M. Sc Engineering Thesis

Point-Set Embeddings of Planar Graphs with Fewer Bends

by

Md. Emran Chowdhury

Submitted to

Department of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

Department of Computer Science and Engineering

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Dhaka-1000, Bangladesh

December, 2010

The thesis titled "**Point-Set Embeddings of Planar Graphs with Fewer Bends**", submitted by Md. Emran Chowdhury, Roll No. 040805068P, Session April 2008, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on December 15, 2010.

Board of Examiners

Candidate's Declaration

This is to certify that the work presented in this thesis entitled "**Point-Set Embeddings of Planar Graphs with Fewer Bends**" is the outcome of the investigation carried out by me under the supervision of Professor Dr. Md. Saidur Rahman in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka. It is also declared that neither this thesis nor any part thereof has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.

Md. Emran Chowdhury
Candidate

Contents

Board of Examiners **i**

Candidate's Declaration **ii**

Acknowledgments **vii**

Abstract **viii**

1 Introduction **1**

1.1 Point-Set Embeddings of Planar Graphs 3

1.1.1 Straight-line Point-Set Embedding 3

1.1.2 Orthogonal Point-Set Embedding 5

1.1.3 Upward Point-Set Embedding 5

1.2 Applications of Point-Set Embeddings 6

1.2.1 VLSI Layout 6

1.2.2 Hierarchical Structure 8

1.3 Scope of This Thesis 8

1.3.1 Previous Results 8

1.3.2 Results in This Thesis 9

1.4 Thesis Organization 10

2 Preliminaries **11**

2.1 Basic Terminology 11

2.1.1 Graphs 11

2.1.2 Simple Graphs and Multigraphs 12

2.1.3 Directed and Undirected Graphs 12

2.1.4 Paths and Cycles 14

2.1.5 Connectivity 14

2.2 Planar Graphs 15
2.3 Drawing Conventions 17
2.3.1 Planar Drawing 17
2.3.2 Straight-line Drawing 18
2.3.3 Orthogonal Drawing 18
2.4 Topological Book Embedding 19
2.5 Complexity of Algorithms 21
2.5.1 The Notation $O(n)$ 21
2.5.2 Polynomial Algorithms 22
2.5.3 NP-complete Problems 22

3 Upward Point-Set Embedding **24**
3.1 Introduction 24
3.2 Upward Topological Book Embedding 25
3.3 Upward Point-Set Embedding 30
3.4 Conclusion 32

4 Orthogonal Point-set Embedding **33**
4.1 Introduction 33
4.2 3-Connected Cubic Planar Graphs 34
4.3 4-Connected Planar Graphs 42
4.4 Conclusion 50

5 Conclusion **51**

References **53**

Index **57**

List of Figures

1.1 An example of graph drawing in circuit schematics. 2

1.2 (a) Planar graph G, (b) point-set S, (c) straight-line point-set embedding Γ of G on S. 4

1.3 3-connected cubic planar graph with point-set. 4

1.4 An orthogonal drawing of a planar graph G of Fig. 1.3. 5

1.5 An upward planar digraph G and a point-set S. 6

1.6 (a) Interconnection graph G, (b) point-set S, (c) VLSI Layout of G on S. 7

2.1 A graph with six vertices and eight edges. 12

2.2 Multigraphs. 13

2.3 Undirected and directed graphs. 13

2.4 External Hamiltonian cycle of a 3-connected cubic planar graph G. 14

2.5 Connected and disconnected graphs. 15

2.6 Two planar embeddings of the same planar graph. 16

2.7 Inner edges and outer edges of cycle C. 17

2.8 Planar and non-planar drawings. 18

2.9 A straight line drawing. 18

2.10 (a) A planar graph G, (b) orthogonal drawing of G with ten bends. 19

2.11 (a) A planar graph G, (b) a topological book embedding of G with a spine crossing by the edge (b, e). 20

2.12 (a) A planar graph G, (b) a topological book embedding of G with a spine crossing by the edge $(2, 5)$. 21

3.1 (a) An upward planar digraph G, (b) a point-set S, (c) an upward point-set embedding Λ of G on S 24

3.2 (a) Upward planar straight line drawing λ of G, (b) graph G' of λ. 27
3.3 (a) Upward topological book embedding γ' of G', (b) upward topological book embedding γ. 28
3.4 Illustration of the proof of Lemma 3.2.4 29
3.5 (a) An upward planar digraph G, (b) projection of S on line l' and l'' 31

4.1 (a) 3-connected cubic planar graph G, (b) topological book embedding γ with no spine crossing. 35
4.2 (a) Topological book embedding γ, (b) orthogonal drawing Γ. . 36
4.3 3-connected cubic planar graph G and point-set S 38
4.4 Orthogonal drawing Γ of G on S of Fig. 4.3 40
4.5 (a) 4-connected planar graph G, (b) point-set S, (c) hamiltonian cycle C in G' 43
4.6 (a) Topological book embedding γ' of G', (b) orthogonal drawing Γ of G 44
4.7 Different types of edges of case 2(b) 46
4.8 Different types of edges of case 3(b) 47
4.9 Tight example of 4-connected planar graphs for which $6n$ bends are necessary 49

Acknowledgments

I would like to thank, first of all, my supervisor Professor Dr. Md. Saidur Rahman. He introduced me to this wonderful and fascinating world of *graph theory* and *graph drawing*. His wonderful classes both in undergraduate and graduate level stimulated me to work in this field under his supervision. Whenever I found myself in stalemate condition, his helpful direction revived my interest and enthusiasm all the time. Without his constant supervision and words of encouragement, it would almost impossible for me to finish this work.

I also thank the members of my thesis committee: Prof. Dr. Md. Monirul Islam, Dr. Mahmuda Naznin, and Prof. Dr. Suraiya Parveen for their valuable and helpful suggestions. This thesis work is done in the *Graph Drawing and Information Visualization Lab*, Dept. of CSE, BUET. It is my great pleasure to work in such a wonderful environment.

I would like to express my deep gratitude to all the members of our research group for their valuable suggestions and continual encouragements. My special thanks goes to Muhammad Jawaherul Alam. My parents also supported me and encouraged me to the best of their ability. My heart-felt gratitude goes to them.

Abstract

An upward point-set embedding of an upward planar digraph G on a set of points S with a mapping $\Phi : V(G) \rightarrow S$ is a drawing Γ of G where each vertex of G is placed on a point of S according to Φ, each edge is drawn upward and no two edges cross each other. In this thesis we give an algorithm for finding an upward point-set embedding of an upward planar digraph G with a mapping if it exists. Our algorithm finds a drawing with at most $n-3$ bends per edge whereas the best known previous algorithm finds a drawing with at most $2n-3$ bends. Furthermore we also find an upper bound on total number of bends for an upward point-set embedding.

An orthogonal point-set embedding of a planar graph G on a set S of points in Euclidean plane is a drawing of G where each vertex of G is placed on a point of S, each edge is drawn as a sequence of alternate horizontal and vertical line segments and any two edges do not cross except at their common end. Orthogonal point-set embeddings have practical applications in circuit schematics on pre-fabricated printed circuit boards (PCBs), where position of components on the PCB is prescribed and standard cell technology employed during the VLSI layout design process. In both the cases, it is always desirable to minimize the number of bends, points at which the edge changes its direction in the drawing. Because bends increase the manufacturing cost in PCB board and VLSI chip. In this thesis we devise an algorithm for orthogonal point-set embedding of 3-connected cubic planar graphs having a hamiltonian cycle with at most $\left(\frac{5n}{2}+2\right)$ bends. We also give an algorithm for finding an orthogonal point-set embedding of 4-connected planar graphs ($\Delta \leq 4$) with at most $6n$ bends. To the best of our knowledge this is the first work on orthogonal point-set embedding with fewer bends.

Chapter 1

Introduction

A *graph* is a mathematical tool that can efficiently represents the real world objects and the relationships among them. Generally the objects are represented by small dots named *vertices* and the relationships are represented by connecting lines called *edges*. A graph is often used in computer science because of its versatile usefulness to represent any information that can be modeled as objects and relationships between pairs of them. As a consequence, graphs became the most reliable and useful mathematical tool not only in the fields of computer science but also in the areas of electrical, architectural and other field of engineering, genetics, bioinformatics, molecular biology, chemistry, VLSI technology, and even in geology, social sciences. Information visualization is one of the vital parts of graph theory in almost each of its application areas. A drawing of a graph is a sort of visualization of information represented by that graph. The field in which the different aesthetic techniques of drawing graphs are vividly described is "Graph Drawing".

The origin of *Graph Drawing* is not well known. It is relatively a new area in Computer Science. The field of graph drawing is motivated by its abundant applications for information visualization and also for VLSI circuit design, social network analysis, cartography, and bioinformatics. The graph in Fig. 1.1(a) has six vertices and ten edges can easily be visualized if we can represent the same graph as in Fig. 1.1(b) which is one of the drawing techniques generally used in graph drawing. Apart from these, graph drawing, particularly automated generation of the drawings of graphs, nowadays finds inducive applications in software engineering (data-flow diagrams, subroutine-call graphs,

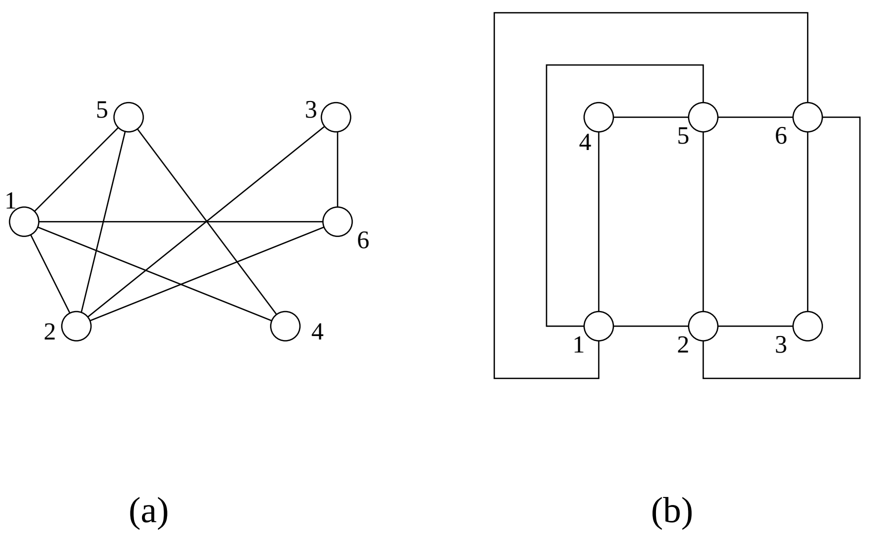


Figure 1.1: An example of graph drawing in circuit schematics.

object-oriented class hierarchies etc.), databases (ER-diagrams), information systems (organizational charts), real-time systems (Petri-nets, state-transition diagrams), Decision support systems (PERT networks, activity trees), electrical and VLSI circuit design (layout design and circuit schematics), artificial intelligence (knowledge-representation diagrams), logic programming (SLD-trees), biology and phylogenetics (evolutionary trees), medical sciences (concept lattices), chemistry (molecular drawings), civil engineering (architectural floorplan) and cartography (map schematics) [Rah99].

In the field of graph drawing, the geometric representations of graphs generated by graph drawing algorithms are constrained by some predefined geometric or aesthetic properties. In another words, the objective of graph drawing is to obtain a nice representation of a graph such that the structure of the graph is easily understandable, moreover the drawing should help to resolve the question arises from the application point of view using predefined properties.

Point-set embedding of a plane graph G on a set of points S is a planar drawing Γ of G where each vertex of G is placed on a point of S. Γ is called orthogonal if each edge of Γ is drawn by as a sequence of alternate horizontal and vertical line segments. Orthogonal drawings have attracted much attention due to their numerous applications in circuit layouts, database diagrams, entity-relationship diagrams, etc. The drawing in Fig. 1.1(b) is an example of an orthogonal drawing of graph in Fig. 1.1(a). Every plane graph with $\Delta \leq 4$

has an orthogonal drawing, but may need bends, that is, points where an edge changes its direction in a drawing. If a graph corresponds to a VLSI circuit, then it is often desirable to find an orthogonal drawing with fewer bends, because bends increase the manufacturing cost of a VLSI chip. In this thesis, we at first address the problem of finding an upward point-set embedding of an upward planar digraph with mapping allowing $n-3$ bends per edge. After that we devise two different linear time algorithms for orthogonal drawings of 3-connected cubic planar graphs having a hamiltonian cycle and 4-connected planar graphs with smaller number of bends on a point-set.

In the rest of this chapter, we provide the necessary background and objectives for this thesis. We describe point-set embeddings of planar graphs in Section 1.1 and Section 1.2 depicts some interesting applications of point-set embeddings. Section 1.3 presents the scope of this thesis with a brief overview of the previous results related to the scope and the new results described in this thesis. Finally the thesis organization is narrated in Section 1.4.

1.1 Point-Set Embeddings of Planar Graphs

In this section we describe the definition of different types of point-set embeddings.

1.1.1 Straight-line Point-Set Embedding

A straight-line drawing of a planar graph G is a drawing of G, where each vertex is drawn as a point and each edge is drawn as a straight-line segment and no two edges do not cross except their common end. Given a set S of n points on the Euclidean plane, a *straight-line point-set embedding* of a planar graph G with n vertices on S is a straight-line drawing of G, where each vertex of G is mapped to a distinct point of S. Figure 1.2(c) is a straight-line point-set embedding of a planar graph G on S (Fig. 1.2(a),(b)) where each v_i of G is mapped to point i in Γ for $1 \leq i \leq 6$.

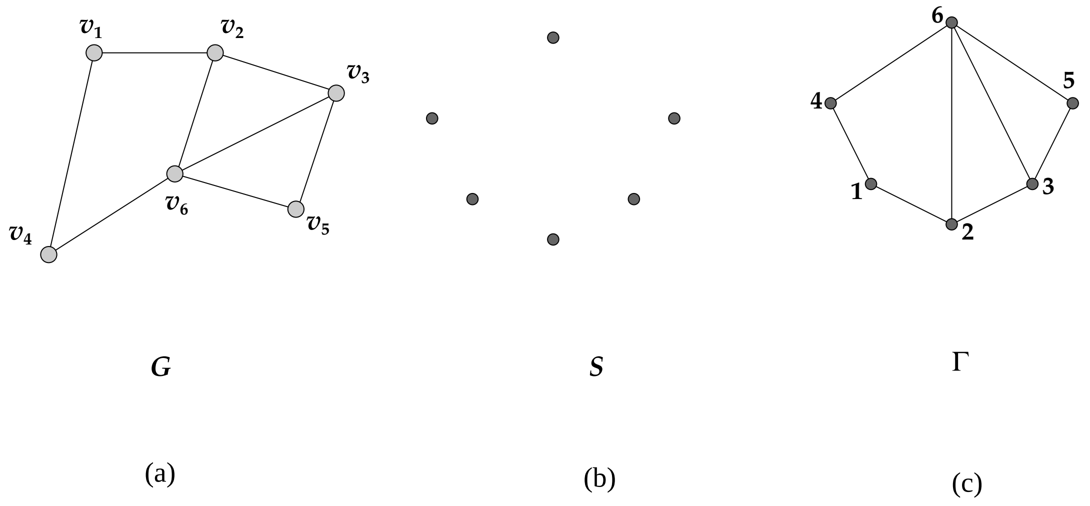


Figure 1.2: (a) Planar graph G, (b) point-set S, (c) straight-line point-set embedding Γ of G on S.

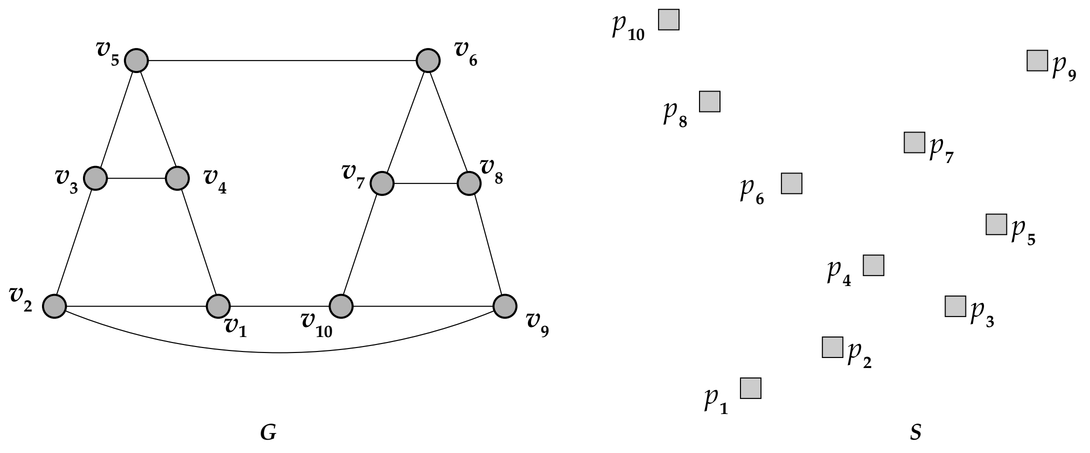


Figure 1.3: 3-connected cubic planar graph with point-set.

1.1.2 Orthogonal Point-Set Embedding

An *orthogonal point-set embedding* of a planar graph G on a set of points S is a drawing of G where each vertex of G is placed on a point of S, each edge is drawn as a sequence of alternate horizontal and vertical line segments and any two edges do not cross except at their common end. A planar graph G is said to be k-connected if removal of any set of $k-1$ vertices does not disconnect the graph G. If all the vertices of G have degree three, then G is cubic. A cubic planar graph G with a point-set S is shown in Fig. 1.3. Figure 1.4 is an orthogonal drawing of a 3-connected cubic planar graph G in point-set S depicted in Fig. 1.3.

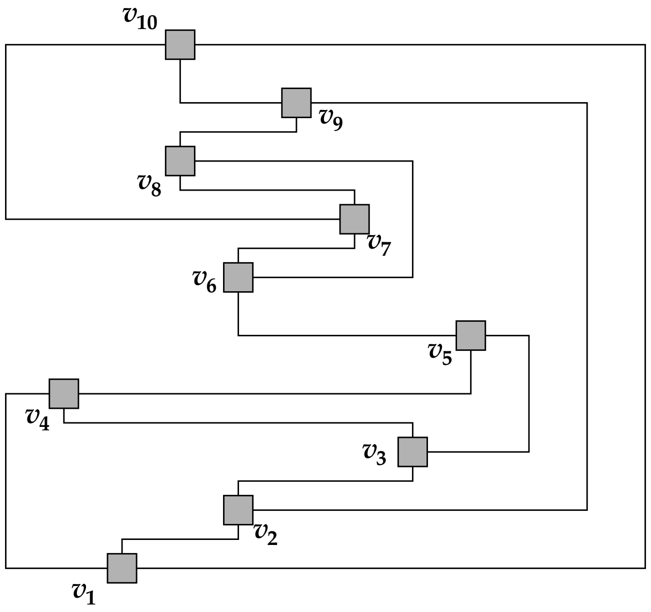


Figure 1.4: An orthogonal drawing of a planar graph G of Fig. 1.3.

1.1.3 Upward Point-Set Embedding

Let G be an upward planar digraph with n vertices and let S be a set of n distinct points in the plane. An *upward point-set embedding* of G on S is a drawing of G where each vertex of G is placed on a distinct point of S, each edge is drawn upward and no two edges cross each other. Suppose Φ is a mapping from the vertices of G to the points of S. An *upward point-set embedding* of G on S with the mapping Φ is an upward point-set embedding of G on S where the vertices of G are located at the points of S according to the mapping Φ

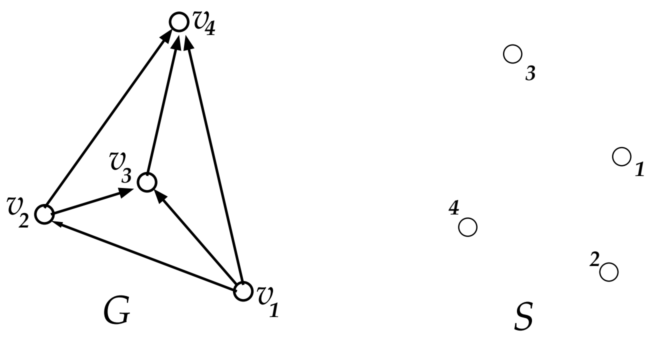


Figure 1.5: An upward planar digraph G and a point-set S.

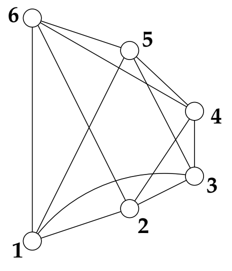
[GLW09]. G does not admit an upward point-set embedding on S with every mapping Φ. For example, one can easily observe that there exists no upward point-set embedding of the upward planar digraph G on the points S in Fig. 1.5 where each vertex v_i of G is mapped to the point i of S for $1 \leq i \leq 4$.

1.2 Applications of Point-Set Embeddings

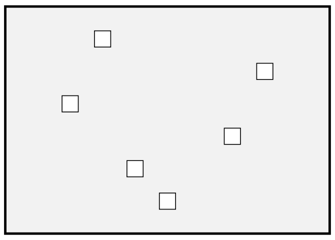
Point-set embeddings of planar graphs have number of applications in the areas of VLSI Layouts, circuit schematics, network flow models, Computer Networks, etc. We present a few applications of point-set embeddings for both undirected and directed graphs in the remainder of this section.

1.2.1 VLSI Layout

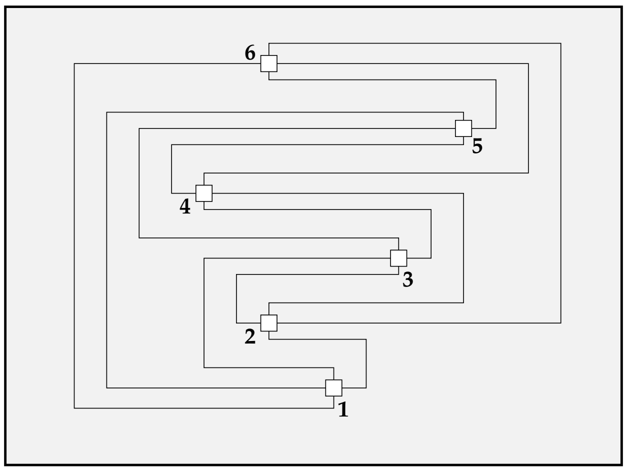
In the *standard cell* technology employed during the VLSI layout design process, the VLSI modules are placed on some constant number of previously fixed rows and columns. An orthogonal drawing of a graph in a point-set can be used to obtain a layout of an interconnection network on a standard cell. A vertex of the graph represents a module of the VLSI circuit and an edge represents an interconnection between two modules. The input graph represents the interconnection graph and the set of points represents the place of modules in VLSI standard cell. Thus we can find an appropriate VLSI layout of smaller bends using the orthogonal point-set embedding of planar graph G. The process is illustrated in Fig. 1.6. The Fig. 1.6(c) is a VLSI layout of a four-connected planar interconnection graph G of Fig. 1.6(a) in the point-set S of Fig. 1.6(b).



(a)



(b)



(c)

Figure 1.6: (a) Interconnection graph G, (b) point-set S, (c) VLSI Layout of G on S.

1.2.2 Hierarchical Structure

Upward point-set embedding has an immidiate application in representing the hierarchical structure of any nature. Our upward topological book embedding drawing is useful in the context of computing drawings of hierarchical structures where it is required to consider not only aesthetic constraints such as the upwardness and the planarity but also semantic constraints expressed in terms of collinearity for a set of vertices, i.e in the application domains of knowledge engineering and of project management, PERT diagrams are typically drawn by requiring that critical sequence of tasks be represented as collinear vertices [GLMS07].

There are also numerous applications of point-set embeddings of planar graphs. Upward point-set embedding turns out to be a classical problem of computational geometry. It is also useful in the context of visualization of self-modifiable code, based on computing a sequence of drawings whose edges are define at run-time [Hal91]. Nonetheless, orthogonal drawing alogorithms became useful tools for finding other important drawing algorithms such as octagonal drawing, box-orthogonal drawing and hexagonal drawing [NR04].

1.3 Scope of This Thesis

In this section we first mention the previous results of point-set embeddings of planar graphs and upward point-set embeddability for directed planar graphs. After that we will discuss the results obtained in this thesis.

1.3.1 Previous Results

The problem of embedding planar graphs in point-set has a rich collection of literature [Bos97, KW02, PGMP91, PW98]. Bose presented an $O(nlog^3n)$ time and $O(n)$ space algorithm for embedding outer-planar graph in point-set [Bos97]. Pach and Wenger developed $O(n^2)$ time drawing algorithm for embedding general planar graphs at fixed vertex locations where each edge is drawn by a polygonal curve with $O(n)$ bends [PW98]. In the later time Kaufman and Wiese developed two algorithms for drawing 4-connected plane graphs with at most one bend per edge and general plane graphs with at most two bends per edge [KW02]. Their both algorithms have $O(n^2)$ time complexity which can be improved to $O(nlogn)$ if three bends per edge are allowed in the drawings. The open problem of deciding whether there is a planar straight-line embedding of planar graph in point-set exists or not [Bos97, KW02] has been proved as NP-complete by Cabello [Cab06].

Point-set embedding of planar graphs with minimum number of bends is another objective for the current researchers. For the class of outer-planar triangulated *st*-digraphs, a recent result gives a linear-time algorithm for an upward topological book embedding with minimum number of bends where at most two bends per edge are allowed [MS09]. Giordano *et al.* proved that any planar digraph has an upward topological book embedding Γ such that every edge of G contains at most two bends in Γ and they have also developed another algorithm to obtain an upward point-set embedding of an upward planar digraph on any set of n distinct points in the plane but their algorithm does not minimize the total number of bends in Γ [GLMS07]. Furthermore, none of the algorithms mentioned so far has considered a given mapping of the vertices to the points. Then Giordano, Liotta and Whitesides modify the previous algorithm to give an algorithm to draw an upward point-set embedding of an upward planar digraph with a given mapping with at most $2n-3$ bends per edge [GLW09]. Moreover, they posted an open problem of minimizing the total number of bends of an upward point-set embedding in their paper.

1.3.2 Results in This Thesis

In this thesis we mainly provide three different algorithms. Our first algorithm is for finding upward point-set embeddings of upward planar digraphs. The next algorithm is for orthogonal point-set embeddings of 3-connected cubic planar graphs having a hamiltonian cycle. Our last algorithm is for orthogonal point-set embeddings of 4-connected planar graphs. We also find the upper bound on number of bends of these drawings. For a brief summary, we now list the results presented in this thesis as follows:

- At first we give an algorithm for upward point-set embeddings of upward planar digraphs. Our algorithm uses at most $n-3$ bends per edge which improves the best known previous upper bound of $2n-3$ bends per edge. We also find the upper bound on number of bends in the drawing.

- We give a linear time algorithm for orthogonal point-set embeddings of 3-connected cubic planar graphs with a hamiltonian cycle. Our algorithm finds a drawing with at most $\left(\frac{5n}{2}+2\right)$ bends.

- At last we provide another linear time algorithm for orthogonal point-set embeddings of 4-connected planar graphs ($\Delta \leq 4$) with fewer bends which has more practical applications in VLSI layout. Our algorithm finds a drawing with at most $6n$ bends.

1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we give some basic terminology of graph theory and algorithmic theory. Chapter 3 describes the algorithm for finding upward point-set embeddings of upward planar digraphs and related results. In Chapter 4, we give two algorithms for finding orthogonal point-set embeddings of 3-connected cubic planar graphs and 4-connected planar graphs. Finally Chapter 5 concludes the thesis with a summary of the results with some future works.

Chapter 2

Preliminaries

In this chapter we define some basic terminology of graph theory, graph drawing, book embedding and algorithm theory, that we will use throughout the rest of this thesis. In Section 2.1, we cover some definitions of standard graph-theoretical terms. We devote Section 2.2 to define terms related to planar graphs. Section 2.3 defines some drawing conventions and Section 2.4 consists of the terms related to topological book embedding. Finally we introduce the notion of time complexity of algorithms in Section 2.5.

2.1 Basic Terminology

In this section we give some definitions of standard graph-theoretical terms used throughout this thesis. For readers interested in more details of graph theory we refer to [NC88, NR04, Wes01].

2.1.1 Graphs

A *graph* G is a tuple (V, E) which consists of a finite set V of *vertices* and a finite set E of *edges*; each edge being an unordered pair of vertices. Figure 2.1 depicts a graph $G = (V, E)$ where each vertex in $V = \{v_1, v_2, \ldots, v_6\}$ is drawn as a small circle and each edge in $E = \{e_1, e_2, \ldots, e_8\}$ is drawn by a line segment.

We denote an edge joining two vertices u and v of the graph $G = (V, E)$ by (u, v) or simply by uv. If $uv \in E$ then the two vertices u and v of the graph G are said to be *adjacent*; the edge uv is then said to be *incident* to the vertices u and v; also the vertex u is said to be a neighbor of the vertex v (and vice

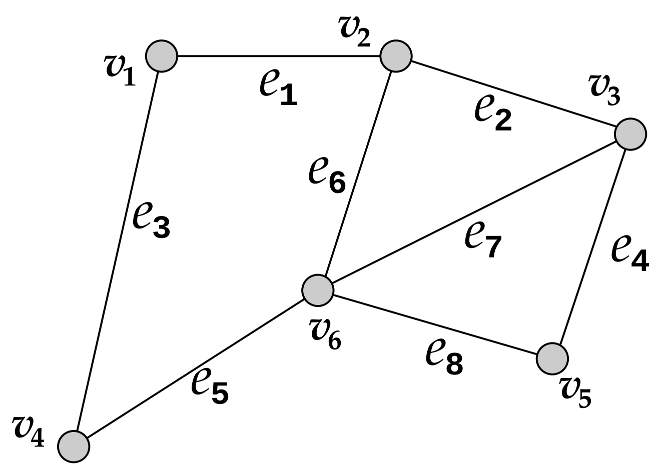


Figure 2.1: A graph with six vertices and eight edges.

versa). The *degree* of a vertex v in G, denoted by $d(v)$ or $deg(v)$, is the number of edges incident to v in G. In the graph shown in Figure 2.1 vertices v_1 and v_2 are adjacent, and $d(v_6) = 4$, since four of the edges, namely e_5, e_6, e_7 and e_8 are incident to v_6.

2.1.2 Simple Graphs and Multigraphs

If a graph G has no "multiple edges" or "loops", then G is said to be a *simple graph*. *Multiple edges* join the same pair of vertices, while a *loop* joins a vertex with itself. The graph in Figure 2.1 is a simple graph.

A graph in which loops and multiple edges are allowed is called a *multigraph*. Multigraphs can arise from various applications. One example is the "call graph" that represents the telephone call history of a network. The graph in Figure 2.2(a) is a call graph that represents the call history among six subscribers. Note that there is no loop in this graph. Figure 2.2(b) illustrates another multigraph with multiple edges and loops.

Often it is clear from the context that the graph is simple. In such cases, a simple graph is called a *graph*. In the remainder of thesis we will only be concerned about simple graphs.

2.1.3 Directed and Undirected Graphs

In a *directed graph*, the edges do have a direction but in an *undirected graph*, the edges are undirected. Strictly speaking, each edge in a directed graph should be

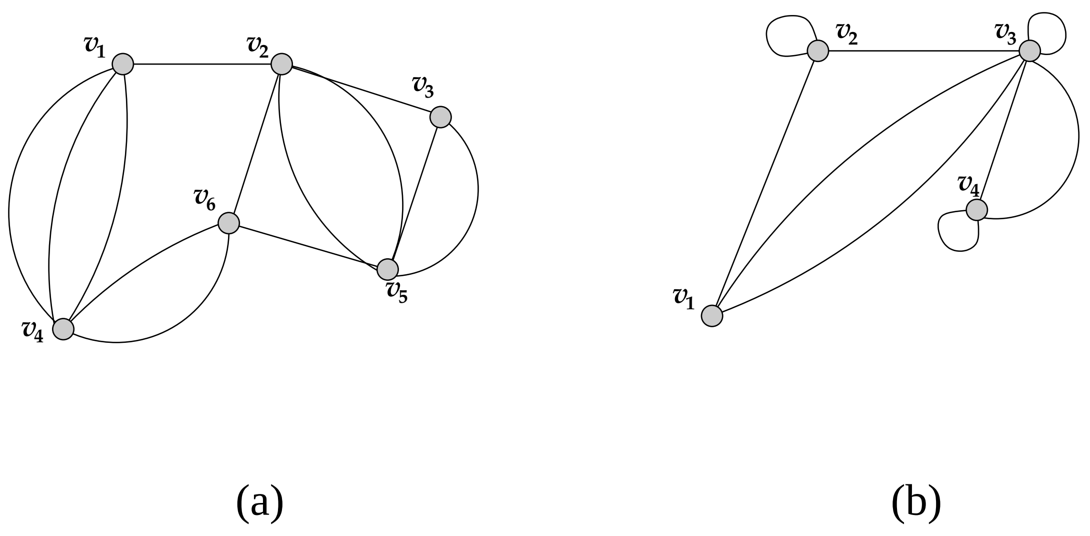


Figure 2.2: Multigraphs.

represented by a 2-tuple while for an undirected graph it should be represented by a 2-member subset of the vertex set. In Figure 2.3(a) and (b), we show an undirected and a directed graphs respectively. In this thesis, we will mean an undirected graph when we say "a graph" unless mentioned otherwise.

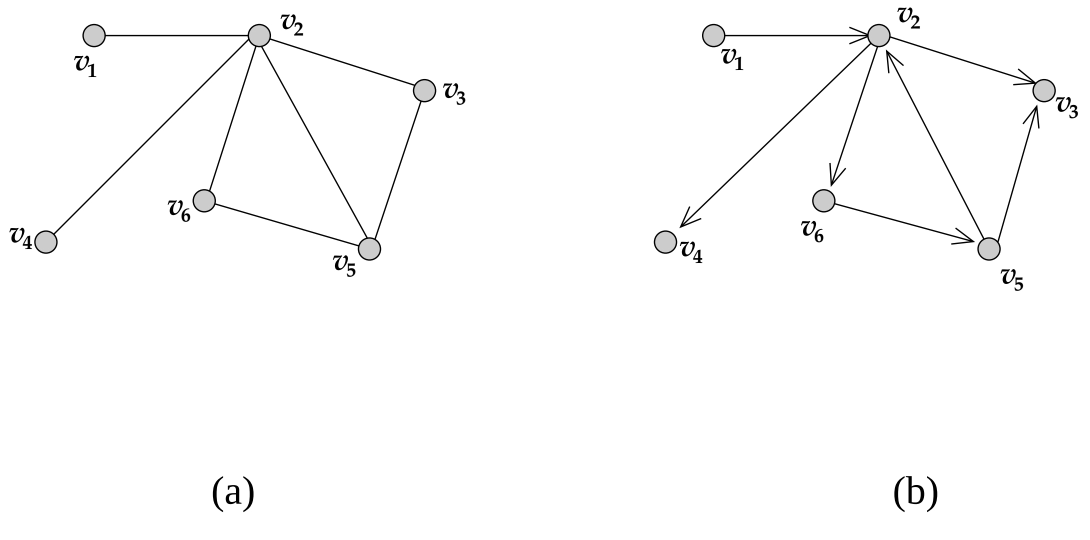


Figure 2.3: Undirected and directed graphs.

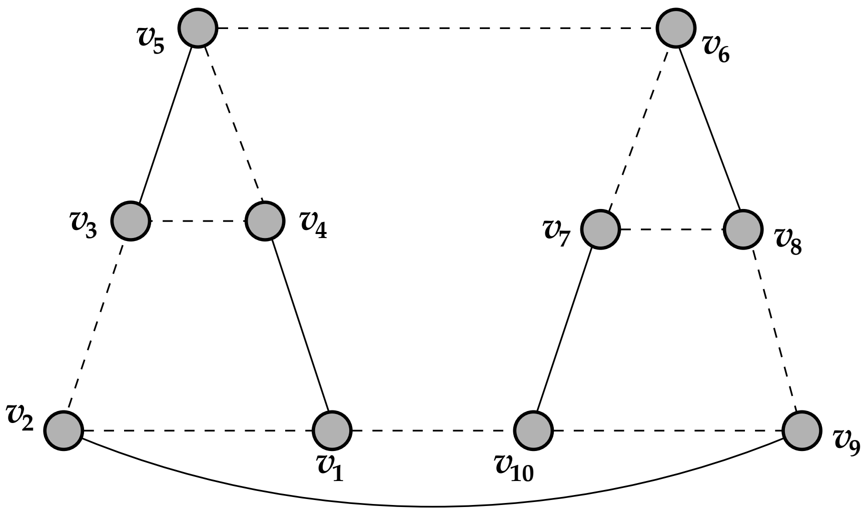


Figure 2.4: External Hamiltonian cycle of a 3-connected cubic planar graph G.

2.1.4 Paths and Cycles

A *walk*, $w = v_0, e_1, v_1, \ldots, v_{l-1}, e_l, v_l$, in a graph G is an alternating sequence of vertices and edges of G, beginning and ending with a vertex, in which each edge is incident to the two vertices immediately preceding and following it. The vertices v_0 and v_l are said to be the end-vertices of the walk w.

If the vertices $v_0, v_1, \ldots, v_l$ are distinct (except possibly v_0 and v_l), then the walk is called a *path* and usually denoted either by the sequence of vertices $v_0, v_1, \ldots, v_l$ or by the sequence of edges $e_1, e_2, \ldots, e_l$. The length of the path is l, one less than the number of vertices on the path. For any two vertices u and v of G, a u, v-path in G is a path whose end-vertices are u and v.

A walk or path w is *closed* if the end-vertices of w are the same. A closed path containing at least one edge is called a *cycle*. A cycle is called *Hamiltonian* if it consists of all the vertices of a graph G. An *external Hamiltonian* cycle C of a graph G is a hamiltonian cycle such that C has at least one edge on the outer face of G. This property is called external hamiltonicity [KW02]. An external Hamiltonian cycle $C = \langle\ v_1,\ v_2,\ \ldots,\ v_{10},\ v_1\ \rangle$ is shown in Fig. 2.4 by dotted lines.

2.1.5 Connectivity

A graph G is *connected* if for any two distinct vertices u and v of G, there is a path between u and v. A graph which is not connected is called a *disconnected* *graph.* A *(connected) component* of a graph is a maximal connected subgraph. The graph in Figure 2.5(a) is a connected graph since there is a path between every pair of distinct vertices of the graph. On the other hand, the graph in Figure 2.5(b) is a disconnected graph since there is no path between, say, v_1 and v_5. The graph in Figure 2.5(b) has two connected components as indicated by the dotted lines. Note that every connected graph has only one component; the graph itself.

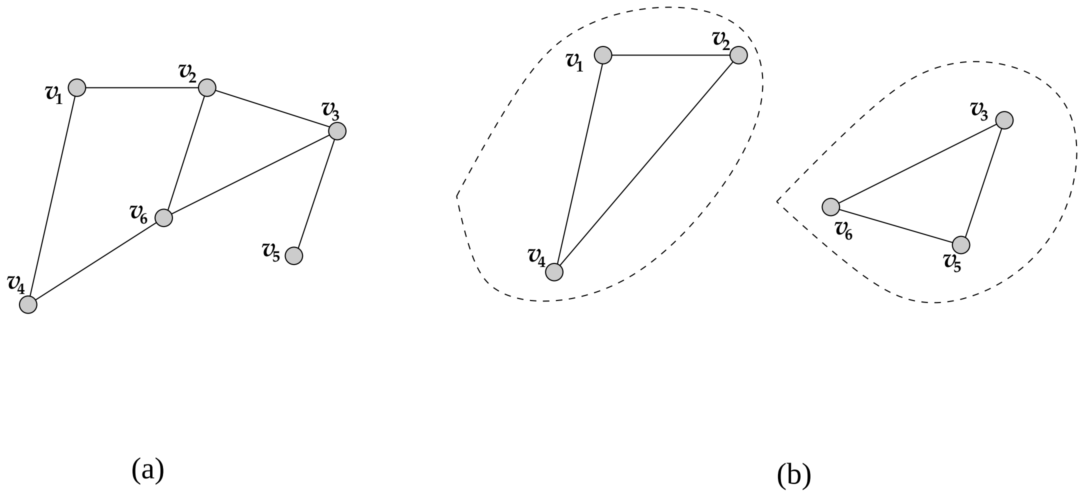


Figure 2.5: (a) A connected graph, (b) A disconnected graph with two connected components.

The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single-vertex graph K_1. We say that G is *k-connected* if $\kappa(G) \geq k$. 2-connected and 3- connected graphs are also called biconnected and triconnected graphs, respectively. A *block* is a maximal biconnected subgraph of G. We call a set of vertices in a connected graph G a *separator* or a *vertex cut* if the removal of the vertices in the set results in a disconnected or single-vertex graph. If a vertex-cut contains exactly one vertex then we call the vertex a *cut vertex*.

2.2 Planar Graphs

In this section we give some definitions related to planar graphs used in the remainder of the thesis.

A *planar drawing* of a graph G is a two-dimensional drawing of G in which no pair of edges intersect with each other except at their common end-vertex. A planar graph is a graph that has at least one planar drawing. A *planar embedding* of a graph G is a data structure that defines a clockwise (or counter clockwise) ordering of the neighbors of each vertex of G that corresponds to a planar drawing of the graph. Note that a planar graph may have an exponential number of embedding. Figure 2.6 shows two planar embeddings of the same planar graph.

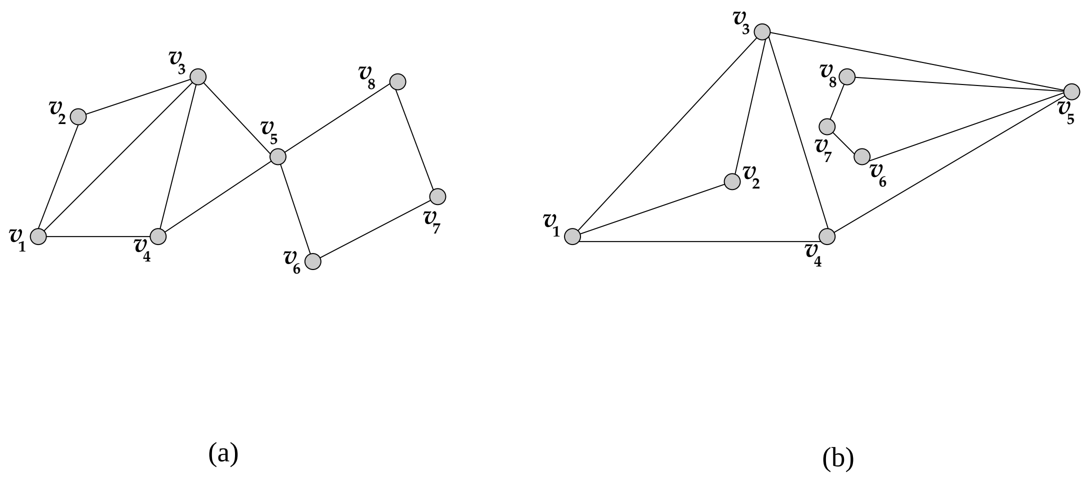


Figure 2.6: Two planar embeddings of the same planar graph.

A *plane graph* is a planar graph with a fixed planar embedding. A plane graph divides the plane into connected regions called *faces*. A finite plane graph G has one unbounded face and it is called the *outer face* of G. G is said to be *triangulated* if each face of G is a triangle, i.e contains exactly three vertices. A triangulated plane graph is 4-connected if and only if it has no separating triangle [Woo82]. Furthermore, all 4-connected planar graphs are Hamiltonian [Tho83].

Let G be a plane graph with a cycle C. Now if $E(G), E(C)$ represent the set of edges of plane graph G and the cycle C respectively, then we define $G - C$ as $G - C = E(G) - E(C)$. The edges of $G - C$ can be partitioned into two classes as inner edges and outer edges of C. The edges of $G - C$ which are stay inside with respect to the cycle C are called the *inner edges* of C and the edges of $G - C$ which are stay outside with respect to the cycle C are called the *outer edges* of C. If a cycle $C = \langle\ v_1,\ v_2,\ \ldots\ ,\ v_6,\ v_1\ \rangle$ of a plane graph G, then

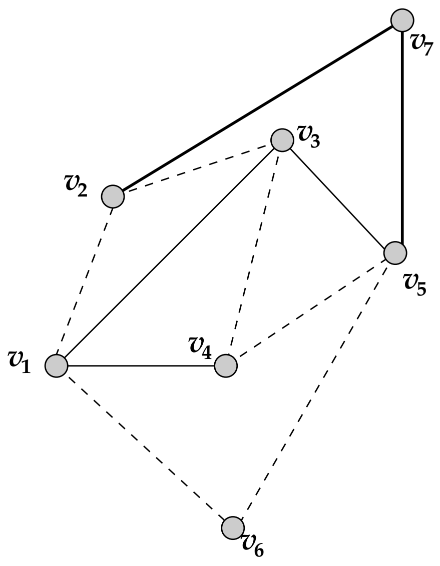


Figure 2.7: Inner edges and outer edges of cycle C.

the edges $(v_1, v_3), (v_1, v_4)$ are inner edges of C and the edges $(v_2, v_7), (v_5, v_7)$ are outer edges of C as shown in Fig. 2.7.

2.3 Drawing Conventions

In this section we introduce some conventional drawing styles, which are found suitable in different application domain. The different drawing styles vary owing to different representations of vertices and edges. Depending on the purpose and objective, the vertices are typically represented with points or boxes and edges are represented with simple Jordan curves [NR04]. A few of the most important drawing styles are introduced below.

2.3.1 Planar Drawing

A drawing Γ of a graph G is planar if no two edges intersect with each other except at their common end-vertices. In Figure 2.8(a) and (b), we show a planar and a non-planar drawing of the same graph.

Planar drawings of graphs are more convenient than non-planar drawings because, as shown empirically in [Pur97], the presence of edge-crossings in a drawing of a graph make it more difficult for a person to understand the information being modeled. Unfortunately, not all graphs have a planar drawing.

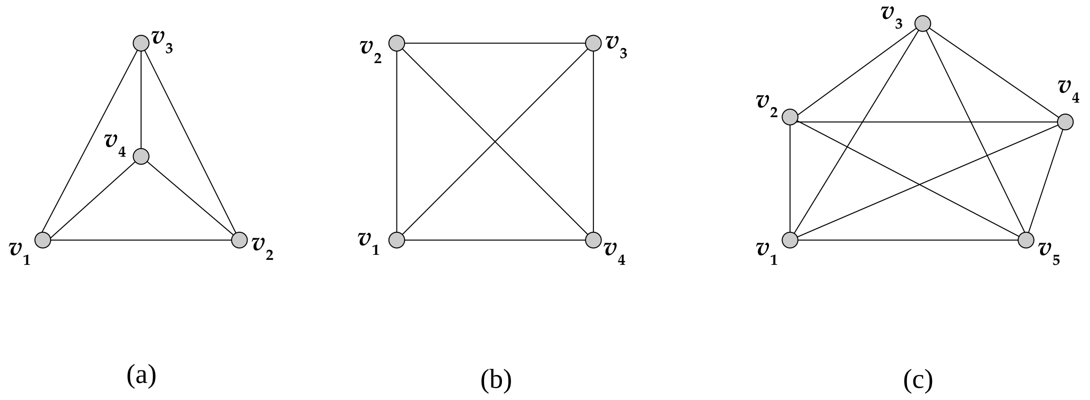


Figure 2.8: (a) A planar drawing, (b) A non-planar drawing of the graph drawn in (a), (c) A graph which does not have a planar drawing.

Figure 2.8(c) is an example of one such graph.

2.3.2 Straight-line Drawing

A *straight-line drawing* of a graph G is a drawing of G in which each edge is drawn as a straight line segment, as illustrated in Figure 2.9. Wagner [Wag36],

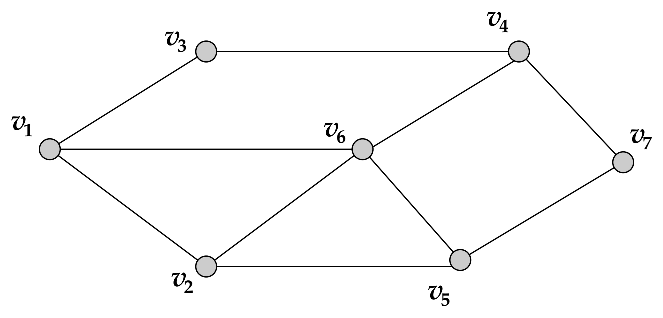


Figure 2.9: A straight line drawing.

Fary [Far48] and Stein [Ste51] independently proved that every planar graph has a straight line drawing.

2.3.3 Orthogonal Drawing

An *orthogonal drawing* of a planar graph G is a drawing of G, in which each vertex of G is mapped to a point, each edge is drawn as a sequence of alternate horizontal and vertical line segments, and any two edges do not cross except at their common end, as illustrated in Fig. 2.10.

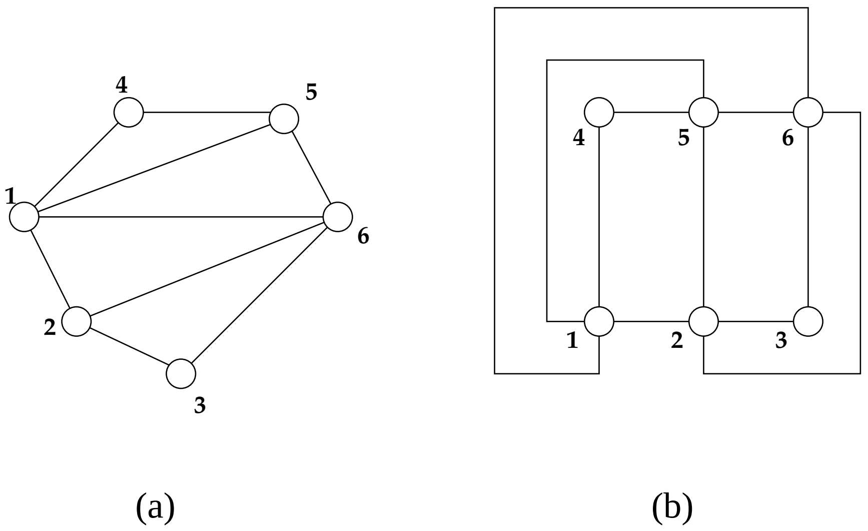


Figure 2.10: (a) A planar graph G, (b) orthogonal drawing of G with ten bends.

Clearly the maximum degree Δ of G is at most four if G has an orthogonal drawing. Conversely, every plane graph with $\Delta \leq 4$ has an orthogonal drawing, but may need bends, that is, points where an edge changes its direction in a drawing. Bend minimization in orthogonal drawing has a rich collection of literature [GL99, RNN99, GT01, RN02, RNN03].

2.4 Topological Book Embedding

In this section we will at first define topological book embedding and the related terms of topological book embedding. Upward topological book embedding is defined in the last part of the section.

A *topological book embedding* γ of a planar digraph G is a planar embedding of G in the form of a 2-page book where the vertices are contained along a straight-line, each edge is drawn within the two half-planes induced by the straight-line and the edges are allowed to cross the spine [Miy06]. The straight-line on which the vertices of G are placed is called the *spine* of γ and the two half-planes induced by the spine is called the *pages* of γ. Figure 2.11(b) shows a topological book embedding of the graph G of Fig. 2.11(a) where the dotted line represents the spine and the two half planes on either side of the spine are the pages of the book. The edge (b, e) in Fig. 2.11(b) crosses the spine.

A *book* is defined as a collection of half planes called pages which join to-

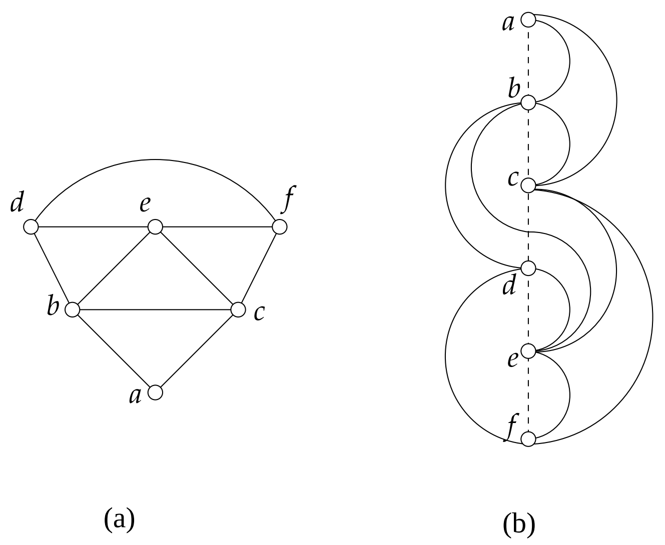


Figure 2.11: (a) A planar graph G, (b) a topological book embedding of G with a spine crossing by the edge (b, e).

gether at a straight-line called the *spine*. For the rest of the thesis, we shall assume that in a topological book embedding of a planar graph, the spine is aligned along the y-axis. In such a drawing the half plane on the left hand side of the spine is called the *left page* and that on the right hand side is called the *right page*.

An *upward topological book embedding* of a planar graph G is a topological book embedding γ where each edge of γ is drawn as upward. Thus an upward book embedding of G takes the form of a 2-page book where the vertices are contained along a straight-line, each edge is drawn in the monotonically increasing direction within the two half-planes induced by the straight-line and the edges are allowed to cross the spine. Figure 2.12(b) shows a upward topological book embedding of the graph G of Fig. 2.12(a) where the dotted line represents the spine. The edge (2, 5) in Fig. 2.12(b) crosses the spine. Let Φ is an ordering of the vertices of G. An upward topological book embedding of G with the ordering Φ is an upward topological book embedding of G where the ordering of the vertices along the spine follows Φ.

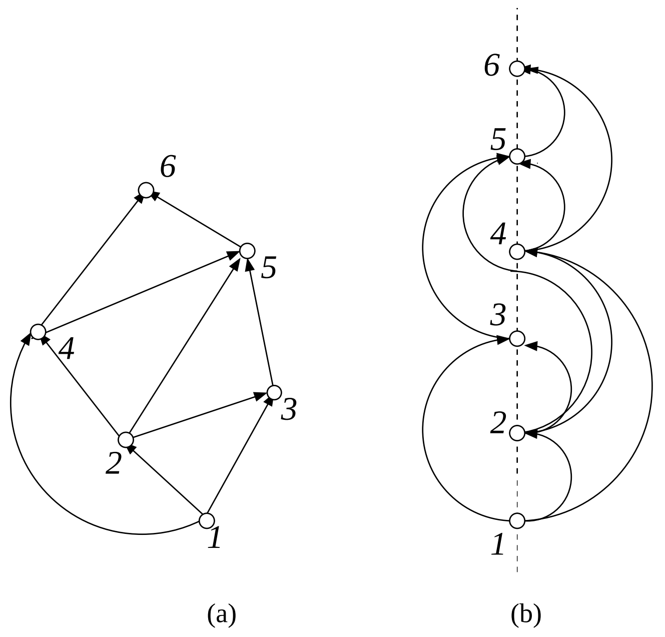


Figure 2.12: (a) A planar graph G, (b) a topological book embedding of G with a spine crossing by the edge $(2, 5)$.

2.5 Complexity of Algorithms

In this section we briefly introduce some terminologies related to *complexity* of algorithms. For interested readers, we refer the book of Garey and Johnson [GJ79].

The most widely accepted complexity measure for an algorithm is the *running time*, which is expressed by the number of operations it performs before producing the final answer. The number of operations required by an algorithm is not the same for all problem instances. Thus, we consider all inputs of a given size together, and we define the *complexity of the algorithm for that input size* to be the worst case behavior of the algorithm on any of these inputs. Then the running time is a function of size n of the input.

2.5.1 The Notation $O(n)$

In analyzing the complexity of an algorithm, we are often interested only in the "asymptotic behavior", that is, the behavior of the algorithm when applied to very large inputs. To deal with such a property of functions we shall use the following notations for asymptotic running time. Let $f(n)$ and $g(n)$ are the functions from the positive integers to the positive reals, then we write $f(n) = O(g(n))$ if there exists positive constants c_1 and c_2 such that $f(n) \leq c_1 g(n) + c_2$ for all n. Thus the running time of an algorithm may be bounded from above by phrasing like "takes time $O(n^2)$".

2.5.2 Polynomial Algorithms

An algorithm is said to be *polynomially bounded* (or simply *polynomial*) if its complexity is bounded by a polynomial of the size of a problem instance. Examples of such complexities are $O(n)$, $O(nlogn)$, $O(n^{100})$, etc. The remaining algorithms are usually referred as *exponential* or *non-polynomial*. Examples of such complexity are $O(2^n)$, $O(n!)$, etc. When the running time of an algorithm is bounded by $O(n)$, we call it a *linear-time* algorithm or simply a *linear* algorithm.

2.5.3 NP-complete Problems

There are a number of interesting computational problems for which it has not been proved whether there is a polynomial time algorithm or not. Most of them are "NP-complete", which we will briefly explain in this section.

The state of algorithms consists of the current values of all the variables and the location of the current instruction to be executed. A *deterministic algorithm* is one for which each state, upon execution of the instruction, uniquely determines at most one of the following state (next state). All computers, which exist now, run deterministically. A problem Q is in the *class P* if there exists a deterministic polynomial-time algorithm which solves Q. In contrast, a *non-deterministic algorithm* is one for which a state may determine many next states simultaneously. We may regard a non-deterministic algorithm as having the capability of branching off into many copies of itself, one for the each next state. Thus, while a deterministic algorithm must explore a set of alternatives one at a time, a non-deterministic algorithm examines all alternatives at the same time. A problem Q is in the *class NP* if there exists a non-deterministic polynomial-time algorithm which solves Q. Clearly, $P \subseteq NP$.

Among the problems in *NP* are those that are hardest in the sense that if one can be solved in polynomial-time then so can every problem in *NP*. These are called *NP-complete* problems. The class of *NP*-complete problems has the following interesting properties.

(a) No *NP*-complete problem can be solved by any known polynomial algorithm.

(b) If there is a polynomial algorithm for any *NP*-complete problem, then there are polynomial algorithms for all *NP*-complete problems.

Sometimes we may be able to show that, if problem Q is solvable in polynomial time, all problems in *NP* are so, but we are unable to argue that $Q \in NP$. So Q does not qualify to be called *NP*-complete. Yet, undoubtedly Q is as hard as any problem in *NP*. Such a problem Q is called *NP-hard.*

Chapter 3

Upward Point-Set Embedding

3.1 Introduction

An *upward point-set embedding* of an upward planar digraph is an upward planar drawing of the graph where the vertices are placed on a predefined set of distinct points in the plane. Mapping is defined from the vertices of upward planar digraph to the points in the plane.

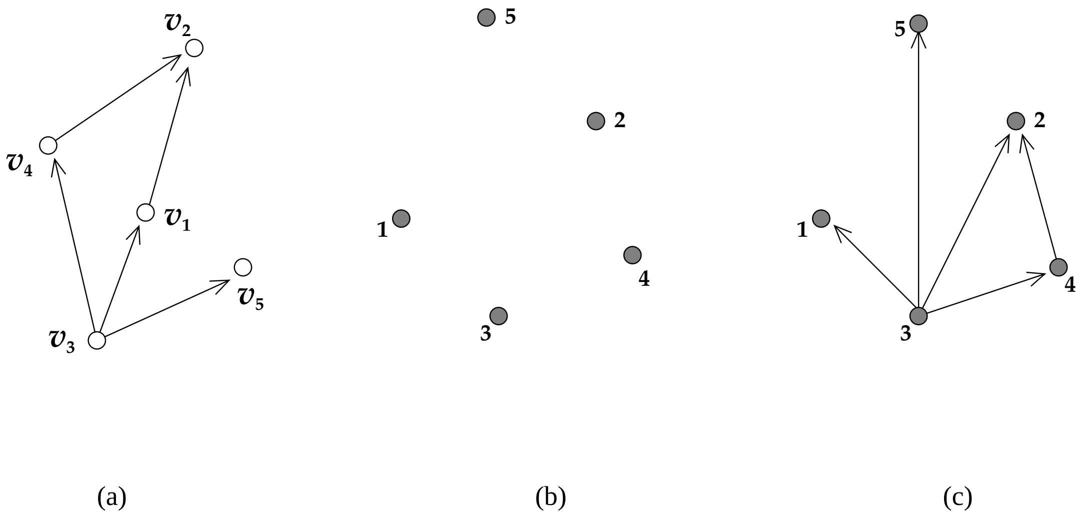


Figure 3.1: (a) An upward planar digraph G, (b) a point-set S, (c) an upward point-set embedding Λ of G on S

.

Let G be an upward planar digraph with n vertices, S be a set of n distinct points in the plane and Φ be a mapping from the vertices of G to the points of S, then an *upward point-set embedding with mapping* is an upward planar drawing Λ of G where each vertex of G is placed on the point of S according to Φ. Figure 3.1(c) shows the upward point-set embedding Λ of upward planar digraph G on S of Fig. 3.1(a),(b) where Φ is defined from vertex v_i of G to the point i of S for $1 \leq i \leq 5$. It is easy to understand that not every mapping Φ has an upward point-set embedding of G on S.

The rest of the chapter is organized as follows. Section 3.2 describes the algorithm for finding an upward topological book embedding of an upward planar digraph with a given ordering. The algorithm for finding upward point-set embedding from upward topological book embedding is illustrated in Section 3.3. Finally Section 3.4 concludes the paper. The results described in this chapter has been presented in [CAR09].

3.2 Upward Topological Book Embedding

It is trivial to see that G admits an upward topological book embedding with the ordering Φ if and only if Φ induces a topological numbering of $V(G)$. We thus assume that Φ induces a topological numbering of $V(G)$. We first rename the vertices so that the vertices are labeled as v_1, v_2, ... , v_n in the order of Φ. We now have the following lemma.

Lemma 3.2.1 *Let G be an upward planar digraph of n vertices with a directed hamiltonian path P and let Φ be a topological ordering of the vertices of G. Then one can find an upward topological book embedding of G in linear time with no spine crossing.*

Proof. Let us rename the vertices so that the vertices are labeled as v_1, v_2, ... , v_n in the order of Φ. Then P contains only the edges (v_i, v_{i+1}) for $1 \leq i < n$ since Φ is a topological ordering of G. Let G' be a plane embedding of G. We now obtain a topological book embedding γ of G with no spine crossing as follows. We first fix the position of the vertices on a vertical straight-line L (the spine of γ) such that v_{i+1} is above v_i for $1 \leq i < n$. We first draw each edge of P by a semi-circle between its two end-vertices on either the left page or the right page of γ. We draw the rests of the edges as follows. If an edge is to the left of P in G', then we draw it on the left page of γ by a semicircle between its end-vertices. And if an edge is to the right of P in G', then we draw it on the right page of γ by a semicircle between its end-vertices. Clearly γ keeps the embedding G' unaltered and hence γ is a planar drawing of G. Furthermore it is obvious from the algorithm that all the vertices of G are on the straight-line L and each edge is either on the left page or on the right page of γ, creating no spine crossing. Thus γ is an upward topological book embedding of G with no spine crossing. It is also trivial to implement the algorithm in linear time.

$\mathcal{Q.E.D.}$

Let G be an upward planar digraph and let Φ be a topological ordering of the vertices of G. If G contains a hamiltonian path, then the proof of Lemma 3.2.1 gives a linear-time algorithm to obtain an upward topological book embedding of G with the ordering according to Φ. We call this algorithm **Draw_Ham** for the rest of this section. We now use this algorithm to obtain a topological book embedding of any upward planar digraph G with the ordering Φ. We have the following theorem.

Theorem 3.2.2 *Let G be an upward planar digraph with n vertices and let Φ be a topological ordering of $V(G)$. Then one can find an upward topological book embedding of G with the ordering Φ in $O(n^2)$ time where each edge of G crosses the spine at most $n-4$ times. Furthermore, if s is the number of edges that crosses the spine, then the total number of spine crossings is at most $2(n-4)+3(n-5)+\ldots+k(n-2-k)+p(n-3-k)$ where k and p are integers, $s=\left(\frac{k(k+1)}{2}-1+p\right)$ and $p<k$.*

In the rest of this section, we give a constructive proof of Theorem 3.2.2.

We rename the vertices of G so that the vertices are labeled as $v_1, v_2, \ldots, v_n$ in the order of Φ. Let λ be an upward planar stright-line drawing of G as illustrated in Fig. 3.2(a). For every vertex v of G, we denote by $p(v)$ the point on which v is placed in λ. We may assume that for any three vertices u, v and w, $p(u)$, $p(v)$ and $p(w)$ are not collinear. We now construct an upward planar digraph G' with a hamiltonian path from G as follows. If G contains the edge (v_i, v_{i+1}) for $1 \leq i < n$, then G itself has a hamiltonian path. We thus assume that there is an index i ($1 \leq i < n$) such that G does not contain the edge (v_i, v_{i+1}). If the straight-line segment between $p(v_i)$ and $p(v_{i+1})$ does not intersect any existing edge in λ, then we add the edge (v_i, v_{i+1}) to G. Otherwise,

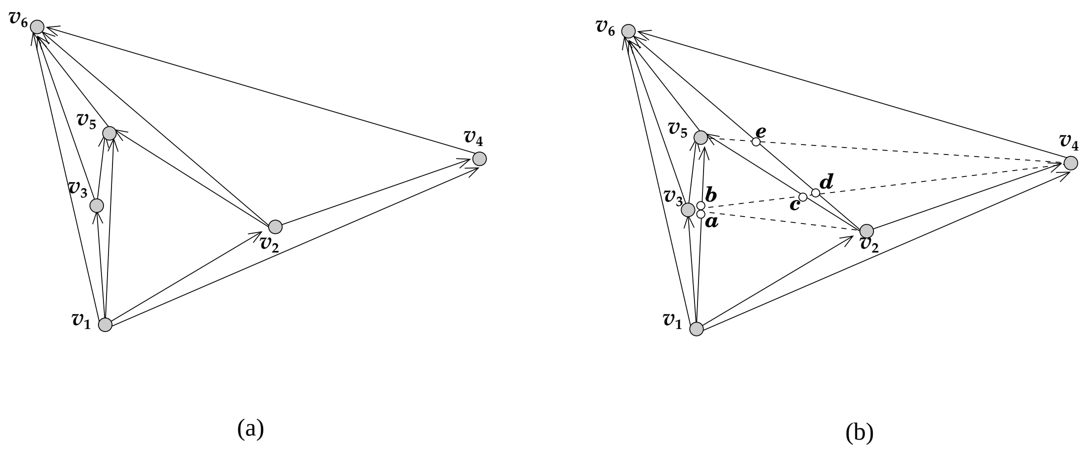


Figure 3.2: (a) Upward planar straight line drawing λ of G, (b) graph G' of λ.

suppose that the straight-line segment between $p(v_i)$ and $p(v_{i+1})$ intersects the edges $e_1 = (u_1, w_1)$, $e_2 = (u_2, w_2)$, …, $e_f = (u_f, w_f)$ in this order from $p(v_i)$ to $p(v_{i+1})$. Then we subdivide all these edges, i.e. we delete the edges e_1, e_2, …, e_f from G; add the dummy vertices x_1, x_2, …, x_f to G; and add the edges (u_1, x_1), (x_1, w_1), (u_2, x_2), (x_2, w_2), … , (u_f, x_f), (x_f, w_f). Finally we add the edges (v_i, x_1), (x_1, x_2), … , (x_{f-1}, x_f) and (x_f, v_{i+1}). For each index i $(1 \leq i < n)$ such that G does not contain the edge (v_i, v_{i+1}), we do the same operations as above. Let G' be the resulting graph obtained by the operations above, and we call G' the *augmented hamiltonian graph* of G. If p is the total number of dummy vertices, then G' has $n + p$ vertices and contains a hamiltonian path as illustrated in Fig. 3.2(b). Then Algorithm **Draw_Ham** gives an upward topological book embedding γ' of G' with no spine crossing as illustrated in Fig. 3.3(a). We now obtain an upward topological book embedding γ of G from γ' by deleting the dummy vertices as well as the edges incident to them and drawing the edges of G that were deleted in G' through the dummy vertices as illustrated in Fig. 3.3(b). Note that the dummy vertices of G' represents the spine crossings in γ. For the rest of this article, we call this algorithm **Draw_General**. We now have the following three lemmas.

Lemma 3.2.3 *Let G be an upward planar digraph and let $\Phi = v_1, v_2, \ldots, v_n$ be a topological ordering of G. Let G' be the augmented hamiltonian graph of G. Then the number of vertices in G' is bounded by $O(n^2)$.*

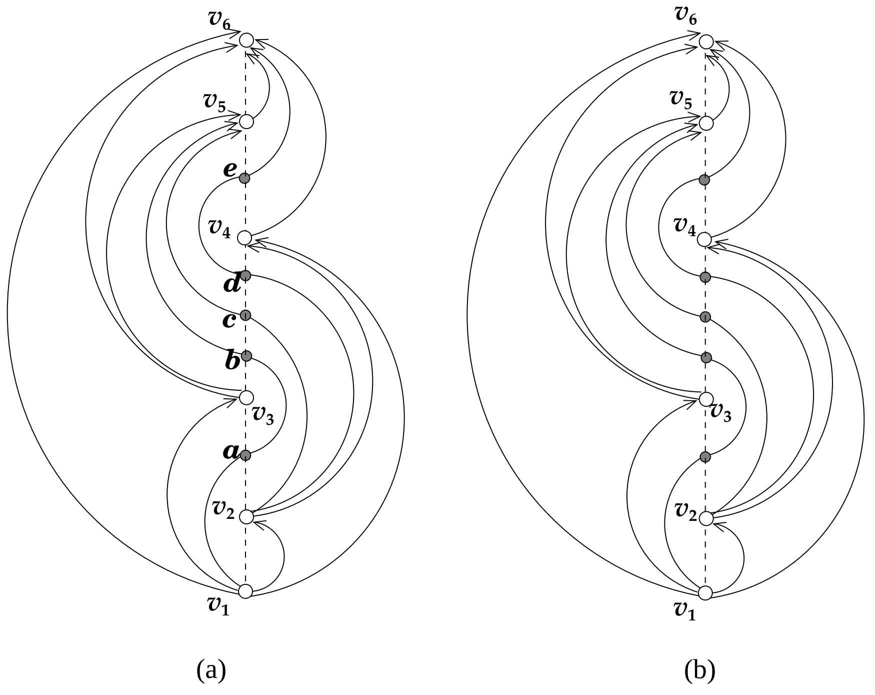


Figure 3.3: (a) Upward topological book embedding γ' of G', (b) upward topological book embedding γ.

Proof. A hamiltonian path of length $n-1$ is embedded in G'. An edge of hamiltonian path can cross at most $n-2$ existing edges of G (see Fig. 3.2(b)). Since each crossing represents a dummy vertex, there are at most $(n-1)(n-2)$ dummy vertices in G'. Hence the number of vertices in G' is bounded by $O(n^2)$. $\mathcal{Q.E.D.}$

Lemma 3.2.4 *Let G be an upward planar digraph and let $\Phi = v_1, v_2, \ldots, v_n$ be a topological ordering of G. Let γ be the upward topological book embedding of G with the ordering Φ obtained by Algorithm **Draw_General**. Then the edge (v_i, v_j) can cross the spine at most $j-i-2$ times.*

Proof. Let λ be any upward planar straight-line drawing of G and let $p(v)$ denote the point on which a vertex v of G is placed in γ. Let L denote the polyline containing the straight-line segments between $p(v_i)$ and $p(v_{i+1})$ for $1 \leq i < n$ as illustrated in Fig. 3.4. Since the drawing is upward, the edge (v_i, v_j) can cross at most $j-i$ line-segments of L between $p(v_i)$ and $p(v_j)$ in γ. However since the edge is drawn as a straight-line segment, it does not cross the two line-segments between $(p(v_i), p(v_{i+1}))$ and between $(p(v_{j-1}), p(v_j))$. Thus

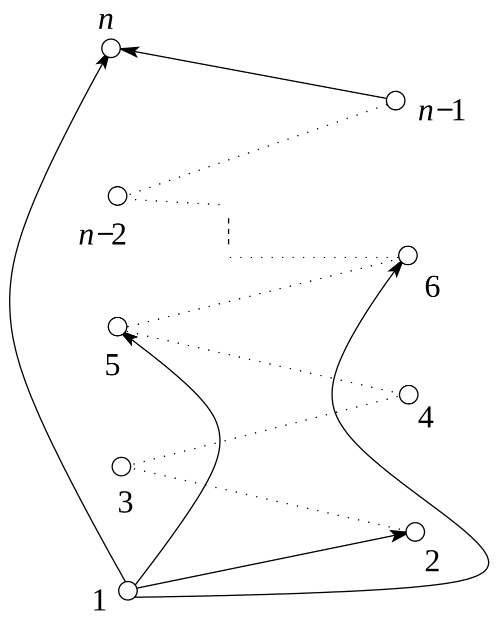


Figure 3.4: Illustration of the proof of Lemma 3.2.4

from the algortihm it is obvious that the edge (v_i, v_j) can cross the spine at most $j - i - 2$ times. $\mathcal{Q.E.D.}$

Lemma 3.2.5 *Let G be an upward planar digraph and let Φ be a topological ordering of G. Let γ be the upward topological book embedding of G with the ordering Φ obtained by Algorithm **Draw_General**. Then there are at most $\left(\frac{k(k+1)}{2} - 1\right)$ edges each of which can cross the spine at least $n - 2 - k$ times for $k \geq 1$.*

Proof. Let (v_i, v_j) be an edge that crosses the spine at least $n - 2 - k$ times. Then by Lemma 3.2.4, $j - i \geq n - 2 - k + 2 = n - k$. Thus there are at most k edges $((v_1, v_{n-k+1}), (v_1, v_{n-k+2}), \ldots, (v_1, v_n))$ from v_1 that crosses the spine at least $n - 2 - k$ times in γ. Similarly there are at most $k - 1$ edges from v_2 that crosses the spine at least $n - 2 - k$ times in γ and so on. Thus the number of edges that crosses the spine at least $n - 2 - k$ times in γ is at most $k + (k - 1) + \ldots + 1 = \frac{k(k+1)}{2}$. However if the edge (v_1, v_n) is contained in G, we may assume that it is on the outer face in γ since otherwise we can redraw it keeping the edge (v_1, v_n) on the outerface. Thus the edge (v_1, v_n) does not cross the spine and the result follows. $\mathcal{Q.E.D.}$

We are now ready to prove Theorem 3.2.2

Proof of Theorem 3.2.2. Let γ be the upward topological book embedding of G with the ordering Φ obatined by Algorithm **Draw_General**. By Lemma 3.2.5, there is no edge that crosses the spine at least $n-3$ times in γ. Thus an edge can cross the spine at most $n-4$ times in γ. Furthermore, if the total number of edges that crosses the spine in γ is at most s, then by Lemma 3.2.5, the total number of spine crossings is at most $2(n-4)+3(n-5)+\ldots+k(n-2-k)+p(n-3-k)$ where k and p are integers, $s=\frac{k(k+1)}{2}-1+p$ and $p<k$. Finally since the number of vertices of G' is bounded by $O(n^2)$ according to Lemma 3.2.3, the time complexity of the algorithm is also $O(n^2)$. $\mathcal{Q.E.D.}$

Let G be an upward planar digraph and let Φ be a topological ordering of the vertices of G. Let γ be the upward topological book embedding of G with the ordering Φ obtained by Algorithm **Draw_General**. Since (v_1,v_2), (v_{n-1},v_n) and (v_1,v_n) does not cross the spine in γ, at most $3n-9$ edges crosses the spine in γ. Theorem 3.2.2 then gives an upper bound on the total number of spine crossings in γ. However in reality, the number of edges that crosses the spine is much less than the trivial bound of $3n-9$. We finish this section with the following conjecture the proof of which can give a much better upper bound on the total number of spine crossings.

Conjecture 3.2.1 *Let G be an upward planar digraph and let Φ be a topological ordering of G. Let γ be the upward topological book embedding of G with the ordering Φ obtained by Algorithm **Draw_General**. Then there are at most $2n-6$ edges of G that crosses the spine in γ.*

3.3 Upward Point-Set Embedding

In this section we address the problem of finding an upward point-set embedding of an upward planar digraph G of n vertices on a set of n distinct points in the plane with a mapping Φ from the vertices of G to the points in S. We can find the upward point-set embedding of an upward planar digraph G using Algorithm **Draw_General**. The following Theorem is the main result of this section.

Theorem 3.3.1 *Let G be an upward planar digraph with n vertices, S be a set of n distinct points in the plane and Φ be a mapping from the vertices of G to the* *points in S. Then G admits an upward point-set embedding with the mapping Φ if and only if there exists a directed line l' such that Φ induces a topological ordering of G on l'. Furthermore, such an upward point-set embedding of G on S can be computed in $O(n^2)$ time with at most $n-3$ bends per edge.*

In the rest of the section, we prove the Theorem 3.3.1.

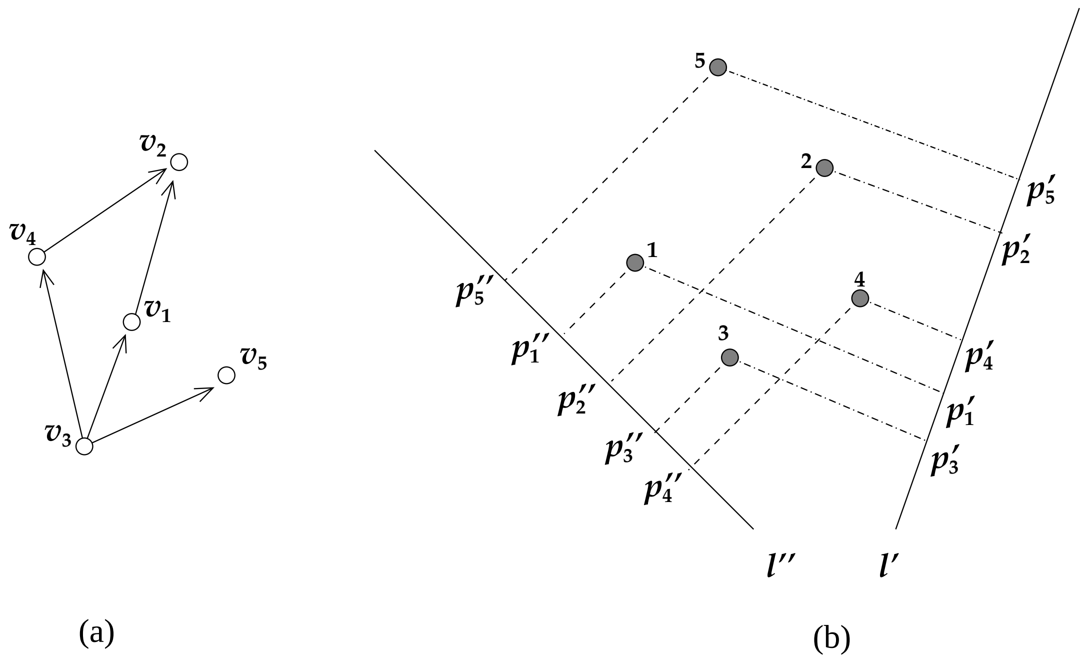


Figure 3.5: (a) An upward planar digraph G, (b) projection of S on line l' and l''

.

Let l' be a directed line and $S_{l'} = \{p'_1, p'_2, \ldots, p'_n\}$ be the collinear set of points obtained by orthogonaly projecting S onto l' and we assume that no two projected points in l' coincide. And let $\Phi_{l'}$ be the mapping from G to $S_{l'}$ that associates each vertex v of G with the projection of $\Phi(v)$ on l'. Now if mapping $\Phi_{l'}$ induces a topological numbering of G, then Φ induces a topological numbering of G [GLW09]. In Figure 3.5(b), l' is a line in which the mapping Φ induces a topological numbering of G whereas the line l'' does not induce. Then we can find the upward point-set embedding using the Algorithm **Draw_General** in the following way.

If the edge of γ does not cross the spine, it can be drawn using one bend in Γ. Otherwise each edge can be drawn using one more bend than the number of spine crossings. So one edge uses at most $n - 4 + 1 = n - 3$ bends. Clearly the algorithm runs in quadratic time.

We can find an upper bound on the total number of bends in an upward point-set embedding of G from the fact that for each edge, the number of bends in the point-set embedding is at most one more than the number of spine crossings for that edge in γ and that there are at most $3n - 6$ edges in G. Furthermore, Giordano *et. al.* also gave an $O(n^3)$-time testing algorithm to check whether an upward planar digraph G of n vertices admits an upward point-set embedding on a set S of n distinct points with a given mapping from the vertices of G to the points in S.

3.4 Conclusion

In this chapter we first gave an algorithm to find an upward topological book embedding of upward planar digraphs with a given mapping. After that we extended our result to find an upward point-set embedding of an upward planar digraphs with mapping and also proved the upper bound on number of bends in the drawing. Recently Mchedlidze and Symvonis have developed an algorithm for *"ρ-constrained upward topological book embedding"* of an embedded planar *st*-digraphs which can also be used to find the same bound on the number of bends per edge for upward point-set embedding of directed planar graphs [MS10].

Chapter 4

Orthogonal Point-set Embedding

4.1 Introduction

An *orthogonal point-set embedding* of a planar graph G on a set S of points in Euclidean plane is a planar drawing Γ of G where each vertex of G is placed on a point of S and each edge is drawn as a sequence of alternate horizontal and vertical line segments. *Bend* is a point at which an edge changes its direction in Γ. If a graph G corresponds a VLSI circuit, then one may be interested in an orthogonal drawing such that the number of bends is as small as possible, because bends increase the manufacturing cost in a VLSI chip. We provide two algorithms of orthogonal point-set embeddings for two sub-classes of planar graphs in this chapter. Our first algorithm is for 3-connected cubic planar graphs with hamiltonian cycle and later one is for 4-connected planar graphs. Both drawing algorithms have practical applications in circuit schematics on pre-fabricated printed circuit boards (PCBs), where position of components on the PCB is prescribed and standard cell technology employed during the VLSI layout design process. We find the upper bound on number of bends for both the drawings. For reducing the number of bends, we have imposed another practical constraint for 3-connected cubic planar graphs that the external hamiltonian cycle to be embedded consecutively on the point-set S. This problem may arise in the floorplanning of VLSI design where a critical cycle should be placed with minimum wire length.

The rest of the chapter is organized as follows. Section 4.2 describes the algorithm for finding orthogonal point-set embeddings of 3-connected cubic planar graphs and the computation of upper bounds on number of bends in the drawing. The algorithm for 4-connected planar graphs and related results are described in Section 4.3. Finally Section 4.4 concludes the chapter.

4.2 3-Connected Cubic Planar Graphs

In this section, we give an algorithm for obtaining an orthogonal point-set embedding of a 3-connected cubic planar graph G with an external hamiltonian cycle C on a point-set S. The following Theorem is the main result of this section.

Theorem 4.2.1 *Let G be a 3-connected cubic planar graph of n vertices with a hamiltonian cycle $C = \langle v_1, v_2, \ldots, v_n, v_1 \rangle$ and let S be a set of n points in Euclidean plane. Then one can find an orthogonal point-set embedding Γ of G on S with at most $\left(\frac{5n}{2} + 2\right)$ bends such that the vertices of C are placed consecutively on the point-set S. Furthermore Γ can be obtained in linear time.*

In the rest of the section, we give a constructive proof of Theorem 4.2.1 which leads to our algorithm.

An outline of our algorithm is as follows. We first rotate the point-set S in such a way that no two points of S has the same x or y coordinates. We find a plane embedding G' of the planar graph G such that an edge of hamiltonian cycle C in the outer face of G'. We now find a topological book embedding γ of G as follows. We separate the inner edges and outer edges of hamiltonian cycle C in G'. The vertices which are incident to the inner edges are called *inner vertices* and incident to the outer edges are called *outer vertices*. We now draw the inner and outer edges in two different pages of γ for consistency. Our idea is to choose an appropriate page of γ in which the inner edges to be drawn by reducing number of bends. From the topological book embedding γ, one can find the orthogonal drawing Γ of G on point-set S by replacing the edges of $G' - C$ of γ in the left side of Γ if they are in the left page of γ and right side of Γ if they are in the right page of γ.

At first we will find a topological book embedding of G using the following lemma.

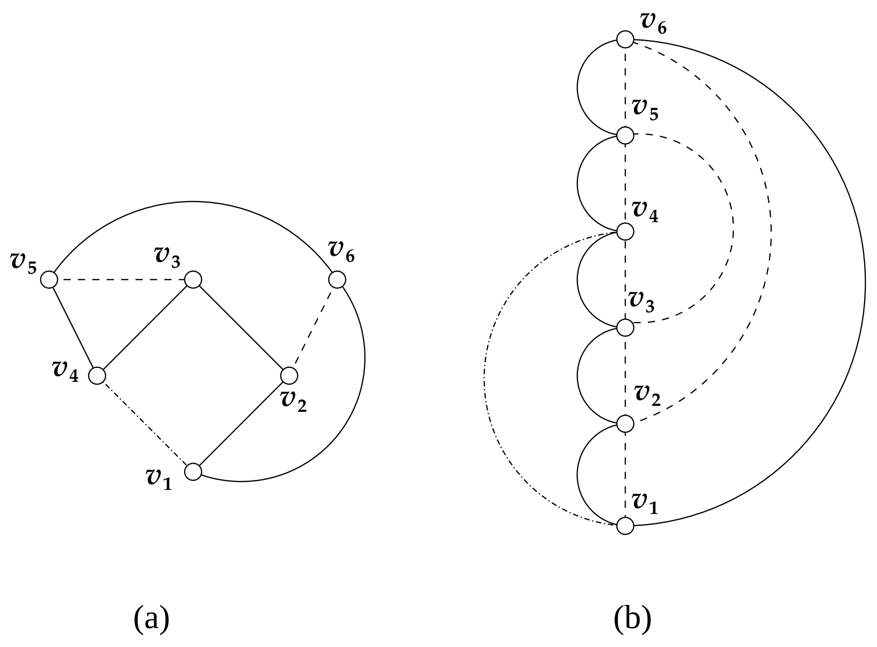


Figure 4.1: (a) 3-connected cubic planar graph G, (b) topological book embedding γ with no spine crossing.

Lemma 4.2.2 *Let G be a 3-connected cubic planar graph of n vertices with a hamiltonian cycle $C = \langle\, v_1,\, v_2,\, \ldots,\, v_n,\, v_1 \,\rangle$. Then one can find a topological book embedding γ of G in linear time with no spine crossing.*

Proof. Let G' be a plane embedding of the planar graph G such that an edge of hamiltonian cycle C in the outer face of G'. At first we place the vertices of G' on the spine of γ according to the order of the vertices of hamiltonian cycle C on the spine. Then we draw the edges of C in the left page of γ except the outer face edge of G'. We draw the inner edges of G' in the right page of γ and the outer edges of G' in the left page of γ. Clearly the edges will not create any spine crossing because the inner and outer edges are in two different sides with respect to the hamiltonian cycle C in G'. Finally we draw the last edge of C in the right page of γ, the procedure of drawing is illustrated in Fig. 4.1. Since G is a planar graph, it is trivial to implement the algorithm in linear time. $\mathcal{Q.E.D}$.

We call the edge (v_1, v_n) which connects the top most point and the bottom most point of γ as *long edge* for the rest of the chapter. From the topological book embedding, one can find the orthogonal drawing Γ on point-set S with smaller number of bends in the following way.

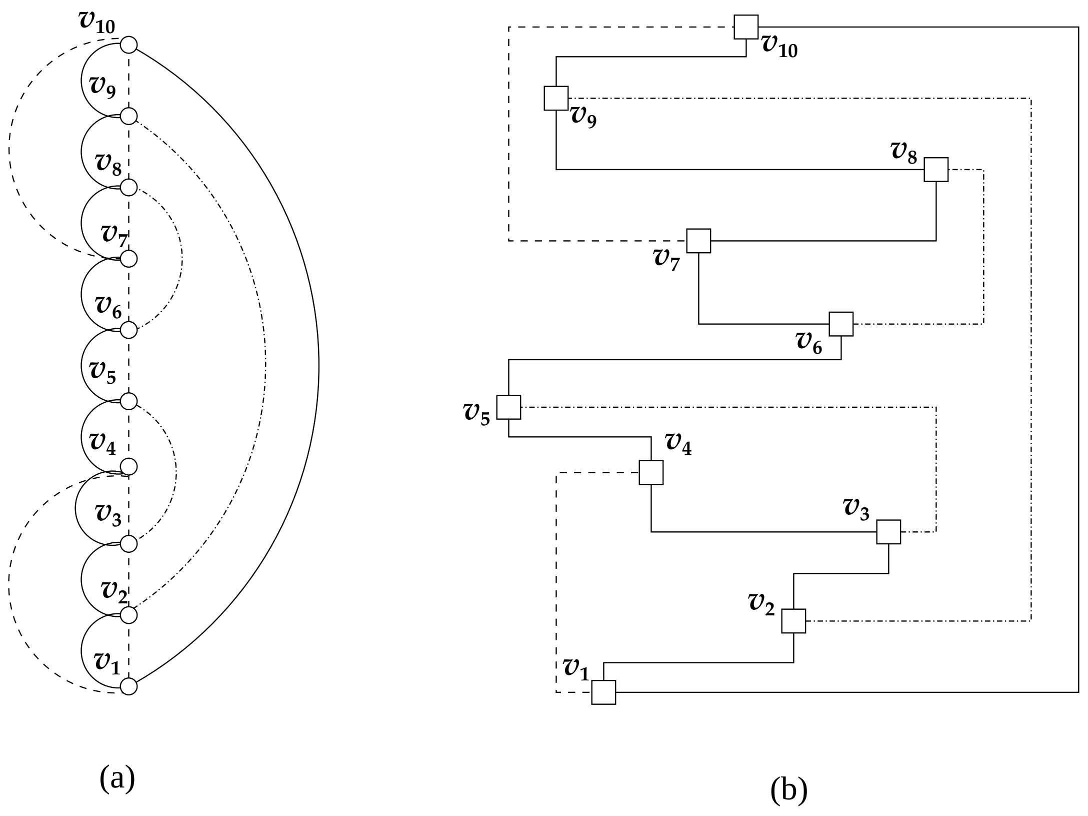


Figure 4.2: (a) Topological book embedding γ, (b) orthogonal drawing Γ.

We place the vertices of γ to the point-set S and draw the arc edges of γ by alternate sequence of vertical and horizontal line segments. Our idea is to reduce the number of bends in the drawing. Then we have the following lemma.

Lemma 4.2.3 *Let G be a 3-connected cubic planar graph of n vertices and γ be a topological book embedding of G with no spine crossing. Then one can find an orthogonal drawing Γ from topological book embedding γ on point-set S with at most three bends per edge in linear time.*

Proof. Without loss of generality, we will assume that no two points of S does not have same x or y coordinate. Let the order of the vertices be $\langle$ p_1, p_2, $\ldots$, p_n $\rangle$ in the monotonically increasing coordinate of y and x_i be x-coordinate of p_i for $1 \leq i \leq n$. If $v_1, v_2, \ldots, v_n$ be the order of the vertices of γ, we can map each vertex v_i of γ to the point p_i of S where $1 \leq i \leq n$. We draw the edges of γ in orthogonal drawing Γ in the following way.

We draw the edge (v_1, v_2) with two bends. The edge is drawn by vertical, horizontal and vertical line segments (see in Fig. 4.2). After that if v_i has an edge in the right page of γ and $x_i < x_{i+1}$, we draw the edge (v_i, v_{i+1}) with two bends. Otherwise we draw the edge with one bend. The edge (v_2, v_3) is drawn with two bends but the edge (v_7, v_8) is drawn with one bend in Fig. 4.2. Similar approach can be adopted for if v_i has an edge in the left page of γ and $x_i > x_{i+1}$. The inner and outer edges of C can be drawn by two bends per edge, because γ has no spine crossing. So the edges which are in the left page of γ are drawn in the left side of Γ by horizontal, vertical and horizontal line segments and the edges in the right page of γ are drawn in the right side of Γ. Now if v_1, v_n are both inner vertices, we draw the edge with two bends in the right side of Γ (see the edge (v_1, v_{10}) in Fig. 4.2) and if v_1, v_n are both outer vertices, we draw the edge with two bends in the left side of Γ. So all the edges can be drawn with at most two bends per edge. But if one of the vertices of v_1, v_n is inner and the other is outer, then it takes three bends to draw the edge in Γ. One can implement the algorithm in linear time. $\mathcal{Q.E.D.}$

One can find an orthogonal point-set embedding of 3-connected cubic planar graphs with total $3n + 1$ bends using the technique described in the proof of Lemma 4.2.3. But the number of total bends in the drawing can be reduced as mentioned in the following lemma.

Lemma 4.2.4 *Let G be a 3-connected cubic planar graph of n vertices with a hamiltonian cycle $C = \langle\ v_1,\ v_2,\ \ldots,\ v_n,\ v_1\ \rangle$ and let S be a set of n points in Euclidean plane. Then one can find an orthogonal point-set embedding Γ of G on S with at most $\left(\frac{5n}{2} + 2\right)$ bends such that the vertices of C are placed consecutively on the point-set S.*

Proof. We at first find a plane embedding G' of G such that an edge of C in the outer face of G'. We then choose an appropriate page of γ in which the inner edges will be drawn in the following way.

Let x_i denotes the x-coordinate of the point p_i of S for $1 \leq i \leq n$, we sort the the points of S by their x-coordinate in non-decreasing order. We have to consider two cases for vertex v_i of C to choose the suitable page.

Case 1: *Inner edges of G' are in the left page of γ.*

Now if vertex v_i $(1 \leq i < n)$ is mapped to the point p_1, we count the number of *nice points* p_j, points for which it is possible to draw the edge (v_{i+j-1}, v_{i+j}) with one bend for $1 \leq j \leq (n\text{-}i)$. If vertex v_{i+j-1} is an inner vertex and

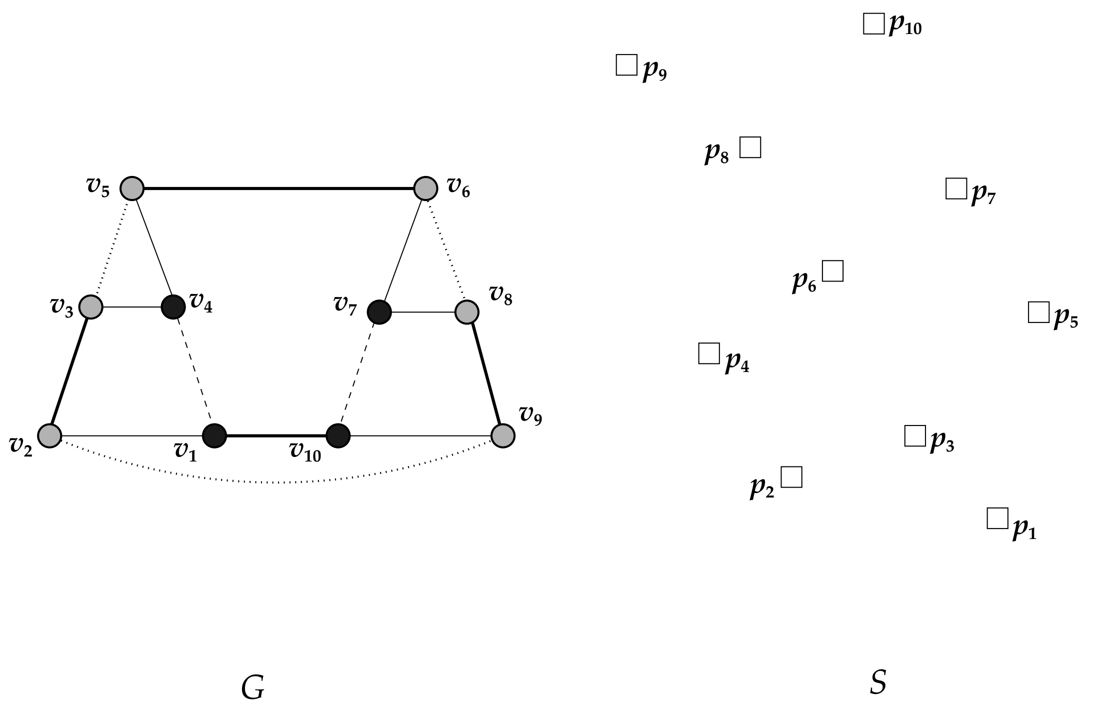


Figure 4.3: 3-connected cubic planar graph G and point-set S

$x_j < x_{j+1}$ or v_{i+j-1} is an outer vertex and $x_j > x_{j+1}$, then it is possible to draw the edge (v_{i+j-1}, v_{i+j}) with one bend. If v_1 of G maps to the point p_1 of S, then p_3, p_4, p_5, p_8 are the nice points for vertex v_1 (see Fig. 4.3). We now count the number of those nice points for which it is possible to draw the inner edge or outer edge with one bend and we call these points as *min points*. A nice point p_j is a min point if

- v_{i+j-1} is an inner vertex, the inner edge is (v_{i+j-1}, v_{i+j+k}) for $1 \leq k < (n$-i-$j)$ and there exists no such point p_{j+l} ($1 \leq l \leq k$) for $x_{j+l} < x_j$.

- v_{i+j-1} is an outer vertex, the outer edge is (v_{i+j-1}, v_{i+j+k}) for $1 \leq k < (n$-i-$j)$ and there exists no such point p_{j+l} ($1 \leq l \leq k$) for $x_{j+l} > x_j$.

In both cases we can draw the edge with one bend. Let $count_L(v_i)$ denotes the sum of nice points and min points for considering v_i as start vertex when inner edges are drawn in the left page of γ. If the long edge of γ connects both inner vertices or outer vertices, then it is possible to draw long edge with two bends. Otherwise we will decrease the $count_L(v_i)$ by 1.

Case 2: *Inner edges of G' are in the right page of γ.*

In this case, a point p_j is a *nice point* if vertex v_{i+j-1} is an inner vertex and $x_j > x_{j+1}$ or v_{i+j-1} is an outer vertex and $x_j < x_{j+1}$. If vertex v_1 maps to the point p_1, then p_2, p_6, p_7, p_9 are the nice points for vertex v_1. Now a nice point p_j is a min point if

- v_{i+j-1} is an inner vertex, the inner edge is (v_{i+j-1}, v_{i+j+k}) for $1 \leq k < (n$-i-$j)$ and there exists no such point p_{j+l} ($1 \leq l \leq k$) for $x_{j+l} > x_j$.
- v_{i+j-1} is an outer vertex, the outer edge is (v_{i+j-1}, v_{i+j+k}) for $1 \leq k < (n$-i-$j)$ and there exists no such point p_{j+l} ($1 \leq l \leq k$) for $x_{j+l} < x_j$.

Let $count_R(v_i)$ denotes the sum of nice points and min points for considering v_i as start vertex when inner edges are drawn in the right page of γ. If long edge of γ does not connect both inner vertices or outer vertices, then we decrease the $count_R(v_i)$ by 1.

We now can find the drawing Γ in the following way.

We map the vertex v_i to the point p_1 and let there is no *min point* for vertex v_i. Though the *nice points* (L) and the *nice points* (R) are complementary except points p_1 and p_n, from pigeonhole principle either $count_L(v_i)$ or $count_R(v_i)$ is at least $\frac{n-2}{2}$. If $count_L(v_i) \geq \frac{n-2}{2}$, then we draw the inner edges of G in the left side of Γ otherwise we draw the inner edges in the right side of Γ.

We now calculate the total number of bends in Γ. There are total $\frac{3n}{2}$ edges in G and among those edges only $\frac{n-2}{2}$ edges have been drawn with one bend and other edges except the long edge are drawn with two bends in the worst case scenerio. Only long edge may need three bends in Γ from Lemma 4.2.3.

$$\begin{aligned}
\text{Total number of bends} &= 1.\left(\tfrac{n-2}{2}\right) + 2.\left(\tfrac{3n}{2} - \tfrac{n-2}{2} - 1\right) + 3 \\
&= \tfrac{n}{2} - 1 + 3n - n + 2 - 2 + 3 \\
&= \tfrac{n}{2} + 2n + 2 \\
&= \tfrac{5n}{2} + 2
\end{aligned}$$

And that completes the proof of Lemma 4.2.4. $\mathcal{Q.E.D.}$

We now formally present the algorithm for finding the orthogonal point-set embedding of 3-connected cubic planar graphs.

Algorithm 3-Connected(G, C, S)

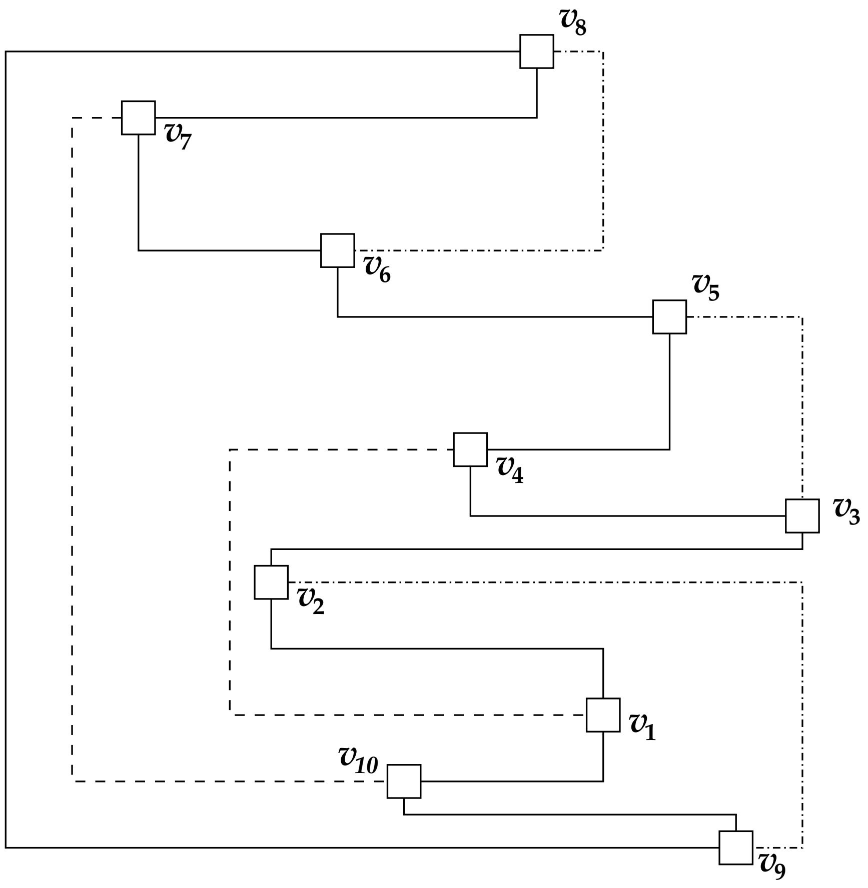


Figure 4.4: Orthogonal drawing Γ of G on S of Fig. 4.3

.

```
   begin
1    Take a plane embedding $G'$ of $G$ such that an edge of $C$ lies on outer
     face of $G'$;
2    Classify the edges of $G' - C$ as inner and outer edges;
3    Classify the vertices of $G'$ as inner and outer vertices;
4    Take an edge $(v_i, v_{i+1})$ of $C$ from the outer face of $G'$;
5    Let vertex $v_{i+1}$ be mapped to the point $p_1$, set $count_L$ to zero;
6    for each $j$, $1 < j < n$, do
       begin
7        if $v_{(i+j)\%n}$ is an inner vertex and $x_{j+1} > x_j$ then
           increase $count_L$ by 1 and mark the edge $(v_{(i+j)\%n}, v_{(i+j+1)\%n})$;
8        if $v_{(i+j)\%n}$ is an outer vertex and $x_{j+1} < x_j$ then
           increase $count_L$ by 1 and mark the edge $(v_{(i+j)\%n}, v_{(i+j+1)\%n})$;
       end
9    if $count_L \geq \left(\frac{n}{2} - 1\right)$ then
       Choose left page for inner edges;
10     for each $j$, $1 < j < n$, do
       begin
11       if the edge $(v_j, v_{j+1})$ is marked then
           Draw the edge with one bend;
       end
12   else
       Choose right page for inner edges;
13     for each $j$, $1 < j < n$, do
       begin
14       if the edge $(v_j, v_{j+1})$ is unmarked then
           Draw the edge with one bend;
       end
15   Draw the other edges with two bends except the edge $(v_i, v_{i+1})$;
16   if $v_i, v_{i+1}$ are both inner vertices or outer vertices then
       Draw the edge $(v_i, v_{i+1})$ with two bends.
     else
       Draw the edge $(v_i, v_{i+1})$ with three bends.
   end.
```

We now have the following lemma.

Lemma 4.2.5 *The **Algorithm 3-Connected** runs in linear time.*

Proof. Since G is a 3-connected cubic planar graph with n vertices, the number of edges is $\frac{3n}{2}$ which is linear. We take a plane embedding G' of G such that an edge lies on the outer face of G' in linear time. Since the number of edges of G' is linear, Line 2 and 3 can be executed in linear time. It is easy to observe that the Line 6, 7, 8 run in linear time. Finally we draw the edges in linear time. So the overall time complexity of the **Algorithm 3-Connected** is linear. $\mathcal{Q.E.D.}$

We now prove the Theorem 4.2.1. Given a 3-connected cubic planar graph G with a hamiltonian cycle and a set S of points in the Euclidean plane, we can find the orthogonal point-set embedding Γ of G with at most $\left(\frac{5n}{2}+2\right)$ bends using the Lemma 4.2.4. The drawing can be found in linear time using Lemma 4.2.5 and hence the proof of the theorem is complete.

4.3 4-Connected Planar Graphs

In this section we describe the algorithm for obtaining an orthogonal point-set embedding of 4-connected planar graphs with $\Delta \leq 4$. The following theorem is the main result of this section.

Theorem 4.3.1 *Let G be a 4-connected planar graph of n vertices with $\Delta \leq 4$ and S be a set of n points in Euclidean plane. Then one can find an orthogonal point-set embedding Γ of G on S with at most $6n$ bends in linear time.*

In the rest of the section we give a constructive proof of Theorem 4.3.1 which leads to our main algorithm.

We first present the following lemma.

Lemma 4.3.2 *Let G be a 4-connected planar graph of n vertices with $\Delta \leq 4$ and S be a set of n points in Euclidean plane. Then one can find an orthogonal point-set embedding Γ of G on S with at most $6n$ bends.*

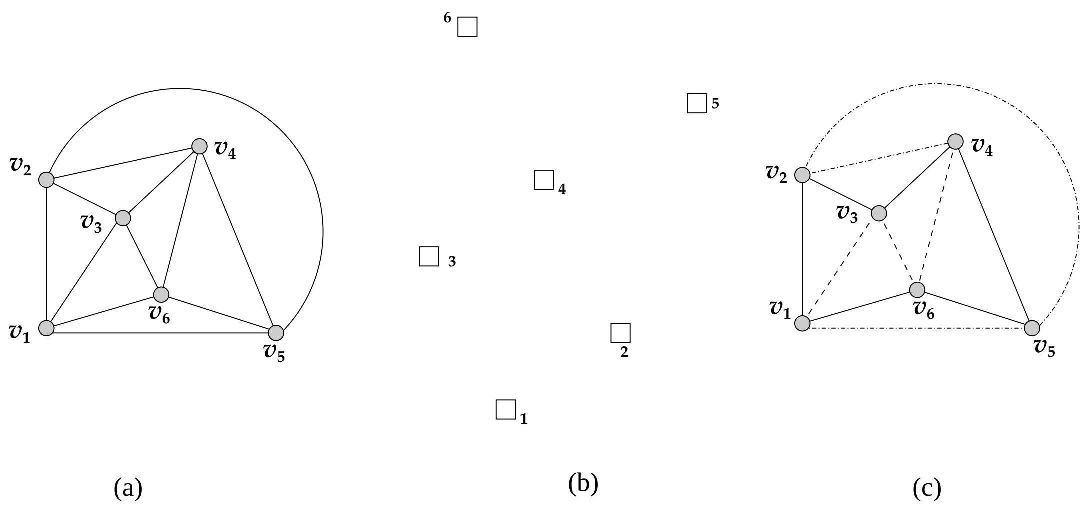


Figure 4.5: (a) 4-connected planar graph G, (b) point-set S, (c) hamiltonian cycle C in G'

.

Proof. Let G' be a plane embedding of the planar graph G. G' contains a hamiltonian cycle, since every 4-connected plane graph is hamiltonian [Tho83]. We can find a hamiltonian cycle C in G' using the algorithm of Chiba and Nishizeki [CN89] in linear time (see Fig. 4.5). We now obtain a topological book embedding γ' of G' without any spine crossing using the technique described in the proof of Lemma 4.2.2. Figure 4.6(a) illustrates the topological book embedding γ' of plane graph G' of Fig. 4.5(c).

We find an orthogonal point-set embedding Γ of G from the topological book embedding γ' as follows.

We classify the vertices of γ' in three types. *Left vertices* are those vertices which have two edges in the left page of γ' and *right vertices* are those vertices which have two edges in the right page of γ' except the edges of hamiltonian cycle C. We call the other vertices *middle vertices*. Clearly a middle vertex has one edge in the left page and another edge in the right page of γ'.

Let x_i, y_i be denote the x and y-coordinates of the point p_i of S. We use four letters L, R, U, D to represent the change of direction of an edge in the left, right, up and down, respectively. Left and right movement change the x coordinate whereas up and down movement change the y coordinate only. So if an edge (v_i, v_j) with $i < j$ is drawn by LUR sequence, it represents that the edge is drawn with horizontal, vertical and horizontal line segments consecutively and

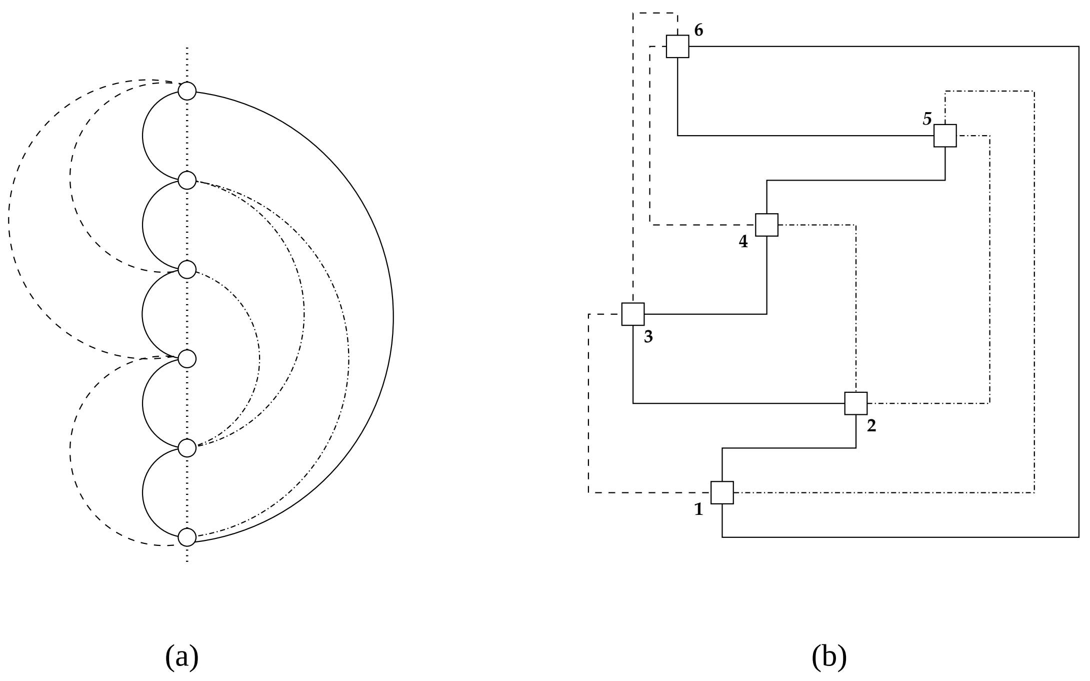


Figure 4.6: (a) Topological book embedding γ' of G', (b) orthogonal drawing Γ of G

.

it needs two bends.

We now draw the edges of C in Γ except the long edge of γ'. Three cases may arise in this context.

- *Case 1: v_i is a middle vertex.*

  We draw the edge (v_i, v_{i+1}) of C with two bends in Γ. The edge at first goes upward, then left or right depending on the position of point $i + 1$ and then goes upward to reach the point and we represent the orthogonal edge by ULU if $x_{i+1} < x_i$ and URU if $x_{i+1} > x_i$ (see the edge (v_1, v_2) and (v_4, v_5) in Fig. 4.6 (b)).

- *Case 2: v_i is an left vertex.*

  We draw the edge (v_i, v_{i+1}) by at first going to the right side of v_i and then upward if $x_{i+1} > x_i$ i.e. the edge (v_3, v_4) in Fig. 4.6. Otherwise we draw the edge by $RULU$ with three bends.

- *Case 3: v_i is an right vertex.*

We draw the edge (v_i, v_{i+1}) by at first going to the left side of v_i and then upward if $x_{i+1} < x_i$ i.e. the edge (v_2, v_3) in Fig. 4.6 (b). Otherwise we draw the edge by $LURU$ with three bends.

We now draw the inner and outer edges of G'. Let (v_i, v_j) be an inner or outer edge where $i < j$, we have to consider the three cases of v_i to draw the edge. In each case of v_i, there may be two or three sub cases depending on the nature of the vertex v_j. For all figure references refer to Fig. 4.6 (b).

- *Case* 1*: v_i is a middle vertex.*

  (a) *v_j is a middle vertex.*

  We draw the edge (v_i, v_j) of C with two bends in Γ. If (v_i, v_j) is an inner edge, we draw it by LUR. Otherwise we draw it by RUL.

  (b) *v_j is a left vertex.*

  Let the other edge of v_j is (v_j, v_k). Now if $y_j > y_k > y_i$, We draw the edge (v_i, v_j) of C with three bends in Γ by $LURD$. Otherwise we draw the edge with two bends by LUR (see the edge (v_1, v_3)).

  (c) *v_j is a right vertex.*

  Let the other edge of v_j is (v_j, v_k). Now if $y_j > y_k > y_i$, We draw the edge (v_i, v_j) of C by $RULD$ (see the edge (v_1, v_5)). Otherwise we draw the edge with two bends by RUL.

- *Case* 2*: v_i is a left vertex.*

  Let the other inner edge of v_i is (v_i, v_l).

  (a) *v_j is a middle vertex.*

  If $y_j > y_l > y_i$ we draw the edge (v_i, v_j) of C with two bends in Γ by LUR. Otherwise we draw it with either one bend by UR when $x_j > x_i$ or three bends by $ULUR$ when $x_j < x_i$.

  (b) *v_j is a left vertex.*

  Let the other edge of v_j is (v_j, v_k). Different types of edges are shown in Fig. 4.7.

  (i) If $y_j > y_k > y_i > y_l$ or $y_l > y_j > y_k > y_i$, we draw the edge with two bends by URD.

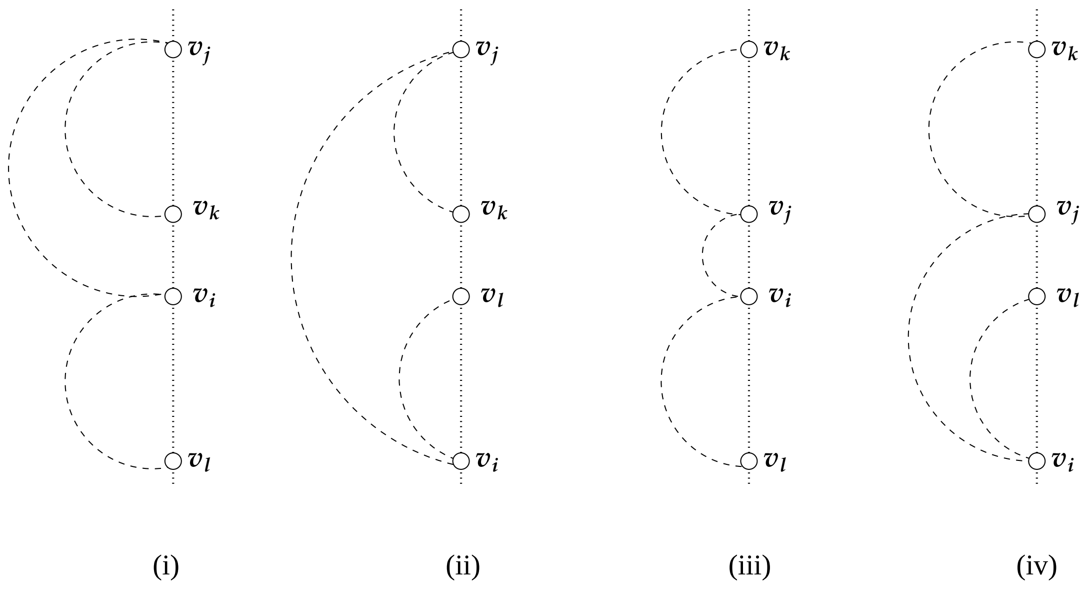


Figure 4.7: Different types of edges of case 2(b)

.

(ii) If $y_j > y_k > y_l > y_i$, the edge is drawn with three bends by $LURD$.

(iii) If $y_k > y_j > y_i > y_l$ or $y_l > y_k > y_j > y_i$, we draw the edge with one bend by UR.

(iv) If $y_k > y_j > y_l > y_i$ or $y_j > y_l > y_i > y_k$, we draw the edgewith two bends by LUR.

- *Case 3: v_i is a right vertex.*

  Let the other outer edge of v_i is (v_i, v_l).

  (a) *v_j is a middle vertex.* If $y_j > y_l > y_i$ we draw the edge (v_i, v_j) of C with two bends in Γ by RUL. Otherwise we draw it either with one bend by UL when $x_j < x_i$ or either three bends by $URUL$ when $x_j > x_i$.

  (b) *v_j is a right vertex.*

  Let the other edge of v_j is (v_j, v_k). Different types of edges are shown in Fig. 4.8.

  (i) If $y_j > y_k > y_i > y_l$ or $y_l > y_j > y_k > y_i$, we draw the edge with two bends by ULD.

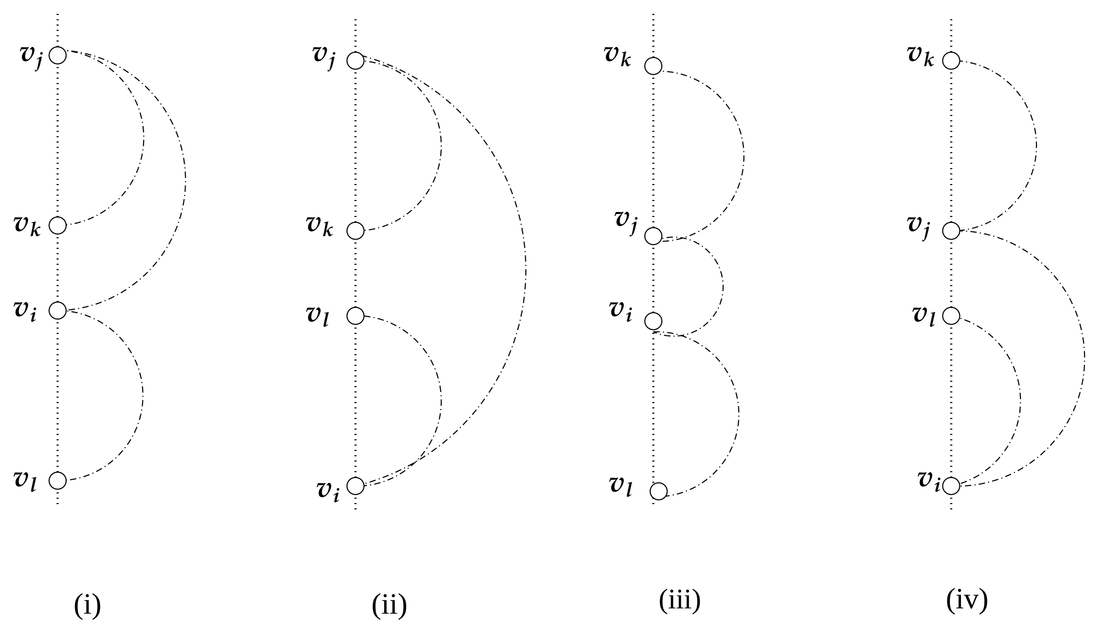


Figure 4.8: Different types of edges of case 3(b)

.

(ii) If $y_j > y_k > y_l > y_i$, the edge is drawn with three bends by $RULD$.

(iii) If $y_k > y_j > y_i > y_l$ or $y_l > y_k > y_j > y_i$, the edge is drawn with one bend by UL.

(iv) If $y_k > y_j > y_l > y_i$ or $y_j > y_l > y_i > y_k$, we draw the edge with two bends by RUL.

At last we draw the long edge (v_1, v_n) of γ' in Γ. Three cases may be arise for the nature of the vertex v_n.

- *Case* 1*: v_n is a middle vertex.*

  We draw the edge with four bends by $DRULD$.

- *Case* 2*: v_n is an left vertex.*

  If v_1 is an *outer vertex*, then we draw the edge with four bends by $DRULD$. Otherwise we draw the edge with three bends by $DRUL$ (see the edge (v_1, v_6) in Fig. 4.6 (b)).

- *Case* 3*: v_n is an right vertex.*

If v_1 is an *outer vertex*, then we draw the edge with three bends by $DLUL$. Otherwise we draw the edge by $DLUR$.

We now compute an upper bound on the number of bends in othogonal point-set embedding of 4-connected planar graphs from the drawing algorithm. The edges of C except the long edge have been drawn with at most three bends per edge, and the long edge may be drawn with at most four bends. For the case of inner and outer edges, each edge may be drawn with at most three bends. If all these edges are drawn with three bends, then the long edge is drawn with three bends.

$$\begin{aligned}\text{Total number of bends} &= 3.(n-1)+3+3.n\\ &= 3n-3+3+3n\\ &= 6n\end{aligned}$$

And that completes the proof. $\mathcal{Q}.\mathcal{E}.\mathcal{D}.$

We now formally present the algorithm for finding the orthogonal point-set embedding of 4-connected 4-regular planar graphs.

```
Algorithm 4-Connected($G, S$)
begin
1  Take a plane embedding $G'$ of $G$ such that an edge of $C$ lies on outer
   face of $G'$;
2  Find a hamiltonian cycle $C$ in $G'$;
3  Classify the vertices of $G'$ as inner and outer vertices;
4  Find a topological book embedding $\gamma$ of $G'$;
5  Classify the vertices of $G'$ as middle, left and right vertices. A vertex
   is middle vertex if has one edge in left page and one edge in right page
   except the edges of $C$. A vertex is left vertex if it has two edges in the
   left page except the edges of $C$;
6  Draw all the edges with necessary bends using case comparison.
end.
```

We now have the following lemma.

Lemma 4.3.3 *The **Algorithm 4-Connected** takes linear time.*

Proof. We can compute the time complexity of **Algorithm 4-Connected** as follows. We can find a hamiltonian cycle C in a 4-connected planar graph in linear time using the algorithm of Chiba and Nishizeki [CN89]. Line 3 and line 4 can be executed in linear time. Since G has $2n$ edges, line 6 runs in linear time also. So the overall time complexity of **Algorithm 4-Connected** is linear. $\mathcal{Q.E.D.}$

It is now left to prove the Theorem 4.3.1. Given a 4-connected 4-regular graph G and a set S of points in Euclidean plane, we can find an orthogonal point-set embedding of G on S with at most $6n$ bends using Lemma 4.3.2. Finally the drawing can be found in linear time using the Lemma 4.3.3 and hence the the proof of Theorem 4.3.1 is complete.

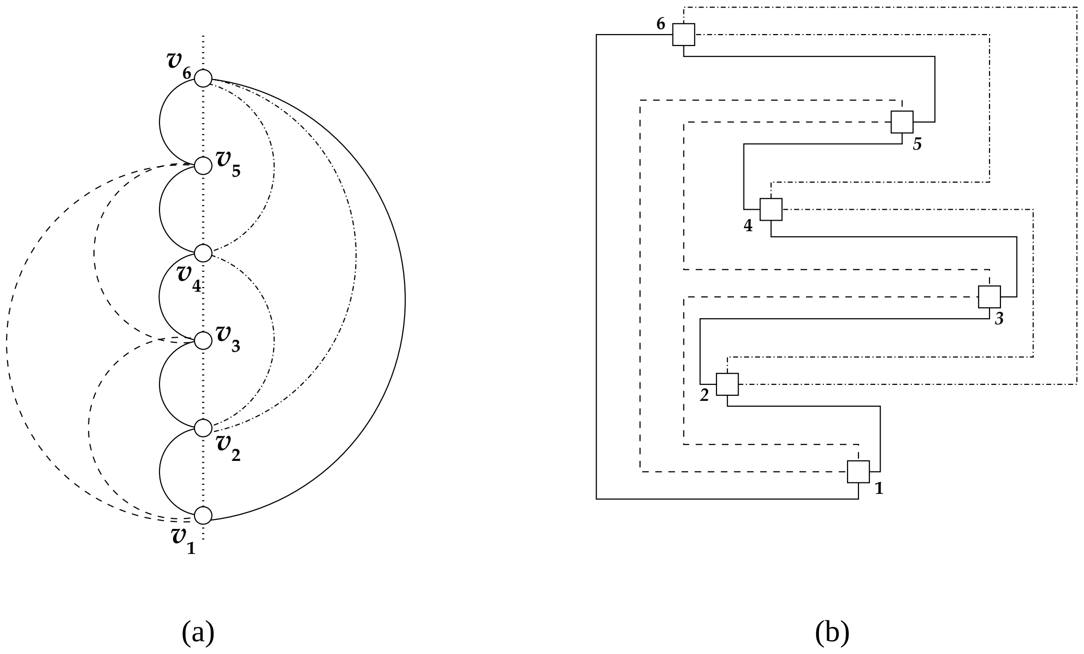


Figure 4.9: Tight example of 4-connected planar graphs for which $6n$ bends are necessary

.

Theorem 4.3.1 finds an orthogonal drawing in the point-set of 4-connected planar graphs with mapping if mapping satisfies certain criteria. We now have the following theorem.

Theorem 4.3.4 *Let G be a 4-connected 4-regular planar graph of n vertices, S be a set of n points in Euclidean plane and Φ be a mapping from the vertices of G to the points in S. Let $p_1, p_2, \ldots, p_n$ be the sorted order of the points of S according to y-axis in non-decreasing order. Then one can find an orthogonal point-set embedding Γ of G on S with the mapping Φ if $\Phi^{-1}(p_1), \Phi^{-1}(p_2), \ldots, \Phi^{-1}(p_n), \Phi^{-1}(p_1)$ represents a hamiltonian cycle in G. Moreover Γ can be obtained in linear time with at most $6n$ bends.*

Figure 4.9 illustrates a tight example of a 4-connected planar graph G, a point-set S with a mapping Φ where each vertex v_i of G is mapped to a point i of S.

4.4 Conclusion

In this chapter we gave two algorithms for finding orthogonal drawings of 3-connected cubic planar graphs and 4-connected planar graphs. Both the algorithms find the orthogonal drawings with fewer bends. Moreover we gave an upper bound on number of bends for both the drawings.

Chapter 5

Conclusion

In this thesis we addressed the problem of finding point-set embeddings of planar graphs. Our first algorithm was for finding an upward point-set embedding of upward planar digraphs with a mapping. After that we gave two algorithms for orthogonal point-set embeddings of 3-connected cubic planar graphs having a hamiltonian cycle and 4-connected planar graphs with $\Delta \leq 4$. In all cases, we first found a topological book embedding of given planar graph using a hamiltonian cycle in the graph. After that we found the required drawing using that topological book embedding. In the case of upward point-set embedding, we gave an upper bound on number of bends in the drawing and conjectured that the upper bound can be improved using our result. Our algorithm for finding an orthogonal point-set embedding of 3-connected cubic planar graphs uses at most $\left(\frac{5n}{2}+2\right)$ bends. Our last algorithm finds a point-set embedding of 4-connected planar graphs ($\Delta \leq 4$) with at most $6n$ bends.

The problem of minimizing the number of bends in a drawing of planar graphs is motivated by both theoretical interest and practical applications. The problem of minimum bends drawing of planar graphs has attracted much interest for the researchers. Rahman and Nishizeki gave a linear time algorithm to find a bend-optimal orthogonal drawing for plane graphs with $\Delta \leq 3$ [RN02]. But for planar graphs, there had been no algorithms for finding bend optimal drawing. Garg and Liotta [GL99] gave an $O(n^2)$ time algorithm for finding orthogonal drawings of planar 2-connected graphs with three bends more than the minimum number of bends. In this respect, our upper bounds on number of bends in the drawing are of good value. The practical applications of point-set embedding of planar graphs arise from the requirement in various fields, especially for the VLSI circuit design and circuit schematics. In VLSI circuit design, it is always desirable to find the orthogonal drawing with minimum bends. The problem of finding point-set embedding arises from the concept of fixed positions of modules in VLSI chip beforehand. Our algorithms are more applicable for the standard cell layout of VLSI design. Thus the result in this thesis is more interesting and motivating for the theoretical prospects rather than practical applications.

The following is a brief list of future works related to our results presented in this thesis.

- We have provided an algorithm for finding an upward point-set embedding of upward planar digraphs with mapping. Our algorithm presents the upper bound of number of bends per edge and we have given a conjecture on the number of edges which will cross the spine in γ. The conjecture can be used to improve the upper bound on total number of bends in the drawing. It would be interesting to either prove or disprove the conjecture. Minimizing the total bends in upward point-set embedding with mapping and also improve the time complexity would be an interesting area of research.

- We have given an algorithm for orthogonal point-set embedding of 3-connected cubic planar graphs with fewer bends. The upper bound on number of bends obtained from our algorithm is $\left(\frac{5n}{2} + 2\right)$. It will be interesting to reduce the number of bends of the drawing and also if possible, find an algorithm for finding the minimum bend orthogonal point-set embedding of planar graphs in restricted cases.

- Algorithm for finding orthogonal point-set embedding of 4-connected planar graphs with fewer bends has been introduced in this thesis. Minimizing the number of bends in the drawing may be a good area for future research.

References

[Bos97] P. Bose, On Embedding an OuterPlanar Graph in a Point Set. *Computational Geometry: Theory and Applications*, 23(3):303–312, 2002.

[Cab06] S. 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.

[CAR09] M. E. Chowdhury, M. J. Alam, and M. S. Rahman, On Upward Point-Set Embedding of Upward Planar Digraphs. In Proceedings of the *16th Mathematics Conference* of Bangladesh Mathematical Society, 2009.

[CN89] N. Chiba and T. Nishizeki, The Hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *Journal of Algorithms*, 10(2):187–211, 1989.

[Far48] I. Fary, On straight line representations of planar graphs. *Acta Sci. Math. Szeged.*, 11:229–233, 1948.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness.* W. H. Freeman and Company, New York, USA, 1979.

[GL99] A. Garg and G. Liotta, Almost bend-optimal planar orthogonal drawings of biconnected degree-3 planar graphs in quadratic time. In *the 7th International Symposium on Graph Drawing (GD 2008)*, volume 1731 of *Lecture Notes in Computer Science*, pp. 38–48. Springer, 1999.

[GLMS07] F. Giordano, G. Liotta, T. Mchedlidze, and A. Symvonis, Computing Upward Topological Book Embeddings of Upward Planar Digraphs. In *International Symposium on Algorithms and Computation (ISAAC 2007)*, volume 4835 of *Lecture Notes in Computer Science*, pp. 172–183. Springer, 2007.

[GLW09] F. Giordano, G. Liotta, and S. H. Whitesides, Embeddability Problems for Upward Planar Digraphs. In *the 16th International Symposium on Graph Drawing (GD 2008)*, volume 5417 of *Lecture Notes in Computer Science*, pp. 242–253. Springer, 2009.

[GT01] A. Garg and R. Tamassia, On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal of Computing*, 31(2):601–625, 2001.

[Hal91] J. H. Halton, On the thickness of graphs of given degree. *Information Sciences*, 54:219–238, 1991.

[HM88] D. A. Holton and B. D. Mckay, The Smallest Non-Hamiltonian 3-Connected Cubic Planar Graps Have 38 Vertices. *Journal of Combinatorial Theory*, 45(3):305–319, 1988.

[KW02] M. Kaufmann and R. Wiese, Embedding Vertices at Points: Few Bends Suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1):115–129, 2002.

[Miy06] M. Miyaguchi, Topological book embedding of bipartite graphs. In*Proceedings of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, volume 89-A(5), pages 1223–1226, 2006.

[MS09] T. Mchedlidze and A. Symvonis, Crossing-Optimal Acyclic Hamiltonian Path Completion and Its Application to Upward Topological Book Embeddings. In*Workshop on Algorithms and Computation (WALCOM 2009)*, volume 5431 of *Lecture Notes in Computer Science*, pp. 250–261. Springer, 2009.

[MS10] T. Mchedlidze and A. Symvonis, On ρ-Constrained Upward Topological Book Embeddings, In*17th International Symposium on Graph Drawing (GD 2009)*, volume 5849 of *Lecture Notes in Computer Science*, pp. 411–412. Springer, 2010.

[NC88] T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms.* North-Holland, Amsterdam, 1988.

[NR04] T. Nishizeki and M. S. Rahman, *Planar Graph Drawing.* World Scientific, Singapore, 2004.

[Pur97] H. C. Purchase, Which aesthetic has the greatest effect on human understanding? In *the Proceedings of the 5th International Symposium on Graph Drawing (GD 1997)*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 1997.

[PGMP91] J. Pach, P. Gritzmann,B. Mohar, and R. Pollack, Embedding a Planar Triangulation with Vertices at Specified Points. *American Mathematical Monthly*, 98 (1991), pp. 165–166.

[PW98] J. Pach and R. Wenger, Embedding Planar Graphs at Fixed Vertex Locations. In *the 6th International Symposium on Graph Drawing (GD 1998)*, volume 1547 of *Lecture Notes in Computer Science*, pp. 263–274. Springer, 1998.

[Rah99] M. S. Rahman, *Efficient Algorithms for Drawing Planar Graphs.* PhD thesis, Graduate School of Information Sciences, Tohoku University, Sendai, Japan, 1999.

[RN02] M. S. Rahman and T. Nishizeki, Bend-minimum orthogonal drawings of plane 3-graphs. In*Proc. International Workshop on Graph Theoretic Concepts in Computer Science (WG '02)*, volume 2573 of *Lect. Notes in Computer Science*, pp. 367-378, Springer, 2002.

[RNN99] M. S. Rahman, S. Nakano and T. Nishizeki, A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs. *Journal of Graph Alg. and Appl.*, 3(4):31–62, 1999.

[RNN03] M. S. Rahman, T. Nishizeki and M. Naznin, Orthogonal drawings of plane graphs without bends. *Journal of Graph Alg. and Appl.*, 7(4):335–362, 2003.

[Ste51] K. S. Stein, Convex maps. In *the Proceedings of American Mathematical Society*, volume 2, pages 464–466, 1951.

[Tho83] C. Thomassen, A theorem on paths in planar graphs. *Journal of Graph Theory*, volume 7, pages 169–176, 1983.

[Wag36] K. Wagner, Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.

[Wes01] D. B. West, *Introduction to Graph Theory.* Prentice-Hall, Upper Saddle River, New Jersey, USA, 2001.

[Woo82] D. Woods, *Drawing Planar Graphs.* PhD Thesis, Stanford University, 1982.

Index

$O(n)$ notation, 21
$\kappa(G)$, 15
k-connected, 15

adjacent, 11
aesthetic properties, 2
asymptotic behavior, 21

Bend, 33
block, 15
book, 19

call graph, 12
class NP, 22
class P, 22
complexity, 21
component, 15
connected graph, 14
connectivity, 15
cut vertex, 15
cycle, 14

degree, 12
deterministic algorithm, 22
directed graph, 12
disconnected graph, 15
drawing convention, 17

edges, 1, 11
exponential, 22
external Hamiltonian, 14

face, 16

geometric properties, 2
graph, 1, 11
Graph Drawing, 1

Hamiltonian, 14

incident, 11
inner edges, 16
inner vertices, 34

left page, 20
Left vertices, 43
linear-time, 22
long edge, 35
loop, 12

middle vertices, 43
min points, 38
multigraph, 12

nice points, 37
non-deterministic algorithm, 22
non-polynomial, 22
NP-complete, 22, 23
NP-hard, 23

orthogonal drawing, 18
orthogonal point-set embedding, 5, 33
outer edges, 16
outer face, 16

outer vertices, 34

pages, 19
path, 14
planar drawing, 16, 17
planar embedding, 16
plane graph, 16
Point-set embedding, 2
polynomial, 22
polynomially bounded, 22

right page, 20
right vertices, 43
running time, 21

separator, 15
simple graph, 12
spine, 19, 20
standard cell, 6
straight-line drawing, 18
straight-line point-set embedding, 3

topological book embedding, 19
triangulated, 16

undirected graph, 12

vertex cut, 15
vertices, 1, 11

walk, 14