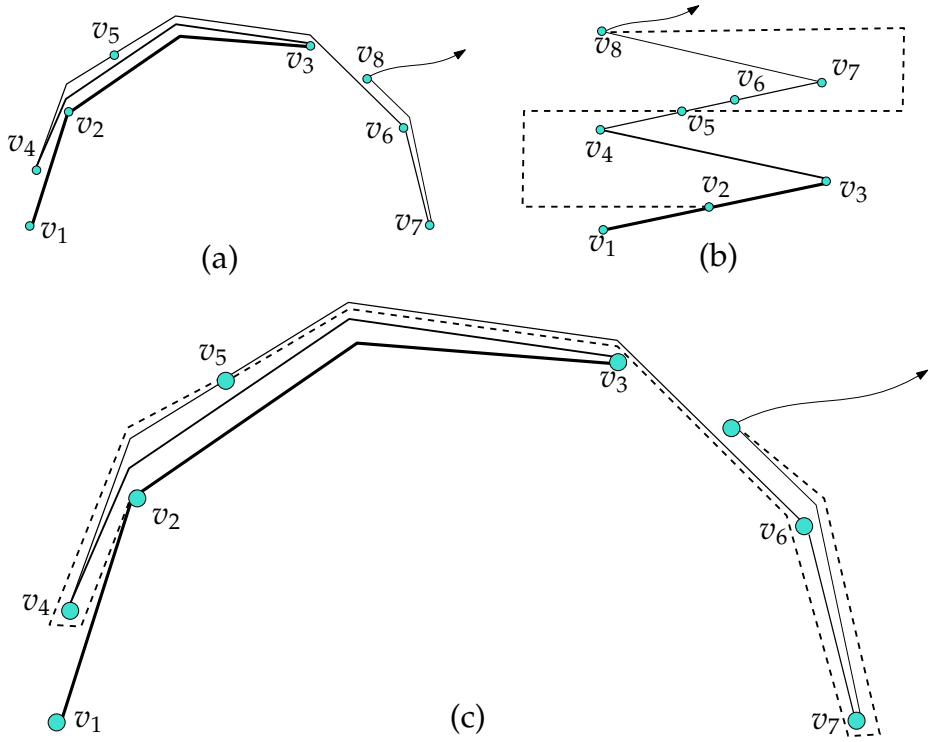


Figure 6.3: An overview of the algorithm.

embedding. The following lemma will be useful to draw the spinal path P_i of G_i.

Lemma 7 *Let $S = \{p_0, p_1, \ldots, p_{n+1}\}$ be a set of points lying on an x-monotone semi-circular arc, and let $P = \{v_1, v_2, \ldots, v_n\}$ be a path of n vertices. Assume that p_0 and p_{n+1} are the leftmost and rightmost points of S, respectively, and the points $p_1, \ldots, p_n$ are equally spaced between them in some random order. Then P admits an uphill drawing Γ with the vertex v_i assigned to p_i, where $1 \leq i \leq n$, and every point p_i satisfies the following properties:*

A. *Both the points p_0 and p_{n+1} are $(3n/4)$-visible to p_i.*

B. *One can draw an x-monotone polygonal chain from p_0 to p_{n+1} with $3n/4$ bends that intersects Γ only at p_i.*

Proof: We prove the lemma by constructing such a drawing Γ for P. The construction assigns a polyline for each edge of P. The resulting drawing may contain edge overlaps, and the bend complexity could be as large as $n-2$. Later we remove these degeneracies and reduce the bend complexity to obtain Γ.

Drawings of Edges: For each point $p_i \in S$, where $1 \leq i \leq n$, we create an *anchor point* p_i' at $(x(p_i), y(p_i)+\epsilon)$, where $\epsilon > 0$. We choose ϵ small enough such that for any j, where $1 \leq i \neq j \leq n$, all the points of S between p_i and p_j lie above (p_i', p_j'). Figure 6.4(a) illustrates this property for the anchor point p_1'.

We first draw the edge (v_1, v_2) using a straight line segment. For each j from 2 to $n-1$, we now draw the edges (v_j, v_{j+1}) one after another. Assume without loss of generality that $x(p_j) < x(p_{j+1})$. We call a point $p \in S$ between p_j and p_{j+1} a *visited point* if the corresponding vertex v appears in $v_1, \ldots, v_j$, i.e., v has already been placed at p. We draw an x-monotone polygonal chain L that starts at v_j, connects the anchors of the intermediate visited points from left to right, and ends at v_{j+1}. Figure 6.4(b) illustrates such a construction.

Since the number of bends on L is equal to the number of visited points of S between p_j and p_{j+1}, each edge contains at most α bends, where α is the number of points of S between p_j and p_{j+1}.

Removing Degeneracies: The drawing D_n of the path P constructed above contains edge overlaps, e.g., see the edges (v_3, v_4) and (v_4, v_5) in Figure 6.4(c). To remove the degeneracies, for each i, we spread the corresponding bend points between p_i and p_i', in the order they appear on the path, see Figure 6.4(d). Consequently, we obtain a planar drawing of P. Let the resulting drawing be D_n'. Since

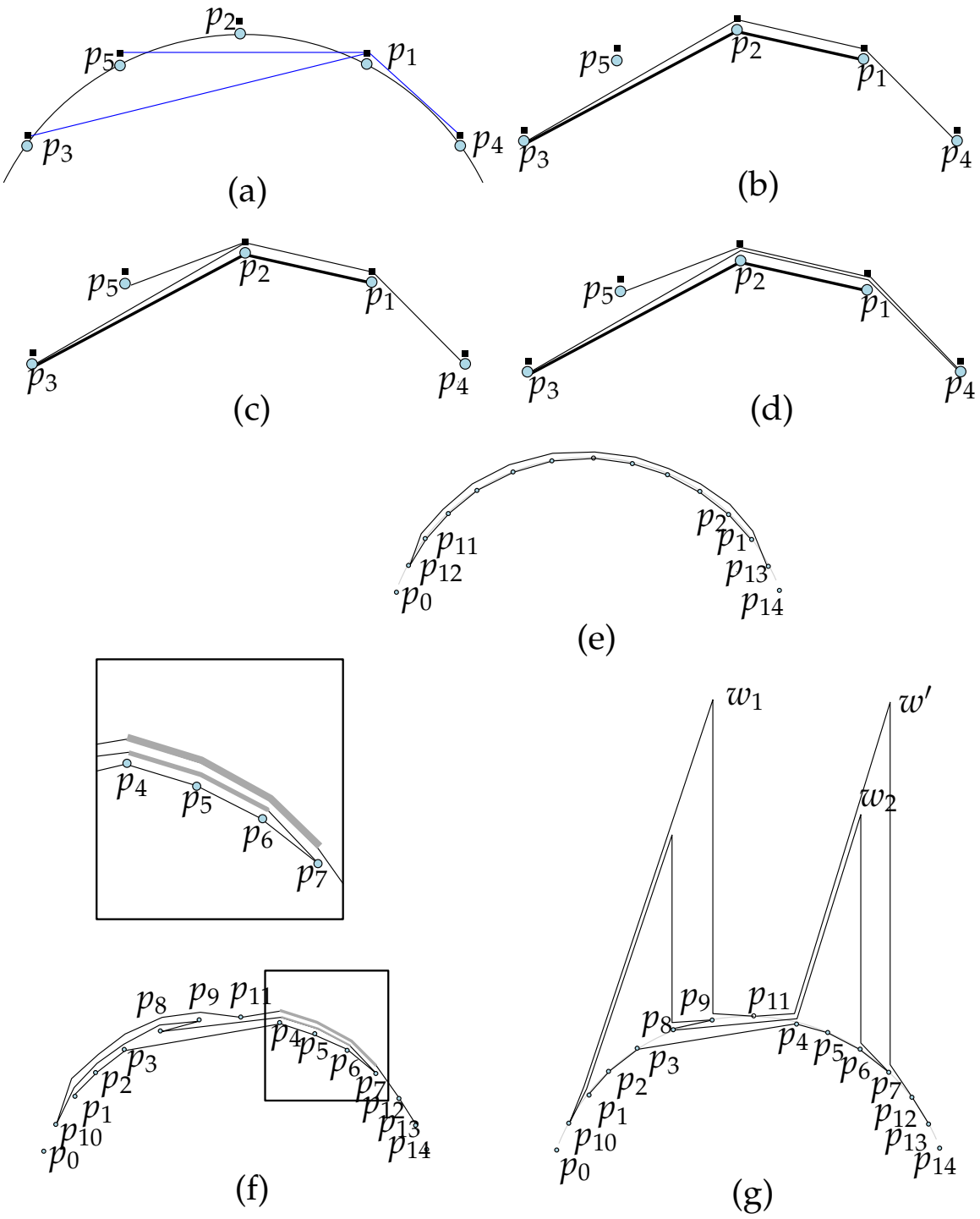


Figure 6.4: Illustration for the proof of Lemma 7. (a) Construction of the point set, and the anchor points. The anchor points are shown in black squares. (b)–(d) Construction of D'_n. (e) A scenario when the number of bends may be large. (f)–(g) Reducing bend complexity.

each edge (p_j, p_{j+1}) is drawn as an x-monotone polyline above the path $p_1, \ldots, p_j$, D'_n satisfies the uphill property. Note that D'_n may have bend complexity $n-2$, e.g., see Figure 6.4(e). We now show how to reduce the bend complexity and satisfy Properties A–B.

Reducing Bend Complexity: A pair of points in S are *consecutive* if they do not contain any other point of S in between. Let e be any edge of P. Let C_e be the corresponding polygonal chain in D'_n. A pair of bends on C_e are called *consecutive bends* if their corresponding points in S are also consecutive. A *bend-interval* of C_e is a maximal sequence of consecutive bends in C_e. Note that we can partition the bends on e into disjoint sets of bend-intervals.

For any bend-interval s, let $l(s)$ and $r(s)$ be the x-coordinates of the left and right endpoints of s, respectively. Let s_1 and s_2 be two bend-intervals lying on two distinct edges e_1 and e_2 in D'_n, respectively, where e_2 appears after e_1 in P. We claim that the intervals $[l(s_1), r(s_1)]$ and $[l(s_2), r(s_2)]$ are either disjoint, or $[l(s_1), r(s_1)] \subseteq [l(s_2), r(s_2)]$. We refer to this property as the *balanced parenthesis property of the bend-intervals*. To verify this property assume that for some s_1, s_2, we have $[l(s_1), r(s_1)] \cap [l(s_2), r(s_2)] \neq \phi$. Since s_2 is a maximal sequence of consecutive bends, the inequalities $l(s_2) \leq l(s_1)$ and $r(s_2) \geq r(s_1)$ hold, i.e., $[l(s_1), r(s_1)] \subseteq [l(s_2), r(s_2)]$. We say that s_1 *is nested by* s_2. Figure 6.4(f) illustrates such a scenario, where s_1, s_2 are shown in thin and thick gray lines, respectively.

We now consider the edges of P in reverse order, i.e., for each j from n to 2, we modify the drawing of $e = (v_j, v_{j-1})$. For each bend-interval $s = (b_1, b_2 \ldots, b_r)$ of C_e, if s has three or more bends, then we delete the bends $b_2, \ldots, b_{r-1}$, and join

b_1 and b_r using a new bend point w. To create w, we consider the two cases of the balanced parenthesis property.

If s is not nested by any other bend-interval in D'_n, then we place w high enough above b_r such that the chain b_1, w, b_r does not introduce any edge crossing, e.g., see the point $w_1(= w)$ in Figure 6.4(g). On the other hand, if s is nested by some other bend-interval, then let s' be such a bend-interval immediately above s. Since $s' = (b'_1, b'_2, \ldots, b'_r)$ is already processed, it must have been replaced by some chain b'_1, w', b'_r. Therefore, we can find a location for b inside $\angle b'_1 w' b'_r$ such that the chain b_1, w, b_r does not introduce any edge crossing, e.g., see the points w' and $w_2(= w)$ in Figure 6.4(g). Let the resulting drawing of P be Γ.

We now show that the above modification reduces the bend complexity to $3n/4$. Let e be an edge of P that contains α points from S between its endpoints. Let C_e be the corresponding polygonal chain in D'_n. Recall that any bend-interval of length ℓ in C_e contributes to $\min\{\ell, 3\}$ bends on e in Γ. Therefore, if there are at most $\alpha/4$ bend-intervals on C_e, then e can have at most $3\alpha/4$ bends in Γ. Otherwise, if there are more than $\alpha/4$ bend-intervals, then there are at least $\alpha/4$ points[1] of S that do not contribute to bends on C_e. Therefore, in both cases, C_e can have at most $3\alpha/4$ bends in Γ.

Satisfying Properties A–B: Let p_i be any point of $S \setminus \{p_0, p_{n+1}\}$. We first show that p_0 is $(3n/4)$-visible to q. Consider the drawing D_i. Observe that one can insert an edge (p_0, p_i) using an x-monotone polyline L such that the bends on L corresponds to the intermediate visited points. Now the drawing of the rest of the path $v_i, v_{i+1}, \ldots, v_n$ can be continued such that it does not cross L. Therefore,

[1]Every pair of consecutive bend-intervals contain such a point in between.

if the number of points of S between p_0 and p_i is α, then L has at most α bends. Finally, the process of reducing bend complexity improves the number of bends on L to $3\alpha/4$. Similarly, we can observe that p_{n+1} is at most $3\alpha'/4$ visible to p_i, where α' is the number of points of S between p_i and p_{n+1}. Since the edges (p_0, p_i) and (p_i, p_{n+1}) are x-monotone, we can draw an x-monotone polygonal chain from p_0 to p_{n+1} with at most $3(\alpha + \alpha')/4 \leq (3n/4)$ bends that intersects Γ only at p_i. ■

Theorem 9 *Every n-vertex graph of thickness t admits a drawing on t layers with bend complexity* $2.25n + O(1)$.

Proof: Let $G_1, \ldots, G_t$ be the planar subgraphs of the input graph G, and let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of G. let $S = \{p_0, p_1, \ldots, p_{n+1}\}$ be a set of $n + 2$ points lying on a semicircular arc as defined in Lemma 7. Let P_i be spinal path of the monotone topological book embedding of G_i, where $1 \leq i \leq t$. We first compute an uphill drawing Γ_i of the path P_i. We then draw the edges of G_i that do not belong to P_i. Let $e = (u, v)$ be such an edge, and without loss of generality assume that u appears to the left of v on the spine.

If e lies above (resp., below) the spine, then we draw two x-monotone polygonal chains; one from u to p_0 (resp., p_{n+1}), and the other from v to p_0 (resp., p_{n+1}). By Lemma 7, these polygonal chains do not intersect Γ_i except at u and v, and each contains at most $3n/4$ bends. Hence e contains at most $1.5n$ bends in total.

If e crosses the spine, then it crosses some edge (w, w') of P_i. Draw the edges (u, w) and (w, v) using the polylines $u, \ldots, p_0, \ldots, w$ and $w, \ldots, p_{n+1}, \ldots, v$, respectively. The polylines $u, \ldots, p_0$ and $p_{n+1}, \ldots, v$ are x-monotone, and have at

most $3n/4$ bends each. The polyline $C = (p_0, \ldots, w \ldots, p_{n+1})$ is also x-monotone and has at most $3n/4$ bends. Hence the number of bends is $2.25n$ in total. It is straightforward to avoid the degeneracy at w, by adding a constant number of bends on C.

Note that we still have some edge overlaps at p_0 and p_{n+1}. It is straightforward to remove these degeneracies by adding only a constant number of more bends per edge. ■

6.3.2 A Construction with Improved Bend Complexity

In this section we give another construction to draw thickness-t graphs on t layers. We first show that every thickness-t graph, where $t \in \{3, 4\}$, can be drawn on t layers with bend complexity $O(\sqrt{n})$, and then show how to extend the technique for larger values of t.

6.3.2.1 Algorithm Overview

The construction here is similar to the previous section, but instead of placing all the points on a semicircle, we create the point set based on the spinal paths to be drawn. Besides, the points are partitioned into subgroups, where the points on the same group are contagious along the x-axis. While drawing a path, each edge may span several groups. Figure 6.5 shows a schematic representation of such a scenario. We show that passing through a group requires constant number of bends. Therefore, the number of bends correspond to the number of groups in the point set. For fixed t, we prove a sublinear upper bound on the number of

groups, which keep the bend complexity small.

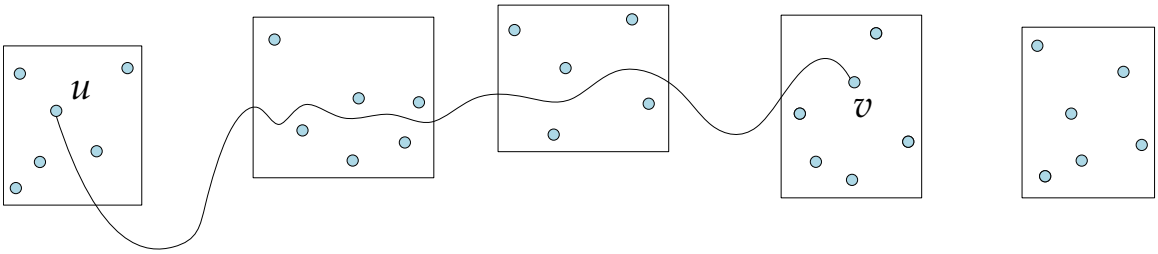


Figure 6.5: Overview of the Algorithm. The edge (u, v) spans four groups.

6.3.2.2 Construction when $t = 3$

Let S be an ordered set of n points, where the ordering is by increasing x-coordinate. A (k, n)-*group* $S_{k,n}$ is a partition of S into k disjoint ordered subsets $\{S_1, \ldots, S_k\}$, each containing contiguous points from S. Label the points of S using a permutation of $p_1, p_2, \ldots, p_n$ such that for each set $S' \in S_{k,n}$, the indices of the points in S' are either increasing or decreasing. If the indices are increasing (resp., decreasing), then we refer S' as a rightward (resp., leftward) set. We will refer to such a labelling as a *smart labelling* of $S_{k,n}$. Figure 6.6 illustrates a $(5, 23)$-group and a smart labelling of the underlying point set $S_{5,23}$.

Note that for any i, where $1 \leq i \leq n$, deletion of the points $p_1, \ldots, p_i$ removes the points of the rightward (resp., leftward) sets from their left (resp., right). The *necklace* of $S_{k,n}$ is a path obtained from a smart labelling of $S_{k,n}$ by connecting the points p_i, p_{i+1}, where $1 \leq i \leq n-1$. The following lemma constructs an uphill drawing of the necklace using $O(k)$ bends per edge.

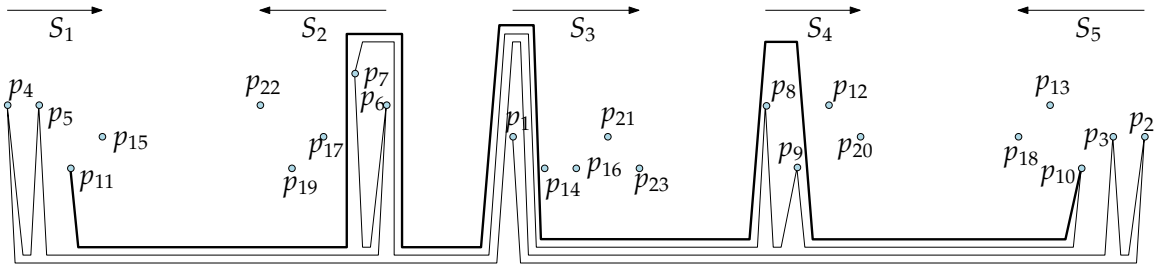


Figure 6.6: Illustration for the proof of Lemma 8. The edge (p_{10}, p_{11}) is shown in bold. Passing through each intermediate set requires at most 4 bends.

Lemma 8 *Let S be a set of n points ordered by increasing x-coordinate, and let $S_{k,n} = \{S_1, \ldots, S_k\}$ be a (k, n)-group of S. Label $S_{k,n}$ with a smart labelling. Then the necklace of $S_{k,n}$ admits an uphill drawing with $O(k)$ bends per edge.*

Proof: We construct this uphill drawing incrementally in a similar way as in the proof of Lemma 7. Let D_j, where $1 \leq j \leq n$, be the drawing of the path $p_1, \ldots, p_j$. At each step of the construction, we maintain the invariant that D_j is an uphill drawing.

We first assign v_1 to p_1. Then for each i from 1 to $n-1$, we draw the edge (p_i, p_{i+1}) using an x-monotone polyline L that lies above D_i and below the points $p_{j'}$, where $j' > i+1$. Figure 6.6 illustrates such a drawing of (p_i, p_{i+1}).

The crux of the construction is that one can draw such a polyline L using at most $O(k)$ bends. Assume that p_i and p_{i+1} belong to the sets $S_l \in S_{k,n}$ and $S_r \in S_{k,n}$, respectively. If S_l and S_r are identical, then p_i and p_{i+1} are consecutive, and hence it suffices to use at most $O(1)$ bends to draw L. On the other hand, if S_l and S_r are distinct, then there can be at most $k-2$ sets of $S_{k,n}$ between them. Let S_m be such a set. While passing through S_m, we need to keep the points that already belong to the path, below L, and the rest of the points above L. By the

property of smart labelling, the points that belong to D_i are consecutive in S_m, and lie to the left or right side of S_m depending on whether S_m is rightward or leftward. Therefore, we need only $O(1)$ bends to pass through S_m. Since there are at most $k-2$ sets between S_l and S_r, $O(k)$ bends suffice to construct L. ∎

We are now ready to describe the main construction. Let G be an n-vertex thickness-3 graph, and let G_1, G_2, G_3 be the planar subgraphs of G. Let P_i be the spinal path of the monotone topological book embedding of G_i, where $1 \leq i \leq 3$. We first create a set of n points and assign them to the vertices of G. Later we route the edges of G.

Creating Vertex Locations: Assume without loss of generality that $P_1 = (v_1, \ldots, v_n)$. For each i from 1 to n, we place a point at (i, j') in the plane, where j' is the position of v_j in P_2. Let the resulting point set be Q. Recall that Q can be partitioned into disjoint monotone subsets $Q_1, \ldots, Q_k$, where $k \leq O(\sqrt{n})$ [7]. Figure 6.7(a) illustrates such a partition.

The sets $Q_1, \ldots, Q_k$ are ordered by the x-coordinate, and the indices of the labels of the points at each set is either increasing or decreasing. Therefore, if we place the points of the ith set between the lines $x = 2(i-1)n$ and $x = (2i-1)n$, then the resulting point set Q' would be a (k, n)-group, labelled by a smart labelling. Finally, we adjust the y-coordinates of the points according to the position of the corresponding vertices in P_3. Let the resulting point set be S. Figure 6.7(b) illustrates the vertex locations, where $P_1 = (v_1, v_2, \ldots, v_n)$, $P_2 = (v_{11}, v_1, \ldots, v_3)$, and $P_3 = (v_6, v_{11}, \ldots, v_{10})$.

Edge Routing: It is straightforward to observe that the path P_1 is a necklace

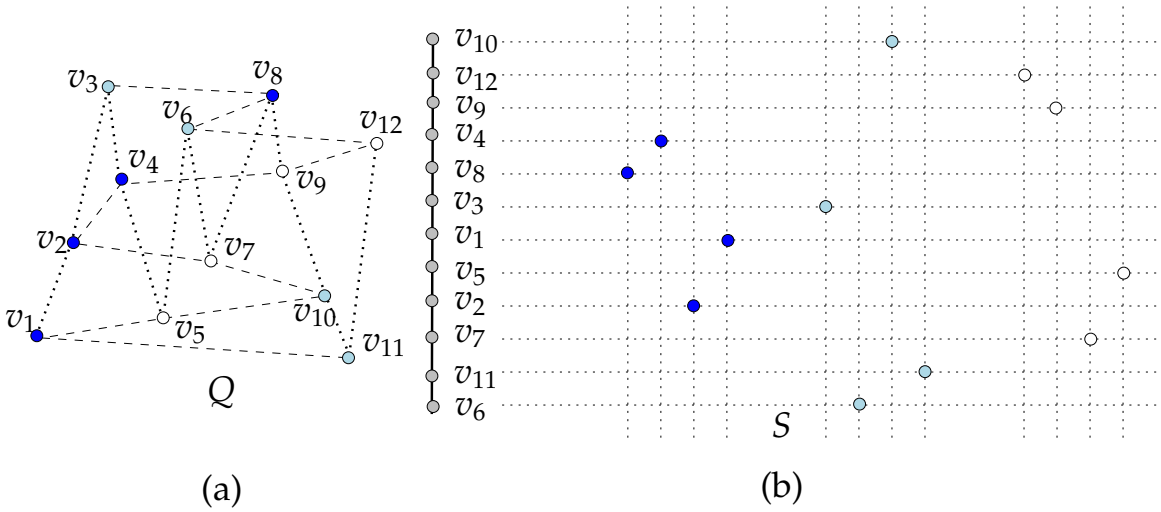


Figure 6.7: Creating vertex locations for drawing thickness-3 graphs, where P_1, P_2 and P_3 are shown in dotted, dashed and thick solid lines, respectively.

for the current labelling of the points of $S_{k,n}$. Therefore, by Lemma 8, we can construct an uphill drawing of P_1 on S. Observe that for every set $S' \in S_{k,n}$, the corresponding points are monotone in Q, i.e., the points of S' are ordered along the x-axis either in increasing or decreasing order of their y-coordinates in Q. Therefore, relabelling the points according to the increasing order of their y-coordinates in Q will produce another smart labelling of S, and the corresponding necklace would be the path P_2. Therefore, we can use Lemma 8 to construct an uphill drawing of P_2 on S. Since the height of the points of S are adjusted according to the vertex ordering on P_3, connecting the points of S from top to bottom with straight line segments yields a y-monotone drawing of P_3.

We now route the edges of G_i that do not belong to P_i, where $1 \leq i \leq 3$. Since P_3 is drawn as a y-monotone polygonal path, we can use the technique of Erten and Kobourov [58] to draw the remaining edges of G_3. To draw the edges of G_2, we insert two points p_0 and p_{n+1} to the left and right of all the

points of S, respectively. Then the drawing of the remaining edges of G_1 and G_2 is similar to the edge routing described in the proof of Theorem 9. That is, if the edge $e = (u, v)$ lies above (resp., below) the spine, then we draw it using two x-monotone polygonal chains from p_0 (resp., p_{n+1}). Otherwise, if e crosses the spine, then we draw three x-monotone polygonal chains, one from u to p_0, another from p_0 to p_{n+1}, and the third one from v to p_{n+1}. Since $k \leq O(\sqrt{n})$, the number of bends on e is $O(\sqrt{n})$. Finally, we remove the degeneracies, which increases the bends per edge by a small constant.

6.3.2.3 Construction when $t = 4$

We now show that the technique for drawing thickness-3 graphs can be generalized to draw thickness-4 graphs with the same bend complexity.

Let $G_1, \ldots, G_4$ be the planar subgraphs of G, and let $P_1, \ldots, P_4$ be the corresponding spinal paths. While constructing the vertex locations, we use a new y-coordinate assignment for the points of S. Instead of placing the points according to the vertex ordering on the path P_3, we create a particular order that would help to construct uphill drawings of P_3 and P_4 with bend complexity $O(\sqrt{n})$. That is, we first create a (k', n)-group $S'_{k',n}$ using P_3 and P_4, where $k' \leq O(\sqrt{n})$, in a similar way we created $S_{k,n}$ using P_1 and P_2. We then adjust the y-coordinates of the points of S according to the order these points appear in $S'_{k',n}$.

Here we illustrate the construction of the point set. Figure 6.8(a) illustrates P_1 and P_2 in black and gray, respectively. Figure 6.8(b) illustrates P_3 and P_4 in black and gray, respectively.

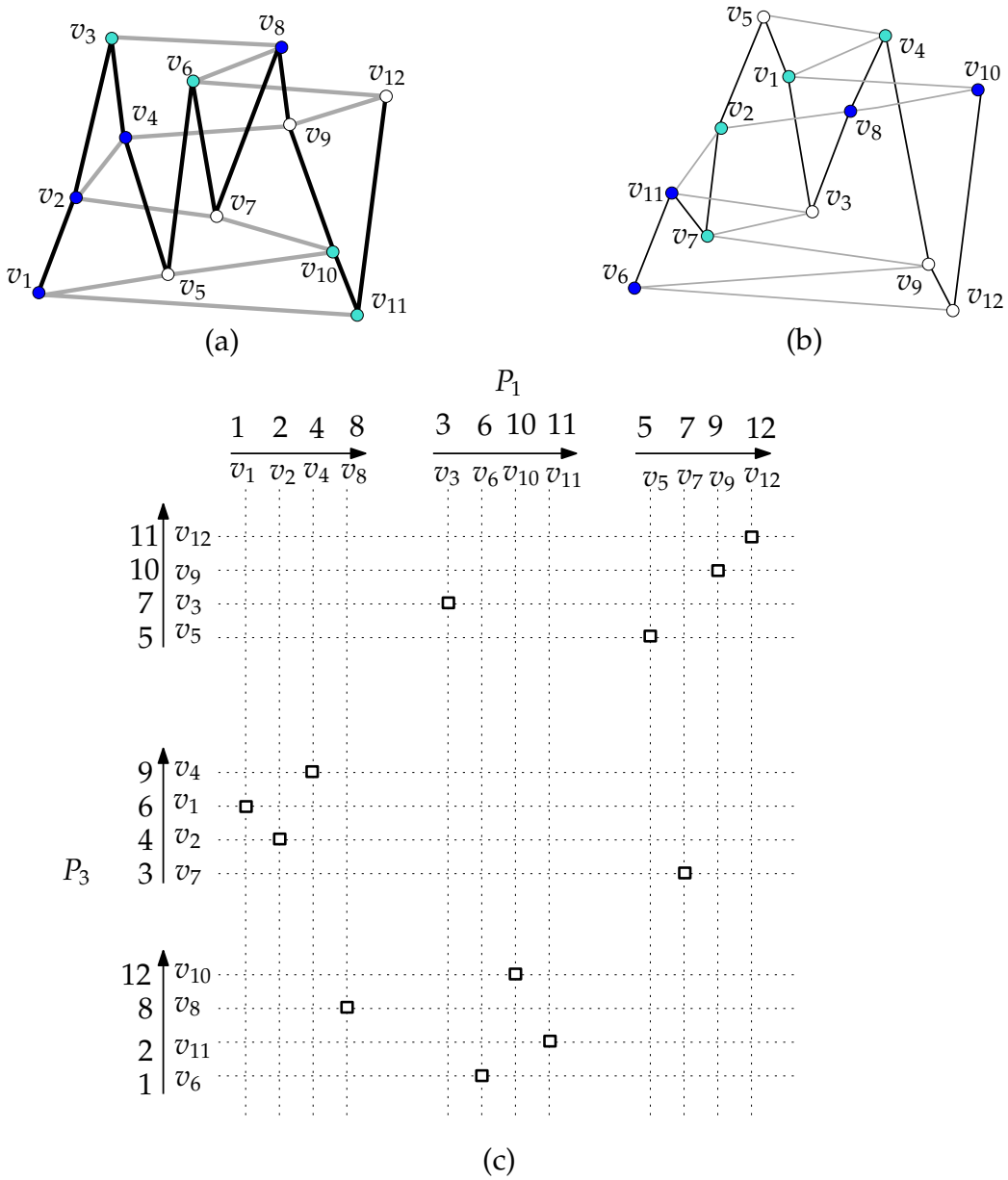


Figure 6.8: (a) A point set, constructed from the paths P_i, where $i \in \{1,2\}$, by placing each vertex v at $(\delta_1(v), \delta_2(v))$. Here $\delta_i(v)$ is the position of v on P_i. (b) A point set, constructed from the paths P_i, where $i \in \{3,4\}$, by placing each vertex v at $(\delta_3(v), \delta_4(v))$. (c) The final point set, and the corresponding (k,n)-groups. The numbers denote the vertex positions on the corresponding spinal path. The arrows illustrate whether the corresponding sets are leftward or rightward.

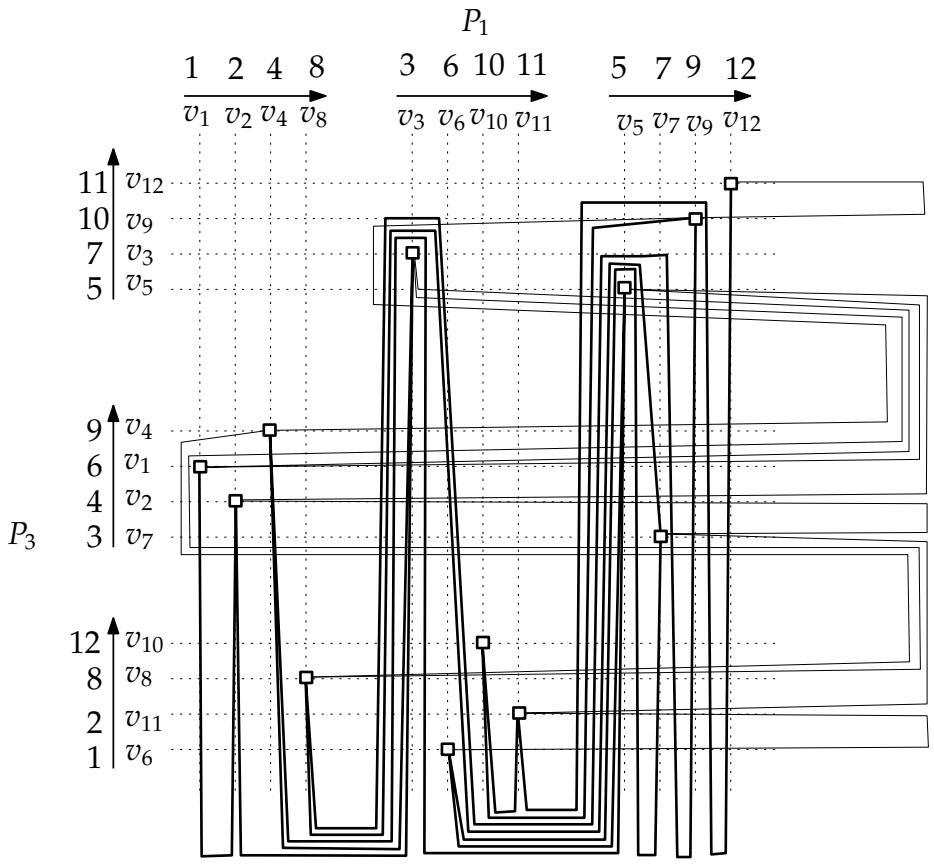


Figure 6.9: Drawings of P_1 and P_3 on the point set of Figure 6.8(c).

The construction of G_1 and G_2 remains the same as described in the previous section. However, since P_3 and P_4 now admit uphill drawings on S with respect to y-axis, the drawing of G_3 and G_4 are now analogous to the construction of G_1 and G_2.

6.3.2.4 Construction when $t > 4$

De Bruijn [85] observed that the result of Erdös-Szekeres [57] can be generalized to higher dimensions. Given a sequence ρ of n tuples, each of size κ, one can find a subsequence of at least $n^{1/\lambda}$ tuples, where $\lambda = 2^{\kappa}$, such that they are

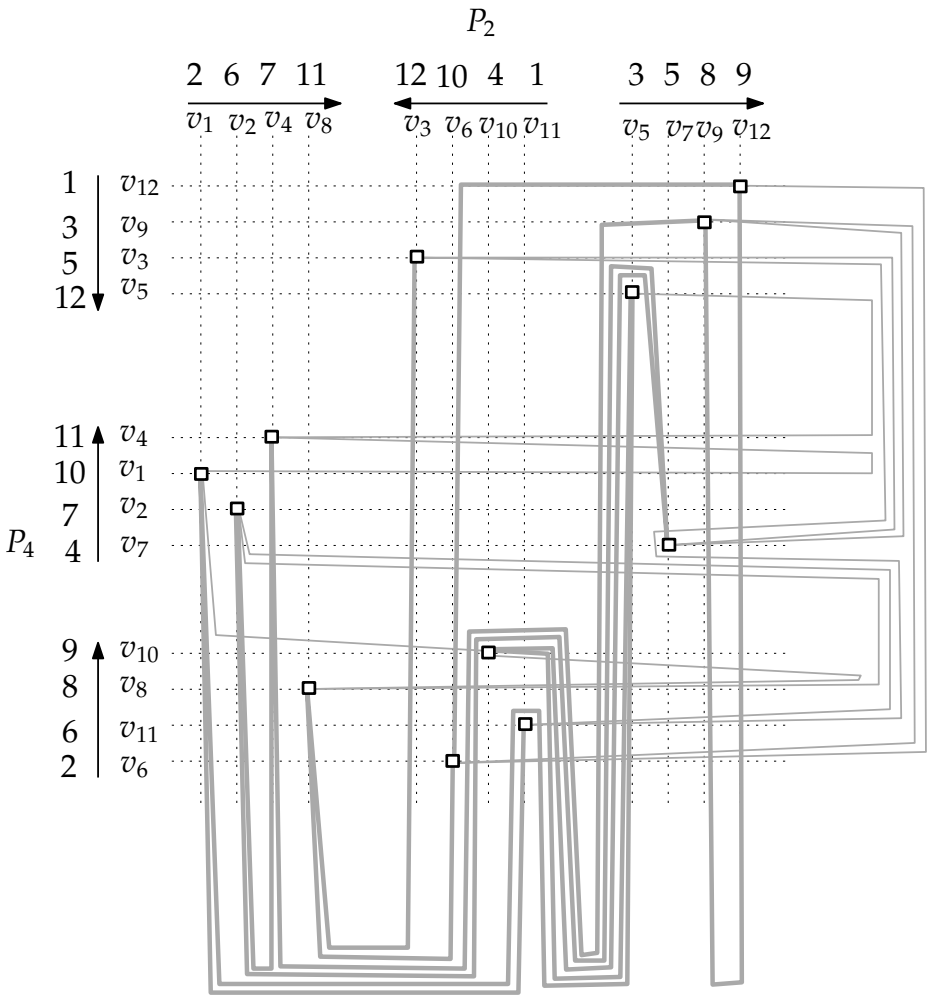


Figure 6.10: Drawings of P_2 and P_4 on the point set of Figure 6.8(c).

monotone (i.e., increasing or decreasing) in every dimension. This result is a repeated application of Erdös-Szekeres result [57] at each dimension. We now show how to partition ρ into few monotone sequences.

We use the partition algorithm of Bar-Yehuda and Fogel [7] that partitions a given sequence of n numbers into at most $2\sqrt{n}$ monotone subsequences. It is straightforward to restrict the size of the subsequences to $\sqrt{n}$, without increasing the number of subsequences, i.e., by repeatedly extracting a monotone sequence

of length exactly $\sqrt{n}$. Consequently, one can partition ρ into $2\sqrt{n}$ subsequences, where each subsequence is of length $\sqrt{n}$, and monotone in the first dimension. By applying the partition algorithm on each of these subsequences, we can find $2\sqrt{n} \cdot 2\sqrt{\sqrt{n}}$ subsequences, each of which is of length $\sqrt{\sqrt{n}}$, and monotone in the first and second dimensions. Therefore, after κ steps, we obtain a partition of ρ into $2^\kappa \cdot (n^{1/2} \cdot n^{1/4} \cdot \ldots \cdot n^{1/2^\kappa} = 2^\kappa \cdot n^{1-(1/\lambda)}$ monotone subsequences, where $\lambda = 2^\kappa$. We use this idea to extend our drawing algorithm to higher thickness.

Let $G_1, \ldots, G_t$ be the planar subgraphs of G, and let $P_1, \ldots, P_t$ be the corresponding spinal paths. Let $v_1, v_2, \ldots, v_n$ be the vertices of G. Construct a corresponding sequence $\rho = (\tau_1, \tau_2, \ldots, \tau_n)$ of n tuples, where each tuple is of size t, and the ith element of a tuple τ_j corresponds to the position of the corresponding vertex v_j in P_i, where $1 \leq i \leq t$ and $1 \leq j \leq n$. We now partition ρ into a set of $2^t \cdot n^{1-1/\beta}$ monotone subsequences, where $\beta = 2^t$.

For each of these monotone sequences, we create an ordered set of consecutive points along the x-axis, where the vertex v_j corresponds to the point p_j. It is now straightforward to observe that these sets correspond to a (k, n)-group $S_{k,n}$, where $k \leq 2^t \cdot n^{1-1/\beta}$. Furthermore, since each group corresponds to a monotone sequence of tuples, for each P_i, the positions of the corresponding vertices are either increasing or decreasing. Hence, every path P_i corresponds to a necklace for some smart labelling of $S_{k,n}$. Therefore, by Lemma 8, we can construct an uphill drawing of P_i on S. We now add the remaining edges of G_i following the construction described in Section 6.3.2.2. Since $k \leq 2^t \cdot n^{1-1/\beta}$, the number of bends is bounded by $O(2^t \cdot n^{1-1/\beta})$.

Observe that all the points in the above construction have the same y-coordinate. Therefore, we can improve the construction by distributing the load equally among the x-axis and y-axis as we did in Section 6.3.2.3. Specifically, we draw the graphs $G_1, \ldots, G_{\lceil t/2 \rceil}$ using the uphill drawings of their spinal paths with respect to the x-axis, and the remaining graphs using the uphill drawings of their spinal paths with respect to the y-axis. Consequently, the bend complexity decreases to $O(2^{t/2} \cdot n^{1-1/\beta'})$, where $\beta' = 2^{\lceil t/2 \rceil}$.

We can improve this bound further by observing that we are free to choose any arbitrary vertex labelling for G while creating the initial sequence of tuples. Instead of using an arbitrary labelling, we could label the vertices according to their ordering on some spinal path, which would reduce the bend complexity to $O(2^{(t-2)/2} \cdot n^{1-1/\beta''})$, where $\beta'' = 2^{\lceil (t-2)/2 \rceil}$.

Theorem 10 *Every n-vertex graph G of thickness $t \geq 3$ admits a drawing on t layers with bend complexity $O(\sqrt{2}^t \cdot n^{1-(1/\beta)})$, where $\beta = 2^{\lceil (t-2)/2 \rceil}$.*

6.4 Trade-off Between Layers and Bends

In the previous sections we have seen that thickness-t graphs admit polyline drawings on t layers with bend complexity $\min\{2.25n + O(1), O(2^{t/2} \cdot n^{1-1/\beta})\}$. The constant hidden in O-notation is small, specifically, for the bound $O(2^{t/2} \cdot n^{1-1/\beta})$, it is at most 6.

For fixed t, the bound $O(2^{t/2} \cdot n^{1-1/\beta})$ is asymptotically smaller than $2.25n + O(1)$. However, for smaller values of n, the bound of $6 \cdot 2^{t/2} \cdot n^{1-1/\beta}$ dominates

the other bound only for small values of t. Figures 6.11(a) and (b) illustrate the cases when $t = 5$ and $n = 10^6$, respectively.

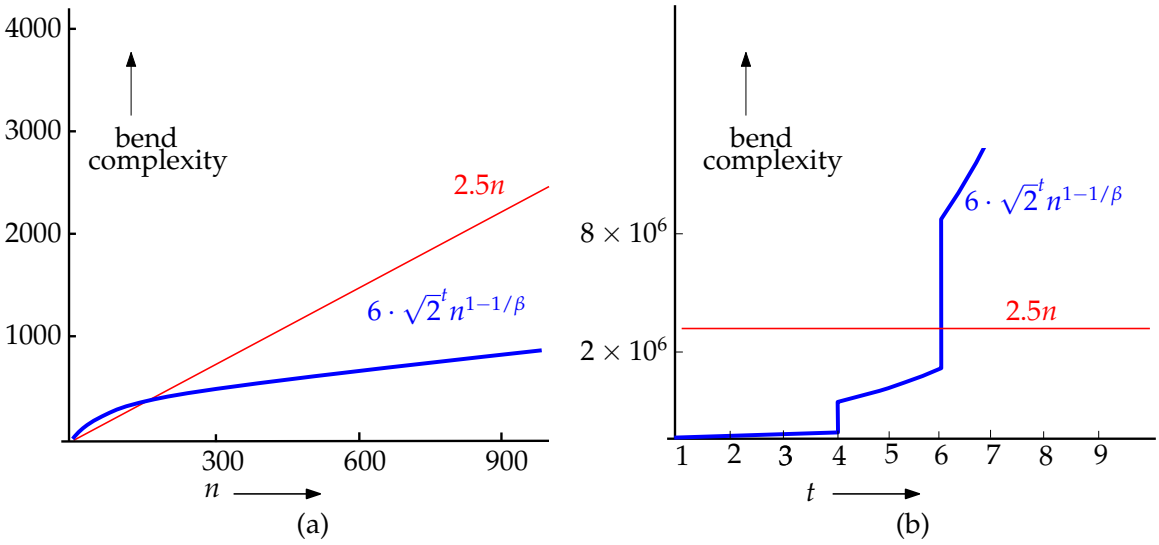


Figure 6.11: Plots for the bounds on bend complexity: (a) $t = 5$, and (b) $n = 10^6$.

We now show that allowing $k \geq t$ layers, where k is the linear arboricity of the input graph, may reduce the bend complexity to $\frac{3(k-1)n}{(4k-2)} < 0.75n$. To construct polyline drawings, where the layer number and bend complexities are functions of the linear arboricity of the input graph, we use the concept of bandwidth. We prove that the bandwidth of a graph can be bounded in terms of its linear arboricity and the number of vertices, and then the result follows from an application of Lemma 7.

The following lemma proves an upper bound on the bandwidth of graphs.

Lemma 9 *Given an n-vertex graph $G = (V, E)$ with linear arboricity k, the bandwidth of G is at most $\frac{3(k-1)n}{(4k-2)}$.*

Proof: Without loss of generality assume that G is a union of k spanning paths $P_1, \ldots, P_k$. For any ordered sequence σ, let $\sigma(i)$ be the element at the ith position, and let $|\sigma|$ be the number of elements in σ. We now construct an ordered sequence $\sigma = \sigma_1 \circ \sigma_2 \circ \ldots \circ \sigma_k \circ \sigma_{k+1}$ of the vertices in V, as follows.

σ_1: We initially place the first x vertices of P_1 in the sequence, where the exact value of x is to be determined later.

σ_2: We then place the vertices that are neighbors of σ_1 in P_2, in order, i.e., we first place the neighbors of $\sigma_1(1)$, then the neighbors of $\sigma_1(2)$ that have not been placed yet, and so on.

σ_i: For each $i = 3, \ldots, k$, we place the vertices that are neighbors of σ_1 in P_i in order.

σ_{k+1}: We next place the remaining vertices of P_1 in order.

Figure 6.12(a) illustrates an example for three paths with $x = 2$. Observe that $|\sigma_1| \leq x$, and $|\sigma_t| \leq 2x$, where $1 < t \leq k$. We now compute an upper bound on the bandwidth of G using the vertex ordering of σ.

For any i, j, where $1 \leq i < j \leq k+1$, let $\sigma_{i,j}$ be the sequence $\sigma_i \circ \ldots \circ \sigma_j$. The edges of P_1 that are in σ_1 have bandwidth 1, and those that are in $\sigma_1(x) \circ \sigma_{2,k+1}$ have bandwidth at most $(n - x)$, e.g., see Figure 6.12(b). Now let (v, w) be an edge of G that does not belong to P_1. We compute the bandwidth of (v, w) considering the following cases.

Case 1. If none of v and w belongs to σ_1, then the bandwidth of (v, w) is at most $(n - x)$.

Case 2. If both v and w belong to σ_1, then the bandwidth of (v, w) is at most x.

Case 3. If at most one of v and w belongs to σ_1, then without loss of generality assume that v belongs to σ_1. Since (v, w) does not belong to P_1, we may assume that w belongs to the path P_t, where $1 < t \leq k$. By the construction of σ, w belongs to $\sigma_{1,t}$, e.g., see Figure 6.12(b). Without loss of generality assume that w belongs to σ_r, where $1 < r \leq t$. Let v be the qth vertex in the sequence σ. Then the position of w cannot be more than $q + 2x \cdot (r-2) + 2q$, where the term $2x \cdot (r-2)$ corresponds to the length of $\sigma_2 \circ \ldots \circ \sigma_{r-1}$. Therefore, the bandwidth of the edge (v, w) is at most $2x \cdot (r-2) + 2q \leq 2x(r-1) \leq 2x(t-1)$.

Observe that the bandwidth of the edges of P_1 is upper bounded by $(n - x)$. The bandwidth of any edge that belongs to P_t, where $1 < t \leq k$ is at most $\max\{n - x, 2x(k-1)\}$. Consequently, the bandwidth of G is at most $\max\{n - x, 2x(k-1)\} \leq \frac{(2k-2)n}{(2k-1)}$, where $x = \frac{n}{(2k-1)}$. ∎

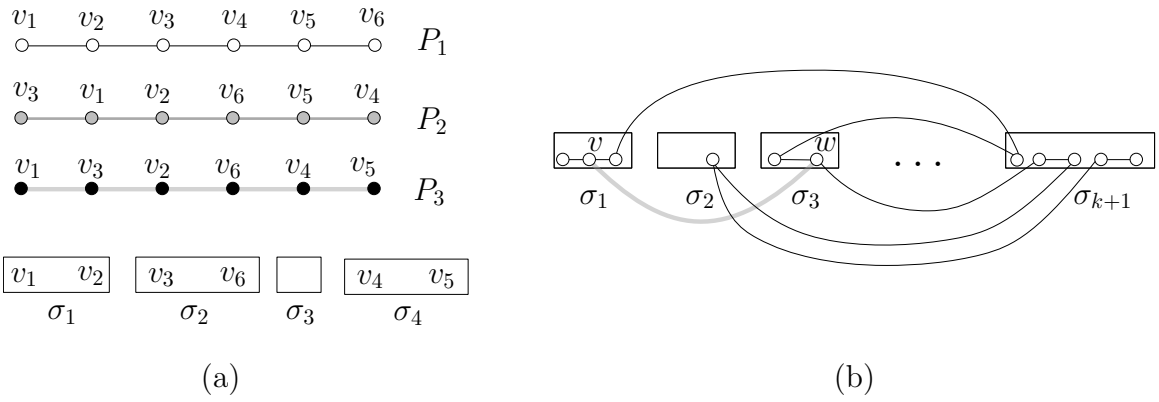


Figure 6.12: (a) Construction of σ. (b) A schematic representation of P_1 and (v, w), where (v, w) belongs to P_3.

The following theorem is immediate from the proof of Lemmas 7 and 9.

Theorem 11 *Every n-vertex graph with linear arboricity k can be drawn on k layers with at most $\frac{3(k-1)n}{(4k-2)} < 0.75n$ bends per edge.*

6.5 Summary and Open Questions

In this chapter we have developed algorithms to draw graphs on few planar layers and with low bend complexity. A natural direction for future research would be to improve the upper bounds on bend complexity while drawing thickness t-graphs on t layers. While developing the algorithms, we showed that the bandwidth of graphs with linear arboricity k is at most $\frac{3(k-1)n}{(4k-2)}$, and there exist graphs with linear arboricity two with bandwidth at least $n/6 - 3$. Therefore, it would be of independent interest to find tight bound on bandwidth of a graph in terms of its linear arboricity and number of vertices.

Open Problem 6.1. Find tight upper bound on the bandwidth of a graph in terms of its number of vertices and linear arboricity.

We believe our upper bounds on bend complexity to be nearly tight, but we require more evidence to support this intuition. The only related lower bound is that of Pach and Wenger [102], who showed that given a planar graph G and a unique location to place each vertex of G, $\Omega(n)$ bends are sometimes necessary to construct a planar polyline drawing of G with the given vertex locations. Therefore, a challenging research direction would be to prove tight lower bounds on the bend complexity while drawing thickness-t graphs on t layers.

Open Problem 6.2. Can we draw every thickness-t graph, where $t > 2$, on t planar layers with bend complexity $o(\sqrt{n})$?

Although our algorithms do not construct drawings with integral coordinates, it is straightforward to see that these drawings can also be constructed on polynomial-size integer grids, where all vertices and bends have integral coordinates. We leave the task of finding compact grid drawings achieving the same upper bounds as a direction for future research.

There exist some negative results that imply interesting lower bounds on the layer complexity of straight-line drawings. Eppstein [56] proved that geometric thickness of a graph is not bounded by the thickness of the graph. Specifically, for any t, he constructed a thickness-3 graph whose geometric thickness is at least t. Thus the layer complexity of the straight-line drawings of a graph G may not be bounded by the thickness of G. However, the case when the thickness of G is 2 is open [56].

Open Problem 6.3.[Eppstein [56]] Can the geometric thickness of a thickness-2 graph be unbounded?

Geometric thickness of a graph with maximum degree Δ is at most $\lceil \Delta/2 \rceil$ [74, 116], and this bound is tight [112]. Eppstein [56] asked whether the geometric thickness of a graph is also bounded. Barát et al. [8] answered the question in the negative by proving that for every $\Delta \geq 9$ and sufficiently large n, there exists a Δ-regular graph with geometric thickness at least $c\sqrt{\Delta}n^{1/2-4/\Delta-\epsilon}$. Consequently, the layer complexity of the straight-line drawings of a graph is not bounded by its

maximum degree. Since the geometric thickness of graphs with maximum degree four is two [44], the case when $\Delta \in \{5, 6, 7, 8\}$ is open.

Open Problem 6.4.[Barát et al. [8]] Can the geometric thickness of graphs with maximum degree Δ be unbounded, where $\Delta \in \{5, 6, 7, 8\}$?

It would be interesting to examine whether similar negative results could be established in the presence of edge bends. For example, can we prove that for every integer $\kappa > 0$, there exists a graph that cannot be drawn with bend complexity κ in a bounded number of layers? Does every graph G with maximum degree Δ admit a polyline drawing such that the bend complexity and layer complexity are both bounded by a function of Δ?

Chapter 7

Drawing Complete Graphs on Few Planes

This chapter focuses on drawing complete graphs on few layers, where the optimization goal is to minimize the sum of the number of edge crossings in all layers. Since the edges are drawn with curves, we can use arbitrary vertex positions in each layer[1]. These drawings are also known as k-planar drawings [29, 30], where k denotes the number of layers in the drawing. Each plane (i.e., layer) may contain a nonplanar subgraph. The term 'k-planar' merely extends the notion of the Euclidean plane.

Let G be an arbitrary graph. The *crossing number of* G is the smallest integer, denoted by $cr(G)$, such that G admits a drawing with $cr(G)$ edge crossings. The k-planar crossing number $cr_k(G)$ of G is $\min\{cr(G_1) + cr(G_2) + \ldots + cr(G_k)\}$, where the minimum is taken over all possible decompositions of G into k subgraphs

[1]Using the results of Chapter 6 it is straightforward to redraw each layer such that each vertex is mapped to the same point in all the layers, but the number of edge crossings remains the same.

$G_1, G_2, \ldots, G_k$.

The problem of computing the crossing number of complete graphs K_n has been extensively studied in the literature [73, 72, 119]. In this chapter we examine $cr_2(K_n)$, i.e., the biplanar crossing number of complete graphs. Since 1971, Owens' construction [99] has been the best known construction for biplanar drawings of K_n. We propose an improved technique for constructing biplanar drawings, which reduces Owens' upper bound on $cr_2(K_n)$. For small fixed n, it was known previously that $cr_2(K_n) = 0$ for $n \leq 8$ and $cr_2(K_9) = 1$; we show that $cr_2(K_{10}) = 2$, $cr_2(K_{11}) \in \{4, 5, 6\}$, and for $n \geq 12$, we improve previous upper and lower bounds on $cr_2(K_n)$.

We first review the motivation behind drawing graphs on few planes. We then present the related research and discuss our contribution. The subsequent sections present the technical background (Section 7.2), the drawing technique (Sections 7.3–7.4), limitations of our approach and directions to future research (Section 7.5).

Some of the results of this chapter appeared in preliminary form at the 28th Canadian Conference on Computational Geometry [54].

7.1 Drawings on Few Planes

Determining the crossing number of a graph is a classic problem in graph theory [107]. In addition to theoretical interest, drawings with few edge crossings are important in many practical applications such as network visualization [113] and multilayer VLSI layout [87]. Drawing on multiple planes helps minimize the

number of crossings in each plane, which improves the readability of visualization by distributing the visual load into multiple displays.

The complete graph is one of the most studied graph classes in the literature on crossing number [2, 18, 33, 97, 105, 107]. Let K_n be a complete graph of n vertices, and consider k-planar drawings of K_n. If $k = \lfloor (n+7)/6 \rfloor$, i.e., the thickness of K_n [5], then we have $cr_k(K_n) = 0$. As the value of k decreases, the value of $cr_k(K_n)$ increases, and for $k = 1$, we have $cr(K_n) \in \Theta(n^4)$. The case when $k = 1$ is particularly interesting since the upper bound on $cr(K_n)$ matches the exact crossing number for all the complete graphs for which the crossing numbers are known [104], i.e., up to K_{12}. However, for $k \geq 2$, currently known upper bounds on $cr_k(K_n)$ [99, 110] are not tight even for small complete graphs. Consequently, we were motivated to examine the case when $k = 2$.

7.1.1 Related Work

The problem of determining $cr(K_n)$, i.e., the crossing number of a complete graph with n vertices, has been studied since the early 1960s [72, 73, 119]. From that time it was known [72] that $cr(K_n)$ is at most Z_n, where $Z_n = \frac{1}{4} \lfloor \frac{n}{2} \rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \lfloor \frac{n-2}{2} \rfloor \lfloor \frac{n-3}{2} \rfloor$.

Given a complete graph of n vertices, there are several construction techniques [34] to produce a drawing of the graph with exactly Z_n crossings. In fact, it is conjectured that the equality $cr(K_n) = Z_n$ holds in general [119, 73]. Pan and Richter [104] showed that the conjecture holds for the case when $n \leq 12$. For general graphs, it is NP-hard to find a drawing that minimizes the number of crossings [63].

In 1971, Owens [99] showed that $cr_2(K_n)$ is at most W_n, where

$$W_n = \begin{cases} Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{n^2(n-4)(n-8)}{384}, & \text{if } n = 4m. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{(n-1)(n-3)^2(n-5)}{384}, & \text{if } n = 4m+1. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{n(n-2)(n-4)(n-6)}{384}, & \text{if } n = 4m+2. \\ Z_{\lceil n/2 \rceil} + Z_{\lfloor n/2 \rfloor} + \frac{(n+1)(n-3)^2(n-7)}{384}, & \text{if } n = 4m+3. \end{cases} \tag{7.1}$$

A rich body of research examines the asymptotic behaviour of the k-planar crossing numbers of different classes of graphs [9, 29, 30, 65, 110]. Czabarka et al. [30] proved that for every graph G, $cr_2(G) \leq (3/8)cr(G)$. Shahrokhi et al. [110] showed that for every k, where $(k-1)$ is a prime power[2], the inequality $cr_k(K_n) \leq \frac{k(n+k^2)^4}{64(k-1)^3}$ holds. While tight bounds for $cr(K_n)$ are known for $n \leq 12$ [72, 104], the value of $cr_2(K_n)$ is known only when $n \leq 9$, i.e., $cr_2(K_n) = 0$ if $n < 9$, and $cr_2(K_9) = 1$ [96]. In a survey on the biplanar crossing number, Czabarka et al. [29] posed an open question that asks to determine $cr_2(K_n)$ when n is small.

A 1-*page drawing* Γ of G is a drawing of G on the Euclidean plane such that all the vertices lie on a circle C in Γ and the edges that do not belong to the boundary of C lie interior to C. The 1-*page crossing number* $\nu(G)$ of G is the minimum number of crossings over all the 1-page drawings of G. The *k-page crossing number* $\nu_k(G)$ of G is $\min\{\nu(G_1) + \nu(G_2) + \ldots + \nu(G_k)\}$, where the minimum is taken over all possible decompositions of G into k subgraphs $G_1, \ldots, G_k$, and the order of vertices along C is the same for all these subgraphs. Observe that a 2-page drawing of K_n with t crossings determines a drawing of K_n into a single plane

[2] A positive integer of the form p^c, where p is a prime number and c is a positive integer.

with exactly t crossings[3]. Recall that the currently best known upper bound on $cr(K_n)$ is Z_n. Several studies proved that $\nu_2(K_n) = Z_n$ for different values of n [21, 33, 34], and very recently Ábrego et al. [1] proved the equality for every $n \in \mathbb{Z}^+$. However, it is still unknown whether $cr(K_n)$ is strictly smaller than $\nu_2(K_n)$, i.e., we only know that $cr(K_n) \leq \nu_2(K_n) = Z_n$.

An analogous relationship between the k-planar crossing number and $2k$-page drawings of K_n is $cr_k(K_n) \leq cp_{2k}(K_n)$. Interestingly, we observe that W_n, which is the best known upper bound on $cr_2(K_n)$ for large values of n, is equal to the best known upper bound [34] on $\nu_4(K_n)$, when $n = 4m$ for some $m \in \mathbb{Z}^+$; see Section 7.2. However, the equality does not hold in general since $cr_2(K_9) = 1 < cp_4(K_9) = 3$ [34].

7.1.2 Our Contribution

We first show that $cr_2(K_{10}) = 2$, $cr_2(K_{11}) \in \{4, 5, 6\}$, and we give an algorithm to compute good upper bounds on $cr_2(K_n)$ for small values of n. We then prove that every K_n, where $n = 16m + 4$ and $m \in \mathbb{Z}^+$, admits a biplanar drawing with at most $W_n - n^3/192 + O(n^2)$ edge crossings, which improves Owens' [99] upper bound of W_n.

[3]Imagine the drawing on a sphere, where the first page is drawn on the upper hemisphere, and the second page is drawn on the lower hemisphere.

7.2 Technical Foundation

In this section we review previous constructions for drawing K_n on one or more planes or pages that achieve few crossings.

De Klerk et al. [34] gave a generalized construction for k-page drawings of complete graphs. Their upper bound on 4-page crossing number (thus the biplanar crossing number) of K_n, where $n = 4m$ for some $m \in \mathbb{Z}^+$, matches exactly the upper bound obtained by Owens [99] for biplanar drawings of complete graphs, which will be described in the later part of this section. We first briefly recall the construction given by Owens [99], and then the construction given by de Klerk et al. [34].

Owens' [99] Construction: Given a complete graph K_n (assume for convenience that $n = 4m$, where $m \in \mathbb{Z}^+$), in each plane Owens constructed two vertex disjoint cycles $C = (v_1, \ldots, v_{n/2})$ and $C' = (u_1, \ldots, u_{n/2})$, each with $n/2$ vertices. He constructed the complete graph induced by the vertices on C using a 2-page drawing of $K_{n/2}$, i.e., placing the edges of the ith page, $i \in \{1, 2\}$, interior to the cycle C in the ith plane. The complete graph induced by the vertices on C' was constructed exterior to C' in a similar way. The remaining edges that form a complete bipartite graph $K_{n/2,n/2}$ connecting the vertices of C with the vertices of C', were drawn as follows: for each v_j on C, the first plane contains the edges from v_j to $n/4 - 1$ consecutive vertices on C' starting at u_j in clockwise order. The remaining edges of $K_{n/2,n/2}$ are drawn in the second plane symmetrically. Figures 7.1 (a)–(b) illustrate such a construction for K_{16}.

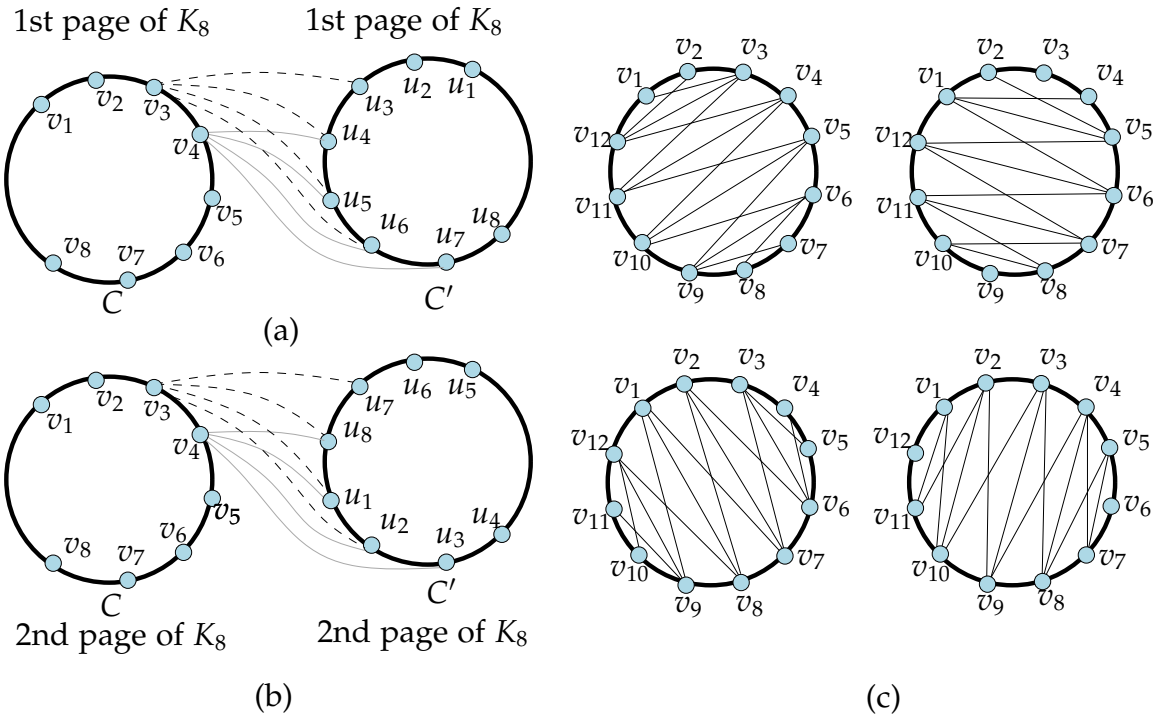


Figure 7.1: (a)–(b) Owens' [99] Construction. (c) De Klerk et al.'s [34] Construction.

De Klerk et al.'s [34] Construction: De Klerk et al. [34] showed that for complete graphs K_n, where $n = km$ with $m, k \in \mathbb{Z}^+$, the k-page crossing number of K_n is

$$\begin{aligned} \nu_k(K_n) &= \frac{1}{12k^2}\left(1 - \frac{1}{2k}\right)n^4 - \frac{1}{4k}n^3 + \left(\frac{7}{24k} + \frac{1}{6}\right)n^2 - \frac{1}{4}n \\ &= \frac{7}{1536}n^4 - \frac{1}{16}n^3 + \frac{23}{96}n^2 - \frac{1}{4}n, \text{ when } k = 4. \end{aligned}$$

We can observe that this is equal to Equation (7.1) when $k = 4$, as follows. Since $n = 4m$, we may assume $n = 2q$ with $q = 2m$. Then we have

$$\begin{aligned} Z_q &= \frac{1}{4}\left\lfloor\frac{q}{2}\right\rfloor\left\lfloor\frac{q-1}{2}\right\rfloor\left\lfloor\frac{q-2}{2}\right\rfloor\left\lfloor\frac{q-3}{2}\right\rfloor \\ &= \frac{1}{4}\left(\frac{q}{2}\right)\left(\frac{q}{2}-1\right)\left(\frac{q}{2}-1\right)\left(\frac{q}{2}-2\right) \\ &= \frac{1}{1024}q(q-4)^2(q-8). \end{aligned}$$

From Equation (7.1), we have

$$\begin{aligned} W_n &= Z_{\lceil q\rceil} + Z_{\lfloor q\rfloor} + \frac{n^2(n-4)(n-8)}{384} \\ &= \frac{7}{1536}n^4 - \frac{1}{16}n^3 + \frac{23}{96}n^2 - \frac{1}{4}n \\ &= \nu_4(K_n). \end{aligned}$$

To construct the k-page drawing, let the vertices of K_n be $v_1,\ldots,v_n$, and let M_i be the set of edges $\{(v_a,v_b) : 1 \le a,b \le n$, and $i = (a+b-2) \mod n\}$. Now draw the edges $M_{(j-1)n/k} \cup \ldots \cup M_{jn/k-1}$ in the jth page. Figure 7.1 (c) in illustrates the construction for K_{12} on 4 pages. Pairing the k pages and placing them in each side of a circle yields a $\lceil k/2\rceil$-planar drawing, which implies that $cr_k(K_n) \le cp_{2k}(K_n)$.

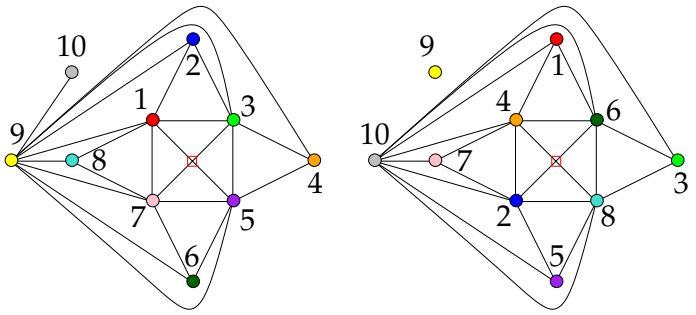
7.3 Computing $cr_2(K_n)$ for Small Values of n

In this section we establish some tight bounds on the biplanar crossing number of K_n when n is small. It has been known since 1971 that $cr_2(K_n) = 0$ if $n < 9$, and $cr_2(K_9) = 1$ [99]. We may thus assume that $n > 9$. We first prove that $cr_2(K_{10}) = 2$ and $cr_2(K_{11}) \in \{4,5,6\}$, and then provide a technique to compute good upper bounds on $cr_2(K_n)$, when $n > 9$.

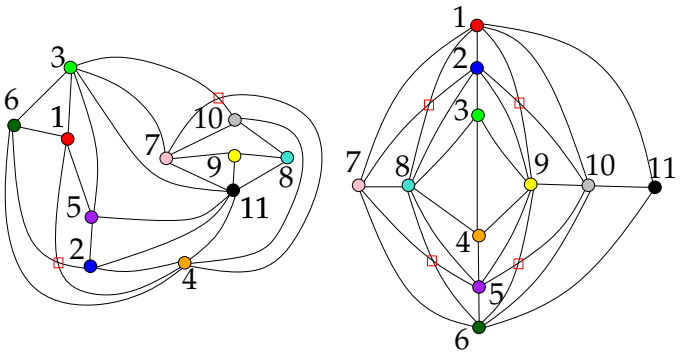
Biplanar Crossing Numbers of K_{10} and K_{11}: Here we prove that $cr_2(K_{10}) = 2$ and $cr_2(K_{11}) \in \{4, 5, 6\}$. To show the sufficiency, we construct biplanar drawings of K_{10} and K_{11} with exactly 2 and 6 edge crossings, respectively, as shown in Figure 7.2. We now show that 2 and 4 edge crossings are necessary for K_{10} and K_{11}, respectively. Suppose for a contradiction that K_{10} admits a biplanar drawing with fewer than two edge crossings, and let Γ be such a biplanar drawing. Since K_{10} contains K_9 as a subgraph, Γ must contain exactly one edge crossing. Let (u, v) be an edge on Γ that is involved in this crossing. Then the deletion of v and its incident edges from Γ would give a biplanar drawing of K_9 without any edge crossing, which contradicts that $cr_2(K_9) = 1$.

For $cr_2(K_{11})$, we prove a lower bound of 4 as follows: Let Γ be an optimal biplanar drawing with at most 3 crossings. Observe that Γ must have at least 3 crossings, otherwise we can delete some vertex which is incident to some crossing in Γ to obtain a biplanar drawing of K_{10} with at most one edge crossing. Observe that no vertex v in Γ can be adjacent to two or more edge crossings, because otherwise deletion of v from Γ would yield a biplanar drawing of K_{10} with at most 1 crossing, which contradicts that $cr_2(K_{10}) = 2$. Since every crossing involves four distinct vertices and every vertex in Γ is incident to at most one crossing, Γ must have at least 12 distinct vertices, which contradicts that Γ is a drawing of K_{11}.

Let S be all the edges of K_n. Let $\mathcal{T}_n$ be the set of triangulations of n vertices. For a triangulation $T \in \mathcal{T}_n$, let $f(S, T)$ be the maximum cardinality subset of edges among the edge set $S \setminus E(T)$ such that $f(S, T)$ induces a planar graph. The following lemma is of independent interest that suggests a lower bound on the



A biplanar Drawing of K_{10}



A biplanar Drawing of K_{11}

Figure 7.2: Biplanar drawings of K_{10} and K_{11} with two and six edge crossings, respectively.

biplanar crossing number using $f(S,T)$.

Lemma 10 *The number of edge crossings in any biplanar drawing of K_n must be at least* $\binom{n}{2} - \max_{T\in\mathcal{T}_n}\{3n-6+|f(S,T)|\}$.

Proof: Suppose for a contradiction that there exists a drawing Γ of K_n with fewer than $\binom{n}{2} - \max_{\forall T}\{3n-6+|f(S,T)|\}$ crossings. Remove the minimum number of edges from Γ to obtain a biplanar drawing Γ' with no edge crossings. Since each layer of Γ' contains a planar graph, the graph G_1 in the first layer of Γ' is a subgraph of some triangulation $T' \in \mathcal{T}_n$ and $E(G_1) \leq 3n-6$. The graph G_2 in

the second layer of Γ' cannot have more than $|f(S,T')| + |E(T')| - |E(G_1)|$ edges. Thus the number of edges in Γ' is $|f(S,T')| + |E(T')|$ in total. Hence Γ must have at least $\binom{n}{2} - |E(T')| - |f(S,T')| \geq \binom{n}{2} - \max_T\{3n-6+|f(S,T)|\}$ crossings. ∎

We performed an exhaustive computer search for K_{11} using Lemma 10, but this also gave a lower bound of 4.

Biplanar Crossing Numbers of K_n, where $n \geq 12$: Let Γ be a biplanar drawing of K_n. Observe that one can construct a biplanar drawing of K_{n+1} by executing the following steps:

S_1. Pick a vertex v in Γ and create a copy v' of v in each of the two layers of Γ.

S_2. In each layer of Γ, place v' arbitrarily close to v and add the edge (v,v') so that this edge does not introduce any new crossing.

S_3. Let $W = \{w_1, w_2, \ldots, w_{\lfloor d_v^i/2 \rfloor}\}$ be the neighbors of v in clockwise order in the ith layer of Γ, where d_v^i denotes the degree of v in the ith layer. For each $w \in W$, we add the edge (v',w) closely following the edge (v,w) such that v' appears after v while examining the neighbors of w in clockwise order. The edges from v' to the remaining neighbors $\{w_{\lfloor d_v^i/2 \rfloor + 1}, \ldots, w_{d_v^i}\}$ of v are added symmetrically.

S_4. Remove the edge (v,v') from the second layer.

Let the resulting drawing be Γ'. It is straightforward to verify that the number of newly created crossings among the edges incident to v and v' is exactly $\sum_{i\in[1,2]} \left(\frac{\lfloor d_v^i/2 \rfloor (\lfloor d_v^i/2 \rfloor - 1) + (\lceil d_v^i/2 \rceil - 1)\lceil d_v^i/2 \rceil}{2} \right)$. Moreover, a crossing between two edges

Table 7.1: Upper and lower bounds on $cr_2(K_n)$, where $n \in [12, 25]$.

n	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$U.B.$	14	26	43	62	81	103	148	176	226	332	469	652	717	958
$L.B.$	6	9	13	19	26	35	46	60	76	95	118	145	176	212

(v, w) and (x, y), where $v \notin \{x, y\}$, corresponds to a crossing between (v', w) and (x, y). Therefore, if v is adjacent to c_v^i crossings in the ith layer, then the number of crossings in Γ' is $\sum_{i\in[1,2]} \left(\frac{\lfloor d_v^i/2 \rfloor (\lfloor d_v^i/2 \rfloor - 1) + (\lceil d_v^i/2 \rceil - 1)\lceil d_v^i/2 \rceil}{2} + c_v^i \right)$ more than the number of crossings in Γ.

To obtain better drawings, we choose the vertex v that minimizes the number of newly introduced crossings (break ties arbitrarily). Table 7.1 shows the number of crossings obtained by the above construction technique, when $n \in [12, 25]$, and the lower bounds using the inequality $cr_2(K_n) \geq \frac{cr_2(K_{n-1})\binom{n}{4}}{\binom{n-1}{4}}$, which is widely used to establish lower bounds on crossing number [30]. Note that the upper bounds of Table 7.1 are significantly smaller than the values $18, 37, 53, 75, 100, 152$, for $n = 12, \ldots, 17$, obtained by Owens' construction [99]. Similarly, the lower bounds are better than the $n(n-1)/2 - (6n-12)$ lower bound implied by Euler's formula [59]. The following theorem summarizes the results of this section.

Theorem 12 *The biplanar crossing numbers $cr_2(K_{10}) = 2$ and $cr_2(K_{11}) \in \{4, 5, 6\}$. Let d_v^i be the degree of some vertex v, and c_v^i be the number of crossings adjacent to v in the ith layer of some biplanar drawing. Then $cr_2(K_n) \leq \alpha + cr_2(K_{n-1})$, where*

$$\alpha = \min_{v\in K_{n-1}} \left\{ \sum_{i\in[1,2]} \left(\frac{\lfloor d_v^i/2 \rfloor (\lfloor d_v^i/2 \rfloor - 1) + (\lceil d_v^i/2 \rceil - 1)\lceil d_v^i/2 \rceil}{2} + c_v^i \right) \right\}.$$

7.4 Upper Bounds on $cr_2(K_n)$

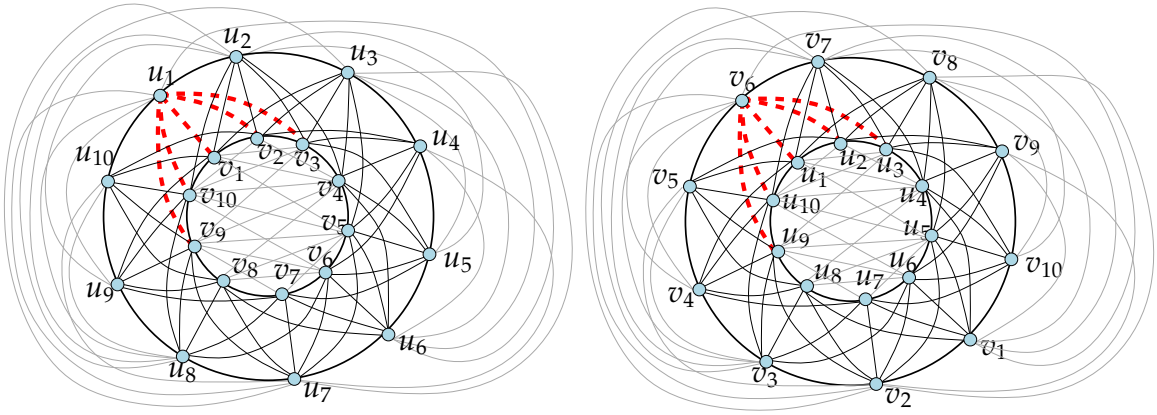
In this section we give improved upper bounds on the biplanar crossing number of complete graphs K_n. Assume that $n = 8m + 4$, where $m \in \mathbb{Z}^+$. We begin with the construction of Owens [99], and later we modify the drawing to improve the number of crossings. We use a slightly different presentation for Owens' [99] construction, which will be more convenient for the subsequent description.

7.4.1 Basic Construction

Let the layers of the drawing be $\mathcal{L}_j$, where $j \in [1, 2]$. In layer $\mathcal{L}_j$, we arrange the vertices into two circles: C^j_{in} and C^j_{out}, where each of them contains $n/2$ vertices. We then embed the cycle C^j_{in} interior to the cycle C^j_{out} such that the resulting embedding of the cycles remains crossing free, as shown in Figure 7.3(a). We now draw the edges that connects the vertices of C^j_{in} and C^j_{out}.

In $\mathcal{L}_1$, let the vertices on C^1_{in} be $v_1, v_2, \ldots, v_{4m+2}$ and the vertices on C^1_{out} be $u_1, u_2, \ldots, u_{4m+2}$ in clockwise order. For each $j \in \{1, 2, \ldots, 4m + 2\}$, connect u_j to the vertices $v_{j-m}, \ldots, v_j, \ldots, v_{j+m}$. Note that the indices wrap around, i.e., for any $v_{j'}$, if $j' < 1$ (respectively, $j' > n/2$), then $v_{j'} = v_{n/2+j'}$ (respectively, $v_{j'} = v_{j'-n/2}$). In the other layer $\mathcal{L}_2$, let the vertices on C^2_{in} be $u_1, u_2, \ldots, u_{2m+1}$ and the vertices on C^2_{out} be $v_1, v_2, \ldots, v_{2m+1}$ in clockwise order. For each $j \in \{1, 2, \ldots, 4m + 2\}$, connect v_j to those vertices of C^2_{in} that are not incident to v_j in $\mathcal{L}_1$. As illustrated in Figure 7.3, all these edges lie in the closed region between C^j_{in} and C^j_{out}.

Note that we may now complete the drawing of K_n by adding the edges among



(a) (b)

Figure 7.3: (a)–(b) Two layers of K_{20} with $W_{20} = 324$ edge crossings.

$\{u_1, \ldots, u_{n/2}\}$ and the edges among $\{v_1, \ldots, v_{n/2}\}$. For the set $\{v_1, \ldots, v_{n/2}\}$, we construct a 2-page drawing of $K_{n/2}$ using de Klerk et al.'s [34] construction, where the edges of one page lie inside C^1_{in} and the edges of the other page lie outside of C^2_{out}. Similarly, for the set $\{u_1, \ldots, u_{n/2}\}$, we construct a 2-page drawing of $K_{n/2}$, where the edges of one page lie inside C^2_{in} and the edges of the other page lie outside of C^1_{out}. Let the resulting drawing be Γ. Since this construction is equivalent to that of Owens [99], the number of crossings in Γ is W_n. An example is illustrated in Figure 7.3, where $m = 2$.

7.4.2 The First Improvement

We now modify the drawing Γ to obtain a biplanar drawing with fewer crossings.

We first delete the incident edges of v_2 that lie inside C^1_{in}, and then add these edges outside of C^2_{out}, as illustrated in thick (blue) lines in Figure 7.4. We then remove the edges that lie on the boundary of C^1_{in}, and finally, move the vertex

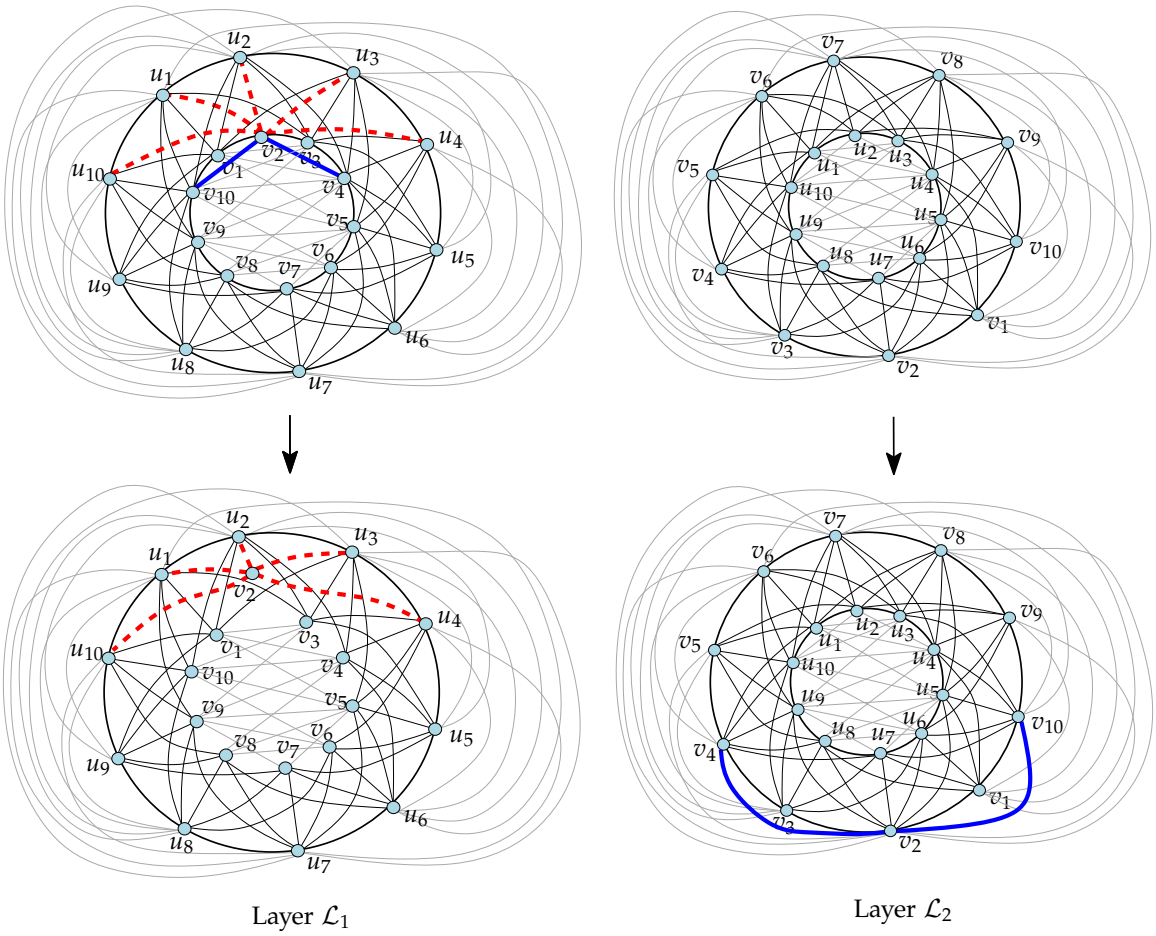


Figure 7.4: Illustrating the modification of Γ, where the blue and red edges are shown in bold and dashed lines, respectively.

v_2 infinitesimally close to u_2 inside the cycle u_2, v_1, v_3, as shown in dashed (red) lines in Figure 7.4. Let the resulting drawing be Γ', which has smaller number of crossings than that of Γ.

We now show that this modification improves the upper bound on the biplanar crossing number when $n \in [20, 52]$. Let the edges incident to v_2 that lie inside C^1_{in} in Γ but moved outside of C^2_{out} in Γ', be the blue edges. Denote the incident edges of v_2 that lie outside of C^1_{in} in Γ as the red edges. Let the number of edge crossings on the blue edges in Γ and Γ' be α and α', respectively. Similarly, let the

number of edge crossings on the red edges in Γ and Γ' be β and β', respectively. Then the number of edge crossings in Γ' is $W_n + (\alpha' + \beta') - (\alpha + \beta)$. Assume that $n = 16m + 4$, $n' = n/2$ and $p = \lfloor n'/4 \rfloor + 1$. We now briefly describe the computation of $\alpha, \alpha', \beta, \beta'$.

Crossings on the Blue Edges in Γ: Here we compute α, i.e., the number of crossings removed by deleting the blue edges from $\mathcal{L}_1$ in Γ. Consider the vertices on the boundary of C^1_{in}. Assume that $S_1 = \{v_3, v_4, \ldots, v_{n'/2-1}\}$, $S_2 = \{v_3, v_4, \ldots, v_{p-1}\}$, $S_3 = S_1 \setminus S_2$, and $S_4 = \{v_{n'/2}, v_{n'/2+1}, \ldots, v_{n'}, v_1\}$. Note that v_2 is adjacent to the vertices in $S_1 \cup \{v_1, v_{n'}\}$ in $\mathcal{L}_1$. We define four types of edge crossings as follows:

- A denotes the number of crossings on the edges (v_2, w) and (x, y), where $w \in S_1, x \in \{v_3, \ldots v_{w-1}\}$ and $y \in S_4$, as shown in bold in Figure 7.5(a). Therefore,

$$A = \sum_{i=4}^{n'/2-1} \sum_{j=3}^{i-1} j.$$

- B denotes the number of crossings on the edges (v_2, w) and (x, y), where $w, y \in S_3$ and $x \in S_2$, as shown in bold in Figure 7.5(b). Therefore,

$$B = \frac{1}{2} \sum_{j=p}^{n'/2-3} \left(\frac{n'}{2} - j - 2\right)\left(\frac{n'}{2} - j - 1\right).$$

- C denotes the number of crossings on the edges (v_2, w) and (x, y), where $w \in S_2, x \in \{v_2, \ldots, v_{w-1}\}$ and $y \in \{v_{w+1}, \ldots, v_{n'/2-1}\}$, as in Figure 7.5(c). Therefore,

$$C = \frac{1}{2} \sum_{j=3}^{p-2} \left((\frac{n'}{2} - 2j)(\frac{n'}{2} - 2j + 1) - (\frac{n'}{2} - j - p + 1)(\frac{n'}{2} - j - p + 2) \right).$$

- D denotes the number of crossings on the edges incident to v_1, which are created by the edge $(v_2, v_{n'})$, as shown in bold in Figure 7.5(d). Therefore, $D = \frac{n'}{2} - 2$.

Thus the number of crossings removed by moving the blue edges from the inner layer is exactly $\alpha = (A + B + C + D)$.

Crossings on the Blue Edges in Γ': We now compute the number of crossings α' created by moving the blue edges to the outer layer of $\mathcal{L}_2$ in Γ'. Consider the vertices on the boundary of C^2_{out}. Assume that $S_1 = \{v_3, v_4, \ldots, v_{n'/2-1}\}$, $S_2 = \{v_3, v_4, \ldots, v_{p-1}\}$, $S_3 = S_1 \setminus S_2$, and $S_4 = \{v_{n'/2}, v_{n'/2+1}, \ldots, v_{n'}, v_1\}$. Note that v_2 is adjacent to the vertices in $S_1 \cup \{v_1, v_{n'}\}$ of $\mathcal{L}_2$ in Γ'. We define four types of crossings as follows.

- A' denotes the number of crossings on the edges (v_2, w) and (x, y), where $w \in S_2 \cup v_p, x \in \{v_3, \ldots v_{w-1}\}$ and $y \in \{v_{w+1}, \ldots v_{n'}\}$, as shown in bold in Figure 7.5(e). Therefore,

$$A' = \sum_{i=4}^{p} \sum_{j=3}^{i-1} \frac{n'}{2} = \frac{n'(p^2 - 5p + 6)}{4}.$$

- B' is an upper bound on the number of crossings on the edges (v_2, w) and (x, y), where $w \in S_3 \setminus v_p, x \in S_2$ and $y \in \{v_{w+1}, \ldots v_{n'}\}$, as shown in bold in Figure 7.5(f). Therefore,

$$B' = \frac{1}{2} \sum_{j=p+1}^{n'/2-1} \left(\left(\sum_{i=3}^{p-1} n' \right) - (j - p - 1)(j - p) \right).$$

- C' is an upper bound on the number of crossings on the edges (v_2, w) and (x, y), where $w \in S_3 \setminus v_p, x \in \{v_p, \ldots, v_{w-1}\}$ and $y \in \{v_{w+1}, \ldots v_{n'}\}$, as

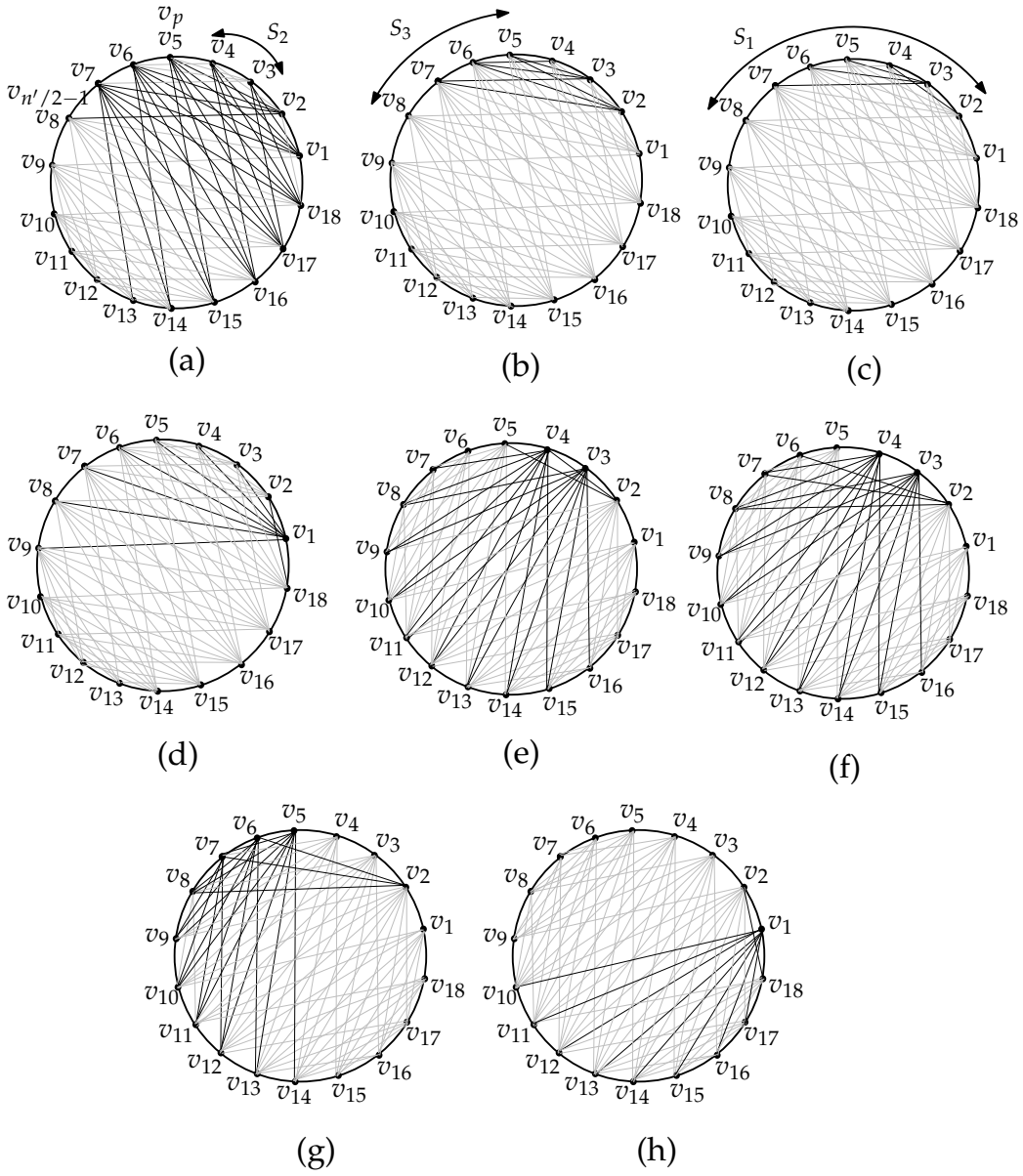


Figure 7.5: Computation of the crossings on the blue edges in (a-d) Γ, and (e-h) Γ'.

shown in Figure 7.5(g). Therefore,

$$C' = \frac{1}{2}\sum_{j=p+1}^{n'/2-1}\left(\left(\sum_{i=p}^{j-1}(n'-2j+2p)\right)-(j-p-1)(j-p)\right).$$

- D' denotes the number of crossings on the edges incident to v_1, which are

created by the edge $(v_2, v_{n'})$, as shown in bold in Figure 7.5(h). Therefore, $D' = \frac{n'}{2} - 1$.

Thus the number of crossings created by moving the blue edges to the outer layer is at most $\alpha' = (A' + B' + C' + D')$.

Crossings on the Red Edges in Γ: The number of crossings created by the edges (v_2, u') and (v_{2+j}, u''), where $1 \leq j \leq 2m-1$ and u', u'' lie on C^1_{out}, is $(2m-j)(2m-j+1)/2$. Figure 7.6(a) illustrates a scenario where $m = 4$. Symmetrically, the number of crossings created by the edges (v_2, u') and (v_{2-j}, u'') is $(2m-j)(2m-j+1)/2$. Hence the number of crossings in the red edges is $\beta = \sum_{j=1}^{2m-1}(2m-j)(2m-j+1)$.

Crossings on the Red edges in Γ': It is straightforward to observe that the number of such crossings is $\beta' = 2m + 2\sum_{i=1}^{m-1} 2mi$, as illustrated in Figure 7.6(b) when $m = 4$.

Figure 7.6(c) plots the difference of the number of edge crossings between Γ and Γ' against the number of vertices. Note that this construction improves the construction of Owens when $n \in [20, 52]$. In the following section we show that instead of choosing v_2 in Γ, one can choose a suitable vertex depending on n so that the corresponding modification always yields improved result.

7.4.3 Choosing a suitable candidate vertex

Let $n = 16m + 4$, $n' = n/2$, $p = \lfloor n'/4 \rfloor + 1$ and $q = \lceil p/2 \rceil$. Instead of choosing v_2, we choose v_q to carry out the modifications. To compute the number of crossings,

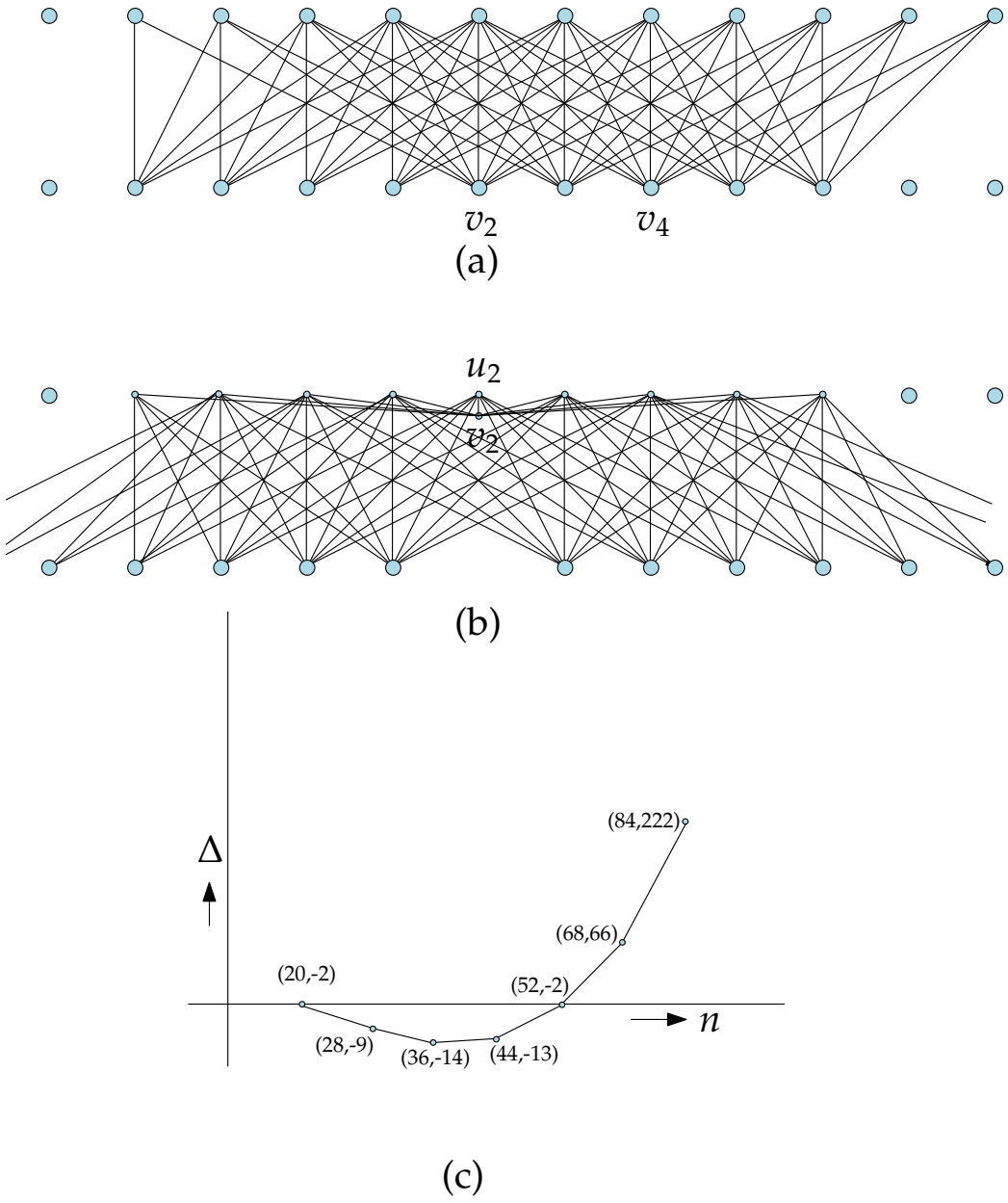


Figure 7.6: Computation of the crossings on the red edges in (a) Γ, and (b) Γ'. (c) Plot of $\Delta = (\alpha' + \beta') - (\alpha + \beta)$ with respect to n.

we define red and blue edges with respect to v_q, and hence we get different number of crossings on the blue and red edges. Note that β and β' remains the same as the previous section, but we now recompute α and α'.

Crossings on the Blue Edges in Γ: We partition edge crossings into the following three types.

- A denotes the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{q+2, \ldots, 2p-q\}$, $x \in \{v_{q+1}, \ldots, v_{w-1}\}$, and $y \in \{v_{2p}, \ldots, v_{n'}\}$. Therefore,

$$A = \sum_{i=q+2}^{2p-q} \sum_{j=q+1}^{i-1} (j + (q-2)),$$

as shown in Figure 7.7(a).

- B denotes the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{q+2, \ldots, p\}$, $x \in \{v_{q+1}, \ldots, v_{w-1}\}$, and $y \in \{v_{w+1}, \ldots, v_{2p}\}$. Therefore,

$$B = \sum_{i=q+2}^{p} \sum_{j=q+1}^{i-1} ((2p-j) - i),$$

as shown in Figure 7.7(b).

- C denotes the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{p+1, \ldots, 2p-q\}$, $x \in \{v_{q+1}, \ldots, v_p\}$, and $y \in \{v_{w+1}, \ldots, v_{2p}\}$. Therefore,

$$C = \sum_{i=p+1}^{2p-q} \frac{(2p-q-i-1)(2p-q-i)}{2},$$

as shown in Figure 7.7(c).

In this case the drawing is symmetric with respect to the axis through v_q and its diametrically opposite vertex. Thus the number of crossings removed by moving the blue edges from the inner layer is exactly $\alpha = 2(A + B + C)$.

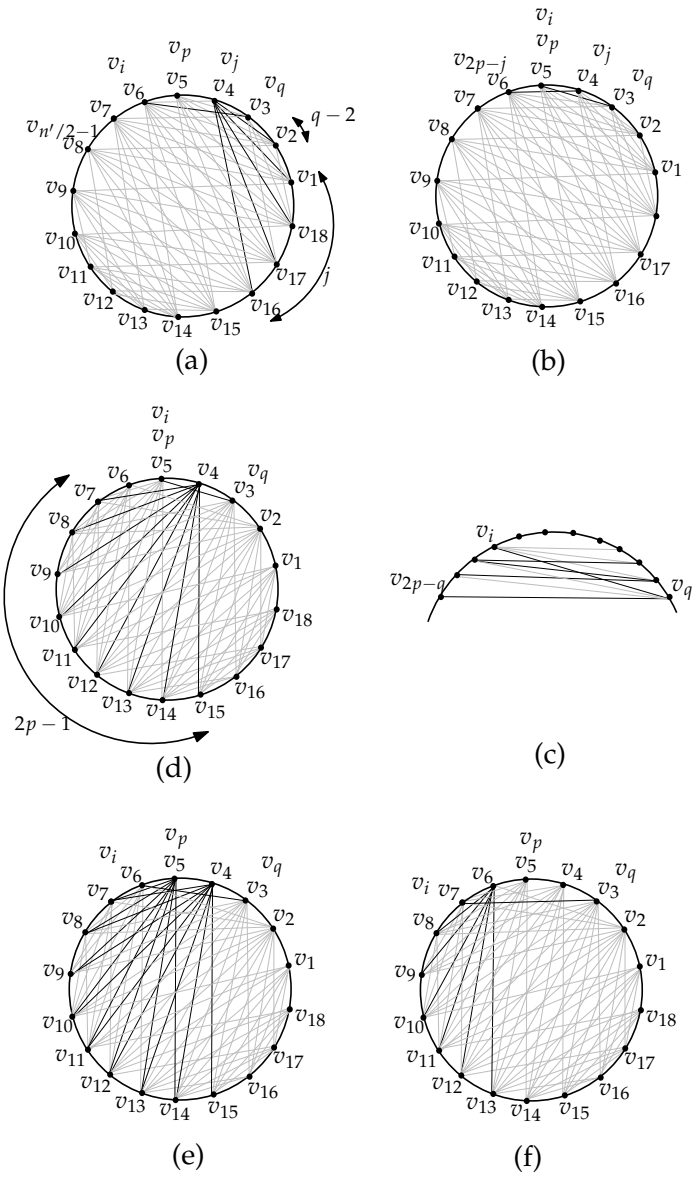


Figure 7.7: Illustration for the computation of crossings with respect to v_q: (a)–(c) Computation of α. (d)–(f) Computation of α'.

Crossings on the Blue Edges in Γ': We partition these edge crossings into the following three types.

- A' denotes the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{q+2, \ldots, p\}$, $x \in \{v_{q+1}, \ldots, v_{w-1}\}$, and $y \in \{v_{w+1}, \ldots, v_{n'}\}$.

Therefore,

$$A' = \sum_{i=q+2}^{p} \sum_{j=q+1}^{i-1} 2p-1,$$

as shown in Figure 7.7(d).

- B' is an upper bound on the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{p+1, \ldots, 2p-q\}$, $x \in \{v_{q+1}, \ldots, v_p\}$, and $y \in \{v_{w+1}, \ldots, v_{n'}\}$. Therefore,

$$B' = \sum_{i=p+1}^{2p-q} \left((p-q)(2p-1) - \frac{(i-p)(i-p+1)}{2} \right),$$

as shown in Figure 7.7(e).

- C' denotes the number of crossings between the edges (v_q, v_w) and (x, y), where $w \in \{p+2, \ldots, 2p-q\}$, $x \in \{v_{p+1}, \ldots, v_{w-1}\}$, and $y \in \{v_{w+1}, \ldots, v_{n'}\}$. Therefore,

$$C' = \sum_{i=p+2}^{2p-q} \sum_{j=p+1}^{i-1} ((2p-1) - 2(j-p) - (i-j)),$$

as shown in Figure 7.7(f).

In this case the drawing is symmetric with respect to the axis through v_q and its diametrically opposite vertex. Hence the number of crossings created by moving the blue edges to the outer layer is at most $\alpha' = 2(A' + B' + C')$.

Now the number of crossings in Γ'' is $W_n + (\alpha' + \beta') - (\alpha + \beta)$, which can be simplified using Maple [92] to get an upper bound of $W_n - \frac{1}{384}n^3 + O(n^2)$. Since the modification we carried out for v_q can also be applied around independently to its diametrically opposite vertex, we can obtain a bound of $W_n - \frac{1}{192}n^3 + O(n^2)$. The following theorem summarizes the result of this section.

Theorem 13 *Every K_n, where $n = 16m + 4$ and $m \in \mathbb{Z}^+$, admits a biplanar drawing with at most $W_n - n^3/192 + O(n^2)$ edge crossings.*

7.5 Summary and Open Questions

In this chapter we have given bounds on the biplanar crossing number of K_n. For small values of n, our technique for computing $cr_2(K_n)$ is incremental. Hence it is natural to ask whether every optimal biplanar drawing of K_{n+1} contains an optimal drawing of K_n.

Open Problem 7.1. Does every optimal biplanar drawing of K_{n+1} contain an optimal biplanar drawing of K_n?

We proved that $cr_2(K_{11}) \in \{4, 5, 6\}$. It would be interesting to find an analytical argument to prove a better lower or upper bound on $cr_2(K_{11})$. For arbitrary values of n, we have proved a $W_n - n^3/192 + O(n^2)$ upper bound on $cr_2(K_n)$. It seems challenging to improve this upper bound to $W_n - cn^4$, where $c > 0$.

Open Problem 7.2. Does there exist a constant $c > 0$ such that every complete graph with n vertices admits a biplanar drawing with at most $W_n - cn^4$ edge crossings?

Chapter 8

Conclusion

In this thesis we explored potential trade-offs among different graph drawing aesthetics from an algorithmic perspective. The various interactions that we observed are specific to the algorithms we developed and the results available in the literature. Consequently, our expected trade-offs may sometimes be different than the actual trade-offs among these aesthetics, and in some cases a trade-off may not even exist. Therefore, it would be an intriguing research direction to prove the existence of such trade-offs, and if they exist, then to determine the exact nature of their interaction. In the following we summarize the result of the thesis and discuss further directions for future research.

Minimizing Segments and Area: In Chapter 3 we proved that every n-vertex planar triangulation admits a straight-line drawing with at most 7n/3 segments, whereas the best known lower bound is $2n + O(1)$ [41]. We also showed that if the vertices are restricted to have integer coordinates, then every triangulation

with maximum degree Δ can be drawn with at most $2n + t - 3$ segments and $O(8^t \cdot \Delta^{2t})$ area, where t is the minimum number of leaves over all the trees of the minimum realizer. For future research, it would be compelling to prove a better upper bound on the number of segments to draw planar triangulations, as well as to establish a better trade-off.

Duncan et al. [46] showed that trees can be drawn with minimum number of segments in polynomial area. Extending this result to larger classes of graphs seems challenging. In fact, many questions on computing minimum-segment drawings respective of area are still open. Dujmović et al. [41] asked whether there exists a polynomial-time algorithm to compute minimum-segment outerplanar drawings of outerplanar graphs. Minimum segment drawings have been achieved for 3-connected planar cubic graphs [94] and series-parallel graphs of maximum degree three [106]. A natural direction of research would be to extend these results to maximum degree four graphs.

Since a k-segment drawing is an arrangement of a set of k straight line segments, an interesting generalization would be to represent planar graphs as arrangement of other objects such as circles, ellipses and lower order splines. Recently, Schulz [109] presented such a generalization considering circular arcs.

Compact Polyline Drawings of Planar Graphs: In Chapter 4 we proved that every n-vertex planar graph with maximum degree $o(n)$ admits a polyline drawing with bend complexity two and at most $(3/8)n^2 + o(n^2)$ area, where the output combinatorial embedding can be freely chosen. Recall that the best known lower bound on area of straight-line grid drawings is $n^2/9 + \Omega(n)$ [61], which holds for

nested triangles graphs. Frati and Patrignani [61] conjectured that for straight-line drawings, it may be possible to improve the lower bound on area to $2n^2/9 + \Omega(n)$. An intriguing direction for future research would be to examine the lower bound allowing bends.

In our algorithm, we first partitioned the input graph into two subgraphs by deleting an edge separator. We then computed the drawings of these subgraphs separately, and finally combined those drawings adding the edges of the edge separator. Such a technique has previously been applied to find compact straight-line drawings of partial 2-trees [122]. It would be interesting to apply this technique to other well known subclasses of planar graphs such as planar cubic graphs, planar 3-trees, etc. Applying this technique may improve the area in many other drawing styles [113] such as orthogonal drawings and visibility drawings.

Polyline Drawings with Good Angular Resolution: In Chapter 5 we proved that every n-vertex planar triangulation admits a drawing with angular resolution at least $r/d(v)$ at each vertex v, and area $f(n,r)$, for any $r \in (0,1]$, where $d(v)$ denotes the degree at v. For $r < 0.389$ or $r > 0.5$, $f(n,r)$ is less than the drawing area required by previous algorithms; $f(n,r)$ ranges from $7.12n^2$ when $r \leq 0.3$ to $32.12n^2$ when $r = 1$. Improving our bounds would be a natural directions for future research.

Our algorithm computes the polyline drawing by transforming a plus-contact representation of the input graph. Along the way we found some interesting results on c-balanced plus-contact representations, where every hand touches at most $\lceil c\Delta \rceil$ other hands. In particular, we proved that for 2-trees, $1/4 \leq c \leq 1/3 (=$

c_2), and for planar 3-trees, $1/3 < c \leq 1/2 (= c_3)$. A consequence of these results is that planar k-trees, $k \in \{2,3\}$, admit 1-bend polyline drawings with at most $2\lceil c_k \Delta \rceil$ slopes. Since this thesis focuses on drawing planar triangulations and arbitrary nonplanar graphs, we did not include those results. However, it would be interesting to examine whether there exist c-balanced plus-contact representations for planar triangulations with $c < 1$.

An interesting line of research would be to extend the results using contact representations of star shapes, which may lead to a better interaction among the aesthetics of polyline drawings.

Drawing Graphs on Multiple Layers: In Chapter 6 we proved that every n-vertex thickness-t graph can be drawn on t layers with bend complexity $\min\{O(2^{t/2} \cdot n^{1-1/\beta}), 2.25n + O(1)\}$, where $\beta = 2^{\lceil (t-2)/2 \rceil}$. Previously, the bend complexity was not known to be sublinear for $t > 2$. We also showed that every graph with linear arboricity k can be drawn on k layers with bend complexity $\frac{3(k-1)n}{(4k-2)}$.

Although determining thickness of an arbitrary graph is an NP-hard problem [91], one can approximate the thickness within a constant factor by repeatedly extracting a spanning forest with maximum number of edges [68]. The number of forest extracted in this way is at most $3t$, where t is the thickness of the input graph. We now can apply our algorithm on these extracted forests. Thus a better algorithm to approximate thickness would help construct better drawings. However, finding a better approximation algorithm is a long standing open question in graph theory.

Every geometric thickness-2 graph admits a polyline drawing on two planar

layers with bend complexity two [58, 66]. While examining thickness-2 graphs, we observed that sometimes one bend per edge is necessary [52]. However, it is not known whether the bend complexity can be reduced to one. We also proved that it is NP-hard to determine whether a geometric thickness-2 graph can be drawn on two layers without using any bend [52]. Since the NP-hardness result does not give much insight into the interaction among bend and layer complexities, we do not include it in this thesis. Nevertheless, it would be interesting to examine such computational complexity questions on simpler classes of graphs. For example, there exists a polynomial-time algorithm to draw graphs with maximum degree four on two layers without any bend [44]. A challenging question is to extend this result to the graphs with maximum degree five or more, while keeping the layer and bend complexities as small as possible, e.g., see [8, 56].

Drawing Complete Graphs on Few Planes: In Chapter 7 we developed an improved technique to construct biplanar drawings of K_n, which reduces Owens' [99] upper bound on $cr_2(K_n)$. Furthermore, we show that $cr_2(K_{10}) = 2$ and $cr_2(K_{11}) \in \{4, 5, 6\}$.

Besides improving the current bounds, it would be interesting to seek for nice polyline drawings with the same number of crossings. Observe that we can construct a planar graph from a non-planar drawing by considering each intersection point as a dummy vertex. The resulting planar graph can be drawn using the known techniques for drawing planar graphs. However, these drawings may have $\Omega(n)$ bend complexity. There have been several attempts [35, 67] to produce nice polyline drawings of complete graphs that ensure low bend complexity and

high angular resolution at the intersection points. But the number of crossings in such a drawing is relatively larger than the crossing number of the underlying graph.

Bibliography

[1] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar. The 2-page crossing number of K_n. *Discrete & Computational Geometry*, 49(4):747–777, 2013. (Cited on page 148.)

[2] B. M. Ábrego, S. Fernández-Merchant, and G. Salazar. The rectilinear crossing number of K_n: closing in (or are we?). In J. Pach, editor, *Thirty essays in Geometric Graph Theory*, pages 5–18. Springer, 2013. (Cited on pages 11 and 146.)

[3] M. J. Alam, T. C. Biedl, S. Felsner, A. Gerasch, M. Kaufmann, and S. G. Kobourov. Linear-time algorithms for hole-free rectilinear proportional contact graph representations. *Algorithmica*, 67(1):3–22, 2013. (Cited on page 29.)

[4] L. C. Aleardi, É. Fusy, and T. Lewiner. Schnyder woods for higher genus triangulated surfaces, with applications to encoding. *Discrete & Computational Geometry*, 42(3):489–516, 2009. (Cited on page 35.)

[5] V. B. Alekseev and V. S. Gonchakov. Thickness of arbitrary complete graphs. *Math USSR Sbornik*, 30(2):187–202, 1976. (Cited on page 146.)

[6] M. Badent, E. D. Giacomo, and G. Liotta. Drawing colored graphs on colored points. *Theoretical Computer Science*, 408(2-3):129–142, 2008. (Cited on pages 115 and 116.)

[7] R. Bar-Yehuda and S. Fogel. Partitioning a sequence into few monotone subsequences. *Acta Informatica*, 35(5):421–440, 1998. (Cited on pages 119, 130, and 135.)

[8] J. Barát, J. Matoušek, and D. R. Wood. Bounded-degree graphs have arbitrarily large geometric thickness. *Electronic Journal of Combinatorics*, 13(R3), 2006. (Cited on pages 115, 142, 143, and 172.)

[9] L. Beineke. Biplanar graphs: A survey. *Computers & Mathematics with Applications*, 34(11):1–8, 1997. (Cited on pages 11 and 147.)

[10] O. Bernardi and É. Fusy. Schnyder decompositions for regular plane graphs and application to drawing. *Algorithmica*, 62(3-4):1159–1197, 2012. (Cited on page 35.)

[11] T. Bläsius, S. G. Kobourov, and I. Rutter. Simultaneous embedding of planar graphs. In R. Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 11, pages 349–380. CRC Press, 2013. (Cited on page 115.)

[12] H. L. Bodlaender. Treewidth of graphs. In *Encyclopedia of Algorithms*, pages 968–970. Springer, 2015. (Cited on page 24.)

[13] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1–2):1–45, 1998. (Cited on page 24.)

[14] N. Bonichon, B. L. Saëc, and M. Mosbah. Optimal area algorithm for planar polyline drawings. In *Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 2573 of *LNCS*, pages 35–46. Springer, 2002. (Cited on pages 4, 13, 55, 56, 65, 69, 70, 71, 72, 73, 77, 84, 90, and 103.)

[15] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected plane graphs. *Algorithmica*, 47(4):399–420, 2007. (Cited on pages 4 and 87.)

[16] P. Bose. On embedding an outer-planar graph in a point set. *Computational Geometry: Theory and Applications*, 23(3):303–312, 2002. (Cited on page 28.)

[17] F. J. Brandenburg. Drawing planar graphs on $\frac{8}{9}n^2$ area. *Electronic Notes in Discrete Mathematics*, 31:37–40, 2008. (Cited on pages 4, 5, 13, 65, 69, 71, 73, 85, 87, and 89.)

[18] P. Brass, W. Moser, and J. Pach. *Research Problems in Discrete Geometry*. Springer, 2006. (Cited on pages 11 and 146.)

[19] P. Braß, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. *Computational Geometry: Theory and Applications*, 36(2):117–130, 2007. (Cited on page 118.)

[20] E. Brehm. *3-Orientations and Schnyder 3-tree-decompositions*. Diploma Thesis, FB Mathematik und Informatik, Freie Universität Berlin, 2000. (Cited on page 35.)

[21] C. Buchheim and L. Zheng. Fixed linear crossing minimization by reduction to the maximum cut problem. In *Proceedings of the 12th Annual International Conference on Computing and Combinatorics (COCOON)*, volume 4112 of *LNCS*, pages 507–516. Springer, 2006. (Cited on page 148.)

[22] C. C. Cheng, C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Drawing planar graphs with circular arcs. *Discrete & Computational Geometry*, 25(3): 405–418, 2001. (Cited on pages viii, 88, and 89.)

[23] P. Z. Chinn, J. Chvatalova, A. K. Dewdney, and N. E. Gibbs. The bandwidth problem for graphs and matrices - a survey. *Journal of Graph Theory*, 6(3): 223–254, 1982. (Cited on page 26.)

[24] M. Chrobak and T. Payne. A linear-time algorithm for drawing planar graphs. *Information Processing Letters*, 54:241–246, 1995. (Cited on page 109.)

[25] F. R. K. Chung. On the coverings of graphs. *Discrete Mathematics*, 30:89–93, 1980. (Cited on page 39.)

[26] F. R. K. Chung and R. Graham. Recent results in graph decompositions. In *Proceedings of the 8th British Combinatorial Conference (Combinatorics)*, volume 52, pages 103–123. London Mathematical Society, 1981. (Cited on page 39.)

[27] CircuitLogix. https://www.circuitlogix.com, Accessed February 20, 2016. (Cited on pages 86 and 112.)

[28] ConceptDraw. http://www.conceptdraw.com, Accessed February 20, 2016. (Cited on pages 86 and 112.)

[29] E. Czabarka, O. Sykora, L. A. Szekely, and I. Vrťo. Biplanar crossing numbers I: A survey of results and problems. In *Proceedings of More Sets, Graphs and Numbers: A Salute to Vera Sós and András Hajnal*, volume 15 of *Bolyai Society Mathematical Studies*, pages 57–77. Springer, 2006. (Cited on pages 11, 144, and 147.)

[30] É. Czabarka, O. Sýkora, L. A. Székely, and I. Vrťo. Biplanar crossing numbers. II. comparing crossing numbers and biplanar crossing numbers using the probabilistic method. *Random Structures and Algorithms*, 33(4):480–496, 2008. (Cited on pages 11, 144, 147, and 155.)

[31] H. de Fraysseix and P. O. de Mendez. On topological aspects of orientations. *Discrete Mathematics*, 229(1-3):57–72, 2001. (Cited on pages 33 and 58.)

[32] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. (Cited on pages 4, 32, 70, 71, 85, and 87.)

[33] E. de Klerk, J. Maharry, D. V. Pasechnik, R. B. Richter, and G. Salazar. Improved bounds for the crossing numbers of $K_{m,n}$ and K_n. *SIAM Journal on Discrete Mathematics*, 20(1):189–202, 2006. (Cited on pages 11, 146, and 148.)

[34] E. de Klerk, D. V. Pasechnik, and G. Salazar. Improved lower bounds on

book crossing numbers of complete graphs. *SIAM Journal on Discrete Mathematics*, 27(2):619–633, 2013. (Cited on pages ix, 146, 148, 149, 150, and 157.)

[35] W. Didimo, P. Eades, and G. Liotta. Drawing graphs with right angle crossings. *Theoretical Computer Science*, 412(39):5156–5166, 2011. (Cited on page 172.)

[36] K. Diks, H. Djidjev, O. Sýkora, and I. Vrto. Edge separators of planar and outerplanar graphs with applications. *Journal of Algorithms*, 14(2):258–279, 1993. (Cited on pages 36, 74, and 84.)

[37] M. B. Dillencourt, D. Eppstein, and D. S. Hirschberg. Geometric thickness of complete graphs. *Journal of Graph Algorithms and Applications*, 4(3):5–17, 2000. (Cited on pages 9, 10, 115, and 116.)

[38] H. Djidjev and S. M. Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34(3):231–243, 1997. (Cited on page 35.)

[39] D. Dolev, T. Leighton, and H. Trickey. Planar embedding of planar graphs. *Advances in Computing Research*, 2:147–161, 1984. (Cited on pages 4 and 71.)

[40] V. Dujmović and D. R. Wood. Graph treewidth and geometric thickness parameters. *Discrete & Computational Geometry*, 37(4):641–670, 2007. (Cited on pages 10, 115, and 116.)

[41] V. Dujmović, D. Eppstein, M. Suderman, and D. R. Wood. Drawings of planar graphs with few slopes and segments. *Computational Geometry: Theory*

and Applications, 38(3):194–212, 2007. (Cited on pages 6, 12, 15, 38, 40, 41, 42, 57, 66, 68, 168, and 169.)

[42] C. A. Duncan. On graph thickness, geometric thickness, and separator theorems. *Computational Geometry: Theory and Applications*, 44(2):95–99, 2011. (Cited on pages 10 and 115.)

[43] C. A. Duncan and S. G. Kobourov. Polar coordinate drawing of planar graphs with good angular resolution. *Journal of Graph Algorithms and Applications*, 7(4):311–333, 2003. (Cited on pages viii, 5, 13, 85, 88, 89, and 109.)

[44] C. A. Duncan, D. Eppstein, and S. G. Kobourov. The geometric thickness of low degree graphs. In *Proceedings of the 20th Annual Symposium on Computational geometry (SoCG)*, pages 340–346. ACM, 2004. (Cited on pages 8, 10, 114, 116, 143, and 172.)

[45] C. A. Duncan, E. R. Gansner, Y. Hu, M. Kaufmann, and S. G. Kobourov. Optimal polygonal representation of planar graphs. *Algorithmica*, 63(3):672–691, 2012. (Cited on page 29.)

[46] C. A. Duncan, D. Eppstein, M. T. Goodrich, S. G. Kobourov, and M. Nöllenburg. Drawing trees with perfect angular resolution and polynomial area. *Discrete & Computational Geometry*, 49(2):157–182, 2013. (Cited on pages 6, 12, 41, 42, and 169.)

[47] S. Durocher and D. Mondal. On balanced +-contact representations. In *Pro-*

ceedings of the 21st International Symposium on Graph Drawing (GD), volume 8242 of *LNCS*, pages 143–154. Springer, 2013. (Cited on page 91.)

[48] S. Durocher and D. Mondal. Drawing plane triangulations with few segments. In *Proceedings of the 26th Canadian Conference on Computational Geometry (CCCG)*, 2014. (Cited on pages 16 and 39.)

[49] S. Durocher and D. Mondal. Trade-offs in planar polyline drawings. In *Proceedings of the 22nd International Symposium on Graph Drawing (GD)*, volume 8871 of *LNCS*, pages 306–318. Springer, 2014. (Cited on pages 16 and 86.)

[50] S. Durocher and D. Mondal. Relating graph thickness to planar layers and bend complexity. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*. LIPIcs, 2016. To appear. (Cited on pages 17 and 112.)

[51] S. Durocher and D. Mondal. Drawing planar graphs with reduced height. In *Proceedings of the 22nd International Symposium on Graph Drawing (GD)*, volume 8871 of *LNCS*, pages 392–403. Springer, 2014. (Cited on pages 16 and 70.)

[52] S. Durocher, E. Gethner, and D. Mondal. Thickness and colorability of geometric graphs. In *Proceedings of the 39th International Workshop Graph-Theoretic Concepts in Computer Science (WG)*, volume 8165 of *LNCS*, pages 237–248. Springer, 2013. (Cited on pages 114 and 172.)

[53] S. Durocher, D. Mondal, R. I. Nishat, and S. Whitesides. A note on

minimum-segment drawings of planar graphs. *Journal of Graph Algorithms and Applications*, 17(3):301–328, 2013. (Cited on pages 6, 12, 41, and 42.)

[54] S. Durocher, E. Gethner, and D. Mondal. On the biplanar crossing number of k_n. In *Proceedings of the 28th Canadian Conference on Computational Geometry (CCCG)*, pages 93–100, 2016. (Cited on pages 17 and 145.)

[55] D. Eppstein. Drawing arrangement graphs in small grids, or how to play planarity. *Journal of Graph Algorithms and Applications*, 18(2):211–231, 2014. (Cited on pages 41 and 69.)

[56] D. Eppstein. Separating thickness from geometric thickness. In J. Pach, editor, *Towards a Theory of Geometric Graphs*. American Mathematical Society, 2004. (Cited on pages 24, 115, 142, and 172.)

[57] P. Erdös and G. Szekeres. A combinatorial theorem in geometry. *Compositio Math.*, 2:463–470, 1935. (Cited on pages 119, 134, and 135.)

[58] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *Journal of Graph Algorithms and Applications*, 9(3):347–364, 2005. (Cited on pages 10, 113, 116, 117, 131, and 172.)

[59] L. Euler. Demonstratio nonnullarum insignum proprietatum quibus solida hedris planis inclusa sunt praedita. *Novi commentarii academiae scientiarum Petropolitanae*, 4:140–160, 1758. (Cited on pages 22 and 155.)

[60] I. Fáry. On straight line representation of planar graphs. 11:229–233, 1948. (Cited on pages 13, 17, 111, and 113.)

[61] F. Frati and M. Patrignani. A note on minimum area straight-line drawings of planar graphs. In *Proceedings of the 15th International Symposium on Graph Drawing (GD)*, volume 4875 of *LNCS*, pages 339–344. Springer, 2008. (Cited on pages 4, 13, 70, 72, 169, and 170.)

[62] M. R. Garey and D. S. Johnson. *Computers and intractability*. Freeman, San Francisco, 1979. (Cited on page 37.)

[63] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):312–316, 1983. (Cited on page 146.)

[64] A. Garg and R. Tamassia. Planar drawings and angular resolution: Algorithms and bounds (extended abstract). In *Proceedings of the Second Annual European Symposium on Algorithms (ESA)*, volume 855 of *LNCS*, pages 12–23. Springer, 1994. (Cited on pages 87 and 110.)

[65] E. Gethner, L. Hogben, B. Lidický, F. Pfender, A. Ruiz, and M. Young. Crossing numbers of complete tripartite and balanced complete multipartite graphs, 2014. http://arxiv.org/abs/1410.0720. (Cited on pages 11 and 147.)

[66] E. D. Giacomo and G. Liotta. Simultaneous embedding of outerplanar graphs, paths, and cycles. *International Journal of Computational Geometry & Applications*, 17(2):139–160, 2007. (Cited on pages 10, 113, 116, 117, and 172.)

[67] E. D. Giacomo, W. Didimo, G. Liotta, and H. Meijer. Area, curve complexity,

and crossing resolution of non-planar graph drawings. *Theory of Computing Systems*, 49(3):565–575, 2011. (Cited on page 172.)

[68] T. F. Gonzalez, editor. *Handbook of approximation algorithms and metaheuristics*. computer & information science. Chapman & Hall/CRC Press, 2007. (Cited on pages 120 and 171.)

[69] M. T. Goodrich and C. G. Wagner. A framework for drawing planar graphs with curves and polylines. *Journal of Algorithms*, 37(2):399–421, 2000. (Cited on pages viii, 5, 85, 88, and 89.)

[70] T. Gordon. Simultaneous embeddings with vertices mapping to pre-specified points. In *Proceedings of the 18th Annual International Conference on Computing and Combinatorics (COCOON)*, pages 299–310, 2012. (Cited on pages 115 and 116.)

[71] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In *Proceedings of the 6th International Symposium on Graph Drawing (GD)*, volume 1547 of *LNCS*, pages 167–182. Springer, 1998. (Cited on pages 5, 87, and 89.)

[72] R. Guy. A combinatorial problem. *Nabla (Bulletin of the Malayan Mathematical Society)*, 7:68–72, 1960. (Cited on pages 11, 14, 145, 146, and 147.)

[73] R. K. Guy. The decline and fall of Zarankiewicz's theorem. In *Proof Techniques in Graph Theory, Proceedings of the Second Ann Arbor Graph Theory*

Conference, pages 63–69. Academic Press, 1969. (Cited on pages 11, 145, and 146.)

[74] J. H. Halton. On the thickness of graphs of given degree. *Journal of Information Sciences*, 54(3):219–238, 1991. (Cited on page 142.)

[75] I. Halupczok and A. Schulz. Pinning balloons with perfect angles and optimal area. *Journal of Graph Algorithms and Applications*, 16(4):847–870, 2012. (Cited on page 41.)

[76] I. S. Hamid and V. M. Abraham. Decomposition of graphs into induced paths and cycles. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 3(11):939–943, 2009. (Cited on page 40.)

[77] V. Jelínek, E. Jelínková, J. Kratochvíl, B. Lidický, M. Tesar, and T. Vyskocil. The planar slope number of planar partial 3-trees of bounded degree. *Graphs and Combinatorics*, 29(4):981–1005, 2013. (Cited on page 7.)

[78] G. Kant. *Algorithms for Drawing Planar Graphs*. PhD thesis, Faculty of Information Science, Utrecht University, the Netherlands, 1993. (Cited on pages 103, 105, and 108.)

[79] M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1): 115–129, 2002. (Cited on page 28.)

[80] B. Keszegh, J. Pach, and D. Pálvölgyi. Drawing planar graphs of bounded

degree with few slopes. *SIAM Journal on Discrete Mathematics*, 27(2):1171–1183, 2013. (Cited on pages 7 and 68.)

[81] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, March 2005. (Cited on page 36.)

[82] T. Kloks. *Treewidth: Computations and Approximations*. Springer, July 1994. (Cited on pages 23 and 24.)

[83] K. B. Knauer, P. Micek, and B. Walczak. Outerplanar graph drawings with few slopes. *Computational Geometry: Theory and Applications*, 47(5):614–624, 2014. (Cited on page 7.)

[84] P. Koebe. Kontaktprobleme der konformen abbildung. *Ber. Schs. Akad. Wiss. Leipzig, Math.-Phys. Kl.*, 88:141–164, 1936. (Cited on pages 6 and 29.)

[85] J. B. Kruskal. Monotonic subsequences. *Proceedings of the American Mathematical Society*, 4:264–274, 1953. (Cited on page 134.)

[86] M. Kurowski. Planar straight-line drawing in an $O(n) \times O(n)$ grid with angular resolution $\Omega(1/n)$. In *Proceedings of the 31st Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 3381 of *LNCS*, pages 250–258. Springer, 2005. (Cited on pages 4, 5, 13, 87, 89, and 110.)

[87] F. T. Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle Exchange Graph and Other Networks*. MIT Press, Cambridge, MA, 1983. (Cited on pages 11 and 145.)

[88] W. Lenhart, G. Liotta, D. Mondal, and R. I. Nishat. Planar and plane slope number of partial 2-trees. In *Proceedings of the 21th International Symposium on Graph Drawing (GD)*, volume 8242 of *LNCS*, pages 412–423. Springer, 2013. (Cited on page 7.)

[89] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computation*, 9(3):615–627, 1980. (Cited on page 35.)

[90] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. (Cited on page 35.)

[91] A. Mansfield. Determining the thickness of a graph is NP-hard. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 93, pages 9–23, 1983. (Cited on pages 120 and 171.)

[92] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, and P. DeMarco. *Maple 10 Programming Guide*. Maplesoft, Waterloo ON, Canada, 2005. (Cited on pages viii, 83, and 166.)

[93] D. Mondal, R. I. Nishat, M. S. Rahman, and M. J. Alam. Minimum-area drawings of plane 3-trees. *Journal of Graph Algorithms and Applications*, 15(2): 177–204, 2011. (Cited on page 71.)

[94] D. Mondal, R. I. Nishat, S. Biswas, and M. S. Rahman. Minimum-segment convex drawings of 3-connected cubic plane graphs. *Journal of Combinatorial Optimization*, 25(3):460–480, 2013. (Cited on pages 6, 12, 41, 42, and 169.)

[95] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs Combin*, 14:59–73, 1998. (Cited on page 25.)

[96] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs and Combinatorics*, 14(1):59–73, 1998. (Cited on pages 14 and 147.)

[97] N. H. Nahas. On the crossing number of $K_{m,n}$. *Electronic Journal of Combinatorics*, 10(N8), 2003. (Cited on pages 11 and 146.)

[98] T. Nishizeki and M. S. Rahman. *Planar Graph Drawing*. World Scientific, Singapore, September 2004. (Cited on page 26.)

[99] A. Owens. On the biplanar crossing number. *IEEE Transactions on Circuit Theory*, 18(2):277 –280, 1971. (Cited on pages ix, 11, 14, 145, 146, 147, 148, 149, 150, 151, 155, 156, 157, 162, and 172.)

[100] J. Pach, editor. *Thirty Essays on Geometric Graph Theory*. Springer, 2012. (Cited on page 29.)

[101] J. Pach and D. Pálvölgyi. Bounded-degree graphs can have arbitrarily large slope numbers. *The Electronic Journal of Combinatorics*, 13(1), 2006. (Cited on page 68.)

[102] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs & Combinatorics*, 17(4):717–728, 2001. (Cited on pages 8, 14, 113, 114, and 141.)

[103] J. Pan and G. J. Chang. Induced-path partition on graphs with special blocks. *Theoretical Computer Science*, 370(1-3):121–130, 2007. (Cited on page 40.)

[104] S. Pan and R. B. Richter. The crossing number of K_{11} is 100. *Journal of Graph Theory*, 56(2):128–134, 2007. (Cited on pages 14, 146, and 147.)

[105] R. B. Richter and G. Salazar. Crossing numbers. In J. L. Gross, J. Yellen, and P. Zhang, editors, *Handbook of Graph Theory*, pages 912–932. Chapman & Hall/CRC, 2nd edition, 2013. (Cited on pages 11 and 146.)

[106] M. A. H. Samee, M. J. Alam, M. A. Adnan, and M. S. Rahman. Minimum segment drawings of series-parallel graphs with the maximum degree three. In *Proceedings of the 16th International Symposium on Graph Drawing (GD)*, volume 5417 of *LNCS*, pages 408–419. Springer, 2009. (Cited on pages 6, 41, 42, and 169.)

[107] M. Schaefer. The graph crossing number and its variants: A survey. *Electronic Journal of Combinatorics*, DS21, 2014. (Cited on pages 11, 145, and 146.)

[108] W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), San Francisco, California, USA*, pages 138–148. ACM, 1990. (Cited on pages 4, 13, 33, 35, 70, 71, 85, and 87.)

[109] A. Schulz. Drawing graphs with few arcs. In *Proceedings of Workshop on*

Graph-Theoretic Concepts in Computer Science (WG), volume 8165 of *LNCS*, pages 406–417. Springer, 2013. (Cited on page 169.)

[110] F. Shahrokhi, O. Sýkora, L. A. Székely, and I. Vrťo. On k-planar crossing numbers. *Discrete Applied Mathematics*, 155(9):1106–1115, 2007. (Cited on pages 11, 146, and 147.)

[111] SmartDraw Software, LLC. http://www.smartdraw.com, Accessed March 18, 2016. (Cited on pages 86 and 112.)

[112] O. Sýkora, L. A. Székely, and I. Vrto. A note on Halton's conjecture. *Journal of Information Sciences*, 164(1-4):61–64, 2004. (Cited on page 142.)

[113] R. Tamassia, editor. *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2014. (Cited on pages 2, 11, 19, 21, 26, 32, 40, 86, 112, 145, and 170.)

[114] W. T. Tutte. Convex representations of graphs. In *London Mathematical Society*, volume 10, pages 304–320, 1960. (Cited on page 23.)

[115] P. Ungar. A theorem on planar graphs. *Journal of the London Mathematical Society*, 1(4):256, 1951. (Cited on page 35.)

[116] W. Wessel. Über die abhängigkeit der dicke eines graphen von seinen knotenpunktvalenzen. In *Proceedings of the 2nd Colloquium for Geometry and Combinatorics*, pages 235–238. Tech. Hochschule Karl-Marx-Stadt, 1983. (Cited on page 142.)

[117] D. B. West. *Introduction to Graph Theory*. Prentice Hall, September 2000. (Cited on pages 19, 23, and 24.)

[118] D. R. Wood. Geometric thickness in a grid. *Discrete Mathematics*, 273(1-3): 221–234, 2003. (Cited on pages 9, 10, 114, and 116.)

[119] K. Zarankiewicz. On a problem of P. Turán concerning graphs. *Fund. Math.*, 41:137–145, 1954. (Cited on pages 11, 145, and 146.)

[120] H. Zhang. Planar polyline drawings via graph transformations. *Algorithmica*, 57(2):381–397, 2010. (Cited on pages 4, 5, 69, 71, 72, 73, 87, and 89.)

[121] H. Zhang and X. He. Canonical ordering trees and their applications in graph drawing. *Discrete & Computational Geometry*, 33(2):321–344, 2005. (Cited on pages 15, 32, 33, 35, 56, and 66.)

[122] X. Zhou, T. Hikino, and T. Nishizeki. Small grid drawings of planar graphs with balanced partition. *Journal of Combinatorial Optimization*, 24(2):99–115, 2012. (Cited on page 170.)

Index