Embedding a Planar Graph on a Given Point Set

by

Debajyoti Mondal

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
August 2012

Thesis advisor

Dr. Stephane Durocher

Author

Debajyoti Mondal

Abstract

A point-set embedding of a planar graph G with n vertices on a set S of n points is a planar straight-line drawing of G, where each vertex of G is mapped to a distinct point of S. We prove that the point-set embeddability problem is NP-complete for 3-connected planar graphs, answering a question of Cabello [20]. We give an $O(n \log^3 n)$-time algorithm for testing point-set embeddability of plane 3-trees, improving the algorithm of Moosa and Rahman [60]. We prove that no set of 24 points can support all planar 3-trees with 24 vertices, partially answering a question of Kobourov [55]. We compute 2-bend point-set embeddings of plane 3-trees in $O(W^2)$ area, where W is the length of longest edge of the bounding box of S. Finally, we design algorithms for testing convex point-set embeddability of klee graphs and arbitrary planar graphs.

Acknowledgments

First and foremost I would like to express my profound gratitude to my thesis supervisor, Dr. Stephane Durocher, for his dedicated time, excellent guidance, endless patience, and financial support throughout my graduate studies. It was December 12, 2009 when I first came in touch with him over email. From that day on, I always found him an honest person and a true gentleman. Working with Dr. Durocher has been one of the most beautiful experiences of my life. He is a real specialist in his field with a wealth of experience, a renowned research scholar, and overall, a genius theoretical computer scientist. It has been a great honor and privilege for me to be his student. He always inspired me to explore various fields of research, and kept his absolute faith in my ability. It amazes me how perfect he has been with his advice and care so that I never felt any frustration at all. Each time I look back on the last few years, I wonder how tremendously blessed I have been to have him as my mentor.

Along with the generous funding I received from Dr. Stephane Durocher, I must acknowledge the Guaranteed Funding Package (GFP) offered by the Department of Computer Science and the University of Manitoba Graduate Fellowship (UMGF), which were critical to the development of my research and collaboration. I want to express my heartfelt thanks to Dr. Therese Biedl, Dr. Helen Cameron (thesis committee member), Dr. Michael Domaratzki (thesis committee member), Dr. Ellen Gethner (thesis committee member), Dr. M. Kaykobad, Dr. Majid Khabbazian, Dr. Stephen G. Kobourov, Dr. Ben Li, Dr. Jason Morrison, Dr. Matthew Skala, and Dr. Sue Whitesides for introducing me with many open problems and providing me with different insights. I want to express my sincere gratitude to Dr. Md. Saidur Rahman, who endowed me with the knowledge of graph drawing. I am extremely thankful to the members of the computational geometry lab in the University of Manitoba, and to all my friends, specially Md. Jawaherul Alam, Sudip Biswas, Tanaeem M. Moosa, Saeed Mehrabi, and M. Abdul Wahid for working together on many problems.

Finally, thank you, Nishat, for inspiring me with numerous research discussions, and for accompanying me from the very beginning of my research career.

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.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter first introduces the reader with the field of graph drawing and gives an overview of the drawing styles that are in the scope of this thesis. It then presents an outline of the relevant results that motivate this thesis and describes our main contributions. Finally, the chapter concludes with a brief description of the organization of the thesis.

1.1 Graph Drawing

Graph drawing is an area of computer science that deals with visualization of graphs arising from diverse applications such as social network analysis, VLSI circuit layout, software engineering, network management and user interface design [5, 38, 48, 63]. Planar graphs, i.e., the graphs admitting planar embedding on the Euclidean plane, are one of the most extensively studied classes of graphs in this research area. Several styles for drawing planar graphs have emerged over the last two decades.

Straight-line drawing is one of the classic and widely studied drawing styles among them.

A *straight-line drawing* of a planar graph G is a planar drawing, where each vertex is drawn as a point and each edge is drawn as straight line segment. The straight-line drawing style is popular since it naturally produces drawings that are easy to read and fit on small screens [77, 80]. A *grid drawing* of G is a straight-line drawing where each vertex is placed on an integer grid point. Many algorithms can compute a grid-drawing of any *plane graph* (i.e., a fixed planar embedding of a planar graph) with n vertices on an $O(n) \times O(n)$ integer grid [15, 29, 67].

To meet the requirements of different practical applications, researchers have examined the straight-line drawing problem under various constraints, e.g., when the vertices are constrained to be placed on a set of pre-specified locations other than integer grid points [13, 20], or when the faces in the drawing are constrained to be convex polygons [25, 82]. If the pre-specified locations for placing the vertices of the input graph are points on the Euclidean plane, then we call the problem a point-set embedding problem. Such problems have applications in VLSI circuit layout, where different circuits need to be mapped on a fixed printed circuit board [51]. Formally, a *point-set embedding* of a graph G with n vertices on a set S of n points is a straight-line drawing of G, where the vertices are placed on distinct points of S. Given a planar graph G with n vertices and a set S of n points, the *point-set embeddability* problem asks whether G admits a point-set embedding on S.

Convexity is sometimes considered as an important criterion of planar graph drawing. A *convex drawing* of a planar graph is a straight-line drawing, where each face in

the output drawing (including the outer face) must be a convex polygon. A point-set embedding is *convex* if the corresponding straight-line drawing is convex. The *convex point-set embeddability* problem asks whether an input planar graph admits a convex point-set embedding on the input point set.

Given a planar graph G with n vertices and a set S of n points, G may not always admit point-set embedding on S. Therefore, many researchers tried to construct a set of k points, $k \geq n$, such that any planar graph with n vertices admits a point-set embedding on a subset of S [26, 29, 57]. Such a point set that *supports* all the planar graphs with n vertices is called a *universal point set for* n.

In this thesis we examine the point-set embeddability and the convex point-set embeddability problems for various subclasses of planar graphs, and review the lower bound on the size of the universal point set.

1.2 Related Results and Motivation

At present, polynomial-time algorithms are known for solving the point-set embeddability problem only on specific restricted classes of planar graphs, which leaves us with some interesting open questions about the time complexity of the problem. Here we give a brief overview of the related results and the questions that motivate this thesis.

1.2.1 Embeddability of 1- and 2-Connected Planar Graphs

The class of 1-connected planar graphs coincides with trees. In 1994, Ikebe et al. [46] gave an $O(n^2)$-time algorithm to embed any tree with n vertices on any set

of n points in general position, i.e., no three points are collinear. Later, Bose et al. [14] devised a divide and conquer algorithm for computing point-set embeddings of trees that runs in $O(n \log n)$ time.

The class of outerplanar graphs is a subclass of 2-connected planar graphs. In 1996, Castañeda and Urrutia [21] gave an $O(n^2)$-time algorithm to construct point-set embeddings of maximal outerplanar graphs. Later, Bose [13] improved the running time of their algorithm to $O(n \log^3 n)$ using a dynamic convex hull data structure. In the same paper Bose posed an open problem that asks to determine the time complexity of testing the point-set embeddability for planar graphs. In 2006, Cabello [20] proved the problem to be NP-complete, even when the input graphs are 2-connected and 2-outerplanar.

These studies on 1- and 2-connected planar graphs motivated much research on the point-set embeddability of 3-connected planar graphs in the last few years.

1.2.2 Embeddability of 3-Connected Planar Graphs

Nishat et al. [62] showed that, unlike trees and outerplanar graphs, not every plane 3-tree with n vertices admits a point-set embedding on any set of n points in general position. They gave an $O(n^2)$-time algorithm that decides whether a plane 3-tree admits a point-set embedding on a given set of points and computes such an embedding if it exists. Recently, Durocher et al. [32] and Moosa and Rahman [60] independently improved the running time of the algorithm of Nishat et al. [62] to $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$.

Since every plane 3-tree with more than three vertices is a 3-connected graph,

these recent findings give rise to the following question: What is the time complexity of deciding the point-set embeddability for 3-connected planar graphs?

1.2.3 Related NP-completeness Results

Kaufmann and Wiese [52] proved that a variation of the point-set embeddability problem is NP-complete, where each edge in the embedding can have at most one bend. Although their proof of NP-completeness holds for 3-connected planar graphs, it does not answer the original question about the point-set embeddability that does not allow the edges to have bends. Later, Katz et al. [51] proved the NP-completeness of another variation of the point-set embeddability problem, where every edge in the output drawing is either a horizontal or a vertical line segment. Their proof holds for subdivisions of 3-connected planar graphs.

Recently, Durocher et al. [32] considered the point-set embeddability problem, where the points of the point set lie in $\mathbb{R}^3$ instead of $\mathbb{R}^2$. Although the point-set embeddability problem is polynomial-time solvable for plane 3-trees in $\mathbb{R}^2$, they proved the problem to be NP-complete in $\mathbb{R}^3$. However, the computational complexity of the point-set embeddability problem for 3-connected planar graphs in $\mathbb{R}^2$ remained unclear.

1.2.4 Lower Bound on the Size of Universal Point Set

In 1990, de Fraysseix et al. [29] proved that any point set that is universal for all planar graphs with n vertices must contain $n+(1-\epsilon)\sqrt{n}$, $\epsilon > 0$, points. Chrobak and Karloff [26] and then Kurowski [57] improved this lower bound to $1.098n$ and $1.235n$,

respectively. In 2002, Kobourov [55] asked for the smallest value m such that no fixed set of m points can be universal for all planar graphs with m vertices. He checked by exhaustive search that for every positive integer t less than 15, there exists a fixed set of t points that supports all planar graphs with t vertices.

1.2.5 Motivation Behind the Thesis

Cabello [20] proved that the point-set embeddability problem is NP-hard for 2-connected planar graphs. He mentioned that his proof technique does not seem possible to extend to show the NP-hardness of the point-set embeddability problem for 3-connected planar graphs and the problem remained open. The problem deserved investigation since many graph drawing problems (e.g., simultaneous embedding of planar graphs with fixed edges [4], upward planarity testing [42] and bend minimization in planar orthogonal drawings [79]) that are NP-hard for 2-connected planar graphs become polynomial-time solvable for 3-connected planar graphs. The problem seemed particularly interesting to us in light of the following two observations. First, Kaufmann and Wiese [52] proved that a variant of the point-set embeddability problem, where each edge can have at most one bend, is NP-complete for 3-connected planar graphs. Second, Nishat et al. [62] proved that the point-set embeddability problem is polynomial-time solvable for plane 3-trees, which is a subclass of 3-connected planar graphs.

Once we established the NP-completeness of the point-set embeddability problem for 3-connected planar graphs, we examined whether the constraint of convexity makes the point-set embeddability problem easier for a *klee graph*, i.e., a graph having a

plane 3-tree as its weak dual. The motivation behind this study was the polynomial-time algorithm of Nishat et al. [62] for testing point-set embeddability of plane 3-trees. Since every plane 3-tree is a maximal planar graph, all of its point-set embeddings are convex. Consequently, klee graphs appeared to us as a potential candidate to study convex point-set embeddability. The current trend to cope with the NP-hard problems with *fixed-parameter tractable* algorithms (i.e., algorithms that run in polynomial time under the assumption that some parameter of the input is constant) motivated us to devise such an algorithm for convex point-set embeddings of 3-connected planar graphs.

The recent improvement over Nishat et al.'s algorithm [62] by Moosa and Rahman [60] drew our attention to design faster algorithm for computing point-set embeddings of plane 3-trees. At this stage we observed that Kurowski [57] achieves the lower bound on the size of the universal point set exploiting the structure of plane 3-trees, which inspired us to examine universal point sets for plane 3-trees.

In summary, the motivation of my thesis was to contribute to a better understanding of the computational complexity of the point-set embeddability problem by examining the following questions.

Question 1: What is the time complexity of deciding point-set embeddability for a 3-connected planar graph?

Question 2: How fast can we decide point-set embeddability of a plane 3-tree?

Question 3: What is the smallest integer m such that no point set of m points is universal for all planar graphs with m vertices?

Question 4: What is the time complexity of deciding convex point-set embeddability of a klee graph?

Question 5: Is there any suitable parameter t such that the convex point-set embeddability problem becomes polynomial-time solvable for general planar graphs under the assumption that t is a constant?

1.3 Contributions and Organization of the Thesis

This section describes the thesis' organization as well as highlights the major contributions of the rest of this thesis.

1.3.1 Technical Foundations (Ch. 2)

In this chapter we introduce the preliminary definitions and notation used throughout the thesis. The readers familiar with the field of planar graph drawing can skip this chapter.

1.3.2 NP-completeness of Point-Set Embeddability (Ch. 3)

Garey et al. [41] proved that the problem of finding a Hamiltonian cycle in a 3-connected cubic planar graph, denoted by HC, is NP-hard. In this chapter we reduce HC to the point-set embeddability problem for 3-connected planar graphs as follows. Let M be an input of HC. We construct a corresponding 3-connected planar graph G and polynomial number of point sets in polynomial time such that G admits a point-set embedding on one of these point sets if and only if M contains

a Hamiltonian cycle. We then show that this reduction proves the NP-hardness of the point-set embeddability problem for 3-connected planar graphs, which answers Question 1.

1.3.3 Point-Set Embeddings of Plane 3-Trees (Ch. 4)

In this chapter we give an algorithm for testing point-set embeddability of a plane 3-tree in $O(n \log^3 n)$ time, which improves the previously best known upper bound $O(n^{4/3}+\epsilon)$ [32, 60]. This answers Question 2. Our algorithm is near optimal since the lower bound on the time complexity is $\Omega(n \log n)$, as established by Nishat et al. [62].

1.3.4 Universal Point Set (Ch. 5)

In this chapter we prove that no set of 24 points can be universal for all planar graphs with 24 vertices. This result sheds light on Kobourov's question on universal point set, i.e., Question 3. Kurowski [57] proved that any point set that is universal for all planar 3-trees with n vertices must contain at least $1.235n$ points. We prove that Kurowski's proof is erroneous, and hence the $1.098n$ lower bound previously obtained by Chrobak and Karloff [26] is still the best known. We also show that every plane 3-tree with n vertices admits a point-set embedding Γ on any set of n points in general position such that Γ uses at most two bends per edge and $O(W^2)$ area, where W is the length of the side of the smallest axis-parallel square that encloses the input point set.

1.3.5 Convex Point-Set Embeddings of Klee Graphs (Ch. 6)

In this chapter we show that the convexity constraint makes the point-set embeddability problem easier for the klee graphs. Every plane 3-tree with $n \geq 4$ vertices admits a decomposition into three smaller plane 3-trees [62]. We characterize such a decomposition in the corresponding klee graph and prove that an input klee graph admits a convex point-set embedding if and only if the smaller graphs obtained from its decomposition admit convex point-set embeddings. We then devise a polynomial-time dynamic programming algorithm to test whether a klee graph admits a convex point-set embedding on a given set of points. This answers Question 4.

1.3.6 Fixed Parameter Tractability (Ch. 7)

In this chapter we give a fixed parameter tractable algorithm to decide convex point-set embeddability for plane graphs. To design such a fixed-parameter tractable algorithm, we choose the number of points interior to the convex hull of the input point set as the fixed parameter. This answers Question 5.

1.3.7 Conclusion and Open Problems (Ch. 8)

This chapter summarizes the main achievements of the thesis and concludes the thesis suggesting possible avenues for future research.

Chapter 2

Technical Foundations

This chapter presents some preliminary definitions and notation that are used in the subsequent chapters of the thesis. We first briefly review relevant introductory concepts on planar graph theory, then discuss the graph drawing problems relevant to this thesis and finally, give a short overview of various types of data structures and algorithms along with their time complexity. The readers familiar with the field of planar graph drawing can skip this chapter.

2.1 Planar Graphs

Let $G = (V, E)$ be a graph with vertex set V and edge set E. By $n(G)$ we denote the number of vertices of G, i.e., $n(G) = |V|$. A graph G is *planar* if it admits a planar embedding on the Euclidean plane. A *plane graph* is a fixed planar embedding of a planar graph. In other words, a planar graph is determined by its adjacency list, whereas a plane graph is determined by its rotation system, i.e., the clockwise order

of the incident edges around each vertex.

A plane graph G delimits the plane into connected regions called *faces*. The unbounded face is the *outer face* of G and all the other faces are the *inner faces* of G. The *length of a face* is the number of vertices on the boundary of the face. The vertices on the outer face are the *outer vertices* and all the other vertices are the *inner vertices*. Let u and v be two outer vertices of G. Then the *outer chain between u and v* is the anticlockwise path between u and v on the outer face of G. By C_{abc} we denote a cycle C that has exactly three vertices a, b, c on its boundary. Let $\{a, b, c\} \subseteq V$. Then by $G(C_{abc})$ we denote the plane subgraph of G induced by the vertices on C_{abc} and the vertices of G interior to C_{abc}.

The *dual graph G^* of G* is a plane graph which has a vertex for each face of G, and two vertices in G^* are adjacent if and only if the corresponding faces in G share an edge. The *weak dual* of G is a subgraph of G^*, which is induced by the vertices corresponding to the inner faces of G. By $\deg(v)$ we denote the *degree* of a vertex v, i.e., the number of edges that are incident to v. A planar graph is called *cubic* if for every vertex v, $\deg(v) = 3$.

We refer the readers to [63, 85] for more details about the planar graph theory. Here we describe some classes of planar graphs that are in the scope of this thesis.

2.1.1 k-Connected Planar Graphs

A graph G is *connected* if for any two distinct vertices u and v there is a path between u and v in G. G is called *k-connected*, $k \geq 1$, if the minimum number of vertices, whose removal results in a disconnected graph or a single-vertex graph, is

k. Every planar graph has a vertex of degree at most five, which is a consequence of Euler's formula [36]. Therefore, planar graphs are at most 5-connected. A planar graph is *maximal* (also called *triangulated*) if addition of any further edge results in a nonplanar graph. Every maximal planar graph with more than three vertices is 3-connected [63].

Let G be a connected planar graph with vertex set V, edge set E and face set F, then by Euler's formula $|V| - |E| + |F| = 2$. If G is maximal, then the numbers of edges and faces in G are $3|V| - 6$ and $2|V| - 4$, respectively. Consequently, using the degree sum formula one can observe that the average degree of a planar graph is strictly less than six. For a detailed discussion on the combinatorial properties of planar graphs we refer the readers to [63, 85].

2.1.2 k-Outerplanar Graphs

A graph G is *outerplanar* if it admits a planar embedding such that all the vertices of G lie on the outer face. An *outerplane graph* is a fixed outerplanar embedding of an outerplanar graph. An outerplanar graph G is *maximal* if the addition of any edge to G violates outerplanarity. Kane and Basu [50] introduced the concept of k-outerplanar graphs. A *k-outerplane graph* G satisfies the following conditions.

(a) If $k = 1$, then G is an outerplane graph.

(b) If $k > 1$, then deletion of all the outer vertices of G results in a $(k-1)$-outerplane graph.

2.1.3 Plane 3-Trees and Klee Graphs

A *plane* 3*-tree* G with $n \geq 3$ vertices is a plane graph for which the following hold.

(a) G is a triangulated plane graph.

(b) If $n > 3$, then G has a vertex whose deletion gives a plane 3-tree with $n - 1$ vertices.

Every plane 3-tree G contains a vertex which is the common neighbor of all the three outer vertices of G. We call this vertex the *representative vertex* of G. Let x be the representative vertex of G and let a, b, c be the three outervertices of G in clockwise order. Then each of the subgraphs $G(C_{abx}), G(C_{bcx})$ and $G(C_{cax})$ is a plane 3-tree. Plane 3-trees are also known as stacked triangulations and Apollonian networks, as studied by Alon and Caro [3] and Takeo [78], respectively.

A *klee graph* G is a plane cubic graph with $n \geq 4$ vertices, which is defined as follows [10, 35].

(a) If $n = 4$, then G is K_4.

(b) Otherwise, G has a face f of length three such that contraction of the three edges on the boundary of f gives a klee graph with $n-2$ vertices.

In other words, every klee graph can be constructed from K_4 by repeatedly replacing a vertex with a face of length three. The weak dual of every klee graph with exactly three outer vertices is a plane 3-tree.

2.2 Graphs with Bounded Parameters

In this section we briefly review various graph parameters, such as treewidth, pathwidth and carving width. We refer interested readers to Kloks [54] for more details about these graph parameters.

A *tree decomposition* of a graph G is a mapping of the vertices of G to the nodes of a tree T satisfying the following conditions.

(a) Each vertex of G belongs to at least one node of T.

(b) For every edge (u, v) of G, there exists a node in T that contains both u and v.

(c) The nodes in T that contain a common vertex of G induce a connected subgraph of T.

Let $X_1, X_2, \ldots, X_k$ be the nodes of T in a tree decomposition of G. Then the *width of the tree decomposition* is $\max_i |X_i| - 1$, where $1 \leq i \leq k$. The *treewidth* of G is the minimum width over all possible tree decompositions of G.

The *path decomposition* of G is a tree decomposition of G, where the underlying tree is a path. The *pathwidth* of G is the minimum width over all possible path decompositions of G.

A *k-carving decomposition* of a graph G is a mapping of the vertices of G to the leaves of a binary tree T satisfying the following conditions.

(a) Each vertex of G belongs to exactly one leaf of T.

(b) Let e be any edge of T and let the two subtrees obtained by the removal of e from T be T' and T''. Then there exists at most k edges in G, where each edge has one endpoint corresponding to a leaf in T' and the other endpoint

corresponding to a leaf in T''.

The *width of a k-carving decomposition* is k. The *carving width* of G is the minimum width over all possible carving decompositions of G.

Bodlaender [12] proved that the pathwidth of every planar graph G with n vertices is $O(\sqrt{n})$. Furthermore, if the treewidth of G is t, then the pathwidth of G is $O(t \log n)$. Since the treewidth of a planar graph is upper bounded by its pathwidth, $O(\sqrt{n})$ is an upper bound on the treewidth of planar graphs. The upper bound is tight, since the treewidth of an $\sqrt{n} \times \sqrt{n}$ grid graph is $\sqrt{n}$. This lower bound on treewidth implies an $\Omega(\sqrt{n})$ lower bound on the pathwidth of planar graphs in the worst case. The treewidth of a k-outerplanar graph is at most $3k - 1$ [11], which is the best possible since there exist k-outerplanar graphs with treewidth $3k - 1$ [49]. For any planar graph G with treewidth t and maximum degree Δ, the carving width of G is at least $0.5t$ [81] and at most $O(\Delta(t + 1))$ [9]. Table 2.1 summarizes the upper bounds and lower bounds on different graph parameters with respect to planar graphs.

Table 2.1: Upper Bounds and Lower Bounds

Graph Parameters	Lower Bounds	Upper Bounds	References
Treewidth	$\Omega(\sqrt{n}), 3k - 1$	$O(\sqrt{n}), 3k - 1$	[11, 12, 49]
Pathwidth	$\Omega(\sqrt{n})$	$O(\sqrt{n})$	[12, 49]
Carving Width	$0.5t$	$O(\Delta(t + 1))$	[9, 81]

Here n, Δ, t and k denote the number of vertices, maximum degree, treewidth and k-outerplanarity, respectively.

2.3 Graph Drawing Styles

Graphs are natural ways to encode information about maps, social networks, VLSI networks and many other models in various disciplines. Graph drawing aims to construct visualization of these graphs that helps users to understand the underlying information and interact with the corresponding data. To enhance the readability of the visualization and to meet the requirements of different practical applications, various graph drawing styles have emerged over time. Here we describe the graph drawing techniques that are in the scope of this thesis. We refer the interested readers to Nishizeki and Rahman [63].

2.3.1 Straight-Line Grid Drawing

A *straight-line drawing* of a planar graph G is a planar drawing of G, where each vertex is drawn as a point on the Euclidean plane and each edge is drawn as a straight line segment. A *straight-line grid drawing* of G is a straight-line drawing of G, where the vertices are placed on integer grid points. Let Γ be a straight-line drawing of G. Then the *area* of Γ is the area of the smallest axis-aligned rectangle R on the grid that encloses Γ. The *width* and the *height* of Γ are denoted by the width and the height of R, respectively. Γ is a *minimum-area straight-line grid drawing* of G if the area of Γ is the minimum among all possible straight-line grid drawings of G. Wagner [84], Fáry [37] and Stein [75] independently proved that every planar graph admits a straight-line drawing. Later, de Fraysseix et al. [29], Schnyder [67] and Brandenburg [15] independently gave respective $2n^2, n^2$ and $8n^2/9$ upper bounds on the area of straight-line grid drawings of the planar graphs with n vertices.

2.3.2 Convex Grid Drawing

A *convex drawing* of a planar graph G is a straight-line drawing Γ of G such that each face (including the outer face) of G is drawn as a convex polygon in Γ. A *convex grid drawing* is a convex drawing, where all the vertices are placed on integer grid points. In 1960, Tutte [82] proved that every 3-connected planar graph admits a convex drawing. Later, Chrobak and Kant [25] gave an algorithm to construct such a drawing on an $(n-2) \times (n-2)$ grid. Any straight-line drawing of a maximal planar graph is convex.

2.3.3 k-Bend Point-Set Embedding

A *polygonal chain* is a curve P determined by a sequence of points $p_1, p_2, \ldots, p_{n-1}, p_n$ such that P consists of the sequence of line segments connecting the consecutive points. Each of the points $p_2, \ldots, p_{n-1}$ is called a *bend* in P. A *polyline drawing* of a planar graph G is a planar drawing of G, where the each vertex is drawn as a point and each edge is drawn as a polygonal chain. A *k-bend point-set embedding* of a graph G with n vertices on a set S of n points is a polyline drawing of G, where each vertex is mapped to a distinct point of S and each edge can have at most k bends. Straight-line drawings and point-set embeddings are special cases of polyline drawings and k-bend point-set embeddings, respectively. Kaufmann and Wiese [52] proved that every planar graph with n vertices admits a 2-bend point-set embedding in any set of n points in the Euclidean plane.

2.4 Data Structures, Algorithms and Complexity

In the subsequent chapters of this thesis we will see different algorithms to solve various graph drawing problems and discuss the efficiency of these techniques. The efficiency of an algorithm is measured in terms of its running time or storage requirements. Divide and conquer algorithm, dynamic programming and greedy algorithm are some of the common types of algorithmic techniques to solve graph-theoretic problems. We refer interested readers to Kleinberg and Tardos [53] for the details of these algorithmic techniques.

Algorithms sometimes process the relevant data into different structures for fast access and manipulation during the execution. Here we briefly review the data structures that are in the scope of this thesis. We refer interested readers to de Berg et al. [28].

2.4.1 Dynamic Convex Hull Data Structures

Given a set S of n points in $\mathbb{R}^d$, a *dynamic convex hull data structure* processes the points of S to answer the extreme point queries quickly as well as supports efficient update upon an insertion or deletion of a point. By efficient update we denote that the data structure can quickly answer the extreme point queries even after an insertion or deletion of a point, but does not require the recomputation of the convex hull from scratch. A dynamic convex hull data structure that is designed specially for the points in $\mathbb{R}^2$ is called *planar*.

In 1981, Overmars and van Leeuwen [64] designed a dynamic planar convex hull data structure that answers queries in $O(\log n)$ time and supports insertion and dele-

tion of a point in $O(\log^2 n)$ time. Although there have been many attempts to improve the $O(\log^2 n)$ time upper bound, all the improvements currently known are in terms of amortized time bound [17, 18, 22] (an *amortized time upper bound* denotes the average time required for a single insertion or deletion, where the average is taken over all the insertions and deletions performed). The best known dynamic planar convex hull data structure in terms of amortized time bound is by Brodal and Jacob [18]. Their data structure answers queries in amortized $O(\log n)$ time and supports an update in amortized $O(\log n)$ time. However, in the worst-case their data structure can take $O(n \log n)$ time for a single operation [47, Section 1.7].

2.4.2 Range Search Data Structures

A *range search data structure* preprocesses a set S of objects so that given a query object X, the objects of S that intersect X can be reported quickly. If X is a triangle and S is a set of points, then the data structure is called a *triangular range search data structure*.

For n points in $\mathbb{R}^d$, the simplex range search data structure of Chazelle et al. [23] takes $O(m^{1+\epsilon})$ preprocessing time and $O(n^{1+\epsilon}/m^{1/d})$ time for range counting queries, where $n < m < n^d$ units of storage are available and $\epsilon > 0$. Therefore, if $d = 2$ and $m = n^{4/3}$, then the time for preprocessing and range counting become $O(n^{4/3+\epsilon})$ and $O(n^{1/3+\epsilon})$, respectively. This data structure can report all the points inside the query triangle in $O(n^{1/3+\epsilon} + k)$ time, where k is the number of points to be reported. We refer interested readers to Agarwal [1] for the details of range search data structures.

2.4.3 Polynomial-Time Algorithms and NP-completeness

An algorithm *runs in polynomial time* if its running time is bounded by a polynomial function of the size of the input instance. If the input instance is a planar graph G, then the size of the instance is usually measured by the number of nodes in G. Let n be the number of nodes in G. Then a polynomial-time algorithm takes $O(n^{O(1)})$ time to process G and to produce the output.

A *decision problem* X is a problem that is answered with either "yes" or "no". By I_X we denote the set of all instances of the problem X. A *polynomial-time reduction* of a decision problem X into a decision problem X' is to define a function $f : I_X \to I_{X'}$ that satisfies the following conditions.

(a) The function f is computable in polynomial time.

(b) For every instance $I \in I_X$, there exists an affirmative solution to I if and only if there exists an affirmative solution to $f(I) \in I_{X'}$.

The *complexity class* P (respectively, NP) denotes the class of decision problems that can be solved by a deterministic Turing machine (respectively, non-deterministic Turing machine) in polynomial time. A decision problem X is *NP-hard* if some problem in complexity class NP is polynomial-time reducible to X. X is called *NP-complete* if it is NP-hard and any solution to X can be verified in polynomial-time. Observe that a polynomial-time algorithm for any NP-complete problem will imply that every NP-complete problem is polynomial-time solvable. But no such polynomial-time algorithm is known. We refer interested readers to Garey and Johnson [40] for more details on the theory of NP-completeness.

2.4.4 FPT-algorithms

An algorithm is called *fixed-parameter tractable* (FPT-algorithm) if it runs in polynomial time under the assumption that some parameter of the input is constant. The running time of these algorithm is of the form $O(f(t)n^{O(1)})$, where n is the size of the input instance and t is the parameter on which the algorithm is parameterized. Usually the function f grows exponentially or even faster. However, an FPT-algorithm can solve NP-hard problems on large instances in polynomial time as long as the value of the parameter t is a small constant.

Many graph algorithms achieve fixed-parameter tractability by choosing treewidth, pathwidth or outerplanarity as a fixed parameter. Niedermeier [61], in his book, compiles a collection of fixed-parameter algorithms.

Chapter 3

NP-completeness of Point-Set Embeddability

In this chapter we prove that the point-set embeddability problem is NP-complete for 3-connected planar graphs. We reduce the problem of finding a Hamiltonian cycle in a 3-connected plane cubic graph to the point-set embeddability problem. Given a 3-connected plane cubic graph G, we construct a corresponding 3-connected planar graph $\mathcal{G}$ and a set $\mathcal{S}$ of point sets. We then prove that $\mathcal{G}$ admits a point-set embedding on some point-set in $\mathcal{S}$ if and only if G is Hamiltonian.

Before presenting the details of our reduction technique we briefly describe a related NP-hardness proof by Kaufmann and Wiese [52] for the 1-bend point-set embeddability problem. Although the proof of Kaufmann and Wiese holds for 3-connected planar graphs, we point out the challenges of modifying their proof technique to prove the NP-hardness of the original point-set embeddability problem. Finally, we give the details of our NP-hardness result.

3.1 Hardness of 1-Bend Point-Set Embeddability

Kaufmann and Wiese [52] reduced the problem of finding a Hamiltonian cycle in a planar graph to the 1-bend point-set embeddability problem. Since it is NP-hard to find a Hamiltonian cycle in a triangulated planar graph [27], the NP-hardness proof for the 1-bend point-set embeddability problem holds for 3-connected planar graph. Here we present a brief overview of their reduction.

3.1.1 Kaufmann and Wise's Proof

Let G be a triangulated planar graph with n vertices. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n collinear points that lie along a horizontal line L such that for each index i in the range $[1, n-1]$, the x-coordinate of s_i is less than that of s_{i+1}. Figure 3.1(a) and (c) give an example of G and S, respectively. Kaufmann and Wiese [52] proved that G is Hamiltonian if and only if G admits a 1-bend point-set embedding on S, as described below.

Assume that G is Hamiltonian. We now construct a 1-bend point-set embedding of G on S. Let $C = v_1, v_2, \ldots, v_n$ be a Hamiltonian cycle in G. First take a plane embedding G' of G such that the edge (v_1, v_2) becomes an outer edge of G', as illustrated in Figure 3.1(b). For each index $i, 1 \leq i \leq n$, map the vertex v_i to the point s_i and then draw the edges that belongs to C with straight line segments except the edge (v_1, v_n). Draw the edge (v_1, v_n) using one bend above the horizontal line L. Then draw the edges of G' that are interior to C in the closed half-plane above L using one bend per edge, and draw the remaining edges of G' in the closed half-plane below L using one bend per edge. Figure 3.1(c) illustrates such a 1-bend

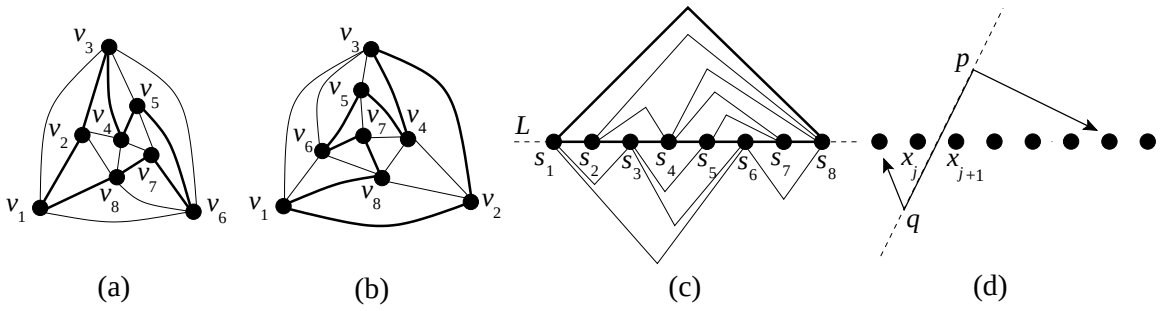

Figure 3.1: (a) A maximal planar graph G. A Hamiltonian cycle in G is emphasized with bold lines. (b) Illustration for G'. (c) A 1-bend point-set embedding of G' on S. (d) A line segment pq separating x_j and x_{j+1}.

point-set embedding of G' on S.

We now assume that G admits a point-set embedding Γ on S and then show that G is Hamiltonian. For each index i, $1 \leq i \leq n$, let x_i be the vertex of G that is mapped to the point s_i in Γ. Since G is a triangulated planar graph and Γ is a 1-bend point-set embedding, x_1 and x_n must be adjacent in Γ. Consequently, it suffices to prove that for each index $j, 1 \leq j \leq n-1$, there exists an edge (x_j, x_{j+1}) in Γ. Suppose for a contradiction that for some index j, x_j and x_{j+1} are not adjacent in Γ. Since G is triangulated, there must be a straight line segment pq in Γ such that the line determined by pq keeps x_j and x_{j+1} to its left and right half-plane, respectively. Figure 3.1(d) illustrates such a scenario. Since all the vertices are mapped along a horizontal line, the edge containing pq must have at least two bends, which contradicts our assumption that Γ is a 1-bend point-set embedding.

3.1.2 Challenges

We now discuss the possibilities and the challenges of modifying the proof of Kaufmann and Wiese [52] to obtain a NP-hardness proof for the original point-set embeddability problem. Observe that if we could avoid the requirement of bends from the Kaufmann and Wiese's proof, then we could establish the NP-hardness of the point-set embeddability problem for 3-connected graphs. Let's attempt to modify the input graph to obtain another graph G' that admits a point-set embedding on S without any bend. Such a graph can be constructed as follows.

Take a fixed planar embedding of G. For each face f of G, insert a vertex v into f and add an edge between v and each vertex on the boundary of f. Now remove the edges that originally belonged to G. Let the resulting graph be H. Figures 3.2(a) and (b) illustrate the graphs G and H, respectively. Observe that if G admits a 1-bend point-set embedding on S, then H admits a straight-line embedding on S such that all the new vertices are either above or below the line L, as illustrated in Figures 3.2(c) and (d).

Although the newly constructed graph H seems promising to avoid the bends used in the NP-hardness proof of Kaufmann and Wiese, we still have several problems to address.

- Observe that H contains $n + |F|$ vertices, where n and $|F|$ are the number of nodes and the number of faces in G, respectively. Since S contains only n points, we need to add $|F|$ more points to construct a new point set S'.

- There are many Hamiltonian graphs with n vertices, which implies that there are many graphs similar to H with $n + |F|$ vertices. Constructing a single

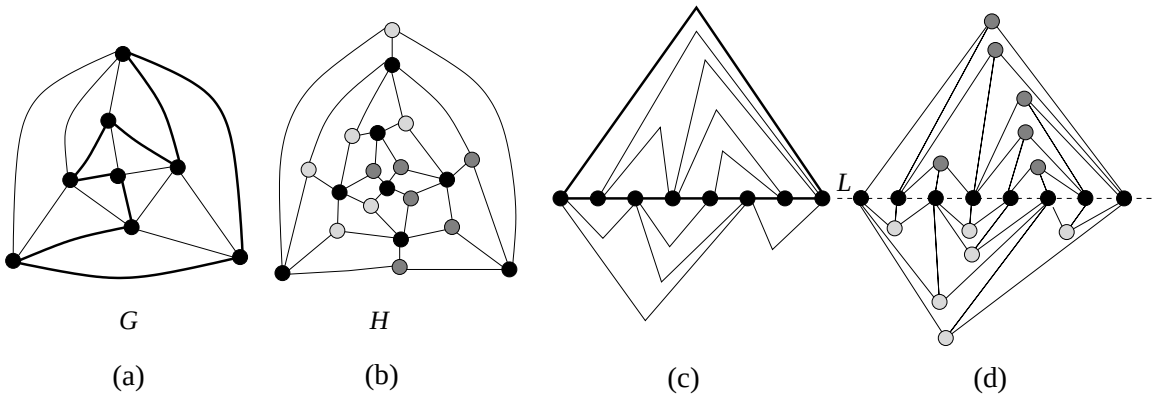

Figure 3.2: (a) A triangulated plane graph G, where a Hamiltonian cycle C is emphasized with bold lines. (b) The graph H obtained from G, where the vertices that correspond to the faces interior (exterior) to C are shown in dark-gray (light-gray). (c)–(d) A 1-bend point-set embedding of G on S and a corresponding straight-line drawing of H, where the black vertices are mapped onto distinct points of S.

suitable point set S' for all these graphs is a difficult task.

- Even if we could construct such a point set S', it is not clear how to ensure that the vertices of H corresponding to the vertices of G (i.e., the black vertices of H in Figure 3.2(b)) must be mapped onto the points of S.

- Let $C_1, C_2, \ldots, C_k$ be k different Hamiltonian cycles of G. Let $H_i, 1 \leq i \leq k$, be the subgraph of H induced by the vertices on C_i and the vertices that correspond to the faces of G interior to C_i (i.e., the black vertices and the dark-gray vertices of H in Figure 3.2(b)). Every H_i will need to have a point-set embedding on the set of points that lie on the closed half-plane above the horizontal line L, which makes it more challenging to the construct a suitable S'.

In the following section we give an NP-completeness proof for the point-set em-

beddability problem that overcomes these challenges and holds for 3-connected planar graphs.

3.2 NP-completeness of Point-Set Embeddability

In this section we prove that the point-set embeddability problem is NP-complete for 3-connected planar graphs. The decision version of the problem is as follows:

Problem: POINT-SET EMBEDDINGS OF 3-CONNECTED PLANAR GRAPHS (PSE)
Instance: A 3-connected planar graph $\mathcal{G}$ with n vertices and a set $\mathcal{S}$ of n points not necessarily in general position.
Question: Does $\mathcal{G}$ admit a point-set embedding on $\mathcal{S}$?

We prove the NP-hardness of PSE by reducing the problem of deciding whether a given 3-connected cubic planar graph is Hamiltonian or not, which has been proved to be NP-complete by Garey et al. [41].

Problem: HAMILTONIAN CYCLE IN 3-CONNECTED CUBIC GRAPHS (HC)
Instance: A 3-connected cubic planar graph M.
Question: Does M contain a Hamiltonian cycle?

Here is an outline of our proof for NP-hardness. For a given instance M of HC with n vertices, we construct a 3-connected planar graph $\mathcal{G}$ with $9n^2 + 7n$ vertices and a set $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ of k point sets, where $k = 3n - 1$ and each point set $\mathcal{S}_i$, $1 \leq i \leq k$, contains $9n^2 + 7n$ points. We prove that $\mathcal{G}$ admits a point-set embedding on some $\mathcal{S}_i$ if and only if M contains a Hamiltonian cycle. We first assume that M

contains a Hamiltonian cycle, and then show a construction of a point-set embedding of $\mathcal{G}$ on some $\mathcal{S}_i$. We then assume that $\mathcal{G}$ admits a point-set embedding Γ on some $\mathcal{S}_i$ and then prove that M contains a Hamiltonian cycle. To prove this, we show that the subgraphs of $\mathcal{G}$, each of which corresponds to a distinct vertex of M, must be embedded in some ordered fashion in Γ. This will in turn correspond to an order of the vertices of M determining a Hamiltonian cycle.

We now describe the formal reduction. In Sections 3.2.1 and 3.2.2 we describe the construction of the graph $\mathcal{G}$ and the set $\mathcal{S}$, respectively. In Section 3.2.3 we prove that $\mathcal{G}$ admits a point-set embedding on some $\mathcal{S}_i \in \mathcal{S}$ if and only if M contains a Hamiltonian cycle.

3.2.1 Construction of $\mathcal{G}$

Let M be an instance of HC with n vertices. We now construct a plane graph $\mathcal{G}$ with $9n^2 + 7n$ vertices. To construct $\mathcal{G}$, we first insert a cycle in each face of M and connect the vertices on the cycle with the vertices on the boundary of the corresponding face. We then delete the original edges of M. Finally, we replace the vertices of the resulting graph with some special graph structure. A formal description of the construction of $\mathcal{G}$ is as follows.

(a) Take any plane embedding of M. For each face f in M, place a cycle R of l vertices inside f, where l is the length of f. We call each cycle R a *ring*. Let the vertices on the boundaries of f and R be $f_1, f_2, \ldots, f_l$ and $r_1, r_2, \ldots, r_l$, respectively. Then add the edges $(f_i, r_i), 1 \leq i \leq l$, and remove the edges that originally belonged to M. Let the resulting graph be M'. Figures 3.3(a)–(c)

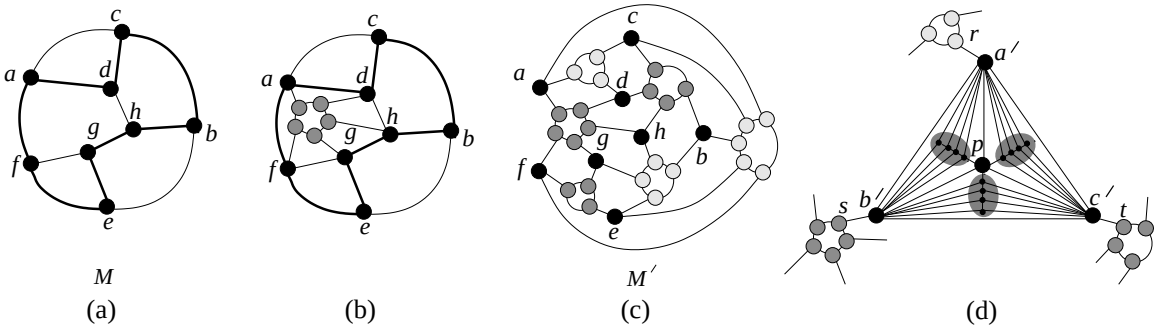

Figure 3.3: (a)A 3-connected cubic planar graph M, where a Hamiltonian cycle C is emphasized with bold lines. (b) Illustration for the construction of M'. (c) M', where the rings corresponding to the faces interior (exterior) to C are shown in dark-gray (light-gray). (d) Replacement of the vertex d with a supernode. Although each of the three paths inside the three dark regions must contain $3n$ vertices, only four of them are shown for simplicity.

illustrate the construction of M' from M.

(b) Let N be a plane 3-tree with the three outer vertices a', b', c' in anticlockwise order and let the representative vertex of N be p. Assume that the subgraph induced by the vertices interior to each of $C_{a'b'p}, C_{b'c'p}$ and $C_{c'a'p}$ is a path of $3n$ vertices. We call such a plane 3-tree a *supernode*. Now for each vertex u of M' that originally belonged to M, replace u with a copy of N as follows. Let r, s, t be the three neighbors of u in anticlockwise order. Delete u and the edges adjacent to u and then add the edges $(a', r), (b', s), (c', t)$. An example is illustrated in Figure 3.3(d). The resulting plane graph determines the required planar graph $\mathcal{G}$.

The number of vertices in all the rings of $\mathcal{G}$ is twice the number of edges of M, i.e., $3n$. Since there are n supernodes in $\mathcal{G}$, the number of vertices in $\mathcal{G}$ is $3n+n(9n+4) =$

$9n^2 + 7n$ vertices. We now have the following lemma. Readers who are interested only in the central idea may skip its proof.

Lemma 1 *$\mathcal{G}$ is a 3-connected planar graph.*

Proof. We first define two operations on a 3-connected planar graph G.

Cycle Replacement: Let u be a vertex in G and let $v_1, v_2, \ldots, v_k$ be the neighbors of u in clockwise order in any plane embedding of G. Let R be a cycle of k vertices, where the vertices of R are $w_1, w_2, \ldots, w_k$ in clockwise order. Replace u with a copy of R as follows. First delete u and its incident edges from G. Then add a copy of R to G. Finally, add the edges $(v_i, w_i), 1 \leq i \leq k$.

Supernode Replacement: Let u be a vertex of degree three in G and let a, b, c be the neighbors of u in clockwise order in any plane embedding of G. Let N be a supernode, where the outer vertices of N are p, q, r in clockwise order. Replace u with a copy of N as follows. First delete u and its incident edges from G. Then add a copy of N to G. Finally, add the edges $(a, p), (b, q), (c, r)$.

In the following we first observe that $\mathcal{G}$ can be obtained by applying the Cycle Replacement and Supernode Replacement operations on some 3-connected graph repeatedly as necessary. We then prove that any Cycle Replacement or Supernode Replacement on a 3-connected graph yields another graph, which is also 3-connected. This will imply that $\mathcal{G}$ is a 3-connected planar graph.

Let M be a given instance of HC. Take any plane embedding Γ of M. For each face f in Γ, place a new vertex u interior to f and connect each vertex on the boundary of f to u introducing a new edge. Then delete all the edges from Γ that originally belonged to M. Let G be the planar graph determined by the resulting plane embedding. Since

M is 3-connected, G is also 3-connected, as proved by Tutte [83]. Now observe that $\mathcal{G}$ can be constructed by applying Cycle Replacement and Supernode Replacement operations on G as follows. First apply Cycle Replacement on all the vertices of G that correspond to the faces in Γ one after another. Then apply Supernode Replacement on all the vertices that originally belonged to M one after another.

We now prove that any Cycle Replacement or Supernode Replacement operation on a 3-connected graph yields another graph which is also 3-connected.

Let G be a 3-connected graph and let G' be a graph obtained from G by a Cycle Replacement on some vertex u of G. Let R be the cycle that replaces u. Let $\{x, y\}$ be a pair of vertices of G', where $x \neq y$. If at most one vertex in $\{x, y\}$ belongs to R, then there are three vertex-disjoint paths between x and y in G', which are determined by the three vertex-disjoint paths between the two nodes of G that correspond to x and y. We now consider the case when both x and y belong to R. Observe that two vertex-disjoint paths between x and y lie on R. Let x' and y' be the neighbors of x and y, respectively, where neither x' nor y' belong to R. Then any cycle containing the path x', u, y' in G determines another vertex-disjoint path between x and y in G'. Since there are three vertex-disjoint paths between any pair of vertices in G', G' is 3-connected by Menger's theorem [58].

Let G be a 3-connected graph and let G' be a graph obtained from G by a Supernode Replacement on some vertex u of G. Let N be the supernode that replaces u. Let $\{x, y\}$ be a pair of vertices of G', where $x \neq y$. If at most one vertex in $\{x, y\}$ belongs to N, then there are three vertex-disjoint paths between x and y in G', which are determined by the three vertex-disjoint paths between the two nodes of

G that correspond to x and y. We now consider the case when both x and y belong to N. Since N is a triangulated planar graph with more than three vertices, N is 3-connected. Therefore, there are three vertex-disjoint paths between x and y in G'.

Since there are three vertex-disjoint paths between any pair of vertices in G', G' is 3-connected [58]. ■

3.2.2 Construction of $\mathcal{S}$

We now construct a set $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ of k point sets, where $k = 3n - 1$. The points in each $\mathcal{S}_j, 1 \leq j \leq k$, lie on four parallel straight lines and two parabolas, as described below.

(a) The set P_t of n points on line $y = 1$, where $P_t = \{(3i, 1) \mid 0 \leq i \leq n-1\}$. The set P_m of n points on line $y = 0$, where $P_m = \{(3i+1, 0) \mid 0 \leq i \leq n-1\}$. The set P_b of n points on line $y = -1$, where $P_b = \{(3i, -1) \mid 0 \leq i \leq n-1\}$. Observe that each triple of points $\{(3i, 1), (3i+1, 0), (3i, -1)\}$ forms an isosceles triangle. We denote such a triangle by T_i, where we call the points $(3i, 1), (3i+1, 0)$ and $(3i, -1)$ the *top*, *middle* and *bottom* points of T_i, respectively. Figure 3.4(a) illustrates the triangles $T_0, T_1, \ldots, T_{n-1}$.

(b) The set $A = \{(9n - x, 10n + x^2) \mid 0 \leq x \leq j-1\}$ of j points on a parabola and the set $B = \{(9n - x, -10n - x^2) \mid 0 \leq x \leq 3n - j - 1\}$ of $3n - j$ points on a parabola, as illustrated in Figure 3.4(a). Observe that for each T_i the following holds:

(i) Every straight line joining the bottom of T_i with a point in A properly

intersects the edge between the top point and the middle point of T_i. Let A_i be the triangle determined by the points $(9n, 10n), (9n - j + 1, 10n + (j-1)^2)$ and the bottom point of T_i. Figure 3.4(b) illustrates the triangles $A_0, A_1, \ldots, A_{n-1}$. Observe that every straight line joining the bottom point of T_i with a point in A is contained in A_i.

(ii) Every line joining the top of T_i with a point in B properly intersects the edge between the bottom point and the middle point of T_i. By B_i we denote the triangle determined by the points $(9n, -10n), (9n-3n+j+1, -10n-(3n-j-1)^2)$ and the top of T_i.

(iii) For every i and every $\epsilon \in (0, 1)$, line $y = \epsilon$ intersects both A_i and the edge e_i between the top point and the middle point of T_i. Let r, s and t denote the respective points of intersection between $y = \epsilon$ and A_i, and between $y = \epsilon$ and e_i. Assume that s is closer to t than r. Then one can choose the ϵ such that for each T_i, s lies interior to T_i and $s \neq t$. Figure 3.4(c) illustrates such a scenario. We will show how to choose ϵ later in Lemma 2. Let l_i be the open line segment from s to t.

(c) The sets $W_i, 0 \leq i \leq n-1$, where each W_i contains $6n + 1$ points on the line $y = \epsilon$ interior to T_i. The points in each W_i are divided into two subsets X_i and Y_i. X_i contains $3n$ points that are on segment l_i. Y_i contains the remaining $3n + 1$ points that are interior to T_i, but not on l_i. The $3n + 1$ points in Y_i includes the intersection points of the lines joining the points of A and B with the bottom point and the top point of T_i, respectively. Figure 3.4(c) gives a hypothetical illustration of these sets.

(d) The sets $Z_i, 0 \leq i \leq n-1$, each containing $3n$ points, which are the perpendicular projections of the points of X_i on the line $y = 0$, as shown in Figure 3.4(c).

Observe that the total number of points in $\mathcal{S}_j$ is $|P_t| + |P_m| + |P_b| + |A| + |B| + n(|W_i| + |Z_i|) = 6n + n(9n+1) = 9n^2 + 7n$. Let $P_i = W_i \cup Z_i \cup T_i, 0 \leq i \leq n-1$, and let $C = \mathcal{S}_j \setminus (A \cup B)$. Any two points $\{u, v\} \in \mathcal{S}$ are *visible to each other* if the straight line segment joining u and v does not contain any other point $w \in \mathcal{S}$. Otherwise, the point w is a *blocking point* between u and v. We now have the following lemma.

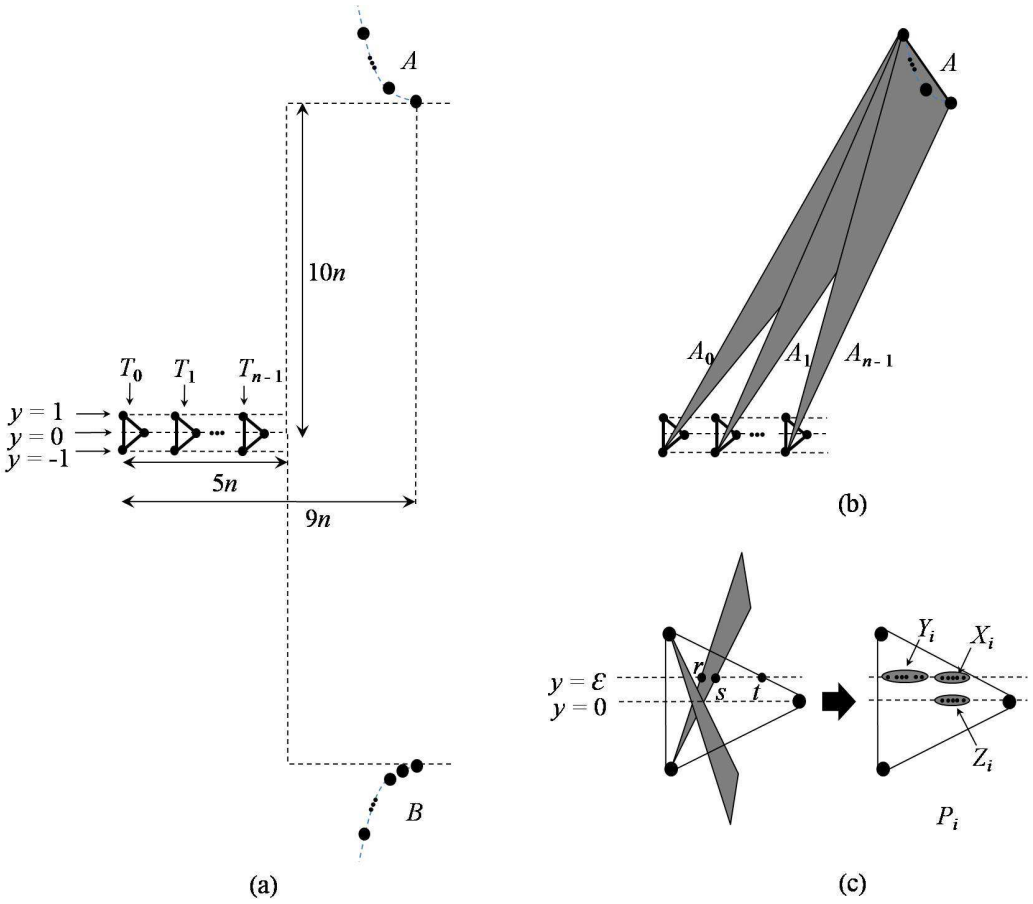

Figure 3.4: (a) Construction of $T_i, 0 \leq i \leq n-1, A$ and B. (b) Illustration for A_i. (c) Construction of X_i, Y_i and Z_i.

Readers who are interested only in the central idea may skip its proof.

Lemma 2 *$\mathcal{S}_j$ has the following properties:*

(a) The coordinates of the points in $\mathcal{S}_j$ are expressible using a number of bits that is polynomial in n.

(b) If $\mathcal{G}$ admits a point-set embedding on $\mathcal{S}_j$, then the supernodes of $\mathcal{G}$ must be mapped to some subset of points of C, where each supernode is mapped to a distinct point set P_i.

(c) For any given mapping of the outer vertices of a supernode N to the three points on the convex hull of P_i, N admits a point-set embedding on P_i.

Proof. (a) Let $a = (a_x, a_y)$ and $b = (b_x, b_y)$ be two points in $\mathbb{R}^2$. Let $c = (c_x, c_y)$ be a point on the line segment joining a and b. Let $l_1 = |c_y - b_y|$ and $l_2 = |a_y - b_y|$. Then it is straightforward to observe that

$$c_x = \frac{l_1}{l_2} b_x + \frac{l_2 - l_1}{l_2} a_x,$$
$$c_y = \frac{l_1}{l_2} b_y + \frac{l_2 - l_1}{l_2} a_y.$$

If each coordinate of a and b can be expressed using a number of bits polynomial in n, then l_1 and l_2 are bounded by a polynomial in n. Consequently, the coordinates of point c are also bounded by a polynomial in n.

Each coordinate of the points in the sets P_t, P_m, P_b, A and B is an integer that is bounded by a polynomial in n. We now consider the remaining sets W_i and Z_i, $0 \leq i \leq n-1$. Recall that we used a positive fixed real ϵ to define W_i and Z_i. We first show how to select a value for ϵ. Assume that the line between points $(0, 1)$ and $(1, 0)$

intersects the line between points $(0, -1)$ and $(9n, 10n)$ at some point $z = (z_x, z_y)$. Then $z_x = 18n/(19n + 1)$ and $z_y = (n + 1)/(19n + 1)$. Observe that $\epsilon = z_y/2$ can serve our purpose, where ϵ is bounded by a polynomial in n. Consequently, for each point $u \in (W_i \cup Z_i)$, we can define each coordinate of u as a convex combination of the endpoints of some straight line segment, where those endpoints are expressible using a number of bits that is polynomial in n. Therefore, each coordinate of u is bounded by a polynomial in n.

(b) Assume that $\mathcal{G}$ admits a point-set embedding Γ on $\mathcal{S}_j$. We now claim that the supernodes of $\mathcal{G}$ must be mapped to some subset of C in Γ. Otherwise, assume that there exists some supernode N such that the three outer vertices of N are mapped to some points $\{r, s, t\} \subset \mathcal{S}_j$ in Γ, where $\{r, s, t\} \not\subset C$.

Since the convex hull of $\mathcal{S}_j$ contains more than three points, no inner face of N in $\mathcal{G}$ can be the outer face in Γ. Therefore, the triangle rst must contain exactly $9n + 1$ points in its interior and for some point $q \in \mathcal{S}_j$ interior to rst, each of the triangles qrs, qst and qtr must contain $3n$ interior points. We now consider the following cases depending on the different choices for $\{r, s, t\}$.

Case 1: Assume that $\{r, s, t\} \subset (A \cup B)$.

Since $|A| + |B| = 3n$, the triangle rst cannot contain $9n + 1$ points in its interior. Therefore, the three outer vertices of N cannot be mapped to any $\{r, s, t\}$, where $\{r, s, t\} \subset (A \cup B)$.

Case 2: Assume that A, B and C each contains one point from $\{r, s, t\}$.

Without loss of generality assume that $r \in A, s \in B$ and $t \in C$. By the construction of $\mathcal{S}_j$, no point of $\mathcal{S}_j$ on line $y = -1$ is visible to r and no point of $\mathcal{S}_j$ on line

$y = 1$ is visible to s. The set Y_i contains all the blocking points. Therefore, t must lie on line $y = 0$ or line $y = \epsilon$.

First consider the case when t lies on line $y = \epsilon$. Observe that $t \in P_{n-1}$, otherwise triangle rst will contain more than $9n + 1$ points in its interior. Moreover, if $q \in A$ or $q \in B$, then the triangle qrs contains less than $3n$ points. Therefore, q must lie on line $y = 0$. Under these circumstances it is straightforward to observe that triangles qst and qrs cannot contain exactly $3n$ points each.

Now consider the case when t lies on line $y = 0$. Observe that $t \in P_{n-1}$, otherwise triangle rst will contain more than $9n + 1$ points in its interior. However, if $t \in P_{n-1}$, then triangle rst contains less than $9n + 1$ points in its interior.

Case 3: Assume that one point of $\{r, s, t\}$ belongs to C and the remaining two points belong to either A or B.

Without loss of generality assume that $t \in C$. Then either $\{r, s\} \subset A$ or $\{r, s\} \subset B$. Consider first the case when $\{r, s\} \subset A$. By the construction of $\mathcal{S}_j$, no point of $\mathcal{S}_j$ on line $y = -1$ is visible to r and s. Therefore, t must lie on line $y = 0$, $y = \epsilon$ or $y = 1$. It is now straightforward to observe that for every choice of t, triangle rst contains less than $9n+1$ points in its interior. The proof for the case when $\{r, s\} \subset B$ is similar.

Case 4: Assume that two points of $\{r, s, t\}$ belong to C and the remaining point belongs to either A or B.

Without loss of generality assume that $\{r, s\} \in C$. Then either $t \in A$ or $t \in B$. It suffices to prove our claim for the case when $t \in A$, since the proof for the other case is similar. By the construction of $\mathcal{S}_j$, no point of $\mathcal{S}_j$ on line $y = -1$ is visible to

t. Therefore, neither r nor s can lie on line $y = -1$. We now consider the following subcases depending on the different choices for r and s.

Subcase 4.1: Assume that r lies on line $y = 1$.

If s lies on line $y = 1$, then triangle rst contains less than $9n + 1$ points in its interior. If s lies on line $y = \epsilon$, then q must lie on line $y = 1$ and triangle qrt must contain less than $3n$ points. If s lies on line $y = 0$, then q must lie on line $y = \epsilon$ or $y = 1$. In both cases we can observe that the triangle qrt contains less than $3n$ points.

Subcase 4.2: Assume that r lies on line $y = \epsilon$.

If s lies on line $y = 1$ or line $y = \epsilon$, then triangle rst contains less than $9n + 1$ points in its interior. If s lies on line $y = 0$, then q must lie on line $y = \epsilon$ or $y = 1$. In both cases we can observe that the triangle qrt contains less than $3n$ points.

Subcase 4.3: Assume that r lies on line $y = 0$.

If s lies on line $y = 0$, then triangle rst contains less than $9n + 1$ points in its interior. If s lies on line $y = \epsilon$, then q must lie on line $y = \epsilon$ or line $y = 1$. In both cases we can observe that the triangle qst contains less than $3n$ points. Similarly, if s lies on line $y = 1$, then q must lie on line $y = \epsilon$ or $y = 1$. In both cases we can observe that the triangle qst contains less than $3n$ points.

The above case analysis proves that if $\mathcal{G}$ admits a point-set embedding on $\mathcal{S}_j$, then the supernodes must be mapped to some subset of C. We now prove that each supernode must be mapped to a distinct point set P_i.

A *layered drawing* of a plane graph G is a straight-line drawing of G, where the vertices are placed on horizontal lines called layers. Observe that every supernode takes at least four layers in any of its layered drawing. With this observation the proof becomes simple, as follows. The number of vertices in all the supernodes is equal to the number of points in C. Therefore, C must be covered by n disjoint triangles each containing $9n + 4$ points. Let these triangles be $t_0, t_1, \ldots, t_{n-1}$ such that any horizontal ray along $y = 0$ starting from $(0, 0)$ crosses t_{i-1} before t_i, $1 \leq i \leq n - 1$. Observe that t_0 must contain both the points $(0, 1)$ and $(0, -1)$. Otherwise, none of $t_1, \ldots, t_{n-1}$ will contain those two points and $t_0, t_1, \ldots, t_{n-1}$ will not cover C. If t_0 contains both the points $(0, 1)$ and $(0, -1)$, then it is straightforward to observe that these two points will lie on the boundary of t_0 and the other point on the boundary of t_0 will be $(1, 0)$. Therefore, t_0 will cover the points in P_0. We now can prove recursively that t_i must cover the points of P_i, which implies that each supernode of $\mathcal{G}$ must be mapped to a distinct P_i.

(c) For any given mapping of the outer vertices of a supernode N to the three points on the convex hull of P_i, we can construct a point-set embedding of N on P_i as follows. Let a, b, c and p be the outer vertices and the representative vertex of N, respectively. Map the vertex p to the point $m \in P_i$, where m is the $(3n + 1)$-th point of Y_i in the increasing order of x-coordinates. Observe that the mappings of a, b, c and p partition the interior region of the convex hull of P_i into three new regions, R_1, R_2 and R_3, each containing $3n$ collinear points. On the other hand, the subgraph induced by the vertices interior to each of C_{abp}, C_{bcp} and C_{cap} is a path of $3n$ vertices. Therefore, it is straightforward to find the mappings for these three subgraphs inside

regions R_1, R_2 and R_3, consistently with the mappings of the vertices a, b, c and p. ■

3.2.3 NP-completeness

We now use the reduction described above to prove the following theorem.

Theorem 1 *PSE is NP-complete.*

Proof. Given a mapping of the vertices of the input 3-connected planar graph G to the input point set S, it is straightforward to check if the drawing determined by this mapping is a straight-line drawing of G in polynomial time. Therefore, the problem is in NP.

We now prove NP-hardness. Let M be a given instance of HC, where M has n vertices. We construct the corresponding graph $\mathcal{G}$ with $9n^2 + 7n$ vertices and a set $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{3n-1}\}$ of $3n - 1$ point sets, where each $\mathcal{S}_j, 1 \leq j \leq 3n - 1$, contains $9n^2 + 7n$ points. The number of vertices of $\mathcal{G}$ and the number of points in $\mathcal{S}$ are polynomials in n. Moreover, by Property (a) of Lemma 2, the coordinates of the points are bounded by polynomials in n. Consequently, we can construct $\mathcal{G}$ and $\mathcal{S}$ in polynomial time. We now prove that M contains a Hamiltonian cycle if and only if $\mathcal{G}$ admits a point-set embedding on some $\mathcal{S}_j$.

We first assume that M contains a Hamiltonian cycle H and then give a construction of a point-set embedding of $\mathcal{G}$ on some $\mathcal{S}_j$ as follows. Choose a plane embedding Γ of M, where Γ contains an edge of H on the outer face. Let $H = (u_0, u_1, \ldots, u_{n-1}, u_0)$ and let (u_0, u_{n-1}) be the edge of H that lies on the outer face in Γ. H partitions the faces of M into two sets F_1 and F_2, where F_1 contains the faces outside of H and F_2

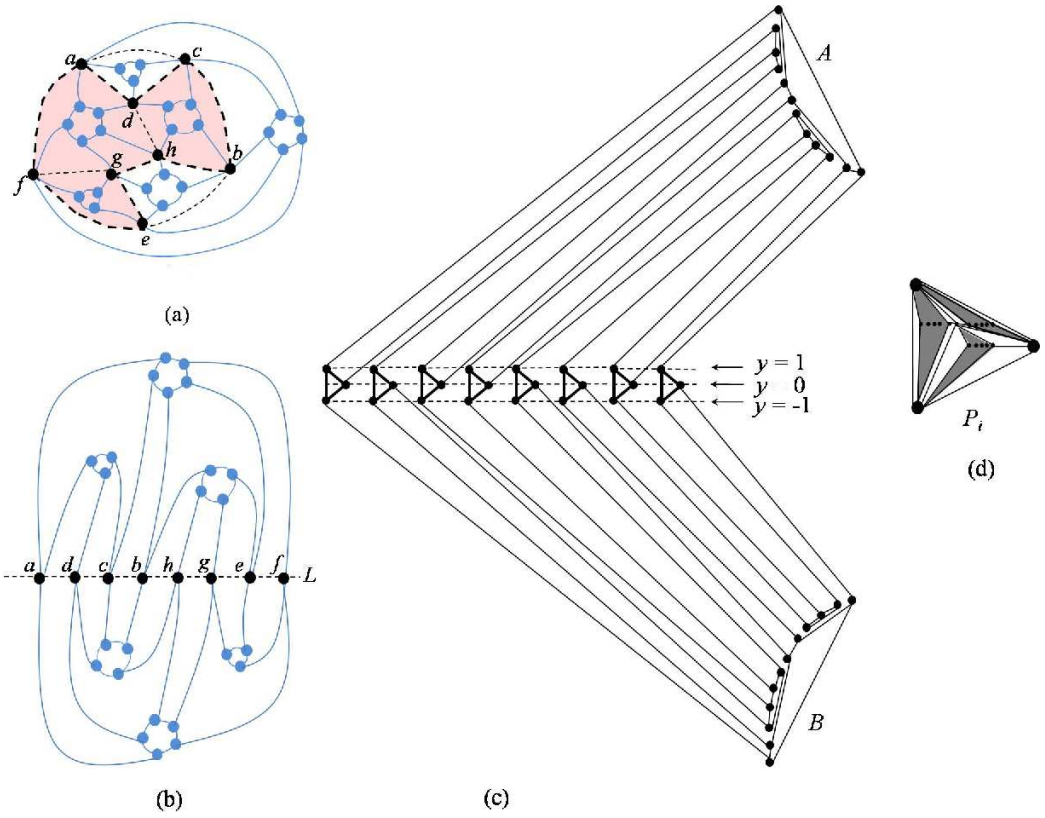

Figure 3.5: (a) Illustration for M and $\mathcal{G}$, where the supernodes are shown in black disks. The edges of M and $\mathcal{G}$ are shown in dashed and solid lines, respectively. A Hamiltonian cycle of M is shown in bold. (b) A plane embedding of M, where the vertices that belong to M are on line L. (c) Illustration of a point-set embedding of $\mathcal{G}$ on $\mathcal{S}_{l_1}$. (d) Mapping of a supernode on some P_i.

contains the faces interior to H. In any plane embedding Γ' of M, where the outer face is similar to the outer face of Γ, if the path $u_0, u_1, \ldots, u_{n-1}$ lies along a horizontal line L, then the faces interior to H will be on one side (above or below) of L and all the other faces of M will be on the other side of L. See Figures 3.5(a) and (b). Recall that each face of M corresponds to a ring of $\mathcal{G}$. Let l_1 and l_2 be the number of vertices in all the rings that correspond to the faces in F_1 and F_2, respectively. Then

$l_1 + l_2$ is twice the number of edges of M, i.e., $l_1 + l_2 = 3n$. In the following we show a construction of a point-set embedding of $\mathcal{G}$ on $\mathcal{S}_{l_1}$.

Recall that $\mathcal{S}_{l_1}$ consists of the point sets A, B and $P_i, 0 \leq i \leq n-1$, where $|A| = l_1, |B| = l_2$ and $|P_i| = 9n + 4$. First map the rings that correspond to the faces in F_1 and F_2 on the point set A and B, respectively, such that the order of the rings corresponds to the order of the faces in Γ'. We now only need to map the supernodes of $\mathcal{G}$ into P_i, where we can draw the edges between the rings and the supernodes avoiding edge crossings. Recall that each supernode corresponds to a node u_i of M. Since H passes through two adjacent edges at each $u_i, 0 \leq i \leq n-1$, the three faces incident to u_i cannot be all in F_1 or F_2. Hence, each ring must be connected with a supernode with at most two edges. Using this observation along with Property (c) of Lemma 2, it is now straightforward to map the supernode corresponding to u_i into P_i avoiding any edge crossings. An example is illustrated in Figures 3.5(c) and (d).

We now assume that $\mathcal{G}$ admits a point-set embedding Γ on some $\mathcal{S}_j$ and then prove that M contains a Hamiltonian cycle as follows. By Property (b) of Lemma 2, each supernode of $\mathcal{G}$ is mapped to a distinct P_i. Let u_i be the vertex of M that corresponds to the supernode mapped to P_i. We then claim that $u_0, u_1, \ldots, u_{n-1}, u_0$ is a Hamiltonian cycle in M. Otherwise, assume that $(u_i, u_{(i+1) \bmod n})$ is not an edge of M. From the construction of $\mathcal{G}$, observe that if we consider each ring and each supernode of $\mathcal{G}$ as a giant node, then each face of $\mathcal{G}$ is a quadrilateral that contains two rings and two supernodes on its boundary. Therefore, if $(u_i, u_{(i+1) \bmod n})$ is not an edge of M, then two rings will be inside the same face of M, deriving a contradiction

to the fact that each ring of $\mathcal{G}$ corresponds to a distinct face of M. ■

Although our NP-hardness proof is inspired by the NP-hardness proofs for some other problems presented in [39, 51], our reduction technique is significantly different from the techniques used in [39, 51] in several aspects. It may initially appear that we have proved NP-hardness for a version of the point-set embeddability problem in which the number of points is greater than the number of vertices. Note that we have reduced an instance of HC to a polynomial number of instances of PSE. If there exists a polynomial-time algorithm for PSE, then we can simulate that algorithm for those instances of PSE to obtain the answer to the instance of HC in polynomial time, which is impossible if P $\neq$ NP.

Since we reduce an instance of HC to a polynomial number of instances of PSE, our reduction is a Cook reduction [65, Page 177]. Traditionally NP-completeness is defined in terms of Karp reduction or polynomial-time many-one reduction [44, Page 98]. Recently, Biedl and Vatshelle [9] have strengthened our result by proving that PSE is NP-complete under Karp reduction even when the input points are in general position.

Chapter 4

Faster Point-Set Embeddings of Plane 3-Trees

In the last few years researchers have examined the point-set embeddability problem restricted to plane 3-trees. Nishat et al. [62] first gave an algorithm for deciding point-set embeddability of plane 3-trees in $O(n^2)$ time. They also proved an $\Omega(n \log n)$ lower bound on the time complexity of any algorithm that decides point-set embeddability for plane 3-trees. Later, Durocher et al. [32] and Moosa and Rahman [60] independently improved the running time of the algorithm of Nishat et al. to $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$. Since $\Omega(n^{4/3})$ is a lower bound on the worst-case time complexity for solving various geometric problems [33, 34], it may be natural to accept the possibility that the $O(n^{4/3+\varepsilon})$-time algorithm could be asymptotically optimal. In fact, Moosa and Rahman mention that an $o(n^{4/3})$-time algorithm does not seem to be likely using the currently known techniques. However, here we prove that the $\Omega(n \log n)$ lower bound of Nishat et al. is tight by giving an $O(n \log^3 n)$-time algorithm for deciding

point-set embeddability of plane 3-trees.

Both Durocher et al. [32] and Moosa and Rahman [60] use the basic ideas developed by Nishat et al. [62] and achieve the speed up using a triangular range search data structure. We also use the basic techniques of Nishat et al.'s algorithm, but we do not use any range search data structure. Instead, we make some simple but crucial observations that help exploit a dynamic convex hull data structure to achieve even a faster algorithm.

We first briefly describe the techniques used in the three known algorithms mentioned above and then give the details of our $O(n \log^3 n)$-time algorithm for deciding point-set embeddability of plane 3-trees.

Before going into the details, we review a few more definitions. Let S be a set of n points in the plane. We denote by $|S|$ the number of points in S. Let p, q and r be three points that do not necessarily belong to S. Then $S(pqr)$ consists of the points of S that lie either on the boundary or in the interior of the triangle pqr.

4.1 Overview of Known Algorithms

We first describe the basic techniques developed by Nishat et al. [62] for testing point-set embeddability of plane 3-trees.

Let G be a plane 3-tree with n vertices, and let a, b, c and p be the three outer vertices and the representative vertex of G, respectively. Nishat et al. used the following steps to test and compute point-set embeddings of G on S.

Step 1. Let C be the convex hull of S. If the number of points on the boundary of C is not exactly three, then G does not admit a point-set embedding on S. Otherwise, let x, y, z be the points on C.

Step 2. For each of the possible six different mappings of the outer vertices a, b, c to the points x, y, z, execute Step 3.

Step 3. Let n_1, n_2 and n_3 be the number of vertices of $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$, respectively. Without loss of generality assume that the current mapping of a, b and c is to x, y and z, respectively. Find the unique mapping of the representative vertex p of G to a point $w \in S$ such that the triangles xyw, yzw and zxw properly contain exactly n_1, n_2 and n_3 points, respectively. If no such mapping of p exists, then G does not admit a point-set embedding on S for the current mapping of a, b, c to x, y, z; hence go to Step 2 for the next mapping. Otherwise, recursively compute point-set embeddings of $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$ on $S(xyw), S(yzw)$ and $S(zxw)$, respectively. See Figure 4.1.

The time complexity of any algorithm that uses Steps 1–3 is dominated by the

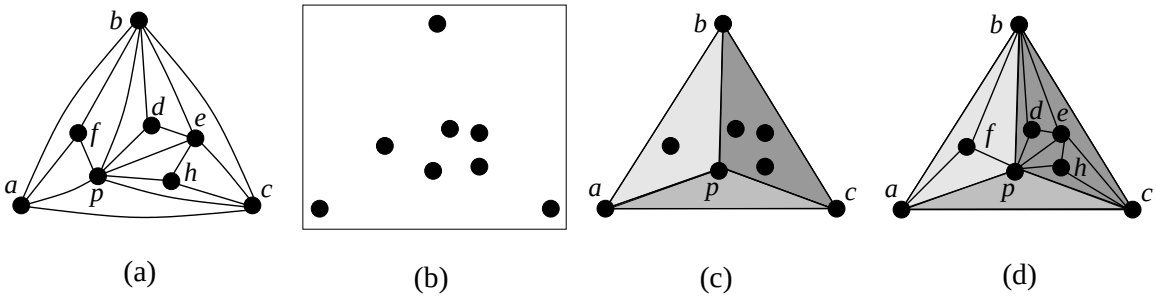

Figure 4.1: (a) A plane 3-tree G. (b) A point set S. (c) A valid mapping of the representative vertex of G. (d) Illustration for the recursive computation.

cost of Step 3 and the bottleneck of this step is the recursive computation of the mappings of the representative vertices. It is straightforward to observe that the recurrence relation for the time taken in Step 3 is $T(n) = T(n_1) + T(n_2) + T(n_3) + \mathcal{T}$, where $\mathcal{T}$ denotes the time required to find the mapping of the representative vertex.

Using brute force, one can find a mapping of the representative vertex by checking for each point w, interior to the triangle xyz, the number of points of S inside the triangles xyw, yzw and zxw. But such an approach takes $\mathcal{T} = O(n^2)$ time implying an overall time complexity of $O(n^3)$.

The algorithm of Nishat et al. [62] preprocesses the set S in $O(n^2)$ time so that the computation for the mapping of a representative vertex takes $O(n)$ time. Hence $\mathcal{T} = O(n)$ and the overall time complexity of their algorithm becomes $O(n^2)$.

Moosa and Rahman [60] used a binary search technique with the help of a triangular range search data structure of Chazelle et al. [23] to prune a constant fraction of points from consideration so that one can find the mapping of a representative vertex by testing only $\min\{n_1, n_2, n_3\}$ points interior to the triangle xyz. A triangular range counting query using the data structure of Chazelle et al. takes $O(n^{1/3+\epsilon})$ time, which implies that $\mathcal{T} = \min\{n_1, n_2, n_3\} \cdot n^{1/3+\epsilon}$ and $T(n) = O(n^{4/3+\epsilon})$, which coincides with the preprocessing time. The algorithm of Durocher et al. [32] uses the same idea, but instead of using a binary search their pruning technique uses a randomized search.

4.2 Embedding Plane 3-Trees in $O(n \log^3 n)$ time

We now give an algorithm for testing point-set embeddability of a plane 3-tree in $O(n \log^3 n)$ time.

We speed up the mapping of the representative vertex as follows. We first select $O(\min\{n_1+n_2, n_2+n_3, n_1+n_3\})$ points interior to the triangle xyz in $O(\min\{n_1, n_2, n_3\} \log^2 n)$ time using a dynamic convex hull data structure. We prove that these are the only candidates for the mapping of the representative vertex. We then make some tricky observations to test and compute a mapping for the representative vertex in $O(\min\{n_1, n_2, n_3\})$ time. As a consequence, we obtain $\mathcal{T} = O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ and a running time of $T(n) = O(n \log^3 n)$, which dominates the $O(n \log^2 n)$ time for building the initial dynamic convex hull data structure.

We are now ready to describe our algorithm. In the following we use three lemmas to obtain our main result. Lemma 3 selects a region R containing the candidate points inside the triangle xyz. Lemma 4 reduces the problem of finding a mapping inside the triangle xyz to the problem of finding a point satisfying specific criteria inside R. Lemma 5 gives an efficient technique to find such a point. Finally, we use these lemmas to obtain a mapping for the representative vertex in $O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ time, which results in an $O(n \log^3 n)$-time algorithm for testing point-set embeddability of plane 3-trees.

Let G be a plane 3-tree with $n \geq 4$ vertices, let a, b, c be the three outer vertices of G, and let p be the representative vertex of G, respectively. Assume that $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$ contain n_1, n_2 and n_3 vertices, respectively. Without loss of generality assume that $n_3 \leq n_2 \leq n_1$. Observe that $n_1 + n_2 + n_3 - 5 = n$. Let S be a set of n points in general position such that the convex hull of S contains exactly three points x, y, z on its boundary. Without loss of generality assume that the vertices a, b, c are mapped to the points x, y, z, respectively.

Let u and v be two points on the straight line segment xz such that $|S(uxy)| = n_1 - 1$ and $|S(vzy)| = n_2 - 1$, as shown in Figure 4.2(a). It is straightforward to verify that if a *valid mapping* for the representative vertex exists (i.e., there exists a point $w \in S$ such that $|S(wxy)| = n_1$, $|S(wyz)| = n_2$ and $|S(wzx)| = n_3$), then the corresponding point (i.e., the point w) must lie inside $S(uvy)$. Let r and s be two points on the straight line segments uy and vy, respectively, such that $|S(rux)| = |S(svz)| = n_3 - 1$. We call the region defined by the simple polygon x, u, v, z, s, y, r, x the *region of interest*. An example is illustrated in Figure 4.2(b). We now have the following lemma.

Lemma 3 *Assume that there exists a valid mapping for the representative vertex of G and $w \in S$ is the point that corresponds to this valid mapping. Then the straight line segments wx, wy and wz lie inside the region of interest R. Furthermore, the number of points in R that belongs to S is $O(n_3)$, and the following properties hold.*

(*a*) *If the points s, y, z (respectively, points r, x, y) are distinct, then $|S(syz)| = n_2 - n_3 + 2$ (respectively, $|S(rxy)| = n_1 - n_3 + 2$).*

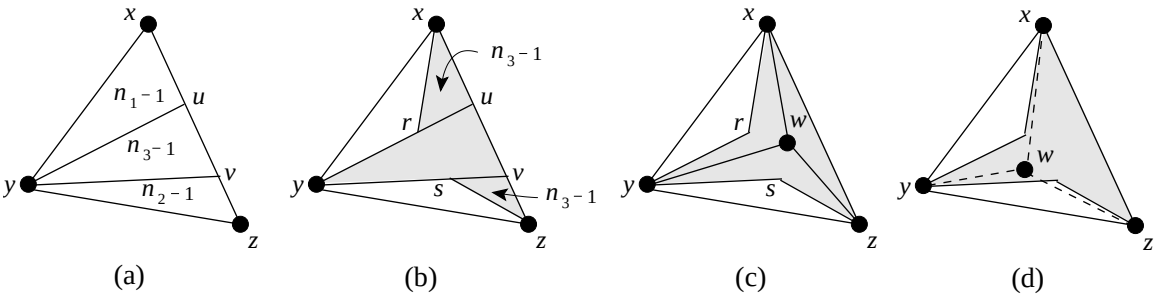

Figure 4.2: (a)–(b) Illustration for the lines uy, vy, xr and zs. The region of interest is shown in gray. (c)–(d) Illustration for the proof of Lemma 3.

(*b*) *Otherwise, point* s (*respectively, point* r) *coincides with* y (*respectively,* y) *and* $|S(syz)| = 2$ (*respectively,* $|S(rxy)| = 2$).

Proof. The point w must be in $S(uvy)$. Otherwise, either $|S(wxy)| < n_1$ or $|S(wyz)| < n_2$ holds, which implies that w does not correspond to a valid mapping. We now claim that the straight line segments wx, wy and wz lie interior to R, as shown in Figure 4.2(c). Since $w \in S(uvy)$, the straight line segment wy must lie inside R. Suppose for a contradiction that either one of wx, wz or both properly crosses the boundary of R. In this situation $S(wxz)$ must contain one of $S(rux)$ and $S(svz)$ implying that $|S(wxz)| > n_3$, as shown in Figure 4.2(d). Consequently, wx, wy and wz must lie interior to R. By the construction of R, the number of points that lie on the boundary and the interior of R is at most $(n_3 - 1) + 2n_3 = O(n_3)$.

We now determine the value of $|S(syz)|$. If the points s, y, z are distinct, then the triangle syz contains $(n_2 - 1) - (n_3 - 1) - 1$ points of S in its proper interior and three points (i.e., y, z and the other point on line sz) of S on its boundary. Therefore, $|S(syz)| = n_2 - n_3 + 2$. Otherwise, if point s coincides with point y, then $n_3 = n_2$ and $S(syz)$ consists of exactly two points of S, i.e., y and z. Since y and z are distinct by assumption, we are only left with the case when s coincides with z. But this case does not arise since $n \geq 4$ and hence $n_3 \geq 3$.

We can determine the value of $|S(rxy)|$ in a similar way. ∎

Let $S' \subseteq S$ be the set that consists of the points lying on the boundary of R and the points lying in the proper interior of R. We call S' the *set of interest*. By Lemma 3, $|S'| = O(n_3)$. We now reduce the problem of finding a valid mapping for

the representative vertex in S to the problem of finding a point with certain properties in S', as shown in the following lemma.

Lemma 4 *There exists a valid mapping for the representative vertex of G in S if and only if there exists a point $w' \in S'$ such that $|S'(w'yz)| = n_2 - |S(yzs)| + 3$, $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$.*

Proof. Assume that there exists a valid mapping for the representative vertex of G in S and the point corresponding to the valid mapping is w. By Lemma 3, $w \in S'$. We now prove that if we choose $w = w'$, then $|S'(w'yz)|$, $|S'(w'xy)|$ and $|S'(w'xz)|$ must have the required number of points.

Observe that the number of points in the proper interior of triangle yzs is $|S(yzs)|-3$. All the points on the boundary of yzs are also on the boundary of R. Since w corresponds to a valid mapping in S, $|S(wyz)| = n_2$. Consequently, $|S'(w'yz)| = |S(wyz)| - |S(yzs)| + 3 = n_2 - |S(yzs)| + 3$. We can prove that $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$ in a similar way.

Assume now that there exists a point $w' \in S'$ such that $|S'(w'yz)| = n_2-|S(yzs)|+3$, $|S'(w'xy)| = n_1-|S(xyr)|+3$ and $|S'(w'xz)| = n_3$. We now prove that $|S(wxy)| = n_2$. Since the number of points in the proper interior of triangle yzs is $|S(yzs)| - 3$, $|S(wxy)| = |S'(w'xy)| + |S(yzs)| - 3 = n_2$. Similarly, it is straightforward to verify that $|S(wxy)| = n_1$ and $|S(wyz)| = n_3$. ∎

Since a valid mapping for the representative vertex is unique, w' must be unique. We call the point w' the *principal point* of S'. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S. We now

have the following lemma.

Lemma 5 *Let S be a set of $t \geq 4$ points in general position such that the convex hull of S is a triangle xyz. Let i, j, k be three non-negative integers, where $i \geq 3, j \geq 3$ and $k = t + 5 - i - j$. Then we can decide in $O(t)$ time whether there exists a point $w \in S$ such that $|S(wxy)| = i, |S(wyz)| = j$ and $|S(wxz)| = k$, and compute such a point if it exists.*

Proof. Consider first a variation of the problem, where we want to construct a point $m \notin S$ interior to xyz such that $|S(mxy)| = i + 1, |S(myz)| = j - 1$ and $|S(mxz)| = k - 1$. Steiger and Streinu [74] proved that such a point always exists and gave an $O(t)$-time algorithm to find m.

We now claim that if there exists a point $m' \neq m$ such that $|S(m'xy)| = i + 1, |S(m'yz)| = j - 1$ and $|S(m'xz)| = k - 1$, then the sets $S(m'xy), S(m'yz)$ and $S(m'xz)$ must coincide with the sets $S(mxy), S(myz)$ and $S(mxz)$. To verify the claim assume without loss of generality that $m' \in S(myz)$. Since the triangle $m'yz$ lies interior to the triangle myz, the sets $S(m'yz)$ and $S(myz)$ must be identical.

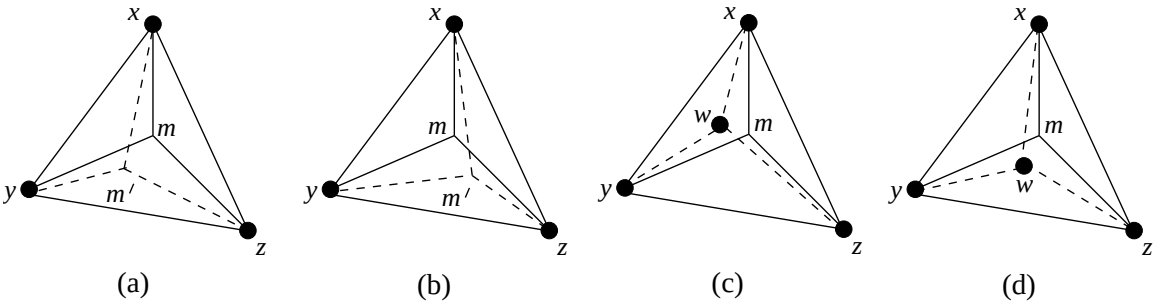

Figure 4.3: Illustration for the proof of Lemma 5, where only those points that are in S are shown in solid disks.

On the other hand, either the triangle mxz lies interior to the triangle $m'xz$, or the triangle mxy lies interior to the triangle $m'xy$, as shown in Figures 4.3(a)–(b). Therefore, either the sets $S(mxz)$ and $S(m'xz)$, or the sets $S(mxy)$ and $S(m'xy)$ must be identical. Consequently, the remaining pair of sets must also be identical.

Observe that if the point $w \in S$ we are looking for exists, then w must lie interior to $S(mxy)$, as shown in Figure 4.3(c). Otherwise, if $w \in S(myz)$ (respectively, $w \in S(mxz)$), then $|S(myz)| \geq |S(wyz)| = j$ (respectively, $|S(mxz)| \geq |S(wxz)| = k$), which contradicts our initial assumption that $|S(myz)| = j - 1$ (respectively, $|S(mxz)| = k - 1$). Figure 4.3(d) depicts such a scenario.

It is now straightforward to observe that if w exists, then the convex hull of $S(mxy)$ must be a triangle xym'', where $m'' \in S(mxy)$. If $|S(m''xy)| = i, |S(m''yz)| = j$ and $|S(m''xz)| = k$, then m'' is the required point w. Otherwise, no such w exists.

We can test whether the convex hull of $S(mxy)$ is a triangle in $O(t)$ time (e.g., find the leftmost point a, the rightmost point b and the point c with the largest perpendicular distance to the line determined by the line segment ab, and then test whether triangle abc contains all the points). It is also straightforward to compute the values $|S(m''xy)|, |S(m''yz)|$ and $|S(m''xz)|$ in $O(t)$ time. ∎

Given the set of interest $S' \subseteq S$, we can use Lemma 5 to find the principal point $w' \in S'$ in $O(n_3)$ time. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S. We now show how to compute the set S' in $O((n_2 + n_3) \log^2 n)$ time using the dynamic planar convex hull data structure of Overmars and van Leeuwen [64]. Recall that such a data structure supports a single update (i.e., a single insertion or deletion) in $O(\log^2 n)$ time. We refer the reader to

Figure 4.4(a) to recall the definition of the region of interest.

Step A. Assume that the points of S are placed in a dynamic convex hull data structure $\mathcal{D}$. We recursively delete the neighbor of y on the boundary of the convex hull of S starting from z in anticlockwise order, as shown in Figures 4.4(c)–(d). After deleting $n_2 - 2$ points, we insert all the deleted points into a new dynamic convex hull data structure $\mathcal{D}'$. We then insert a copy of the point y into $\mathcal{D}'$. Observe that all the points of $S(vyz)$ are placed in $\mathcal{D}'$. In a similar way we construct another dynamic convex hull data structure $\mathcal{D}''$ that maintains all the points of $S(uvy)$. Consequently, $\mathcal{D}$ now only maintains the points of $S(uxy)$. Since a single insertion or deletion takes $O(\log^2 n)$ time, all the above $O(n_2 + n_3)$ insertions and deletions take $O((n_2 + n_3)\log^2 n)$ time in total.

Step B. We now construct two other dynamic convex hull data structures $\mathcal{D}_1$ and $\mathcal{D}_2$ using $\mathcal{D}$ and $\mathcal{D}'$ such that they maintain the points of $S(rux)$ and $S(svz)$, respectively. Since $|S(rux)| + |S(svz)| = O(n_3)$, construction of $\mathcal{D}_1$ and $\mathcal{D}_2$ takes $O(n_3 \log^2 n)$ time.

Step C. We now construct the point set S' using the points maintained in $\mathcal{D}'', \mathcal{D}_1$ and $\mathcal{D}_2$, which also takes $O(n_3 \log^2 n)$ time. Note that in a similar way we can restore the original point set S and the initial data structure $\mathcal{D}$ in $O((n_2 + n_3)\log^2 n)$ time.

The time for the construction of S' using Steps A–C is $O((n_2 + n_3)\log^2 n)$, which dominates the time required for the computation of the valid mapping of the representative vertex p. Let w be the point that corresponds to the valid mapping. We now need to construct the point sets $S(wxy), S(wyz)$ and $S(wzx)$ for recursively testing the point-set embeddability of $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$, respectively. We can

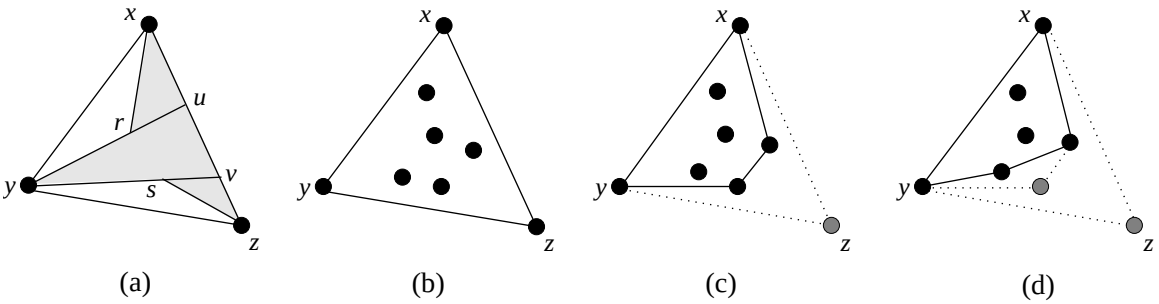

Figure 4.4: (a) Illustration for triangles uvy, rux and svz. The region of interest is shown in gray. (b)–(d) Recursive deletion of the neighbor of y on the boundary of the convex hull starting from z in anticlockwise order.

construct $S(wxy), S(wyz)$ and $S(wzx)$ and their corresponding dynamic convex hull data structures in $O((n_2+n_3)\log^2 n)$ time as follows. Let l be the point of intersection of the straight lines determined by the line segments wy and xz. First construct the set $S(lyz)$ and then modify it to obtain the sets $S(wyz)$ and $S(lwz)$, which takes $O((n_2+n_3)\log^2 n)$ time. Now modify the set $S(lxy)$ to construct the set $S(lwx)$, and then use the sets $S(lwx)$ and $S(lwz)$ to construct $S(wxz)$, which takes $O(n_3 \log^2 n)$ time. Observe that after the modification of the set $S(lxy)$, we are left with the set $S(wxy)$.

We now show that the total time taken by our algorithm is $T(n) \leq dn \log^3 n$, for

some constant d as follows. Let $c > 0$ be the constant hidden in the O-notation. Then

$$\begin{aligned}
T(n) &= T(n_1) + T(n_2) + T(n_3) + c \cdot \min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n \\
&\leq T(n_1) + T(n_2) + T(n_3) + c(n_2 + n_3) \log^2 n \\
&\leq d(n_1 \log^3 n_1) + d(n_2 \log^3 n_2) + d(n_3 \log^3 n_3) + c(n_2 + n_3) \log^2 n \\
&< d(n_1 \log^3 n) + d(n_2 \log^3 \tfrac{n}{2}) + d(n_3 \log^3 \tfrac{n}{3}) + c(n_2 + n_3) \log^3 n \\
&< d(n_1 + n_2 + n_3) \log^3 n - d(n_2 + n_3) \log^3 n + c(n_2 + n_3) \log^3 n \\
&= dn \log^3 n - (d - c)(n_2 + n_3) \log^3 n,
\end{aligned}$$

where the last step holds as long as $d \geq c$. Observe that the construction of the initial data structure $\mathcal{D}$ takes $O(n \log^2 n)$ time, which is dominated by $T(n)$.

The dynamic planar convex hull of Brodal and Jacob [18] takes amortized $O(\log n)$ time per update. Therefore, using their data structure instead of Overmars and van Leeuwen's data structure [64] we can test the point-set embeddability for plane 3-trees in $O(n \log^2 n)$ time. The following theorem summarizes the result of this section.

Theorem 2 *Let G be a plane 3-tree with n vertices and let S be a set of n points in general position in $\mathbb{R}^2$. We can decide in $O(n \log^3 n)$ time (or, in amortized $O(n \log^2 n)$ time) whether G admits a point-set embedding on S and compute such an embedding if it exists.*

Observe that we assume the input points are in general position. However, under the assumption that the algorithms of Overmars and van Leeuwen [64] and Steiger and Streinu [74] can handle degenerate cases, it is straightforward to modify our algorithm for the case when the input points are not necessarily in general position.

Chapter 5

Universal Point Set

Let F_n be a subclass of planar graphs such that for each graph $G \in F_n$, $|V(G)| = n$ and let S be a set of k points on the Euclidean plane. If every graph $G \in F_n$ admits a point-set embedding on S, then S is called a *universal point set* for F_n. Similarly, if every graph $G \in F_n$ admits a k-bend point-set embedding on S, then S is called a *k-bend universal point set* for F_n.

In 1990, de Fraysseix et al. [29] proved that every planar graph admits a straight-line drawing on an $O(n) \times O(n)$ integer grid. Therefore, a point set of size $O(n^2)$ is universal for all planar graphs with n-vertices. They also proved a $n + (1 - \epsilon)\sqrt{n}$, $\epsilon > 0$, lower bound on the size of the universal point set that supports all the planar graphs with n vertices. Chrobak and Karloff [26] and then Kurowski [57] improved this lower bound to $1.098n$ and $1.235n$, respectively. A long-standing open question in graph drawing asks to design a set of $O(n)$ points, which is universal for all planar graphs with n vertices [16]. In 2002, Kobourov [55] asked for the smallest value m for which a universal point set S of size m does not exist, where $|S| = |V(G)| = m$.

He checked by exhaustive search that such a universal point set exists for all the cases when $|S| = |V(G)| \leq 14$. On the other hand, Chrobak and Karloff's result [26] implies an upper bound of 38 on m.

We begin this chapter by proving a tighter bound on m. We prove that no universal point set exists if $|S| = |V(G)| = 24$, which implies that $15 \leq m \leq 24$. To justify this claim we show that the number of plane 3-trees supported by any set of 24 points (i.e., a subset of triangulations of 24 points) is smaller than the number of planar 3-trees of 24 vertices. We then show that the proof for $1.235n$ lower bound on the size of the universal point set given by Kurowski [57] is erroneous, and hence the $1.098n$ lower bound previously obtained by Chrobak and Karloff [26] is still the best known.

Kaufmann and Wiese [52] proved that any set S of n points is 2-bend universal for all the planar graphs with n vertices. However, a 2-bend point-set embedding on S may take $O(W^3)$ area [43], where W is the length of the side of the smallest axis parallel square that encloses S. We give an algorithm to compute 2-bend point-set embeddings of plane 3-trees on any set of n points in general position in $O(W^2)$ area.

5.1 Negative Results on Universal Point Set

In this section we prove that no universal point set exists if $|S| = |V(G)| = 24$. Let $R(n)$ be the maximum number of triangulations, which are non-isomorphic planar 3-trees of n vertices, supported by any set of n points on the Euclidean plane. Since $|S| = |V(G)| = n$, we restrict our attention to the point sets that are *triangular*, i.e., the convex hull of the point set must contain exactly three extreme points. We

compute an upper bound of $R(24)$ and then show that this upper bound is smaller than the number of non-isomorphic planar 3-trees with 24 vertices.

5.1.1 Counting Triangulations that are Plane 3-Trees

Computing the number of triangulations supported by a triangular set of points is a very difficult task. Krasser [56] in his PhD thesis computed the maximum number of triangulations supported by any triangular point set with $n \leq 11$ points. For $n = 3, 4, \ldots, 11$, the maximum number of triangulations supported by any triangular point set of n points are $1, 1, 2, 8, 30, 150, 780, 4550, 26888$, respectively. It is straightforward to observe that these numbers are the upper bounds on $R(n)$, where $3 \leq n \leq 11$. However, there appears to be no nice formula for the maximum number of triangulations supported by a triangular point set of n points. In 1982, Ajtai et al. [2] proved a 10^{13n} upper bound on the maximum number of triangulations supported by n points. This upper bound was improved many times [30, 68, 71] and the currently best known upper bound is 30^n, which was established by Sharir and Sheffer [70].

Since $R(n)$ represents the number of triangulations that are non-isomorphic plane 3-trees, we can bound $R(n)$ by the following recurrence.

$$R(n) \leq \sum_{i+j+k=n-4} R(i+3) \cdot R(j+3) \cdot R(k+3), \tag{5.1}$$

where i, j, k are the number of points interior to the subtriangles formed by joining one of the $n-3$ points interior to the convex hull with the extreme points using straight line segments. However, this recurrence relation is not tight. Observe that since there are only $n-3$ interior points, at most $n-3$ distinct triples (i, j, k) can

appear. Let A be an array sorted in descending order that contains the products $R(i) \cdot R(j) \cdot R(k)$ for all possible i, j, k that sum to $n-4$, i.e., there are $\binom{(n-4)+3-1}{n-4}$ entries in A. Then

$$R(n) \leq \sum_{1 \leq t \leq n-3} A_t, \tag{5.2}$$

where A_t is the t-th entry of array A. Consequently, we can compute an upper bound on $R(n)$ by a dynamic program using the values $1, 1, 2, 8, 30, 150, 780, 4550, 26888$, computed by Krasser [56] for the maximum number of triangulations of $3 \leq n \leq 11$ points, as the base cases. The first few values that bounds $R(n)$ are given in Table 5.1.

Table 5.1: Upper Bounds on $R(n)$

n	Upper Bound on $R(n)$	n	Upper Bound on $R(n)$
12	103414	17	205451270
13	471570	18	984576198
14	2088970	19	4920478614
15	9509986	20	24719921462
16	43893198	21	124657496862

Observe that we can obtain tighter bounds on $R(n)$ if the numbers we use in the base cases correspond to the maximum number of triangulations that are plane 3-trees. Counting the exact values for $R(n)$ is a very difficult task since the number of point configurations grows very rapidly [56].

We perform an exhaustive search to compute the maximum number of triangulations that are plane 3-trees (not necessarily non-isomorphic) for small values of n. For $n = 3, 4, \ldots, 10$, these values are $1, 1, 2, 6, 18, 60, 222, 794$, respectively. We now

use Inequality (5.2) to have tighter bounds on $R(n)$, as listed in Table 5.2.

Table 5.2: Refined Upper Bounds on $R(n)$

n	Upper Bound on $R(n)$	n	Upper Bound on $R(n)$
11	3492	20	3593084172
12	15240	21	17613793476
13	68260	22	86924276676
14	310188	23	431716715388
15	1431564	24	2157412358124
16	6701900	25	10826649528624
17	31796964	26	54597730710384
18	152781060	27	276656076428724
19	738137148	28	1405392102247284

5.1.2 Counting Non-isomorphic Planar 3-Trees

Although it is a difficult task to find a tight bound on the maximum number of non-isomorphic plane 3-trees supported by any set of n points, the number $P(n)$ of non-isomorphic planar 3-trees with n vertices has a nice closed formula. Beineke and Pippert [6] proved that

$$P(n+3) = \frac{1}{12(n+1)}X(n) + \frac{5}{24}X(\tfrac{n}{2}) + \frac{1}{3}X(\tfrac{n-1}{3}) + \frac{1}{4}X(\tfrac{n-1}{4}) + \frac{1}{6}X(\tfrac{n-2}{6}) + \frac{3}{8}Y(n) + \frac{1}{6}Y(\tfrac{2n-1}{3}), \tag{5.3}$$

where any term $X(w)$ or $Y(w)$ is zero if w is not an integer. Otherwise, $X(z) = Y(2z) = \frac{(3z)!}{z!(2z+1)!}$ and $Y(2z+1) = \frac{(3z+1)!}{(z+1)!(2z+1)!}$.

Table 5.3 shows the values for $P(n)$ determined by Equation (5.3), where $12 \leq n \leq 29$. The corresponding sequence in Sloane's database [72] is $A027610$.

Table 5.3: $P(n), 12 \leq n \leq 29$

n	$P(n)$	n	$P(n)$
12	2110	21	11489753730
13	11002	22	68054102361
14	58713	23	405715557048
15	321776	24	2433003221232
16	1792133	25	14668536954744
17	10131027	26	88869466378593
18	57949430	27	540834155878536
19	334970205	28	3304961537938269
20	1953890318	29	20273202069859769

5.1.3 Comparison Between $R(n)$ and $P(n)$

We now prove a bound on the smallest value $m = |S| = |V(G)|$ for which a universal point set does not exist. Kobourov [55] showed that $m \geq 15$. Tables 5.2 and 5.3 show that the smallest integer n for which $R(n) < P(n)$ is 24. Therefore, we obtain the following theorem.

Theorem 3 *Let $m = |S| = |V(G)|$ be the smallest value for which a universal point set of size m does not exist. Then $15 \leq m \leq 24$.*

5.2 Kurowski's Proof

Kurowski [57] proved that any point set that is universal for all planar 3-trees with n vertices must contain at least $1.235n$ points. We prove that Kurowski's proof is erroneous.

Kurowski [57] defined a set T_n that contains all plane 3-trees with n vertices, and a parameter t_n that denotes the cardinality of T_n. He estimated that $t_3 = t_4 = 1$ and for $n > 4$,

$$\begin{aligned} t_n &= 3 \cdot 5 \cdot 7 \ldots (2n-9) \cdot (2n-7) \\ t_n &\geq \sqrt{(2n-7)!} \end{aligned} \tag{5.4}$$

The reason behind this estimate is as follows. Given an embedding of a plane 3-tree G with $n \geq 4$ vertices, one can construct a plane 3-tree with $n+1$ vertices by inserting a vertex into any of the $2n-5$ faces of G.

Kurowski claimed that if a point set with a points is universal for all the plane 3-trees with n vertices, then the number of plane 3-trees with n vertices supported by a is at most ${}^{a}\mathrm{P}_n = a \cdot (a-1) \cdot (a-2) \ldots (a-n+1)$. Consequently,

$$\begin{aligned} & a \cdot (a-1) \cdot (a-2) \ldots (a-n+1) \geq t_n \\ \implies & a \cdot (a-1) \cdot (a-2) \ldots (a-n+1) \geq \sqrt{(2n-7)!} \\ \implies & \left(a - \frac{n-1}{2}\right)^n > \left(\frac{2n-7}{e}\right)^{n-7/2}. \end{aligned} \tag{5.5}$$

Kurowski showed that Inequality (5.5) implies $a > 1.235n$ when n is sufficiently large.

Although t_n is the number of distinct plane 3-trees, in Inequality (5.4) Kurowski over-counted many isomorphic plane 3-trees. To justify our claim we review a bijection between plane 3-trees and ordered rooted ternary trees [59]. Given a plane 3-tree G_n with n vertices, the corresponding ternary tree T_{n-3} is an ordered rooted

tree of $n-3$ vertices, which we call the *representative tree* of G_n. Let a, b, c and p be the outer vertices and the representative vertex of G_n, respectively. Then the *representative tree of* G_n satisfies the following conditions.

(a) If $n = 3$, T_{n-3} consists of a single vertex.

(b) If $n > 3$, then the root p of T_{n-3} is the representative vertex of G_n and the subtrees rooted at the three counter-clockwise ordered children p_1, p_2 and p_3 of p in T_{n-3} are the representative trees of $G_n(C_{abp})$, $G_n(C_{bcp})$ and $G_n(C_{cap})$, respectively.

Figures 5.1(a) and (b) show a plane 3-tree and its representative tree, respectively. Zaks [86] showed that the number of full k-ary trees with t internal nodes is exactly $\frac{1}{kt+1}\binom{kt+1}{t} = \frac{1}{(k-1)t+1}\binom{kt}{t}$. Therefore,

$$\begin{aligned} t_n &\leq \frac{1}{2(n-3)+1}\binom{3(n-3)}{n-3} \\ &\leq \frac{1}{2n-5}\binom{3n-9}{n-3} \\ &\leq \frac{1}{2n-5}\binom{3n}{n} \\ &\leq \frac{6.75^n}{2n-5}\sqrt{\frac{6}{8\pi n}}, \end{aligned} \tag{5.6}$$

applying Stirling's approximation [76] $n! \approx \sqrt{2\pi n}(n/e)^n$. Observe that Inequality (5.6) contradicts Inequality (5.4) since $\sqrt{(2n-7)!} > \frac{6.75^n}{2n-5}\sqrt{\frac{6}{8\pi n}}$ for every $n > 20$.

5.3 Chrobak and Karloff's Proof

In this section we give a brief overview of Chrobak and Karloff's proof. Although Chrobak and Karloff's result [26] proves that no set of 38 points is universal for all

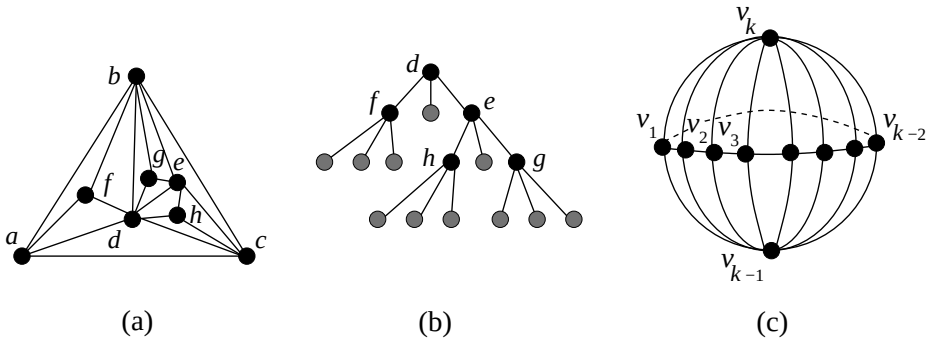

Figure 5.1: (a) A plane 3-tree G_n, where $n = 8$. (b) The representative tree T_{n-3} of G_n, where the nodes of T_{n-3} that correspond to the internal faces of G_n are shown in gray. (c) The graph G. (d) A point-set embedding Γ of G on S'.

planar graphs with 38 vertices, their bound is weaker than our upper bound of 24 as proved in Section 5.1.

Let k be a positive integer whose value will be specified later and let $C = v_1, v_2, \ldots, v_{k-2}, v_1$ be a cycle of $k-2$ vertices. Add two vertices v_{k-1} and v_k to C joining the edges (v_{k-1}, v_i) and (v_k, v_i), for all $1 \leq i \leq k-2$. Let G be the resulting graph. Observe that G contains exactly k vertices. Figure 5.1(c) illustrates the construction of G on a sphere to reflect the property that every face of G is symmetric.

Lemma 6 (Chrobak and Karloff [26]) *For any set S of a points, $a \geq k$, G admits at most $6(2k-4)(k-3)\binom{a}{k}$ different point-set embeddings on S.*

Chrobak and Karloff constructed a set $\mathcal{G}$ of graphs such that each graph in $\mathcal{G}$ contains exactly n vertices. Every graph $H \in \mathcal{G}$ is constructed from a copy of G. Recall that G is a triangulated graph of k vertices. Therefore, G has exactly $2k-4$ faces in any of its plane embeddings. Take a fixed plane embedding of G and assign

the labels $f_1, f_2, \ldots, f_{2k-4}$ to its distinct faces in some arbitrary order. For each solution to the equation $x_1 + x_2 + \ldots + x_{2k-4} = n - k$, insert a plane graph G_j with $x_j, 1 \leq j \leq 2k-4$, vertices into the face f_j of G such that the outer cycle of G_j coincides with the boundary of face f_j. Since the equation $x_1+x_2+\ldots+x_{2k-4} = n-k$ has at most $\binom{(n-k)+(2k-4)-1}{(2k-4)-1}$ solutions, the number of graphs in $\mathcal{G}$ is

$$|\mathcal{G}| = \binom{n+k-5}{2k-5}. \tag{5.7}$$

Fix an embedding for every graph H in $\mathcal{G}$. By Lemma 6, there are at least

$$\frac{\binom{n+k-5}{2k-5}}{6(2k-4)(k-3)\binom{a}{k}} \tag{5.8}$$

embeddings that correspond to a fixed point-set embedding of G on S.

Chrobak and Karloff then counted the number of ways a fixed point-set embedding of G on S can be extended to an embedding of a graph with n vertices. Observe that every extended embedding Γ corresponds to a unique sequence of integers $r_1, r_2, \ldots, r_{2k-4}$, where $r_j, 1 \leq j \leq 2k-4$, denotes the number of points of S that are interior to the i-th face of G and not chosen by Γ. Since for each solution to $\sum_{1 \leq j \leq 2k-4} r_j = a - n$, the number of possible ways a fixed point-set embedding of G on S can be extended is at most $\binom{a-n+2k-5}{2k-5}$. If S is an universal point set for planar graphs with n vertices, then the following inequality must hold.

$$\binom{a-n+2k-5}{2k-5} \geq \frac{\binom{n+k-5}{2k-5}}{6(2k-4)(k-3)\binom{a}{k}}. \tag{5.9}$$

Chrobak and Karloff showed that if $k = 0.133n$ and $a = 1.098n$, then for sufficiently large n the Inequality (5.9) does not hold. This implies that any universal point-set must have at least $1.098n$ points. If $a = n$, then the smallest value for n

such that the Inequality (5.9) does not hold is $a = n = 38$ and $k = 15$. Therefore, Chrobak and Karloff's result [26] implies that no set of 38 points is universal for all planar graphs with 38 vertices.

Observation 1 *Let $m = |S| = |V(G)|$ be the smallest value for which a universal point set of size m does not exist. Then $m \leq 38$.*

5.4 Universal Point Set for Plane 3-Trees

Kaufmann and Wiese [52] proved that any set S of n points is 2-bend universal for all the planar graphs with n vertices. However, a 2-bend point-set embedding on S may take $O(W^3)$ area [43, Theorem 7], where W is the length of the side of the smallest axis parallel square that encloses S. In this section we give an algorithm to compute 2-bend point-set embeddings of plane 3-trees on a set of n points in general position in $O(W^2)$ area.

Here is an outline of the algorithm. Given a plane 3-tree G and a set of points S in general position, we first construct a straight-line drawing Γ of G such that every point of S other than a pair of points on the convex hull of S lies in the proper interior of some distinct inner face in Γ, as shown in Figures 5.2(a)–(c). While constructing Γ, we compute a bijective function ϕ from the vertices of Γ to the points of S. We then extend each edge (u, v) in Γ using two bends to place the vertices u and v onto the points $\phi(u)$ and $\phi(v)$, respectively, as shown in Figure 5.2(d). We prove that Γ and ϕ maintains certain properties so that the resulting drawing Γ' remains planar.

In the following we describe the algorithm in detail. Let H be the convex hull of S. Construct a triangle xyz with $O(W^2)$ area such that xyz encloses H and the side

yz passes through a pair of consecutive points y', z' on the boundary of H. Assume that y' is closer to y than z'. Set $\phi(y) = y'$ and $\phi(z) = z'$. Set $\phi(x) = x'$, where x' is the point on the convex hull of $S(xyz)$ for which the angle $\angle xyx'$ is smallest. Figure 5.2(e) illustrates the triangle xyz and the function ϕ. We call the straight line segments $x\phi(x), y\phi(y), z\phi(z)$ the *wings* of xyz. Observe that only $x\phi(x)$ among the three wings of xyz lie in the proper interior of xyz. We use this invariant throughout the algorithm, i.e., every face f in the drawing will contain at most one wing that is in the proper interior of f. We call such a wing the *major wing* of f.

Let a, b, c be the outer vertices of G in anticlockwise order and let p be the representative vertex of G. Map the vertices a, b, c to the points x, y, z. Let $S \setminus \{x', y', z'\}$ be the point set S'. Let n_1, n_2 and n_3 be the number of inner vertices of $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$, respectively. Since the major wing of xyz is incident to x, we construct a point $w \notin S$ such that $S'(wxy) = n_1, S'(wyz) = n_2 + 1$ and $S'(wxz) = n_3$, as shown in Figure 5.2(e). Steiger and Streinu [74] proved that such a point always exists and gave an $O(|S'|)$-time algorithm to find w. Since the angle $\angle xy\phi(x)$ is the smallest, if wy or wz intersects $x\phi(x)$, then by continuity there must exist another point $\bar{w}$ on the line wz such that $S'(\bar{w}xy) = n_1, S'(\bar{w}yz) = n_2 + 1$, $S'(\bar{w}xz) = n_3$ holds, and we choose $\bar{w}$ as the point w. Figures 5.2(f)–(g) depict such scenarios. Set $\phi(w) = w'$, where w' is the point on the convex hull of $S'(wyz)$ for which the angle $\angle wyw'$ is smallest. Since wyz does not contain $x\phi(x)$, the mapping we compute maintains the invariant that every face contains at most one major wing.

We now recursively construct the drawings of $G(C_{abp}), G(C_{bcp})$ and $G(C_{cap})$ with the point sets $S'(wxy), S'(wyz) \setminus w'$ and $S'(wxz)$, respectively. Note that while re-

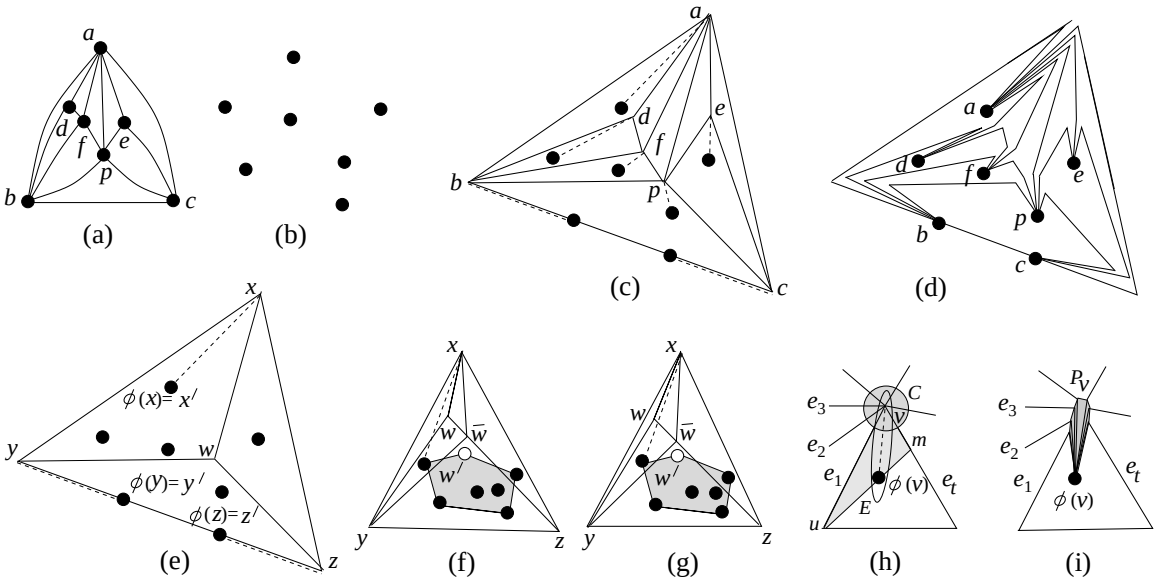

Figure 5.2: (a) A plane 3-tree G. (b) A set of points S. (c) Γ and ϕ, where ϕ is illustrated with dashed lines. (d) A 2-bend point-set embedding of G on S. (e) Illustration for the triangle xyz. (f)–(g) Construction of w and $\phi(w)$, where $\phi(w)$ is shown in white and the convex hull of $S(xyz)$ is shown in gray. (h) The region R and ellipse E, where R is shown in gray. (i) Illustration for P_v.

cursively constructing a point w for the representative vertex inside some triangle xyz, then the triangle may not have any major wing. Also in this case, it suffices to compute w such that $S'(wxy) = n_1, S'(wyz) = n_2 + 1$ and $S'(wxz) = n_3$ holds. Once we complete the recursive computation, we obtain a straight-line drawing Γ of G. Furthermore, we obtain a bijective function ϕ from the vertices of Γ to the points of S.

For each vertex v in Γ we now do the following. Let $e_1, e_2, \ldots, e_t$ be the edges adjacent to v in clockwise order such that e_1 and e_t forms the smallest angle that contains $\phi(v)$. Construct a strictly convex polygon $P_v = \phi(v), v_1, v_2, \ldots, v_t$, where

$v_i, 1 \leq i \leq t$, is a point on e_i and no point of S other than $\phi(v)$ lie inside the polygon. Now delete the straight line segments vv_i and draw the segments $\phi(v)v_i$, as shown in Figure 5.2(i). Since every face in Γ contains at most one major wing, we can construct the P_vs such that for two different vertices v_1 and v_2 in Γ, the corresponding convex polygons P_{v_1} and P_{v_2} are disjoint. Later in this section, We describe such a construction for P_vs.

We claim that the resulting drawing Γ' is a 2-bend point-set embedding of G on S. Since every edge in Γ has only two endpoints, the corresponding edge e in Γ' has exactly two bends. Since ϕ is a bijective function, e does not create any loop in Γ'. It now suffices to prove that Γ' is a planar drawing of G. Observe that Γ is a planar straight-line drawing. Therefore, if Γ' is not a planar drawing, then either two of the newly added segments properly intersect (Case 1), or a newly added segment intersects an old segment that originally belongs to Γ (Case 2). Case 1 cannot appear since all P_vs are disjoint. Case 2 cannot appear since every newly added segment lie in some convex polygon P_v that does not contain any old segment in Γ'.

We can construct Γ in $O(n \log^3 n)$ time in a similar technique as in Section 4.2. We now describe how to construct Γ' from Γ in $O(n \log n)$ time. Let $\mathcal{S}$ be a set that consists of the points corresponding to the vertices in Γ and the points that belong to S. Let η be the Euclidean distance between the closest pair of points in $\mathcal{S}$. We can compute η in $O(n \log n)$ time [69]. For each vertex v in Γ, we now construct the convex polygon P_v in $O(\deg(v))$ time as follows.

Let C be the circle centered at v with radius $\eta/2$. Assume that $e_1 = (u, v)$, and m is the intersection point of edge e_t and the line determined by $u, \phi(v)$. Let R be the

region determined by the union of C and the triangle uvm. Consider now an ellipse E with foci v and $\phi(v)$ such that the half of the ellipse (determined by the minor axis of E) that contains v lie interior to R. Then $v_1, v_2, \ldots, v_t$ are the intersection points of E with $e_1, e_2, \ldots, e_t$, respectively, as shown in Figures 5.2(h)–(i). Since R does not contain any point of S other than $\phi(v)$, the polygon $P_v = \phi(v), v_1, v_2, \ldots, v_t$ does not contain any point of S other than $\phi(v)$. Since every face in Γ contains at most one major wing and C is a circle with radius $\eta/2$, any two P_vs must be disjoint.

Consequently, the algorithm takes $O(n \log^3 n) + O(n \log n) + \sum_{\forall v} O(\deg(v)) = O(n \log^3 n)$ time. The following theorem summarizes the results of this section.

Theorem 4 *Given a plane 3-tree G with n vertices and and a point set S of n points in general position, we can compute a 2-bend point-set embedding of G in $O(n \log^3 n)$ time with $O(W^2)$ area, where W is the length of the side of the smallest axis parallel square that encloses S.*

Chapter 6

Convex Point-Set Embeddings of Klee Graphs

In this chapter we examine convex point-set embeddability of a subclass of 3-connected planar graphs, namely klee graphs. Given a klee graph G with n vertices and a set S of n points in general position on the plane, we give an $O(n^{8+\epsilon})$ expected time algorithm for computing a convex point-set embedding of G on S.

Recall that every klee graph can be constructed from K_4 by repeatedly replacing a vertex with a face of length three. Figure 6.1 illustrates the construction of some klee graphs.

We first introduce a few more definitions and notation. Let G be a planar graph. *Contraction of an edge* (u, v) of G is an operation that removes edge (u, v) from G by merging u and v, and then replaces any resulting multiedge (if exists) by a single edge. Let G' be the dual graph of G and let v be the vertex of G' that corresponds to the outer face of G. Then the *weak dual* G^* *of* G is the plane graph obtained by

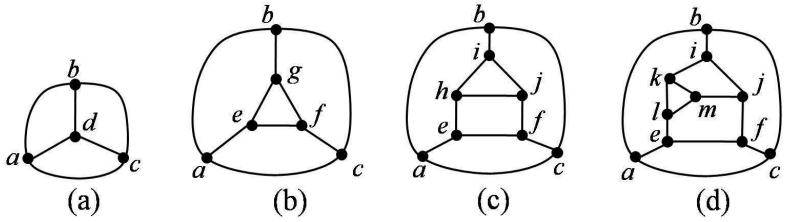

Figure 6.1: (a) The smallest klee graph, K_4. (b)–(d) A sequence of klee graphs constructed from K_4.

deleting the vertex v from G'.

Let S be a set of points on the Euclidean plane and let x, y, z be three points that do not necessarily belong to S. Recall that $S(xyz)$ consists of the points of S that lie either on the boundary or in the interior of the triangle xyz. Let $S'(xyz)$ be the set that consists of the points of S that are in the proper interior of triangle xyz. Let $s_1, s_2, \ldots, s_n$ be an ordered sequence of $n > 1$ points. By $\langle s_1, s_2, \ldots, s_n \rangle$ we denote a drawing of a path obtained by drawing a straight line segment between s_i and s_{i+1}, $1 \leq i < n$. We call $\langle s_1, s_2, \ldots, s_n \rangle$ a *convex path*, if joining s_1 and s_n with a straight line segment results in a convex polygon.

6.1 The Embedding Algorithm

In this section we describe our algorithm for testing convex point-set embeddability of klee graphs. We first give a polynomial-time algorithm to decide convex point-set embeddability for klee graphs that have exactly three outer vertices on the outer face. We then extend the algorithm to decide convex point-set embeddability for any klee graph in $O(n^{8+\epsilon})$ expected time. Our algorithm is motivated by the

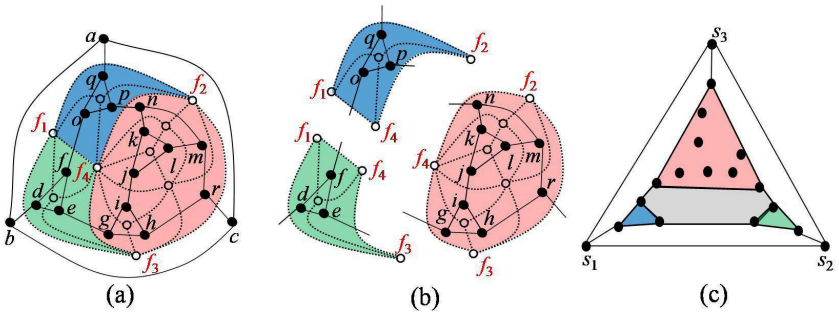

Figure 6.2: Illustration of the algorithm for convex point-set embedding.

recursive structure of plane 3-trees, i.e., every plane 3-tree with $n \geq 4$ vertices admits a decomposition into three smaller plane 3-trees [59].

Here is an outline of our algorithm when the klee graph G has exactly three outer vertices. We use dynamic programming to test whether G admits a convex point-set embedding on S. We observe that the weak dual of every klee graph with exactly three outer vertices is a plane 3-tree, which helps us to recursively divide the decision problem into three simpler subproblems. We solve those subproblems recursively and combine their results to obtain the result of the original decision problem. Figure 6.2(a) depicts a klee graph G, where the weak dual G^* is shown in dotted lines. The decomposition of G^* into three smaller plane 3-trees determines a decomposition of G into three subgraphs, as shown in Figure 6.2(b) in red, blue and green. Figure 6.2(c) depicts a point set S. If G admits a point-set embedding on S, where the outer vertices a, b, c of G are mapped respectively to the points s_1, s_2, s_3 of S, then those three subgraphs must be mapped to the point sets shown in the corresponding colors.

We now focus on some properties of a klee graph and its convex point-set embed-

ding.

Lemma 7 *Let G be a plane graph with exactly three outer vertices. Then G is a klee graph with exactly three outervertices if and only if the weak dual G^* of G is a plane 3-tree.*

Proof. We first assume that G is a klee graph and then prove that G^* is a plane 3-tree. Let n be the number of vertices of G. We employ an induction argument on n. If $n = 4$, then the weak dual G^* of G is a triangle which is a plane 3-tree. We may thus assume that $n \geq 4$ and the claim is true for all klee graphs with fewer than n vertices. We now consider the case when G has n vertices. By definition of a klee graph, G has a face f of length three, where contraction of the three edges on the boundary of f gives another klee graph G' with $n - 2$ vertices.

Since G has exactly three outer vertices, the vertices on f must be inner vertices of G. Therefore, the corresponding vertex x in G^* must be an inner vertex of degree three. The contraction of the edges on the boundary of f corresponds to the deletion of x from G^*. Observe that the graph obtained by deletion of x is the weak dual G'^* of G'. By induction G'^* is a plane 3-tree. Since G^* is obtained from G'^* by adding a vertex of degree three, G^* is triangulated. By definition G^* is a plane 3-tree.

We now assume that G^* is a plane 3-tree and then prove that the graph G having G^* as its weak dual is a klee graph with exactly three outervertices. We employ an induction argument on the number of vertices n^* of G^*. If $n^* = 3$, then G is the klee graph K_4. We thus assume that $n^* \geq 3$ and the claim holds for all G^* with fewer than n^* vertices. We now consider the case when G^* has n^* vertices. Since G^* is a

plane 3-tree, G^* has an inner vertex x of degree three whose removal yields another plane 3-tree G'^* of $n^* - 1$ vertices [59].

Since G has only three outer vertices, the vertex x in G^* corresponds to a face f of length three in G where the vertices on the boundary of f are inner vertices. The deletion of x from G^* corresponds to the contraction of the edges on f. The graph obtained after contraction of the edges on f is G', where G'^* is a weak dual of G'. By induction G' is a klee graph, and by definition G is a klee graph. ■

A *near klee graph* K is a plane graph that is obtained by deleting the outer vertices of some klee graph having exactly three outer vertices. Observe that every near klee graph with $n \geq 3$ vertices contains exactly three outer vertices of degree two. We call these vertices the *poles* of K. If K is obtained from a klee graph G, then the three outer vertices of G are the *legs* of K. The legs and the poles of the near klee graph obtained from the klee graph of Figure 6.2(a) are a, b, c and q, d, r, respectively. The following lemma proves that every near klee graph can be decomposed into three smaller near klee graphs.

Lemma 8 *Let K be a near klee graph with $n \geq 3$ vertices, which is obtained from a klee graph G with exactly three outer vertices. Let G^* be a weak dual of G, where the outer vertices of G^* are f_1, f_2, f_3 and the representative vertex of G^* is f_4. Then the weak duals of $G^*(C_{f_1f_2f_4})$, $G^*(C_{f_2f_3f_4})$ and $G^*(C_{f_3f_1f_4})$ are three vertex-disjoint near klee graphs, which are subgraphs of K.*

Proof. Let the the weak duals of $G^*(C_{f_1f_2f_4})$, $G^*(C_{f_2f_3f_4})$ and $G^*(C_{f_3f_1f_4})$ be H_1, H_2 and H_3, respectively. Since the faces of $G^*(C_{f_1f_2f_4})$, $G^*(C_{f_2f_3f_4})$ and $G^*(C_{f_3f_1f_4})$ are

disjoint, the vertex sets of H_1, H_2 and H_3 are disjoint.

By definition, $G^*(C_{f_1f_2f_4})$, $G^*(C_{f_2f_3f_4})$ and $G^*(C_{f_3f_1f_4})$ are plane 3-trees. If $H_i, 1 \leq i \leq 3$, is a single-vertex graph, then it is a subgraph of K. We may thus assume that H_i has more than one vertex. In this case the corresponding plane 3-tree has more than three vertices. Since the dual graph of the dual of a 3-connected graph $\mathcal{G}$ is $\mathcal{G}$ itself, and every plane 3-tree with more than three vertices is a 3-connected graph, H_i is a subgraph of K. We now prove that H_1 is a near klee graph. The proofs for H_2 and H_3 are similar.

The case when H_1 is a single-vertex graph is straightforward. We thus assume that H_1 has more than one vertex. Since H_1 is a weak dual of a plane 3-tree $G^*(C_{f_1f_2f_4})$, H_1 has exactly three outer vertices p, q and r of degree two. We first insert H_1 into the interior of a cycle C of length three. Let u, v and w be the vertices on C. We then add the edges $(u, p), (v, q)$ and (w, r). Let the resulting graph be G'. Observe that the weak dual of G' is the plane 3-tree $G^*(C_{f_1f_2f_4})$. By Lemma 7, G' is a klee graph. Since H_1 can be obtained from G' by removing the outer vertices of G, therefore H_1 is a near klee graph. ■

Let G be a klee graph with exactly three outer vertices and let K be the corresponding near klee graph. By a *klee partition* of G we denote the three klee graphs that correspond to the near klee graphs obtained from the decomposition of K according to Lemma 8. Figures 6.2(a)–(b) illustrate a klee partition. We now have the following lemma.

Lemma 9 *Let G be a klee graph with $n \geq 6$ vertices and exactly three outer vertices. Let S be a set of n points. If G admits a convex point-set embedding on S, then the*

following conditions hold.

(a) The convex hull of S has exactly three points s_1, s_2 and s_3 on its boundary.

(b) The convex hull $\mathcal{C}$ of $S \setminus \{s_1, s_2, s_3\}$ has exactly three points on its boundary.

(c) Let K be the near klee graph obtained from G. Then for a fixed mapping of the legs of K to $\{s_1, s_2, s_3\}$, the mapping of the poles of K to the three vertices on $\mathcal{C}$ is uniquely defined.

Proof. Let a, b, c be the outer vertices of G, which are also the legs of K by definition. In any point-set embedding of G on S, the outer face of G is drawn as a triangle. Therefore, if G admits a point-set embedding on S, then the convex hull of S must contain exactly three points on its boundary.

Let p, q, r be the poles of K. Assume that in a convex point-set embedding of G on S the vertices a, b, c, p, q, r are mapped to points $s_1, s_2, s_3, s_4, s_5, s_6$, respectively. Then the three quadrangles $s_1s_2s_5s_4, s_2s_3s_6s_5, s_3s_1s_4s_6$ must be convex. This convexity constraint ensures that the convex hull $\mathcal{C}$ of $S \setminus \{s_1, s_2, s_3\}$ contains exactly three points on its boundary, and for a fixed mapping of the legs of K to $\{s_1, s_2, s_3\}$, the mapping of the poles of K to the three vertices on $\mathcal{C}$ is uniquely defined. ■

Let G be a klee graph and let a, b, c be the outer vertices of G. Let v_p be the mapping of vertex v on point p. Then by $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ we denote be the problem of finding a convex point-set embedding of G respecting $a_{s_i}, b_{s_j}, c_{s_k}$. The following theorem gives a recursive solution for $P(G, a_{s_i}, b_{s_j}, c_{s_k})$.

Theorem 5 *Let G be a klee graph with n vertices and exactly three outer vertices, and let S be a set of n points whose convex hull $\mathcal{C}$ has exactly three points s_i, s_j, s_k.*

Let $a_{s_i}, b_{s_j}, c_{s_k}$ be the mapping of the outer vertices a, b, c of G. Let the klee partition of G be G_1, G_2, G_3 and let the corresponding near klee graphs be H_1, H_2, H_3, where the legs of H_1, H_2, H_3 are respectively $\{a, a', a''\}$, $\{b, b', b''\}$ and $\{c, c', c''\}$ in anticlockwise order. Let $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ be nine points in $S'(s_i s_j s_k)$ that satisfy the following conditions.

C_1: The triangle $x_1x_2x_3$ determines the convex hull of $S \setminus \{s_1, s_2, s_3\}$.

C_2: $|S(x_1y_1z_1)| = n(H_1)$, $|S(x_2y_2z_2)| = n(H_2)$ and $|S(x_3y_3z_3)| = n(H_3)$. If $n(H_u) = 1$, for some $u \in \{1, 2, 3\}$, then x_u, y_u and z_u coincide. Otherwise, x_u, y_u and z_u are distinct.

C_3: $\langle s_i, x_1, y_1, z_2, x_2, s_j\rangle$, $\langle s_j, x_2, y_2, z_3, x_3, s_k\rangle$ and $\langle s_k, x_3, y_3, z_1, x_1, s_i\rangle$ are convex paths inside $\mathcal{C}$.

Let $\mathcal{X}$ be a quantifier that denotes "All possible choices for $x_1, y_1, z_1,\ x_2, y_2, z_2,\ x_3, y_3, z_3$ that satisfy (C_1)–(C_3)". Then $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ can be defined recursively as follows.

$$P(G, a_{s_i}, b_{s_j}, c_{s_k}) = \begin{cases} True & \text{if } n(G) = 4 \text{ and } |S'(s_i s_j s_k)| = 1; \\ False & \text{if } \mathcal{X} \text{ is empty;} \\ \bigvee\limits_{(x_1,\ldots,z_3)\in\mathcal{X}} (P(G_1, a_{s_i}, a'_{z_2}, a''_{y_3}) \wedge & \\ \qquad P(G_2, b_{s_j}, b'_{z_3}, b''_{y_1}) \wedge & \\ \qquad P(G_3, c_{s_k}, c'_{z_1}, c''_{y_2})) & \text{otherwise.} \end{cases}$$

Proof. We first refer the reader to Figure 6.3 for an illustration of the recurrence relation. We now proceed to prove the claim for $P(G, a_{s_i}, b_{s_j}, c_{s_k})$.

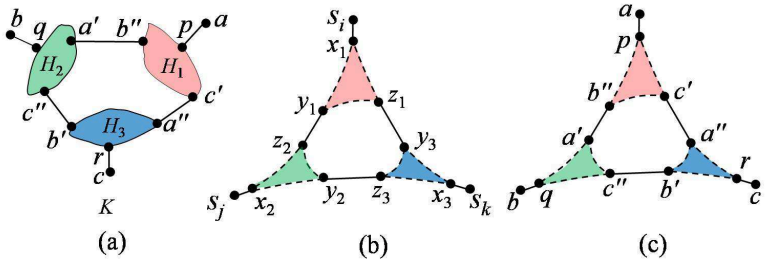

Figure 6.3: (a) The near klee graph of G. (b)–(c) Recursive embedding.

Consider first the case when $n(G) = 4$ and $|S'(s_i s_j s_k)| = 1$. In this case the single inner vertex in G is mapped to the single point interior to triangle $s_i s_j s_k$. This yields a convex point-set embedding of G on S respecting $a_{s_i}, b_{s_j}, c_{s_k}$.

Consider now the case when $\mathcal{X}$ is empty, i.e., there is no choice for x_1, y_1, z_1, $x_2, y_2, z_2, x_3, y_3, z_3$ that satisfies conditions C_1–C_3. In this case $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ is False. Otherwise, assume for a contradiction that $\mathcal{X}$ is empty and G admits a convex point-set embedding Γ on S respecting $a_{s_i}, b_{s_j}, c_{s_k}$.

Let K be the near klee graph obtained from G and let p, q, r be the poles of K, where p, q, r are adjacent to a, b, c, respectively. Then the poles of H_1, H_2 and H_3 are $\{p, b'', c'\}, \{q, c'', a'\}$ and $\{r, a'', b'\}$ in anticlockwise order. Let the mappings of these poles in Γ be $\{p_{l_1}, b''_{m_1}, c'_{n_1}\}$, $\{q_{l_2}, c''_{m_2}, a'_{n_2}\}$ and $\{r_{l_3}, a''_{m_3}, b'_{n_3}\}$. In the following (a)–(c) we prove that the conditions C_1–C_3 hold for $l_1, m_1, n_1, l_2, m_2, n_2, l_3, m_3, n_3$, which will contradict that $\mathcal{X}$ is empty.

(a) Since Γ is a convex point-set embedding of G on S, by Lemma 9, l_1, l_2, l_3 determine the convex hull of $S \setminus \{s_i, s_j, s_k\}$.

(b) $|S(l_1 m_1 n_1)| = n(H_1)$, $|S(l_2 m_2 n_2)| = n(H_2)$ and $|S(l_3 m_3 n_3)| = n(H_3)$. Other-

wise, either $\langle m_1, \ldots, n_1, m_3, \ldots, n_3, m_2, \ldots, n_2 \rangle$ is not a convex path, or at least one outer chain among $\langle s_i, l_1, \ldots, l_2, s_j \rangle$, $\langle s_j, l_2, \ldots, l_3, s_k \rangle$, $\langle s_k, l_3, , \ldots, \ l_1, s_i \rangle$ is not a convex path, which contradicts our assumption that Γ is a convex point-set embedding.

(c) Since each set of vertices among $\{s_i, l_1, m_1, n_2, l_2, s_j\}$, $\{s_j, l_2, m_2, n_3, l_3, s_k\}$ and $\{s_k, l_3, m_3, n_1, l_1, s_i\}$ appears on the boundary of the same face in Γ, $\langle s_i, l_1, m_1, n_2, l_2, s_j \rangle$, $\langle s_j, l_2, m_2, n_3, l_3, s_k \rangle$ and $\langle s_k, l_3, m_3, n_1, l_1, s_i \rangle$ are convex paths inside $\mathcal{C}$.

In the remaining case, $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ is True if and only if there exists x_1, y_1, z_1, $x_2, y_2, z_2, x_3, y_3, \ z_3$ for which G_1, G_2 and G_3 admit convex point-set embeddings with mappings $\{a_{s_i}, a'_{z_2}, a''_{y_3}\}$, $\{b_{s_j}, b'_{z_3}, b''_{y_1}\}$ and $\{c_{s_k}, c'_{z_1}, c''_{y_2}\}$, respectively.

If G admits a convex point-set embedding on S respecting the mapping $a_{s_i}, b_{s_j}, c_{s_k}$, then straightforward reasoning using the definition of a near convex point-set embedding shows that G_1, G_2 and G_3 admit convex point-set embeddings. We thus assume that G_1, G_2 and G_3 admit convex point-set embeddings, and then prove that these embeddings determine a convex point-set embedding Γ of G on S. The following (a)–(c) prove that all the faces of Γ are convex.

(a) The embeddings of G_1, G_2 and G_3 respect the mappings a_{s_i}, b_{s_j} and c_{s_k}. Therefore, the outer face of Γ must be a convex polygon.

(b) The interior faces of G consist of the interior faces of G_1, G_2 and G_3 and the face $(y_1, \ldots, z_1, y_3, \ldots, \ z_3, y_2, \ldots, z_2, y_1)$. By our initial assumption, the interior faces of G_1, G_2 and G_3 are drawn as convex polygons. Furthermore, since $\langle y_3, z_1, \ldots, y_1, z_2 \rangle$, $\langle y_1, z_2, \ldots, y_2, z_3 \rangle$ and $\langle y_2, z_3, \ldots, y_3, z_1 \rangle$ are convex paths and

each pair of these paths share a common edge, face $(y_1, \ldots, z_1, y_3, \ldots, z_3, y_2, \ldots, z_2, y_1)$ is convex. Hence, the internal faces of G are drawn as convex polygons in Γ.

(c) Since the path $\langle s_i, x_1, \ldots, y_1, z_2\rangle$ of G_1 and the path $\langle y_1, z_2, \ldots, x_2, s_j\rangle$ of G_2 are convex paths that share a common edge, $\langle s_i, x_1, \ldots, x_2, s_j\rangle$ must be a convex path in Γ. Similarly, we can prove that $\langle s_j, x_2, \ldots, x_3, s_k\rangle$ and $\langle s_k, x_3, \ldots, x_1, s_i\rangle$ are also convex paths in Γ. Consequently, the corresponding faces of G must be convex polygons in Γ.

Hence $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ is True if and only if G_1, G_2 and G_3 admit convex point-set embeddings with mappings $\{a_{s_i}, a'_{z_2}, a''_{y_3}\}$, $\{b_{s_j}, b'_{z_3}, b''_{y_1}\}$ and $\{c_{s_k}, c'_{z_1}, c''_{y_2}\}$. Otherwise, $P(G, a_{s_i}, b_{s_j}, c_{s_k})$ is False. ■

Theorem 5 leads us to a dynamic programming algorithm for computing a near convex point-set embedding of G on S respecting the given mapping of the outer vertices of G. To avoid the recomputation of the solutions to the subproblems, we use a table $T[1{:}n, 1{:}n, 1{:}n, 1{:}n]$, where each entry $T[f(a, b, c), i, j, k]$ stores the solution to $P(G, a_{s_i}, b_{s_j}, c_{s_k})$. We use another table $T'[1{:}n, 1{:}n, 1{:}n, 1{:}n]$, where each entry $T'[f(a, b, c), i, j, k]$ stores the necessary mapping depending on the value of $P(G, a_{s_i}, b_{s_j}, c_{s_k})$. Table T' helps us to find for an actual embedding of G on S if such an embedding exists.

6.2 Time Complexity

In this section we describe an implementation of the dynamic programming algorithm of Section 6.1 that takes $O(n^{5+\epsilon})$ expected time.

Initially we build a triangular range search data structure with the points of S. We use the data structure of Chazelle et al. [23] that preprocesses the points in $O(m^{1+\epsilon})$ time and answers triangular range reporting and range counting queries in $O(n^{1+\epsilon}/m^{1/2}+k)$ and $O(n^{1+\epsilon}/m^{1/2})$ time, respectively, where $n<m<n^2, \epsilon \in (0, 2/3]$ is fixed and k is the number of points reported. We choose $m=n^{2-\epsilon}$ so that the time complexities for preprocessing, range reporting and range counting become $O(n^{2+\epsilon})$, $O(n^{\epsilon'}+k)$ and $O(n^{\epsilon'})$, respectively, where $\epsilon' = 3\epsilon/2 > 0$.

Observe that there are $O(n^4)$ different entries in table T, which correspond to $O(n^4)$ different subproblems. We fill each entry according to the three cases as described in Theorem 5. In the worst case we need to check all possible choices for $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ that satisfy conditions C_1–C_3. By Lemma 9, the mapping of a, b, c fixes the mapping for x_1, x_2 and x_3. Therefore, we need to check all possible choices for $y_1, y_2, y_3, z_1, z_2, z_3$ in $S'(s_i s_j s_k)$. Although there are $\Theta(n^6)$ choices for six points, we show that there is an efficient way to choose $y_1, z_1, y_2, z_2, y_3, z_3$ in $O(n^{1+\epsilon} \log n)$ time. We use the observation of Biedl [8] that any choice for x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3 satisfying conditions C_1–C_3 is determined by the point y_1. In other words, if a point y_1 corresponds to some choice $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$, then points x_2, y_2, z_3 are the unique points in triangle $y_1 s_j s_k$ such that $|S(y_1 s_j x_2)| = 3, |S(y_1 s_j y_2)| = n(G_1) - 1$ and $|S(y_1 s_j z_3)| = n(G_1)$ holds. In a similar way, y_2 determines the unique choice for x_3, y_3, z_1.

We sequentially consider each point of $S'(s_i s_j s_k)$ as the point y_1 and for each choice we find x_2, y_2, z_3 in $O(n^\epsilon \log n)$ expected time using an algorithm of Durocher et al. [32, Lemma 1]. We then find the points x_3, y_3, z_1 and x_1, y_1, z_2 in a similar way.

Recall that we need to fill $O(n^4)$ entries of table T and $O(n^4)$ entries of table T'. Since we do not recompute the subproblems, and each entry is computed in $O(n^{1+\epsilon} \log n)$ expected time, the time necessary to fill all the entries is $O(n^4 \cdot n^{1+\epsilon} \log n) = O(n^{5+\epsilon} \log n)$, which dominates all the preprocessing costs and the cost for finding the solution embedding from the table.

Observe that for any $\epsilon > 0$, $n^{5+\epsilon} \log n = O(n^{5+\epsilon'})$ for any $\epsilon' > \epsilon$. We thus have the following theorem.

Theorem 6 *Let G be a n-vertex klee graph with exactly three outer vertices and let S be a set of n points in general position. Then in $O(n^{5+\epsilon})$ expected time, for any fixed $\epsilon > 0$, we can test whether G admits a convex point-set embedding on S and find an embedding if one exists.*

6.3 Generalization

In this section extend the algorithm of Section 6.1 to test convex point-set embeddability of arbitrary klee graphs in $O(n^{8+\epsilon})$ expected time. We need the following lemma.

Lemma 10 *Let G be a klee graph with $n \geq 4$ vertices. Then G has an inner edge and two outer edges such that deletion of those edges (not the incident vertices) yields two connected components, C_1 and C_2, each of which is a near klee graph.*

Proof. We use an induction argument on n. The claim is straightforward to verify when $n = 4$. We now assume that the claim holds for every klee graph G with fewer than n vertices and consider the case when G has n vertices.

By definition of a klee graph, G has a face f of length three, where contraction of the three edges on the boundary of f gives another klee graph G' with $n-2$ vertices. By induction G' has an inner edge (a, x) and two outer edges $(b, y), (c, z)$ such that deletion of those edges yields two connected components C'_1, C'_2, each of which is a near klee graph. Without loss of generality assume that the vertices a, b, c and the vertices x, y, z belong to C'_1 and C'_2, respectively. We will also assume in our induction that if C'_1 (respectively, C'_2) is a single vertex then a, b, c (respectively, x, y, z) coincide, otherwise they are distinct vertices of G'. This assumption holds in the base case and we assume that it holds for every klee graph with fewer than n vertices.

If f replaces some vertex of G' other than a, b, c, x, y, z, then that vertex is interior to either C'_1 or C'_2. By definition the resulting component after the replacement will be a near klee graph. Consequently, $(a, x), (b, y), (c, z)$ will be the required edges of G such that deletion of those edges yields two connected components C_1, C_2, each of which is a near klee graph. We can also use induction to verify that if C_1 (respectively, C_2) is a single vertex then a, b, c (respectively, x, y, z) coincide, otherwise they are distinct vertices of G.

Otherwise, without loss of generality assume that f replaces the vertex x. If the vertex x coincides with vertices y and z, i.e., C'_2 is a single vertex, then replacing the vertex with f will give a cycle of length three which is a near klee graph. It is straightforward to label the three vertices on f with x, y, z such that $(a, x), (b, y), (c, z)$

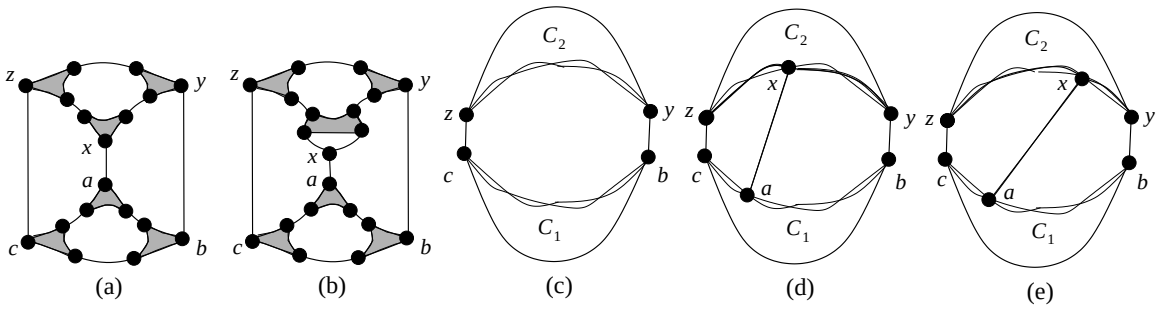

Figure 6.4: Illustration for Lemma 10. (a) G' and (b) G. (c)–(e) Candidate mappings for (a, x).

become the required edges for G so that our claim holds. If C_2' is a single vertex, then C_1' consists of three or more vertices. Consequently, both C_1 and C_2 consist of three or more vertices and a, b, c, x, y, z are distinct vertices of G. We can prove our claim for the remaining case when f replaces the vertex x and the vertices x, y, z are distinct in a similar way. Figures 6.4(a)–(b) illustrate such an example. ■

We now use Theorem 6 and Lemma 10 to obtain the following theorem.

Theorem 7 *Let G be a klee graph with n vertices and let S be a set of n points in general position. Then in $O(n^{8+\epsilon})$ expected time, $\epsilon > 0$, we can test whether G admits a convex point-set embedding on S and find an embedding if one exists.*

Proof. We first find an inner edge of (a, x) and two outer edges $(b, y), (c, z)$ of G such that deletion of those edges yields two connected components C_1, C_2, each of which is a near klee graph. Without loss of generality assume that a, b, c and x, y, x belong to C_1 and C_2, respectively. Lemma 10 proves the existence of such edges and we can find them in polynomial time by deleting every triple of edges by turn and then testing whether the resulting components are near klee graphs. Since one can

recognize plane 3-trees in polynomial time [59], we also can recognize near klee graphs in polynomial time using Lemma 7. Another technique is to recursively contract a face of length three unless the resulting graph is K_4 (i.e., the input graph is a klee graph), or there is no such face to contract (i.e., the input graph is a klee graph).

We now compute the convex hull of S. If the number of points on the boundary of the convex hull is less than the number of outer vertices of G, then G does not have a convex point-set embedding on S. Observe that there are $O(n)$ possible mappings of the outer vertices to distinct points on the convex hull. For each of those mappings we find all possible mappings of the edge (a, x) inside the convex hull. Let the vertices a, b, c, x, y, z be mapped to the points $s_1, s_2, s_3, s_4, s_5, s_6$, respectively. We then test whether C_1 and C_2 admit a convex point-set embedding inside the point sets enclosed by the infinite regions bounded by angles $\angle s_2 s_1 s_3$ and $\angle s_5 s_4 s_6$, respectively, and whether the combined embedding is convex. One can perform such a test by a simple modification of our algorithm for klee graphs with exactly three outer vertices with the same time bound, i.e., in $O(n^{5+\epsilon})$ time. Such a modification is necessary since there may be a pair of poles in each of C_1 and C_2 such that the outer chain between these poles must be drawn as convex paths outside the triangle $s_1 s_2 s_3$ and $s_4 s_5 s_6$.

There are only $O(n)$ candidate mappings for the outer vertices of G. For a fixed mapping of the outer vertices, the mapping of $(b, y), (c, z)$ is fixed. There are now $O(n^2)$ candidate mappings for a, x. For each of these candidate $O(n^3)$ mappings, we can test the embeddability of C_1 and C_2 in $O(n^{5+\epsilon})$ expected time. Therefore, the algorithm takes $O(n^{8+\epsilon})$ expected time in total. which dominates the time for finding

the candidate mappings of a, b, c, x, y, z. ■

The result stated in Theorem 7 establishes the existence of a polynomial-time solution. We believe a more efficient solution (i.e., one whose running time is bounded by a polynomial of low degree) exists. For example, there may be only $O(n)$ candidate mappings for (a, x), but proving this will require further analysis because of the scenarios shown in Figures 6.4(c)–(e).

Chapter 7

Fixed Parameter Tractability

In this chapter we give a fixed-parameter tractable (FPT) algorithm for deciding point-set embeddability of plane graphs. Given a plane graph G with n vertices and a set S of n points, our algorithm takes $O(k^3 16.71^k n^8)$ time for deciding convex point-set embeddability, where k is the number of points interior to the convex hull of S.

Let w and w' be the carving width of the input graph and its dual, respectively. Biedl and Vatshelle [9] gave an algorithm to decide point-set embeddability in $O^*(|S|^{1.5w'})$ time, and Biedl [8] modified that algorithm to decide convex point-set embeddability in $O^*(|S|^{3w})$ time, where $O^*(.)$ hides the polynomial terms. Since we assume $|S| = |V| = n$, these running times become $O^*(n^{1.5w'})$ and $O^*(n^{3w})$, respectively.

Let S be a set of n points such that the convex hull of S contains exactly k interior points. Then S can support planar graphs of treewidth $\sqrt{k}$, e.g., a $\sqrt{k} \times \sqrt{k}$ grid graph. Since the carving width of a graph with treewidth t is at least $0.5t$ [81], the

running time of Biedl's algorithm for such graphs is $O^*(n^{1.5\sqrt{k}})$. In the worst case, S can support planar graphs of maximum degree cn, where $0 < c < 1$. Since the carving width of a graph is at least its maximum degree [9], the running time becomes $O^*(n^{c'n})$, for some fixed $c' > 0$. Therefore, our FPT-algorithm runs faster than Biedl's algorithm [8] as long as k is less than $n^{c'}$. However, our results as stated here are mostly of theoretical interest since the base in the exponentiation 16.71^k is a large constant.

Given a set S of n points, Spillner [73] gave an $O(k^3 2^k n^3)$-time dynamic programming algorithm to construct a plane graph with the minimum number of faces that admits a convex point-set embedding on S. We adapt the dynamic programming technique of Spillner. We split a problem into independent subproblems by monotone paths. While splitting the point set with monotone paths, we split the corresponding graph G such that G admits a convex point-set embedding if and only if the smaller subgraphs of G admit convex point-set embeddings on their corresponding point sets.

In Section 7.1 we briefly review Spillner's algorithm. We then modify that algorithm to obtain an $O(k^3 16.71^k n^8)$-time algorithm for testing convex point-set embeddability.

7.1 Overview of Spillner's Algorithm

Let S be a set of n points and let H be the convex hull of S. Assume that k is the number of points of S that are interior to H. Spillner [73] gave an algorithm for computing a plane graph with the minimum number of faces that admits a convex point-set embedding on S. Such a plane graph is known as *optimal convex partition of*

S. Before describing Spillner's algorithm we need to introduce a few more definitions.

By a *convex partition* of S we denote a convex point-set embedding Γ of some planar graph on S. A path in Γ is *monotone* in direction $\vec{d}$ if the orthogonal projections of its vertices on any line with direction $\vec{d}$ are monotone. Let f be a convex face in Γ and let u, v be two vertices on the boundary of f. The line segment uv is called a *spanning edge* of f if the polygonal chain on f from u to v in clockwise order is monotone in direction $\vec{uv}$. The following observation is crucial for Spillner's algorithm.

Observation 2 (Spillner [73]) *Let Γ be a convex partition of S and let $a_1, a_2, \ldots, a_l$ be a path in Γ that is monotone in some direction $\vec{d}$. Then there exists a path $u, \ldots, a_1, a_2, \ldots, a_l, \ldots, v$, for some outer vertices u, v in Γ, that is monotone in the direction $\vec{d}$.*

We now give an outline of Spillner's algorithm. Let $H_{u,v}$ be the polygonal chain while walking along the boundary of H clockwise from the point u to the point v. Spillner showed that the subproblems of the optimal convex partition problem can be expressed by the tuple $(t, (u, v), A, M)$, where t denotes the type of the subproblem, A is a monotone path with respect to the x-axis, and M is a polygonal chain of at most three spanning edges. For example, Figure 7.1(a) shows a subproblem, where

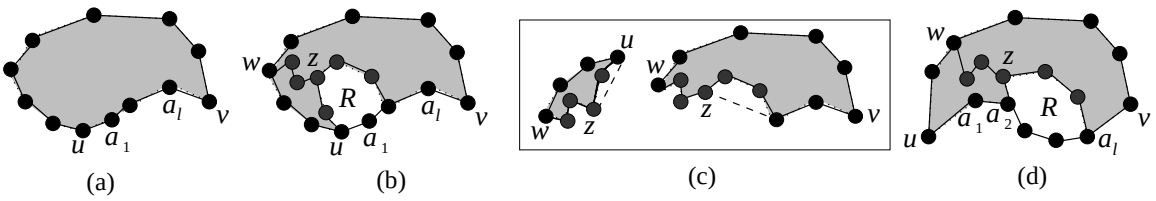

Figure 7.1: (a)–(d) Illustration for Spillner's algorithm.

$A = \{a_0(= u), a_1, \ldots, a_l, a_{l+1}(= v)\}$ and $M = \varnothing$. Let R be a convex face that is incident to some edge of A. Let z be the leftmost point on R that lie interior to the region bounded by $H_{u,v}$ and A, as shown in Figure 7.1(b). By Observation 2, one can find a monotone path from z to some point w on $H_{u,v}$, which splits the problem into two smaller subproblems, as shown in Figure 7.1(c). Observe that both the subproblems of Figure 7.1(c) can be expressed in the form $(t, (u, v), A, M)$, where the spanning edges that belong to M are shown with dashed lines. However, similar break-down may not be possible for the problem of Figure 7.1(d), since the path from u to w (passing through z) that split the point set is not monotone. Spillner proved that every convex partition contains a choice for R such that a scenario like Figure 7.1(d) does not appear. Therefore, he could design a dynamic programming that only examines the subproblems obtained from the feasible choices for R.

We now give some more detail of Spillner's algorithm, which will be helpful to explain our algorithm and its time complexity in Section 7.2. Let $\mathcal{R}$ be a region that is bounded by $H_{u,v}$ and a monotone path $A = \{a_0(= u), a_1, \ldots, a_l, a_{l+1}(= v)\}$. Spillner showed that any convex partition of $\mathcal{R}$ must contain a region R such that one can split $\mathcal{R}$ in one of the six ways, as shown in Figure 7.2. Figures 7.2(a)–(d) (respectively, Figures 7.2(e)–(h)) illustrate the cases when the leftmost (respectively, rightmost) vertex on R is to the left of u (respectively, to the right of v), and R is incident to exactly one of u and v. Figures 7.2(i)–(j) illustrate the case when none of u and v are incident to R. Finally, Figures 7.2(k)–(n) illustrate the case when both u and v belong to R.

Formally, for a particular problem $(t, (u, v), A, M)$, Spillner's algorithm computes

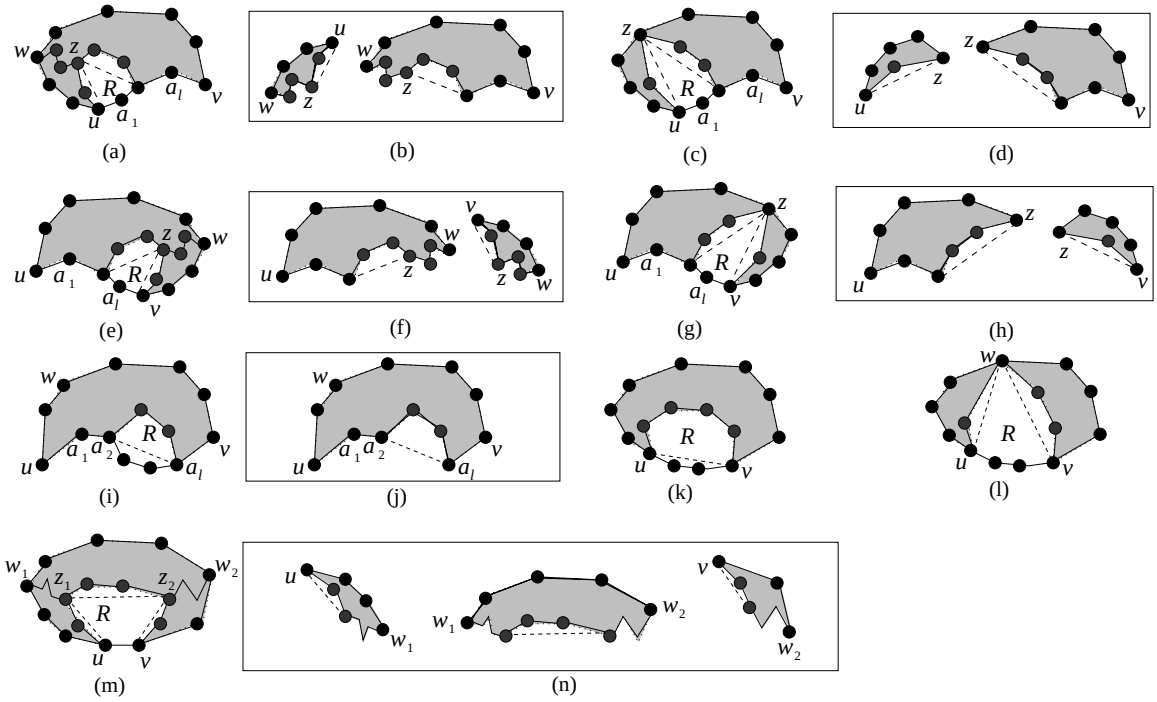

Figure 7.2: Feasible choices for R and the corresponding subproblems.

the subproblems in the following way. If $M = \varnothing$, then the algorithm computes the optimal solution from the subproblems that arise from all possible feasible choices of the leftmost or rightmost vertex of R, as shown in Figure 7.3(a). If $|M| = 1$, then the algorithm examines the subproblems that arise from all possible choices for the polygonal chain that can be spanned by the edge in M, as shown in Figure 7.3(b). If $|M| = 2$, then the algorithm examines the subproblems that arise from all possible choices of the monotone path from the leftmost (or, rightmost) vertex z to the left (or, to the right) along the x-axis. Finally, if $|M| = 3$, then the algorithm examines the subproblems that arise from all possible choices of the monotone path from the leftmost vertex z_1 to the left along the x-axis. Figures 7.3(c) and (d) illustrate the case when $|M| = 2$ and $|M| = 3$, respectively. Observe that the computation of the

subproblems ensures that $|M|$ is always less than three.

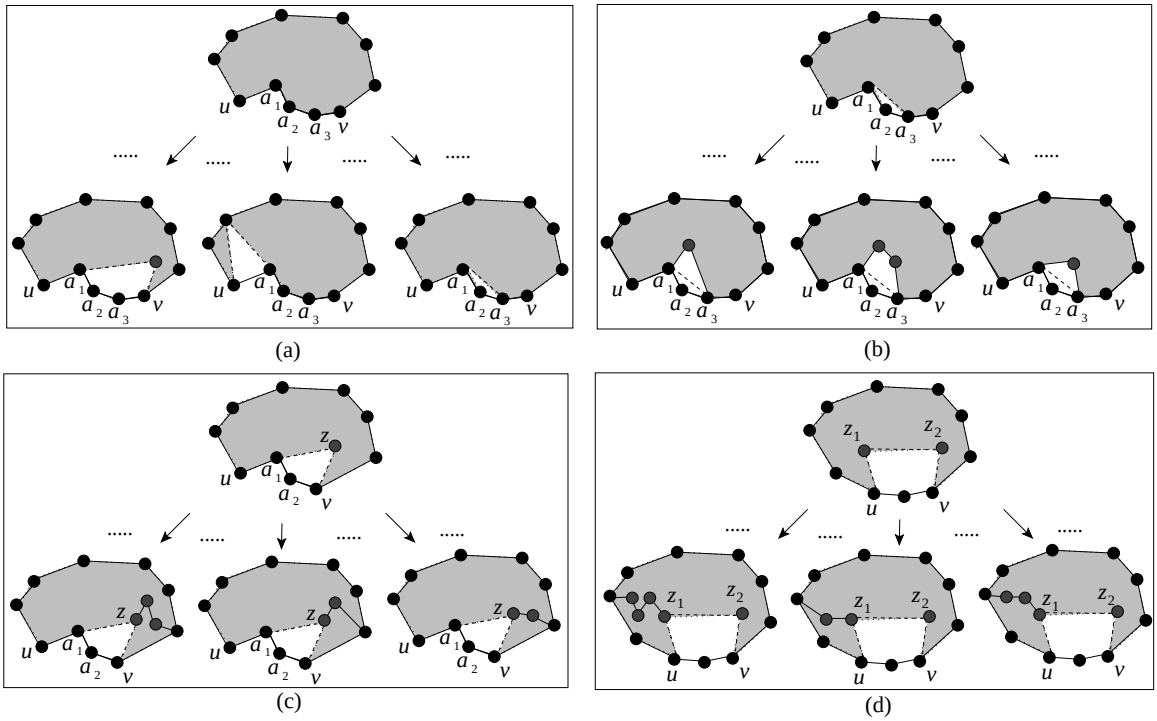

Figure 7.3: Computation of the subproblems.

Spillner proved that the number of different types of the subproblems is a constant. There are only $O(n^2)$ choices for (u, v) and $O(k^2)$ choices for M. Since there are $O(2^k)$ choices for A, the number of subproblems is $O(k^2 2^k n^2)$. The algorithm processes each of these subproblems in $O(n + k^2)k$ time. Therefore, the running time becomes $O(k^4 2^k n^3)$ in total.

7.2 FPT Convex Point-Set Embedding

In this section we adapt Spillner's algorithm [73] to decide convex point-set embeddability for plane graphs.

Observe that if the input plane graph G admits a convex point-set embedding on the input point set S, then G must admit a convex drawing on the Euclidean plane. Therefore, we first test whether G admits a convex drawing in linear time using Chiba et al.'s algorithm [24]. If G admits a convex drawing, then we test its convex point-set embeddability using a modification of Spillner's algorithm [73]. We first define a few notations.

Let x, y be two outer vertices of G. By $P(x, y)$ we denote a path of G with end vertices x, y. If such a path lies on the outer face of G, then we denote it by $P\langle x, y\rangle$. Let $G_{P(x,y)}$ be the subgraph of G induced by the vertices inside the closed cycle determined by $P(x, y)$ and $P\langle x, y\rangle$. Let H be the convex hull of S, and let A be a monotone path interior to H with end vertices u, v, where u, v lie on the boundary of H. Then $S_{u,v}$ denotes the closed subset of S bounded by $H_{u,v}$ and A.

We are now ready to describe our algorithm. Let the tuple $(t, (u, v), A, M, P(x, y), f)$ be the problem of computing a point-set embedding of $G_{P(x,y)}$ on the points of $S_{u,v}$ that satisfies the following conditions.

(a) The inner faces in the output drawing are convex.

(b) The vertices x, y are mapped to the points u, v, respectively.

(c) If $M = \varnothing$, then $f = \varnothing$ and the outer boundary of $G_{P(x,y)}$ is mapped on the outer boundary of the abstract graph induced by $H_{u,v}$ and A, as shown in Figures 7.4(a)–(c).

(c) Otherwise, f is a face of $G_{P(x,y)}$ that corresponds to the edges in M, and a subset of the outer boundary of $G_{P(x,y)}$ is mapped on a subset of the outer boundary of the abstract graph induced by A, M and $H_{u,v}$, as shown in Figures 7.4(d)–(i).

Here, the terms $t, (u, v), A, M$ have the same meaning as in Section 7.1.

Fix an embedding of the outer boundary of G on the convex hull H of S, and choose an arbitray outer edge (x, y) of G. Assume that x, y are mapped to the points u, v, respectively. Then the problem of computing a convex point-set embedding of G on S can be expressed by the tuple $(1, (u, v), \{u, v\}, \varnothing, \{x, y\}, \varnothing)$, where $t = 1$ is the type of the problem according to Spillner's algorithm[73].

We now use Spillner's algorithm to compute the subproblems, but while partitioning the point set, we also compute a partition of the graph that corresponds to those smaller point sets. If $M = \varnothing$, then we compute the solution examining the

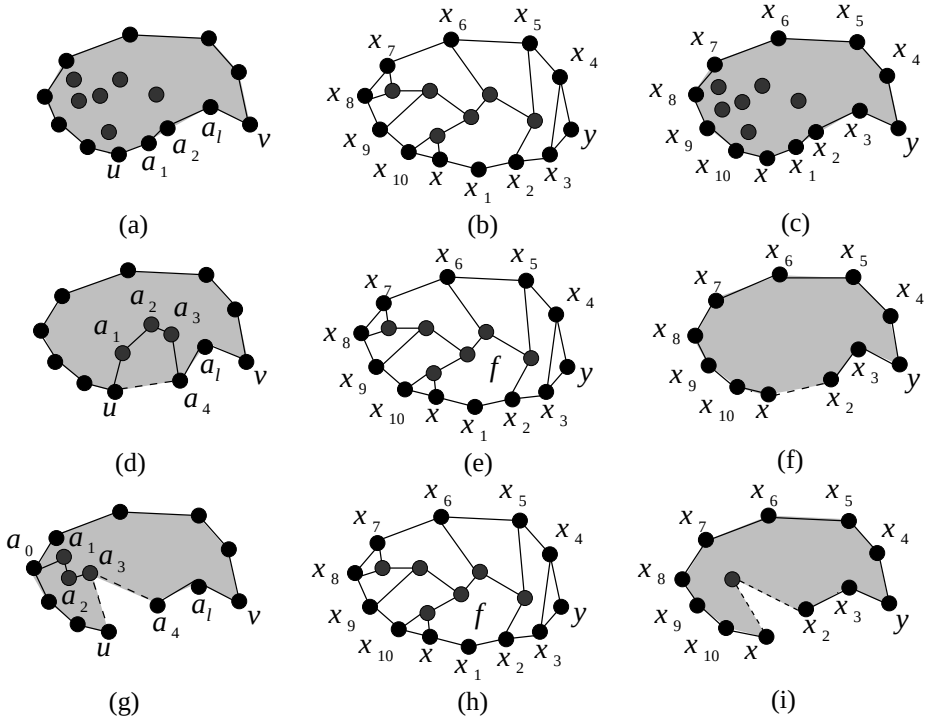

Figure 7.4: Illustration for the mapping when (a)–(c) $M = \varnothing$, and (d)–(i) $M \neq \varnothing$. Figures (a), (d), (g) illustrate $S_{u,v}$, (b), (e), (h) illustrate $G_{P(x,y)}$, and (c), (f), (i) illustrate the partial mapping.

subproblems that arise from all possible feasible choices of the leftmost or rightmost vertex of a convex region R, as shown in Figure 7.5(a). This region R corresponds to the face f in $G_{P(x,y)}$. If $|M| = 1$, then the algorithm examines the subproblems that arise from all possible choices for the polygonal chain that can be spanned by the edge (a_i, a_j) in M, $i < j$, as shown in Figure 7.5(b). Let t, t' be the vertices of f that are mapped to a_i, a_j, respectively. Then the length of this polygonal chain must be equal to the length of the path on f that starts from t and ends at t' in clockwise order. If $|M| = 2$, then the algorithm examines the subproblems that arise from all possible choices of the monotone path from the leftmost (or, rightmost) vertex of R to the left (or, to the right) along the x-axis. Finally, if $|M| = 3$, then the algorithm

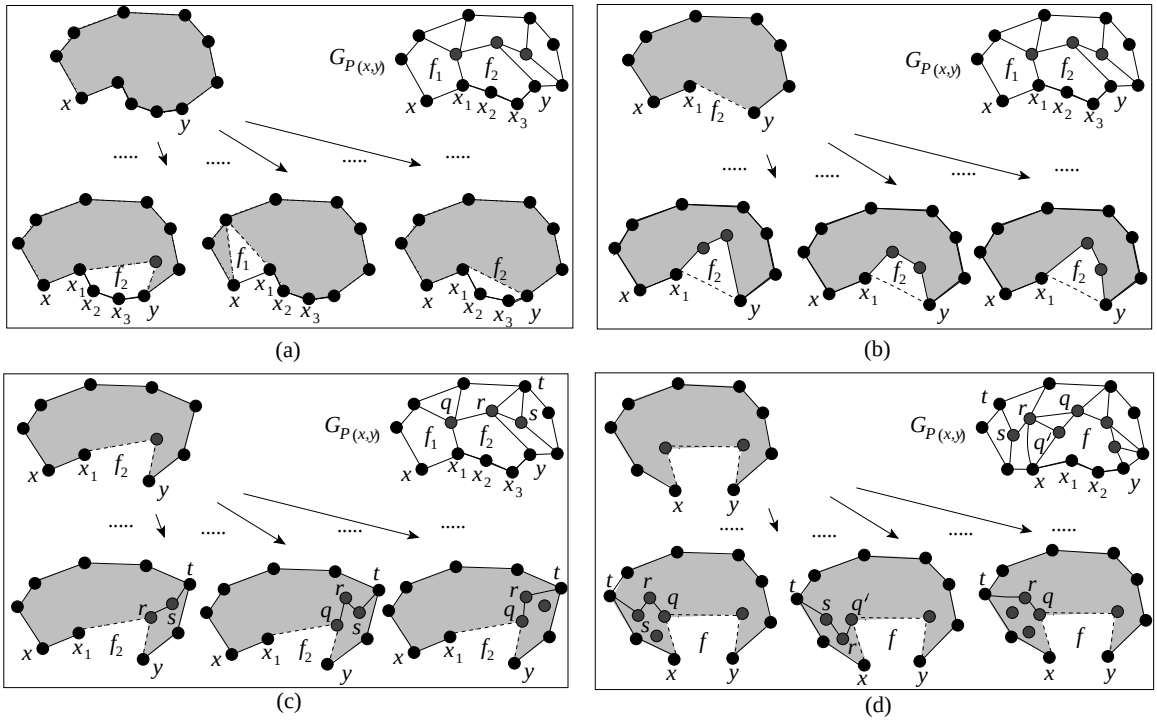

Figure 7.5: Computation of the subproblems using modified Spillner's algorithm.

examines the subproblems that arise from all possible choices of the monotone path from the leftmost vertex of R to the left along the x-axis. Figures 7.5(c) and (d) illustrate the cases when $|M| = 2$ and $|M| = 3$, respectively. Every monotone path in these cases defines some partition of the point set, which may correspond to many simple paths in $G_{P(x,y)}$.

We now compute the time complexity of the dynamic program described above examining the tuple $(t, (u, v), A, M, P(x, y), f)$. There are a constant number of problem types in Spillner's algorithm. Therefore, $t = O(1)$. There are $O(n^2)$ possible ways to choose (u, v). Since A is a monotone path, there are $O(2^k)$ possibilities for A, where k is the number of points interior to H. Given $t, (u, v), A$, there are only $O(k^2)$ choices for M and f. Finally, given $t, (u, v), A, M, f$, we can prove an $O(2.89^k n^2)$ bound on the choices for $P(x, y)$ using the 2.89^t upper bound of Buchin et al. [19, Lemma 1] for simple cycles in planar graphs with t vertices. Consequently, we obtain an $O(k^2 5.78^k n^2)$ upper bound on the number of subproblems. In the following we compute an upper bound on the time to process a single subproblem.

Spillner's algorithm [73] looks up at most $O(n + k^2)$ entries in the dynamic programming table during the processing of a subproblem. However, the modified algorithm sometimes need to look up $O(2.89^k n^2)$ entries since a fixed monotone path that partitions $S_{u,v}$ may correspond to $O(2.89^k n^2)$ simple paths in $G_{P(x,y)}$. Read and Tarjan [66] gave an $O(tC)$-time algorithm for listing all cycles of a planar graph with t vertices, where C is the number of cycles to be enumerated. This algorithm can be modified to enumerate all simple paths with fixed endvertices with the same time bound [7]. In the case of our algorithm, $C = O(2.89^k n^2)$. Hence we can enumerate

all the choices for a feasible path that partitions $G_{P(x,y)}$ in $O(k2.89^k n^2)$ time. Therefore, we can determine all the entries that we need to look up in $O(k2.89^k n^2)$ time. Consequently, the time required to process a subproblem is $O(k2.89^k n^2) \times \mathcal{T}$, where $\mathcal{T}$ denotes the time required to look up an entry in the dynamic programming table.

It is straightforward to store t, (u, v), M and f in some arrays so that one can access them in $O(1)$ time. Spillner used a dictionary data structure to access the sets that correspond to A in $O(\log(2^k)) = O(k)$ time. We use another dictionary data structure to access the sets that correspond to $P(x, y)$ in $O(\log(2.89^k n^2)) = O(k + \log n)$ time. We have some other costs associated with some tests, e.g., whether a face is convex, or whether the number of vertices inside a particular cycle in G is equal to the number of points of S that lie in a particular region. These tests can be performed in $O(n^2)$ time. Consequently, processing of a subproblem takes $O(k2.89^k n^2) \times O(n^2 \log n + n^2 k) = O(k2.89^k n^5)$ time in total.

Since there are $O(k^2 5.78^k n^2)$ subproblems, we obtain an $O(k^3 16.71^k n^7)$ upper bound on the running time of our algorithm that respects a fixed embedding of the outer boundary of G on the convex hull of S. Since the outer boundary of G can be mapped on the convex hull of S in $O(n)$ ways, the running time is $O(k^3 16.71^k n^8)$ in total.

The following theorem summarizes the result of this section.

Theorem 8 *Given a plane graph G with n vertices and a set S of n points, we can decide convex point-set embeddability of G on S in $O(k^3 16.71^k n^8)$ time, where $k > 0$ is the number of points interior to the convex hull of S.*

Chapter 8

Conclusion and Open Problems

This chapter summarizes the main achievements of the thesis and suggests possible avenues for future research.

NP-completeness of Point-Set Embeddability

In Chapter 3 we proved that the point-set embeddability problem is NP-hard for 3-connected planar graphs. Recently, Biedl and Vatshelle [9] have strengthened this result by proving that the problem remains NP-hard for 3-connected planar graphs with constant treewidth. Although the proof of Biedl and Vatshelle is stronger with respect to the structure of the input graph, the point set they used in their reduction is complex. More specifically, while our proof holds for the case when all the points lie on four parallel straight lines and two parabolas, the points in Biedl and Vatshelle's proof lie on linear number of circles. Therefore, we ask the following question.

Open Question 1. What is the time complexity of the point-set embeddability problem for 3-connected planar graphs when the points lie on a constant number of parallel straight lines?

The point-set embeddability problem is polynomial-time solvable for outerplanar graphs [13], but NP-hard for 2-connected and 2-outerplanar graphs [20]. This leaves the case when the input graph is 3-connected and 2-outerplanar open. In particular, this graph class includes Halin graphs, i.e., the 3-connected planar graphs whose internal vertices induce a tree [45].

Open Question 2. Is the point-set embeddability problem polynomial-time solvable for Halin graphs?

Approximate Point-Set Embeddings

In Chapter 4 we gave an algorithm for testing point-set embeddability of a plane 3-tree with n vertices in $O(n \log^3 n)$ time. Our algorithm is near optimal since any such algorithm must take $\Omega(n \log n)$ time [62]. A natural optimization question in this direction is as follows.

Open Question 3. Given a plane 3-tree G with n vertices and a set S of n points in general position, how fast can we compute a straight-line embedding of G such that the number of vertices placed on the points of S is maximized?

Let Γ be a straight-line drawing of G. Then $S(\Gamma)$ denotes the number of vertices in Γ that are mapped to distinct points of S. The *optimal point-set embedding* of G

is a straight-line drawing Γ^* such that $S(\Gamma^*) \geq S(\Gamma')$ for any straight-line drawing Γ' of G. A ρ-approximation point-set embedding algorithm computes a straight-line drawing Γ of G such that $\frac{S(\Gamma)}{S(\Gamma^*)} \geq \rho$. In the following we show that given a plane 3-tree G with n vertices, one can construct a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$, and hence point-set embeddability is approximable with factor $\Omega(\frac{1}{\sqrt{n}})$ for plane 3-trees.

Let S be a partially ordered set of size k. A *chain* C is a subset of S such that every pair of elements in C are comparable. Similarly, an *antichain* C' is a subset of S such that no two elements of C' are comparable. Dilworth [31] proved that in any partially ordered set of k elements, there exists a chain or an antichain of size at least $\sqrt{k}$. Since a tree T with k nodes is a partially ordered set under the 'successor' relation, either the height of T, or the number of leaves in T is at least $\sqrt{k}$. Assume now that G is the input plane 3-tree with n vertices and let T be its representative tree with $n-3$ vertices. If T has $\Omega(\sqrt{n})$ leaves then we use the technique of Theorem 4 to have a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$. Otherwise, the height of T is $\Omega(\sqrt{n})$. In this case we can prove that G has a canonical ordering tree [87] (also, called Schnyder's realizer [67]) with height $\Omega(\sqrt{n})$ and use de Fraysseix et al.'s algorithm [29] to obtain a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$.

Open Question 4. Either design a polynomial-time algorithm that can approximate point-set embeddability for plane 3-trees with a constant factor, or prove that no such algorithm exists.

Observe that we can use the technique of Theorem 4 to have a 1-bend straight-line drawing of G that uses $n/4$ distinct points of S (e.g., choose an independent set of

$n/4$ vertices and place those vertices on $n/4$ distinct points of S using at most one bend per edge). Consequently, 1-bend point-set embeddability is approximable with at least factor 0.25 for plane 3-trees.

Universal Point Set

In Chapter 5 we proved that Kurowski's lower bound on universal point set is erroneous and the $1.098n$ lower bound previously obtained by Chrobak and Karloff [26] is still the best known. However, the best known upper bound on the size of universal point set is $\frac{8}{9}n^2$ [15]. A long-standing open question in graph drawing asks to design a set of $O(n)$ points, which is universal for all planar graphs with n vertices [16]. Here we ask an easier question.

Open Question 5. Design a set of $O(n \log n)$ points, which is universal for all plane 3-trees with n vertices. Given two arbitrary plane 3-trees, each with n vertices, compute a minimal point set (i.e., point set with smallest cardinality) that is universal for the input plane 3-trees.

We have also proved that no set of 24 points can be universal for all planar graphs with 24 vertices. However, the smallest value m such that no set of m points is universal for all planar graphs with m vertices is still unknown. We believe that one can restrict the focus only on plane 3-trees and can refine our proof technique further to have tighter bounds on the value of m.

Open Question 6. What is the smallest value m such that no set of m points is universal for all plane 3-trees with m vertices?

Convex Point-Set Embeddability

In Chapter 6 we gave an $O(n^{8+\epsilon})$-time algorithm to test convex point-set embeddability for klee graphs, which is a subclass of 3-connected planar graphs. Recently, Biedl and Vatshelle [9] proved that the convex point-set embeddability problem is NP-complete for 3-connected planar graphs with unbounded treewidth. In this context, it is natural to ask the following questions.

Open Question 7. How fast can we test convex point-set embeddability of a klee graph? What is the time complexity of the convex point-set embeddability problem for 3-connected planar graphs with constant treewidth?

In Chapter 7 we gave an $O^*(16.71^k)$-time fixed parameter tractable algorithm to decide convex point-set embeddability of arbitrary planar graphs, where k is the number of points interior to the convex hull of the input point set. Since the base in the exponentiation 16.71^k is a large constant, we ask to design a faster algorithm.

Open Question 8. Does there exist an $O(2^k n^c)$-time algorithm to decide convex point-set embeddability of planar graphs with n vertices, where c is a fixed constant and k is the number of points interior to the convex hull?

Bibliography

[1] Pankaj K. Agarwal. Range searching. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Computational Geometry*, chapter 36, pages 809–838. CRC Press, 2nd edition, April 2004. (Cited on page 20.)

[2] Miklós Ajtai, Vašek Chvátal Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. *Annals of Discrete Mathematics*, 12:9–12, 1982. (Cited on page 60.)

[3] Noga Alon and Yair Caro. On the number of subgraphs of prescribed type of planar graphs with a given number of vertices. In *Proceedings of the Conference on Convexity and Graph Theory, Israel*, volume 87, pages 25–36. Elsevier, March 1981. (Cited on page 14.)

[4] Patrizio Angelini, Giuseppe Di Battista, and Fabrizio Frati. Simultaneous embedding of embedded planar graphs. In *Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC 2011), Yokohama, Japan*, LNCS. Springer-Verlag, December 5–8 2011 (In press). (Cited on page 6.)

[5] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis.

Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, 1999. (Cited on page 1.)

[6] Lowell W. Beineke and Raymond E. Pippert. Enumerating dissectible polyhedra by their automorphism groups. *Canadian Journal of Mathematics*, XXVI(1): 50–67, 1974. (Cited on page 62.)

[7] G. J. Bezem and Jan van Leeuwen. Enumeration in graphs. *Technical Report RUU-CS-87-7, Department of Computer Science. University of Utrecht*, May 1987. (Cited on page 99.)

[8] Therese Biedl. 2012. Personal Communication. (Cited on pages 84, 90, and 91.)

[9] Therese Biedl and Martin Vatshelle. The point-set embeddability problem for plane graphs. In *Proceedings of the 27th ACM Symposium on Computational Geometry (SoCG 2012), Paris, France*. ACM, June 13-15 2012, In press. (Cited on pages 16, 44, 90, 91, 101, and 105.)

[10] George D. Birkhoff. On the number of ways of colouring a map. *Proceedings of the Edinburgh Mathematical Society*, 2(2):83–91, 1930. (Cited on page 14.)

[11] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. (Cited on page 16.)

[12] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1–2):1–45, December 1998. (Cited on page 16.)

[13] Prosenjit Bose. On embedding an outer-planar graph in a point set. *Computational Geometry: Theory and Applications*, 23(3):303–312, November 2002. (Cited on pages 2, 4, and 102.)

[14] Prosenjit Bose, Michael McAllister, and Jack Snoeyink. Optimal algorithms to embed trees in a point set. *Journal of Graph Algorithms and Applications*, 1(2): 1–15, 1997. (Cited on page 4.)

[15] Franz J. Brandenburg. Drawing planar graphs on $\frac{8}{9}n^2$ area. *Electronic Notes in Discrete Mathematics*, 31:37–40, August 2008. (Cited on pages 2, 17, and 104.)

[16] Franz J. Brandenburg, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, Giuseppe Liotta, and Petra Mutzel. Selected open problems in graph drawing. In *Proceedings of the 11th International Symposium on Graph Drawing (GD 2003)*, volume 2912 of *LNCS*, pages 515–539. Springer, September 21–24 2004. (Cited on pages 58 and 104.)

[17] Gerth Stølting Brodal and Riko Jacob. Dynamic planar convex hull with optimal query time. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT 2000), Bergen, Norway*, volume 1851 of *LNCS*, pages 57–70. Springer, July 5–7 2000. (Cited on page 20.)

[18] Gerth Stølting Brodal and Riko Jacob. Dynamic planar convex hull. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS 2002), Vancouver, BC, Canada*, pages 617–626. IEEE Computer Society, November 16–19 2002. (Cited on pages 20 and 57.)

[19] Kevin Buchin, Christian Knauer, Klaus Kriegel, André Schulz, and Raimund Seidel. On the number of cycles in planar graphs. In *Proceedings of the 13th Annual International Conference on Computing and Combinatorics (COCOON 2007), Banff, Canada*, volume 4598 of *LNCS*, pages 97–107. Springer, July 16-19 2007. (Cited on page 99.)

[20] Sergio Cabello. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. *Journal of Graph Algorithms and Applications*, 10(2):353–363, 2006. (Cited on pages ii, 2, 4, 6, and 102.)

[21] Netzahualcoyotl Castañeda and Jorge Urrutia. Straight line embeddings of planar graphs on point sets. In *Proceedings of the 8th Canadian Conference on Computational Geometry (CCCG 1996), Ottawa, ON, Canada*, pages 312–318, August 12–15 1996. (Cited on page 4.)

[22] Timothy M. Chan. Dynamic planar convex hull operations in near-logarithmaic amortized time. *Journal of the ACM*, 48(1):1–12, January 2001. (Cited on page 20.)

[23] Bernard Chazelle, Micha Sharir, and Emo Welzl:. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8(5&6):407–429, 1992. (Cited on pages 20, 48, and 84.)

[24] Norishige Chiba, Tadashi Yamanouchi, and Takao Nishizeki. Linear algorithms for convex drawings of planar graphs. *Progress in Graph Theory*, pages 153–173, 1984. (Cited on page 96.)

[25] Marek Chrobak and Goos Kant. Convex grid drawings of 3-connected planar graphs. *International Journal of Computational Geometry & Applications*, 7(3): 211–223, 1997. (Cited on pages 2 and 18.)

[26] Marek Chrobak and Howard J. Karloff. A lower bound on the size of universal sets for planar graphs. *SIGACT News*, 20(4):83–86, 1989. (Cited on pages vii, 3, 5, 9, 58, 59, 65, 66, 67, 68, and 104.)

[27] Vasek Chvátal. Hamiltonian cycles. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem*, chapter 11, pages 403–429. John Wiley and Sons, New York, NY, 1985. (Cited on page 24.)

[28] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, March 2008. (Cited on page 19.)

[29] Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. (Cited on pages 2, 3, 5, 17, 58, and 103.)

[30] Markus Denny and Christian Sohler. Encoding a triangulation as a permutation of its point set. In *Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG 1997), Ontario, Canada*, August 11-14 1997. (Cited on page 60.)

[31] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950. (Cited on page 103.)

[32] Stephane Durocher, Debajyoti Mondal, Rahnuma Islam Nishat, Md. Saidur Rahman, and Sue Whitesides. Embedding plane 3-trees in $\mathbb{R}^2$ and $\mathbb{R}^3$. In *Proceedings of the 19th International Symposium on Graph Drawing (GD 2011), Eindhoven, The Netherlands*, volume 7034 of *LNCS*. Springer, September 21-23 2012. (Cited on pages 4, 5, 9, 45, 46, 48, and 85.)

[33] Jeff Erickson. New lower bounds for Hopcroft's problem. *Discrete & Computational Geometry*, 16(4):389–418, 1996. (Cited on page 45.)

[34] Jeff Erickson. On the relative complexities of some geometric problems. In *Proceedings of the 7th Canadian Conference on Computational Geometry*, pages 85–90, June 13-15 1995. (Cited on page 45.)

[35] Louis Esperet, Frantisek Kardos, and Daniel Král. Cubic bridgeless graphs have more than a linear number of perfect matchings. *Electronic Notes in Discrete Mathematics*, 34:411–415, 2009. (Cited on page 14.)

[36] Leonhard Euler. Demonstratio nonnullarum insignum proprietatum quibus solida hedris planis inclusa sunt praedita. *Novi commentarii academiae scientiarum Petropolitanae*, 4:140–160, 1758. (Cited on page 13.)

[37] István Fáry. On straight line representation of planar graphs. In *Acta Scientiarum Mathematicarum Szeged*, volume 11, pages 229–233, 1948. (Cited on page 17.)

[38] Rudolf Fleischer and Colin Hirsch. Graph drawing and its applications. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs: Methods and Models*, volume 2025 of *LNCS*, chapter 1, pages 1–22. Springer, 2001. (Cited on page 1.)

[39] Ulrich Fößmeier and Michael Kaufmann. Nice drawings for planar bipartite graphs. In *Proceedings of 3rd Italian Conference on Algorithms and Complexity (CIAC 1997), Rome, Italy*, LNCS, pages 122–134. Springer, March 12–14 1997. (Cited on page 44.)

[40] Michael Randolph Garey and David Stier Johnson. *Computers and intractability.* Freeman, San Francisco, 1979. (Cited on page 21.)

[41] Michael Randolph Garey, David Stifler Johnson, and Robert Endre Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. (Cited on pages 8 and 28.)

[42] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001. (Cited on page 6.)

[43] Emilio Di Giacomo and Giuseppe Liotta. The Hamiltonian augmentation problem and its applications to graph drawing. In *Proceedings of the 4th International Workshop on Algorithms and Computation (WALCOM 2010), Dhaka, Bangladesh*, volume 5942 of *LNCS*, pages 35–46. Springer, February 10–12 2010. (Cited on pages 59 and 68.)

[44] Oded Goldreich. *P, NP, and NP-Completeness: The Basics of Computational Complexity*. Cambridge University Press, 1st edition, October 2010. (Cited on page 44.)

[45] Rudolf Halin. Über simpliziale zerfällungen beliebiger (endlicher oder unendlicher) graphen. *Mathematische Annalen*, 156(3):216–225, 1964. (Cited on page 102.)

[46] Yoshiko Ikebe, Micha A. Perles, Akihisa Tamura, and Shinnichi Tokunaga. The rooted tree embedding problem into points in the plane. *Discrete and Computational Geometry*, 11(1):51–63, December 1994. (Cited on page 3.)

[47] Riko Jacob. *Dynamic Planar Convex Hull*. PhD thesis, University of Aarhus, Denmark, May 2002. (Cited on page 20.)

[48] Michael Jünger and Petra Mutzel, editors. *Graph Drawing Software*. Mathematics and Visualization. Springer-Verlag, 1st edition, October 2003. (Cited on page 1.)

[49] Frank Kammer and Torsten Tholey. A lower bound for the treewidth of k-outerplanar graphs. *Technical Report, Institute of Computer Science, University of Augsburg, Germany*, pages 1–7, April 2009. (Cited on page 16.)

[50] Vinay G. Kane and Sanat K. Basu. On the depth of a planar graph. *Discrete Mathematics*, 14(1):63–67, 1976. (Cited on page 13.)

[51] Bastian Katz, Marcus Krug, Ignaz Rutter, and Alexander Wolff. Manhattan-geodesic embedding of planar graphs. In *Proceedings of the 17th International*

Symposium on Graph Drawing, (GD 2009), Chicago, IL, USA, volume 5849 of *LNCS*, pages 207–218. Springer, September 22–25 2010. (Cited on pages 2, 5, and 44.)

[52] Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1): 115–129, 2002. (Cited on pages 5, 6, 18, 23, 24, 26, 59, and 68.)

[53] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison Wesley, March 2005. (Cited on page 19.)

[54] Ton Kloks. *Treewidth: Computations and Approximations*. Springer, July 1994. (Cited on page 15.)

[55] Stephen Kobourov. In *Open-Problem Session at the 12th Annual Fall DIMACS Workshop on Computational Geometry, Piscataway, NJ, USA*, November 14 – 15 2002. http://maven.smith.edu/~orourke/TOPP/P45.html, Accessed May 02, 2012. (Cited on pages ii, 6, 9, 58, and 63.)

[56] Hannes Krasser. *Order types of point sets in the plane*. PhD thesis, Graz University of Technology, Austria, October 2003. (Cited on pages 60 and 61.)

[57] Maciej Kurowski. A 1.235 lower bound on the number of points needed to draw all n-vertex planar graphs. *Information Processing Letters*, 92(2):95–98, October 2004. (Cited on pages vii, 3, 5, 7, 9, 58, 59, 63, 64, and 104.)

[58] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10: 96–115, 1927. (Cited on pages 32 and 33.)

[59] Debajyoti Mondal, Rahnuma Islam Nishat, Md. Saidur Rahman, and Md. Jawaherul Alam. Minimum-area drawings of plane 3-trees. *Journal of Graph Algorithms and Applications*, 15(2):177–204, February 2011. (Cited on pages 64, 75, 77, and 88.)

[60] Tanaeem Md. Moosa and Md. Sohel Rahman. Improved algorithms for the point-set embeddability problem for plane 3-trees. In *Proceedings of the 17th International Computing and Combinatorics Conference (COCOON 2011), Dallas, TX, USA*, volume 6842 of *LNCS*, pages 204–212. Springer, August 14–16 2011. (Cited on pages ii, 4, 7, 9, 45, 46, and 48.)

[61] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, USA, March 2006. (Cited on page 22.)

[62] Rahnuma Islam Nishat, Debajyoti Mondal, and Md. Saidur Rahman. Point-set embeddings of plane 3-trees. *Computational Geometry: Theory and Applications*, 45(3):88–98, April 2012. (Cited on pages 4, 6, 7, 9, 10, 45, 46, 48, and 102.)

[63] Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*. World Scientific, Singapore, September 2004. (Cited on pages 1, 12, 13, and 17.)

[64] Mark H. Overmars and Jan van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23(2):166–204, October 1981. (Cited on pages 19, 54, and 57.)

[65] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1st edition, December 1993. (Cited on page 44.)

[66] Ronald C. Read and Robert Endre Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5(3), 1975. (Cited on page 99.)

[67] Walter Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1990), San Francisco, California, USA*, pages 138–148. ACM, January 22–24 1990. (Cited on pages 2, 17, and 103.)

[68] Raimund Seidel. On the number of triangulations of planar point sets. *Combinatorica*, 18(2):297–299, 1998. (Cited on page 60.)

[69] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science (FOCS 1975), California, USA*, pages 151–162. IEEE, October 13–15 1975. (Cited on page 71.)

[70] Micha Sharir and Adam Sheffer. Counting triangulations of planar point sets. *The Electronic Journal of Combinatorics*, 18(1), 2011. (Cited on page 60.)

[71] Micha Sharir and Emo Welzl. Random triangulations of planar point sets. In *Proceedings of the 22nd ACM Symposium on Computational Geometry (SoCG 2006), Arizona, USA, 2006*, pages 273–281. ACM, June 5–7 2006. (Cited on page 60.)

[72] Neil J. A. Sloane. Sequence a027610. http://oeis.org/A027610, Accessed May 02, 2012. (Cited on page 63.)

[73] Andreas Spillner. A fixed parameter algorithm for optimal convex partitions. *Journal of Discrete Algorithms*, 6(4):561–569, December 2008. (Cited on pages vii, x, 91, 92, 93, 95, 96, 97, 98, 99, and 100.)

[74] William L. Steiger and Ileana Streinu. Illumination by floodlights. *Computational Geometry: Theory and Applications*, 10(1):57–70, April 1998. (Cited on pages 53, 57, and 69.)

[75] K.S. Stein. Convex maps. In *American Mathematical Society*, volume 2, pages 464–466, 1951. (Cited on page 17.)

[76] James Stirling. *Methodus differentialis*. 1730. English translation by J. Holliday, The Differential Method: A Treatise of the Summation and Interpolation of Infinite Series. 1749. (Cited on page 65.)

[77] Kozo Sugiyama. A cognitive approach for graph drawing. *Cybernetics and Systems: An International Journal*, 18(6):447–488, 1987. (Cited on page 2.)

[78] Fujio Takeo. On triangulated graphs I. *Bulletin of Fukuoka University of Education III*, pages 9–21, 1960. (Cited on page 14.)

[79] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987. (Cited on page 6.)

[80] Roberto Tamassia, Giuseppe Di Battista, and Carlo Batini. Automatic graph

drawing and readability of diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):61–79, January 1988. (Cited on page 2.)

[81] Dimitrios M. Thilikos, Maria J. Serna, and Hans L. Bodlaender. Constructive linear time algorithms for small cutwidth and carving-width. In *Proceedings of the 11th International Conference on Algorithms and Computation (ISAAC 2000), Taipei, Taiwan*, volume 1969 of *LNCS*, pages 192–203. Springer, December 18–20 2000. (Cited on pages 16 and 90.)

[82] William Thomas Tutte. Convex representations of graphs. In *London Mathematical Society*, volume 10, pages 304–320, 1960. (Cited on pages 2 and 18.)

[83] William Thomas Tutte. How to draw a graph. *London Mathematical Society*, 13 (52):743–768, 1963. (Cited on page 32.)

[84] Klaus Wagner. Bemerkungen zum vierfarbenproblem. In *Jahresbericht Deutsche Math*, volume 46, pages 26–32, 1936. (Cited on page 17.)

[85] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, September 2000. (Cited on pages 12 and 13.)

[86] Shmuel Zaks. Generation and ranking of k-ary trees. *Information Processing Letters*, 14(1):44–48, 1982. (Cited on page 65.)

[87] Huaming Zhang and Xin He. Canonical ordering trees and their applications in graph drawing. *Discrete & Computational Geometry*, 33(2):321–344, February 2005. (Cited on page 103.)

Index