M. Sc. Engineering Thesis

Algorithms for Bar 1-Visibility Representations of Graphs

By

Shaheena Sultana

Student no.: 100505050F

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(BUET)

Dhaka-1000, Bangladesh

September, 2011

The thesis titled "**Algorithms for Bar 1-Visibility Representations of Graphs**", submitted by Shaheena Sultana, Roll No. 100505050F, Session October 2005, 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 September 10, 2011.

Board of Examiners

1. ______________________________

Dr. Md. Saidur Rahman — Chairman
Professor — (Supervisor)
Department of CSE, BUET, Dhaka 1000.

2. ______________________________

Dr. Md. Monirul Islam — Member
Professor & Head — (Ex-officio)
Department of CSE, BUET, Dhaka 1000.

3. ______________________________

Dr. Mahmuda Naznin — Member
Associate Professor
Department of CSE, BUET, Dhaka 1000.

4. ______________________________

Dr. M. Sohel Rahman — Member
Associate Professor
Department of CSE, BUET, Dhaka 1000.

5. ______________________________

Dr. Md. Rafiqul Islam — Member
Professor — (External)
Department of Computer Science
American International University-Bangladesh (AIUB)
Banani, Dhaka 1213.

Candidate's Declaration

This is to certify that the work presented in this thesis entitled "**Algorithms for Bar 1-Visibility Representations of Graphs**" 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.

Shaheena Sultana
Candidate

Contents

List of Figures

Acknowledgments

I would like to thank my supervisor Professor Dr. Md. Saidur Rahman for introducing me to the field of graph theory and graph drawing problems. I have learned from him how to write, speak and present well. I thank him for his patience in reviewing my so many inferior drafts, for correcting my language, suggesting new ways of thinking and encouraging me to continue my work.

I would like to thank all the members of the examination board, Prof. Dr. Md. Monirul Islam, Dr. Mahmuda Naznin, Dr. M. Sohel Rahman and Prof. Dr. Md. Rafiqul Islam, for their valuable comments.

I would also like to thank Mr. Md. Jawaherul Alam, Mr. Debajyoti Mondal and all the members of my research group for their valuable suggestions and continual encouragements.

My special thanks goes to Prof. Dr. Md. Rezaul Karim of Dhaka University for his helps. My parents, my husband and freinds also supported me to the best of their ability. My heart-felt gratitude goes to them.

Abstract

A *bar visibility representation* of a planar graph G is a drawing of G where each vertex is drawn as a horizontal line segment called bars, each edge is drawn as a vertical line segment where the vertical line segment representing an edge must connect the horizontal line segments representing the end vertices. A *bar 1-visibility representation* is a drawing of G where each vertex is drawn as a horizontal line segment, each edge is drawn as a vertical line segment where the vertical line segment representing an edge must connect the horizontal line segments representing the end vertices and a vertical line segment corresponding to an edge intersects at most one bar which is not an end point of the edge. A graph is a *bar* 1-*visibility graph* if it admits a bar 1-visibility representation. A bar 1-visibility representation is a natural generalization of a visibility representation for a non-planar graph. However, complete characterizations of bar 1-visibility graphs are not known. In this thesis, we introduce "diagonal grid graphs" and "diagonal labeled graphs", which are non-planar graphs, as bar 1-visibility graphs. We first give a linear-time algorithm for finding a bar 1-visibility representation of a diagonal grid graph. We then modify the algorithm for finding a visibility representation on a compact area. We also give an algorithm for finding a bar 1-visibility representation of a diagonal labeled graph in linear time.

Chapter 1

Introduction

The study of graphs has gained itself the identity as a fundamental working tool and a data structure that can be used to gain insight into a real world problem. Graph theory is the study of graphs, the mathematical structures used to model pairwise relations between objects from a certain collection. There are thousands of real world problems each of which underlying structure consisting of entities and their relationships, and the representation of these problems as a suitable graph classification can aid in powerful visualization of the concept. Structures that can be represented as graphs are ubiquitous, and many problems of practical interest can be represented by graphs. The study of graph theory has its widespread applications in a vast majority of fields ranging from computer network analysis, transportation network, social networking concepts to VLSI circuit design, cartography, genetics, bioinformatics, molecular chemistry and condensed matter physics.

In mathematics, a graph is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called edges. Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges. Graph drawing, a drawing of a graph is basically a pictorial representation of an embedding of the graph in the plane, usually aimed at a convenient visualization of certain properties of the graph in question or of the object modeled by the graph. A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights, or weighted graphs, are used to represent structures in which pairwise connections have some numerical values. Besides the weighted graphs, colored graphs can also be used to focus on certain special attributes of a problem.

Although visibility in the plane is a very natural concept, many fundamental problems remain unsolved. Visibility graphs are a much studied approach to these problems. Generally speaking, a visibility graph consists of a set of shapes in the plane, the vertices, and a concept of visibility that defines the edges of the graph. Bar visibility graphs are among the best understood classes of visibility graphs. Here the vertices correspond to horizontal line segments called bars, and visibility runs vertically along lines of sight which connect two bars while being disjoint from all others. These graphs have been completely characterized by Tamassia and Tollis [25]. The concept of bar visibility graphs came up in the early 1980s when many new problems in visibility theory arose, originally inspired by applications dealing with determining visibilities between different electrical components (VLSI-design). Other applications arise when large graphs are to be displayed in a transparent way, and in the rapidly developing field of computer graphics.

Several variations and generalizations of bar visibility graphs have been considered, using different definitions for the type of bars or the kind of visibility or often both. For example, Bose, Dean, Hutchinson and Shermer [19] introduced rectangle visibility graphs, considering rectangles with horizontal and vertical visibility. Hutchinson [14] investigates arc- and circle-visibility graphs, where the vertices correspond to arcs of concentric circles and visibility can go through the origin. Recently, new classes of bar visibility have been introduced by restricting the vertex representations to unit bars or generalizing them to sets of several bars [3].

A *bar k-visibility graph* is another recent generalization of a bar visibility graph. They have been introduced by Dean, Evans, Gethner, Laison, Safari and Trotter [3] on the graph drawing symposium 2005 in Limerick. The new idea is that lines of sight are allowed to intersect at most k other bars. This variation results in a much more complex class of graphs: while it is easy to see that all bar visibility graphs are planar, this is not true for bar k-visibility graphs, and no other immediate property provides an approach to their structure.For the case $k = 1$, a graph is called *bar 1-visibility graph.*

We have focused in this thesis the visibility representation of non-planar graphs. A visibility representation (VR for short) of a plane graph G is a drawing of G, where the vertices of G are represented by non-overlapping horizontal segments (called vertex segments), and each edge of G is represented by a vertical line segment touching the vertex segments of its end vertices. Tamassia & Tollis [25] have given a linear time algorithm for constructing a visibility representation of a planar graph. In a bar k-visibility representation of a graph a horizontal line corresponding to a vertex and the vertical line segment corresponding to an edge intersects at most k bars which are not end points of the edge. Thus a visibility representation is a bar k-visibility representation for $k = 0$. For $k = 1$, a line segment corresponding to an edge intersects at most one bar which is not an end point of the edge, and the representation is called a bar 1-visibility representation. A graph is a *bar 1-visibility graph* if it admits a bar 1-visibility representation. Recently, Fleshner and Massow have investigated some graph theoretic properties of bar 1-visibility graphs [9]. However, there is no algorithm for finding a bar 1-visibility representations of a bar 1-visibility graph.

In this thesis, we study bar 1-visibility graphs. It is easy to see that all bar visibility graphs are planar. But bar k-visibility graphs which are non-planar are recent generalization of bar visibility graphs. We introduce diagonal grid graphs and diagonal labeled graphs which are non-planar graphs as bar 1-visibility graphs. The 0-visibility representation of a graph is also a 1-visibility representation of the graph. Since every planar graph has a 0-visibility representation, algorithms for finding 1-visibility representation of planar graphs are known. Thus the main idea is as follows: first obtain a planar graph by deleting some edges from the input non-planar graph, then obtain a visibility drawing of the planar graph and finally place the deleted edges which gives bar 1-visibility representation. In this thesis, we first give a linear time algorithm for finding a bar 1-visibility representation of non-planar graphs. Then we compact the area for the visibility representation.

We will give the details of the above mentioned algorithm and some of the previous results in this field that have a significant impact on our work. In this chapter, we give some introductory concepts of visibility representation and bar 1-visibility representation that will help realizing the concepts presented here.

Also, we have presented some applications of this topic in various fields. The rest of this chapter is organized as follows. In Section 1.1, we define visibility representation and bar 1-visibility representation of graphs, which is the central idea of this thesis. Section 1.2 depicts some interesting applications of visibility representations. In Section 1.3, we present a brief history of visibility representation and bar 1-visibility representation and on the basis of that, in Section 1.4, we depict the scope and objective of this thesis. In Section 1.5, we present the organization of this thesis.

1.1 Visibility Representations of Planar Graphs

As already stated above, S be a set of disjoint horizontal line segments, or bars, in the plane. We say that a graph G is a *bar visibility graph*, and S a *bar visibility representation* of G, if there exists a one-to-one correspondence between vertices of G and bars in S, such that there is an edge between two vertices in G if and only if there exists an unobstructed vertical line of sight between their corresponding bars. As an example, Figure 1.1(a) shows a plane graph G and Figure 1.1(b) visibility representation S of the graph G.

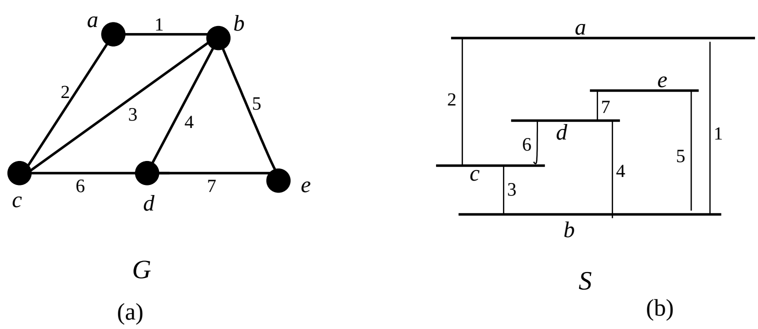


Figure 1.1: (a) A plane graph G and (b) visibility representation S of G.

We define a *bar k-visibility graph* to be a graph with a bar visibility representation in which a sight line between bars X and Y intersects at most k additional bars. A graph is a *bar k-visibility graph* if it admits a *bar k-visibility representation* as follows. Each vertex is represented by a horizontal segment (bar) in the Euclidean plane. Two vertices are joined by an edge if and only if the two corresponding bars can be joined by a vertical line segment (line of sight), which intersects at most k other bars. Lines of sight that do not intersect any bar are called direct, all others are indirect lines of sight. Inspired by this, the corresponding edges are divided into direct and indirect edges. For $k = 1$, the representation is called bar 1-visibility representation.

Let G be a graph. A *bar 1-visibility representation* of G is a drawing of G where

- each vertex is drawn as a horizontal line segment,
- each edge is drawn as a vertical line segment where the vertical line segment representing an edge intersects at most one bar which is not an end point of the edge.

As an example, Figure 1.2(b) is a bar 1-visibility representation of the graph in Figure 1.2(a).

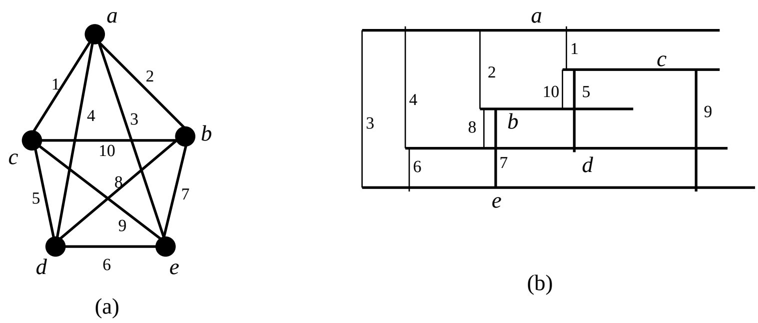


Figure 1.2: (a) A bar 1-visibility graph G and (b) bar 1-visibility representation of G.

1.2 Application of Visibility Representations

The problem of computing a compact Visibility Representation is important not only in algorithmic graph theory, but also in practical applications such as VLSI layout [4]. Modules and their interconnections of a VLSI circuit are given as a graph where a vertex of the graph represents a module of the VLSI circuit and an edge represents an interconnection between two modules. From a visibility representation, a planar polyline drawing can be generated with $O(1)$ bends per edge in linear time [17]. We can construct a planar upward polyline drawing of a planar st-graph G using its visibility representation. We draw each vertex in an arbitrary point inside its vertex segment. We draw each edge (u, v) of G as a three segment polygonal chain. Visibility representations can also be used to generate planar orthogonal drawings.

Every planar graph has bar visibility representation. A bar 1-visibility representation is a natural generalization of the notion of visibility representations for non-planar graphs. However complete characterizations and drawing algorithms of bar 1-visibility graphs are not known. We have been motivated to characterize and to find out the drawing algorithm of these special types of graphs.

1.3 Previous Results

The problem of visibility representation has gained its own inherent interest in related topics and has significant researches based in its specific application and parameters. In this section, we give an outline of the results found in this area. Visibility representation has practical applications in VLSI layout [4] and several researchers concentrated their attention on visibility representations [4, 2]. Otten and Van Wijk [18] have shown that every planar graph admits a visibility representation and Tamassia and Tollis [25] have given a linear-time algorithm for constructing a visibility representation of a planar graph. Battista, Tamassia and Tollis have given constrained visibility representation of graphs [5].

Dean et al. have introduced a generalization of visibility representation for a non-planar graph which is called bar k-visibility representation [3] . While it is easy to see that all bar visibility graphs are planar, this is not true for bar k-visibility graphs, and no other immediate property provides an approach to their structure. Since all bar visibility graphs are planar, they seek measurements of closeness to planarity for bar k-visibility graphs. They have obtained an upper bound on the number of edges in a bar k-visibility graph. As a consequence, they have obtained an upper bound of 12 on the chromatic number of bar 1-visibility graphs, and a tight upper bound of 8 on the size of the largest complete bar 1-visibility graph. They also considered the thickness of bar k-visibility graphs, obtaining an upper bound of 4 when $k = 1$, and a bound that is quadratic in k for $k > 1$

For the case $k = 1$, Dean et al. used the four color theorem to show that their thickness is bounded by 4. They conjectured that no bar 1-visibility graph has thickness larger than 2. Recently, Fleshner and Massow have investigated some graph theoretic properties of bar 1-visibility graphs [9]. They proved the tight upper bound on the thickness of bar 1-visibility graphs is 3.

In recent years, several works are devoted to this field. Fabrici and Madaras [7] study the existence of subgraphs of bounded degrees in 1-planar graphs which is also called bar 1-visibility graph. It is shown that each 1-planar graph contains a vertex of degree at most 7; they also prove that each 3-connected 1-planar graph contains an edge with both end vertices of degrees at most 20. Eades and Liotta study the relationship between "RAC graphs" and 1-planar graphs [6].

1.4 Scope of this Thesis

In this section, we give an overview of the basic intuition of the approach we have taken for dealing with the problem of visibility representation of non-planar graph G. At the end, we list the results obtained by us in this thesis.

If the input graph is a diagonal grid graph then a planar graph is obtained by deleting one edge from each grid cell. Then a source and a sink vertices will be added and a visibility representation of the resulting graph will be obtained using a technique based on st-numbering. The final bar 1-visibility representation will be obtained by suitably drawing a vertical line segment in the drawing for each deleted edge. If the input graph has a diagonal labeling and contains each edge crossing inside each quadrangle then bar 1-visibility representation of that graph is also found. At first, one diagonal which contains lowest and highest number in each quadrangle will be chosen. Then the planar graph is obtained by passing that diagonal within one vertex of the quadrangle and a visibility representation of the resulting graph will be obtained by using the technique of constrained visibility representation of a planar graph [5]. The edges which pass within the vertex will be drawn by keeping the properties of bar 1-visibility representation.

Finally, our findings in this thesis are listed here.

- We have developed an algorithm for finding a bar 1-visibility representation of a *diagonal grid graph*.

- We also compact the area for the visibility representation.
- we have also developed an algorithm for finding a bar 1-visibility representation of *diagonal labeled graph*.

1.5 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we give some basic terminology of graph theory and graph drawing. In Chapter 3, we present previous algorithms on visibility representations and the new algorithm on bar 1-visibility representations of diagonal grid graphs. In Chapter 4, we mention previous algorithms on constrained visibility representations and our algorithms on bar 1-visibility representations of diagonal labeled graphs. Finally, Chapter 5 discusses the open problem in this field and gives this thesis an ending.

Chapter 2

Preliminaries

In this chapter, we define some basic terminology of graph theory, graph drawing and algorithm theory, that we will use throughout the rest of this thesis. Definitions which are not included in this chapter will be introduced as they are needed. We review, in Section 2.1, some definitions of standard graph-theoretical terms. In Section 2.2, we discuss about some special classes of graphs that are important for the ideas and concepts used in the later parts of this thesis. We devote Section 2.3 to define different orientation and numbering of planar graph. Section 2.4 and Section 2.5 define some drawing conventions of planar and non-planar graphs. Finally, we introduce the notion of time complexity in Section 2.6.

2.1 Basic Terminology

In this section we give some definitions of standard graph-theoretical terms used throughout this thesis. For readers interested in graph theory we refer to [26] and [17].

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 $V = \{v_1, v_2, \cdots, v_6\}$ is drawn as a small circle and each edge in $E = \{e_1, e_2, \cdots, e_8\}$ is drawn by a line segment.

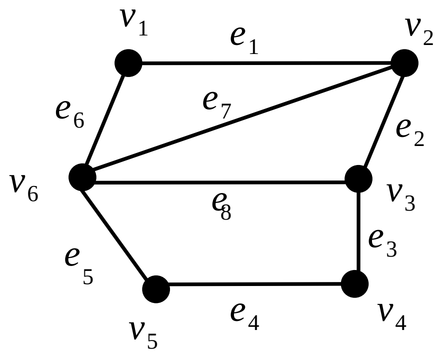


Figure 2.1: A graph with six vertices and eight edges.

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 versa). The *degree* of a vertex v in G, denoted by $d(v)$ is the number of edges incident to v. 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.

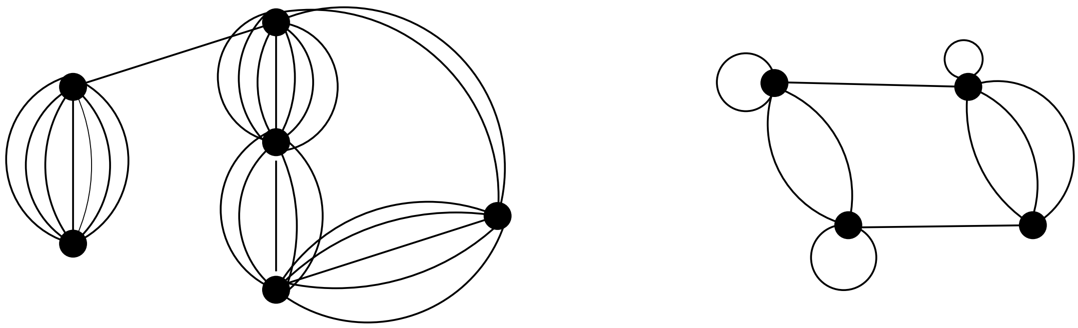
2.1.2 Simple Graphs and Multigraphs

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*. Multi graphs can arise from various application. 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 concern about simple graphs.

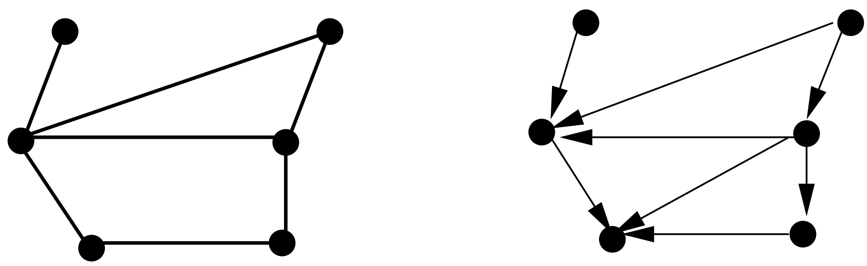


(a) (b)

Figure 2.2: Multigraphs.

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. Mathematically, the edges in a directed graphsare 2-tuple while for undirected graphs they are 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 otherwise mentioned.



(a) (b)

Figure 2.3: Directed and undirected graphs.

2.1.4 Subgraphs

A *subgraph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. If G' contains all the edges of G that join two vertices in V', then

G' is said to be the *subgraph induced by* V'. Figure 2.4 depicts a subgraph of G in Figure 2.1.

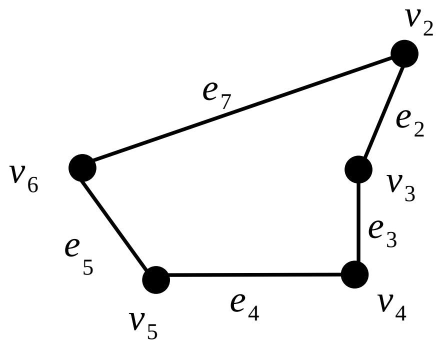


Figure 2.4: A subgraph of the graph in Figure 2.1.

We often construct new graphs from old ones by deleting some vertices or edges. If v is a vertex of a given graph $G = (V, E)$, then $G - v$ is the subgraph of G obtained by deleting the vertex v and all the edges incident to v. More generally, if V' is a subset of V, then $G - V'$ is the subgraph of G obtained by deleting the vertices in V' and all the edges incident to them. Then $G - V'$ is a subgraph of G induced by $V - V'$. Similarly, if e is an edge of a G, then $G - e$ is the subgraph of G obtained by deleting the edge e. More generally, if $E' \subseteq E$, then $G - E'$ is the subgraph of G obtained by deleting the edges in E'.

2.1.5 Paths and Cycles

A *walk*, $w = v_0, e_1, v_1, \cdots, v_{l-1}, e_l, v_l$, 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 $v_0, v_1, \cdots, v_l$ 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 $v_0, v_1, \cdots, v_l$ or by the sequence of edges $e_1, e_2, \cdots, e_l$. 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*.

2.1.6 Connectivity

A graph G is a *connected graph* 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 for 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, for example v_1 and v_{10}. 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.

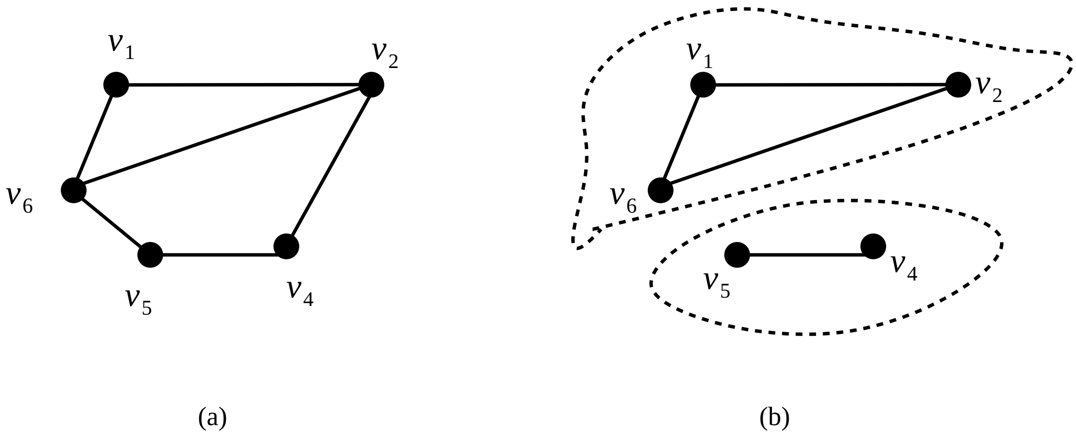


Figure 2.5: (a) A connected graph (b) a disconnected graph with two connected components.

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 Special Classes of Graphs

In this section we give some definitions of special classes of graphs related to planar graphs and non planar graphs used in the remainder of the thesis. For readers interested in planar graphs we refer to [16].

2.2.1 Planar Graphs and Plane Graphs

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.

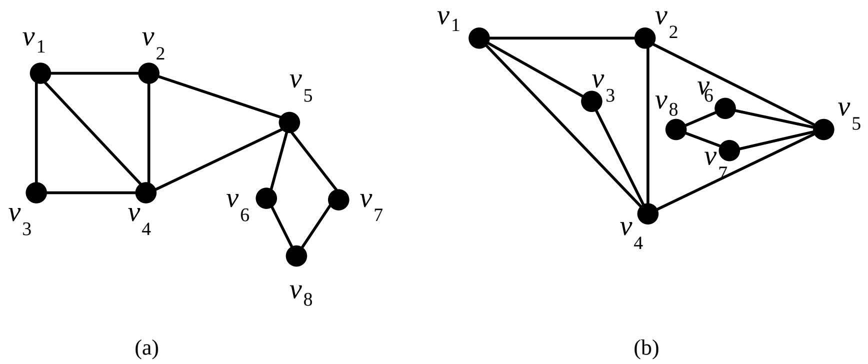


Figure 2.6: Two planar embeddings of the same planar graph.

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.

2.2.2 Dual Graphs

For a plane graph G, we often construct another graph G^* called the (*geometric*) *dual* of G as follows. A vertex v_i^* is placed in each face F_i of G; these are the vertices of G^*. Corresponding to each edge e of G we draw an edge e^* which crosses e (but no other edge of G) and joins the vertices v_i^* which lie in the faces F_i adjoining e; these are the edges of G^*. The construction is illustrated in Figure 2.7

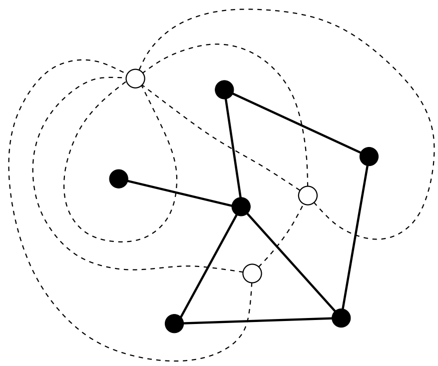


Figure 2.7: A plane graph G and its dual graph G^*.

The vertices v_i^* are represented by small white circles, and the edges e^* of G^* by dotted lines. G^* is not necessarily a simple graph even if G is simple. Clearly the dual G^* of a plane graph G is also plane. One can easily observe the following lemma:

Lemma 2.2.1 *Let G be a connected plane graph with n vertices, m edges and f faces, and let the dual G^* have n^* vertices, m^* edges and f^* faces; then $n^* = f$, $m^* = m$, and $f^* = n$.*

Clearly the dual of the dual of the plane graph G is the original graph G. However a planar graph may give rise to two or more geometric duals since the plane embedding is not necessarily unique.

2.2.3 1-Planar Graphs

A graph is called a *1-planar graph* if it can be drawn in the plane so that each its edge is crossed by at most one other edge. We can also say a 1-planar graph is a graph that has a 1-planar drawing. In Figure 2.8(a) shows the 1-planar graph G_0 . There are infinite family of 1-planar graphs. The graph G_i can be constructed from G_{i-1} which is shown in Figure 2.8(b).

In recent years, several works are devoted to this field. Fabrici and Madaras [7] have studied the existence of subgraphs of bounded degrees in 1-planar graphs. It is shown that each 1-planar graph contains a vertex of degree at most 7; they also proved that each 3-connected 1-planar graph contains an edge

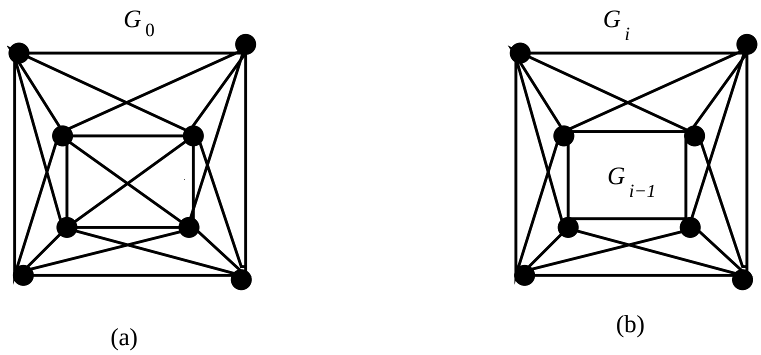


Figure 2.8: (a) A 1-planar graph G_0 and (b) constructing graph G_i from G_{i-1}.

with both endvertices of degrees at most 20. Eades and Liotta have also shown the relationship between RAC graphs and 1-planar graphs [6]. Suzuki [24] have discussed the existence of optimal 1-planar graphs which can be embedded on other closed surfaces as triangulations.

2.2.4 Bar k-Visibility Graphs

In a *bar k-visibility graphs*, bars are allowed to see through at most k other bars. They have been introduced by Dean, Evans, Gethner, Laison, Safari and Trotter [3]. Since all bar visibility graphs are planar, they seek measurements of closeness to planarity for bar k-visibility graphs. They obtain an upper bound on the number of edges in a bar k-visibility graph.

2.3 Orientation and Numbering

In this section we give some techniques of standard graph numbering used throughout this thesis.

2.3.1 st-Orientation and st-Numbering

Let $G = (V, E)$ be a biconnected undirected graph, where V and E are the set of vertices and edges, respectively. The number of vertices in G is denoted by n, that is, $n = |V|$, and the number of edges in G is denoted by m, that is, $m = |E|$. Let s and t be any two vertices of G. An st-numbering of G is a numbering of its vertices by integers $1, 2, \cdots, n$ such that a vertex s receives number 1, a vertex t

receives number n and every other vertex of G is adjacent to at least one lower-numbered vertex and at least one higher-numbered vertex. The st-numbering of a graph is not unique. Figure 2.9(a) shows an undirected biconnected graph G and Figure 2.9(b) and (c) represents two different st-numberings of the same graph G. Lempel et. al. [15] states that every biconnected graph has an st-numbering.

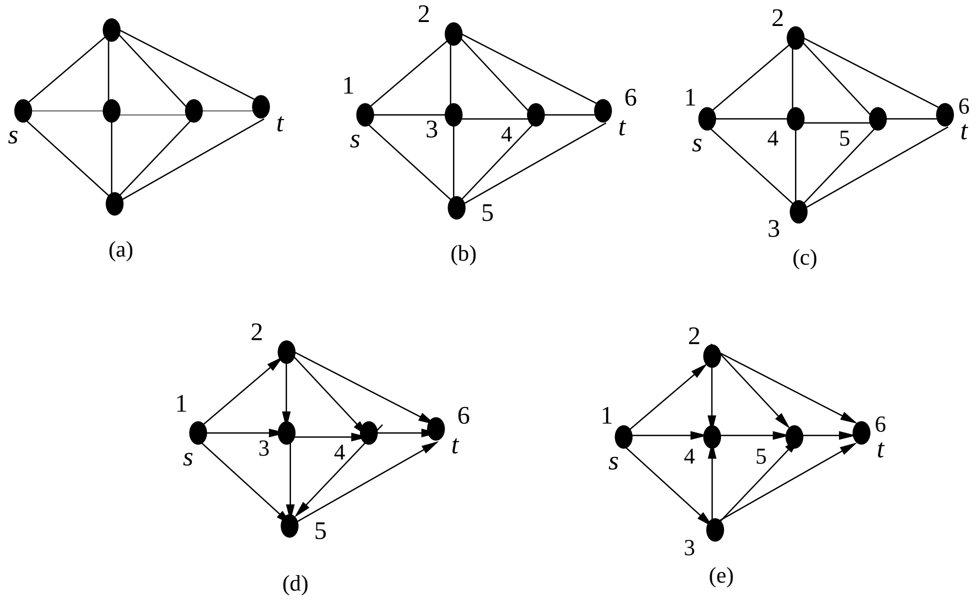


Figure 2.9: (a) A biconnected graph G with s and t, (b) an st-numbering of G, (c) another st-numbering of G, (d) an st-orientation of G, and (e) another st-orientation of G.

In a directed graph, we call a vertex a *source* if all the edges incident to that vertex is outgoing, and we call a vertex a *sink* if all the edges incident to that vertex is incoming. An st-graph is a directed, acyclic graph with a single source s and a single sink t. An st-orientation G' of G with two vertices denoted by s and t is an assignment of directions to its edges such that G becomes an st-graph. Like st-numbering, the st-orientation of a graph is not unique. Figure 2.9(d) and (e) show two different st-orientations of the biconnected undirected graph G drawn in Figure 2.9(a). An st-orientation of an undirected graph G can be easily generated using an st-numbering of G. Using an st-numbering of G, we can orient the edges of G from lower-numbered vertex to higher-numbered vertex and the resultant orientation becomes an st-orientation. We can show that st-numbering can also be generated from st-orientation. There can be more than one path from s to t in the orientation. The path from s to t which is not shorter than any other path is called longest-path. We define the term *orientation length* of an orientation denoting the length of the longest-path of that orientation. The orientation length of the orientation in Figure 2.9(d) is 5 and the orientation length of the orientation in Figure 2.9(e) is 4.

2.3.2 Topological Numbering

We define an st-graph, as a planar acyclic digraph with one source vertex s and one single sink vertex t. If we apply a topological numbering on an st-graph G, we can see that the way the vertices are numbered, give a sense of direction, from a vertex with a low number to a vertex with a higher number, to the edges. The following properties hold:

- Given a topological numbering of an st-graph G, each directed path of G visits vertices with increasing numbers.
- For every vertex v of an st-graph G, there exists at least one directed path P from s to t that contains v.

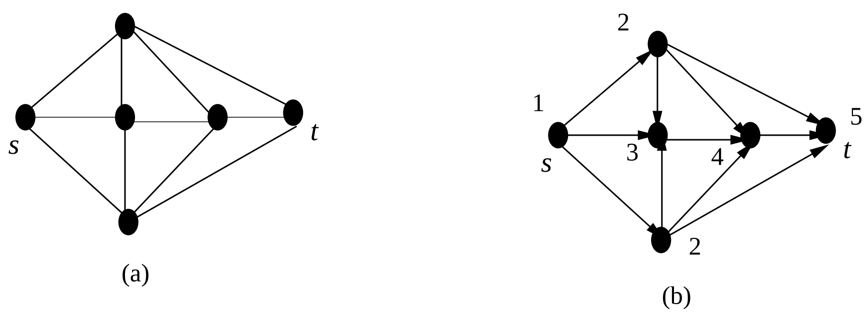


Figure 2.10: (a) A biconnected graph G with s and t, (b) optimal topological numbering of G.

2.4 Drawing Conventions of Planar Graphs

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 [17]. A few of the most important drawing styles are introduced below.

2.4.1 Planar Drawings

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 2.11(a) and (b), we show a planar and a non-planar drawing of the same graph.

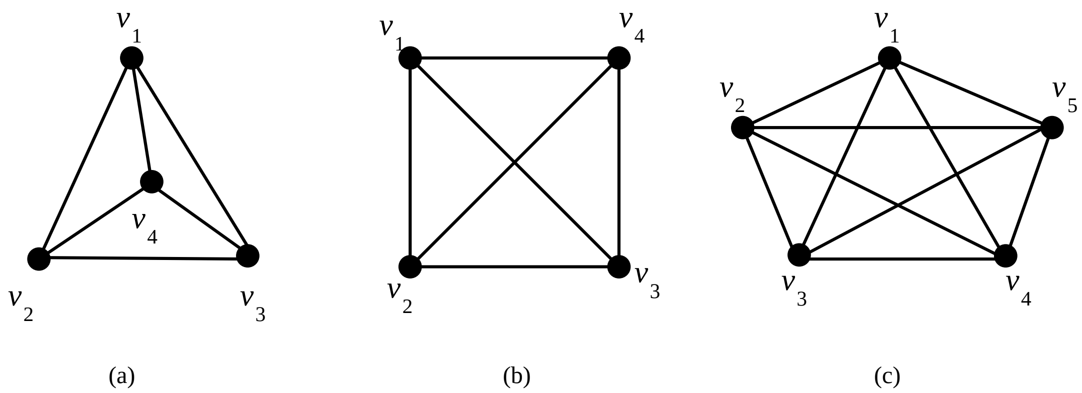


Figure 2.11: (a) A planar drawing, (b) a non-planar drawing of the graph drawn in (a), and (c) a graph which does not have a planar drawing.

Planar drawing of graphs are more convenient than non-planar drawings because, as shown empirically in [20], 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 2.11(c) is an example of one such graph.

2.4.2 Straight-line Drawings

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 2.12.

Fary [8] and Stein [23] independently proved that every planar graph has a straight line drawing.

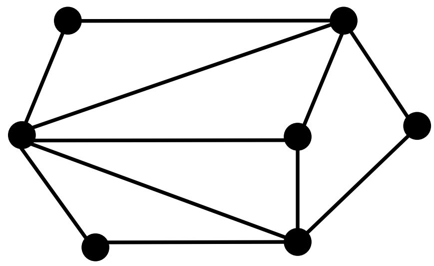


Figure 2.12: A straight-line drawing.

2.4.3 Layered Drawings

A *layered drawing* of a graph G is a drawing of G where the vertices of G are placed on a set horizontal lines called layers, and edges are drawn as straight-line segments between the end-vertices. Depending on the purpose of drawing, it may also satisfy additional constraints. Common constraints include bounds on the number of layers in the drawing, restriction on edge crossing, minimum number of edges whose removal eliminates all crossing, the maximum span of an edge, i.e., the number of layers it crosses, the total span of the edges, the maximum number of vertices in one layer etc. In the following, we define some common variants of layered drawing.

A layered drawing Γ of a graph G is called *proper* if for each edge e of G, the end-vertices of e lie on adjacent layers in Γ.

A layered drawing Γ of a graph G is called *short* if for each edge e of G, the end-vertices of e lie on the same or adjacent layers in Γ.

A layered drawing Γ of a graph G is called *upright* if the edges of G are drawn in such a way that no edge has end-vertices on the same layer in Γ.

A *k-layer planar drawing* of a graph G is a planar drawing of G on k layers. A graph G is called k-layer planar if it admits a k-layer planar drawing.

2.4.4 Grid Drawings

A drawing of a graph is called a *grid drawing* if the vertices are all located at grid points of an integer grid as illustrated in Figure 2.13.

Note that this a special class of layered drawings where not only the vertical but also the vertical position of the vertices are at integer distance from each other. This drawing approach also overcomes the following problems in graph drawing with real number arithmetic [17].

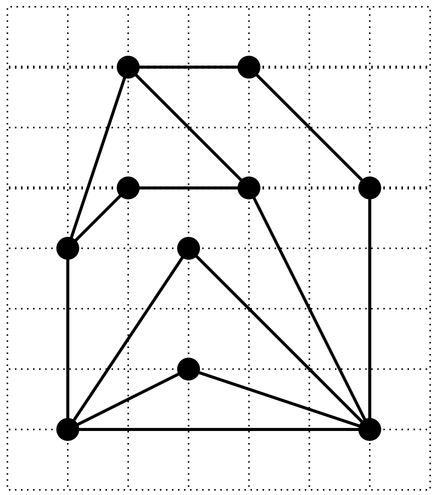


Figure 2.13: A grid drawing.

(i) When the embedding has to be drawn on a raster device, real vertex coordinates have to be mapped to integer grid points, and there is no guarantee that a correct embedding will be obtained after rounding.

(ii) Many vertices may be concentrated in a small region of the drawing. Thus the embedding may be messy, and line intersections may not be detected.

(iii) One cannot compare area requirement for two or more different drawings using real number arithmetic, since any drawing can be fitted in any small area using magnification.

The *size* of an integer grid required for a grid drawing is measured by the size of the smallest rectangle on the grid which encloses the drawing. The *width* W of the grid is the width of the rectangle and the *height* H of the grid is the height of the rectangle. The grid size is usually described as $W \times H$.

It is a very challenging problem to draw a plane graph on a grid of the minimum size. In recent years, several works are devoted to this field [1, 12, 21]; for example, every plane graph of n vertices has a straight line grid drawing on a grid size $W \times H \leq (n-1) \times (n-1)$.

2.4.5 Visibility Drawings

Let S be a set of horizontal nonoverlapping segments in the plane. Two segments s, s' of S are said to be visible if they can be joined by a vertical segment not intersecting any other segment of S. Furthermore, s and s' are called ϵ-visible if they can be joined by a vertical band of nonzero width that does not intersect any other segment of S. This is equivalent to saying that s and s' can be joined by two distinct vertical segments not intersecting any other segment of S [25].

A *w-visibility representation* for a graph $G = (V, E)$ is a mapping of vertices of G into nonoverlapping horizontal segments (called vertex-segments) and of edges of G into vertical segments (called edge-segments) such that, for each edge $(u, v) \in E$, the associated edge-segment has its endpoints on the vertex-segments corresponding to u and v, and it does not cross any other vertex-segment. In order to study the visibility representations in a unified way, we give a definition of ϵ-visibility representations using segments instead of intervals.

An *ϵ-visibility representation* for a graph G is a w-visibility representation with the additional property that two vertex-segments are ϵ-visible if and only if the corresponding vertices of G are adjacent.

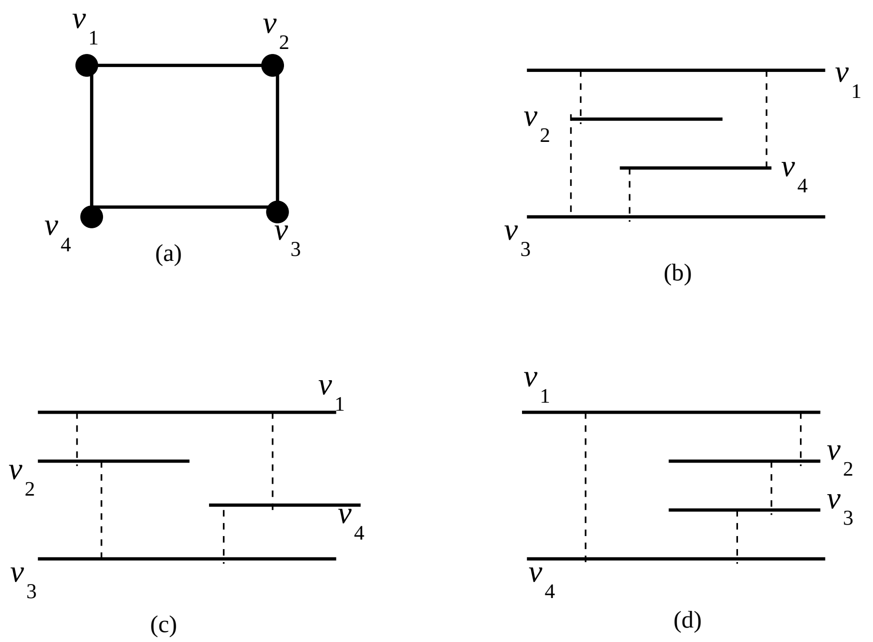


Figure 2.14: The three visibility representations: (a) a cycle of length 4; (b) w-visibility representation; (c) ϵ-visibility representation; (d) s-visibility representation.

An *s-visibility representation* for a graph G is a w-visibility representation with additional property that two vertex-segments are visible if and only if the corresponding vertices of G are adjacent.

If a graph admits any of the three aforementioned visibility representations, then it is planar, since a planar embedding of it can be immediately obtained from the visibility representation by shrinking each vertex-segment into a point.

2.4.6 2-Visibility Drawing

In a *2-visibility drawing* the vertices of a given graph are represented by rectangular boxes and the adjacency relations are expressed by horizontal and vertical lines drawn between the boxes. A *2-visibility drawing* of a graph G is illustrated in Figure 2.15.

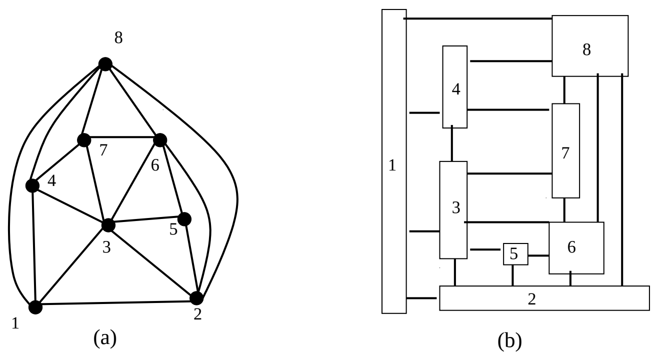


Figure 2.15: (a) A triangulated planar graph G, (b) 2-visibility representation of G .

Fobmeier, Kant and Kaufmann [11] have given a polynomial time algorithm to compute a bend-minimum orthogonal drawing under the restriction that the number of bends at each stage is at most 1.

2.4.7 Upward (Downward) Drawing

In *upward (downward) drawing* (for directed acyclic graphs) each edge is drawn monoton- ically increasing (decreasing) in the vertical direction. An *upward drawing* of a graph G is illustrated in Figure 2.16.

2.5 Drawing Conventions of Non-Planar Graphs

In this section we introduce some non-conventional drawing styles, which are found suitable in different application domain. Depending on the purpose and

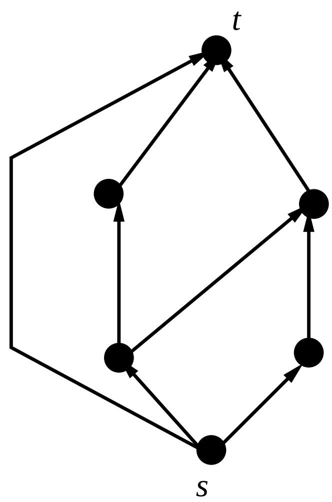


Figure 2.16: Planar upward drawing of an acyclic digraph.

objective, the vertices are typically represented with points or boxes and edges are represented with simple jordan curves. A few of the most important drawing styles are introduced below.

2.5.1 1-Planar Drawing

A *1-planar drawing* is a drawing of a graph where an edge can be crossed by at most another edge. A *1-planar graph* is a graph that has a *1-planar drawing*. A straight-line drawing is a drawing of a graph such that every edge is a straight-line segment.

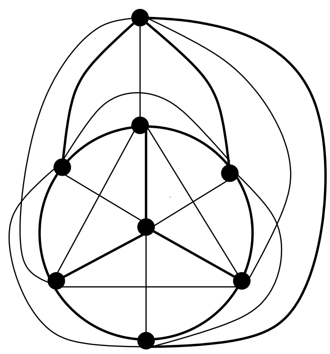


Figure 2.17: 1-planar drawing of a graph.

2.5.2 RAC Drawing

A *right angle crossing drawing* (or *RAC drawing*, for short) is a straight-line drawing where any two crossing edges form right angles at their intersection point. A *Right Angle Crossing graph* (or *RAC graph*, for short) is a graph that has a RAC drawing. Figure 2.18 shows a straight-line RAC drawing with $n = 7$ vertices and $m = 4n - 10$ edges.

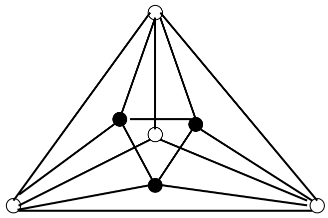


Figure 2.18: A straight-line RAC drawing.

Eades and Liotta [6] has studied the relationship between RAC graphs and 1-planar graphs in the extremal case that the RAC graphs have as many edges as possible. It is known that a maximally dense RAC graph with $n > 3$ vertices has $4n - 10$ edges. They have shown that every maximally dense RAC graph is 1-planar. Also, it is also known that for every integer i such that $i \geq 0$, there exists a 1-planar graph with $n = 8 + 4i$ vertices and $4n - 10$ edges that is not a RAC graph.

2.6 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 [13].

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.

2.6.1 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 $O(n^2)$".

2.6.2 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 *nonpolynomial*. 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.

2.6.3 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 *nondeterministic algorithm* is one for which a state may determine many next states simultaneously. We may regard a nondeterministic 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 nondeterministic algorithm examines all alternatives at the same time. A problem Q is in the *class NP* if there exists a nondeterministic 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

Diagonal Grid Graphs

3.1 Introduction

A *bar visibility representation* of a planar graph G is a drawing of G where each vertex is drawn as a horizontal line segment and each edge is drawn as a vertical line segment where the vertical line segment representing an edge must connect the horizontal line segments representing the end vertices. *Bar 1-visibility representation* is a drawing of G where each vertex is drawn as a horizontal line segment and a vertical line segment corresponding to an edge intersects at most one bar which is not an end point of the edge [3]. Figure 3.1(b) shows bar 1-visibility representations of the graph G of Figure 3.1(a).

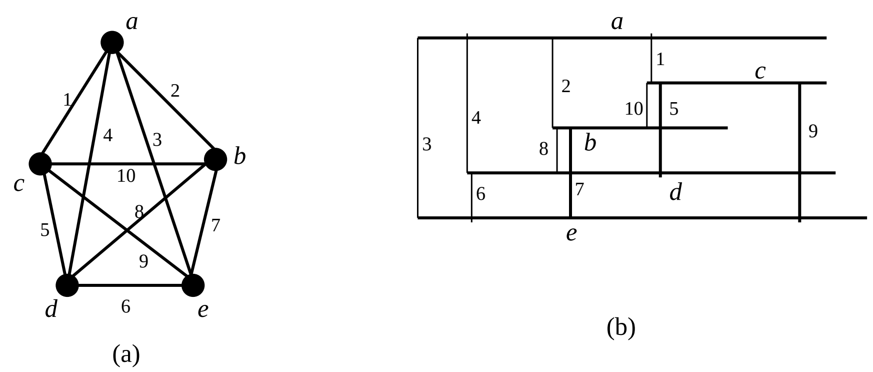


Figure 3.1: (a) Input planar graph G, (b) bar 1-visibility representations of the graph G.

Otten and Van Wijk [18] have shown that every planar graph admits a visibility representation, and Tamassia and Tollis [25] have given a linear-time algorithm for constructing a visibility representation of a planar graph. Alice M. Dean et al. have introduced a generalization of visibility representation for a non-planar graph which is called bar k-visibility representation [3]. In a bar k-visibility representation of a graph a horizontal line corresponding to a vertex is called a bar, and the vertical line segment corresponding to an edge intersects at most k bars which are not end points of the edge. Thus a visibility representation is a bar k-visibility representation for $k = 0$. For $k = 1$, a line segment corresponding to an edge intersects at most one bar which is not an end point of the edge, and the representation is called a bar 1-visibility representation. A graph is a bar 1-visibility graph if it admits a bar 1-visibility representation. Recently, Fleshner and Massow have investigated some graph theoretic properties of bar 1-visibility graphs [9, 10]. However, there is no algorithm for finding a bar 1-visibility representations of a bar 1-visibility graph. In this chapter, we develop an algorithm for finding a bar 1-visibility representation of a diagonal grid graph.

The rest of the chapter is organized as follows. In Section 3.2 contains some basic definitions and an outline of our approach. In Section 3.3 depicts an algorithm for bar 1-visibility representations of diagonal grid Graph. Finally, we summarizes the result in Section 3.4.

3.2 Preliminaries

In this section we present some basic definitions and terminologies related to graph drawing.

Let, G=(V, E) be a simple planar graph where V is the set of vertices and E is the set of edges of graph G. If u and v are two vertices of graph G then (u, v) denotes an *edge* between them.

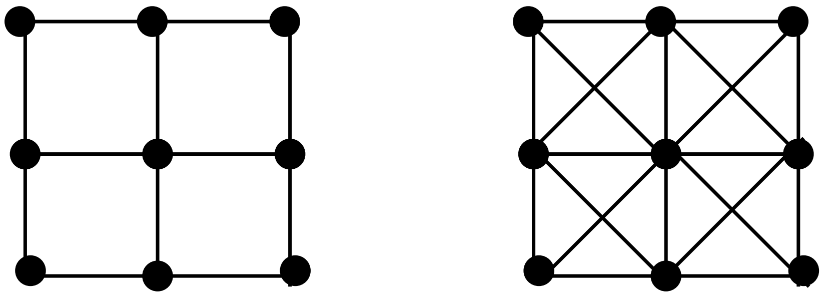
3.2.1 Diagonal Grid Graphs

Recently, several researchers have concentrated their attention on diagonal grid graphs for cordial labeling [22]. We use this graph for its 1-planarity properties.

A $p \times q$*-grid graph* is the graph whose vertices correspond to the grid points of a $p \times q$-grid in the plane and edges correspond to the grid lines between two consecutive grid points.

A diagonal grid graph $G_{p,q}$ is a $p \times q$-grid graph with diagonal edges are introduced in each cell.

For example, Figure 3.2(a) shows a $p \times q$-grid graph and (b) shows a diagonal grid graph $G_{p,q}$.



(a) (b)

Figure 3.2: (a) Grid graph, (b) diagonal grid graph.

3.2.2 Properties of Planar Acyclic Digraphs

Let G is an st-graph, as a planar acyclic digraph with one source vertex s and one single sink vertex t. If we apply a topological numbering on an st-graph G, we can see that the way the vertices are numbered, give a sense of direction, from a vertex with a low number to a vertex with a higher number, to the edges [25]. The following properties hold:

- Given a topological numbering of an st-graph G, each directed path of G visits vertices with increasing numbers.
- For every vertex v of an st-graph G, there exists at least one directed path P from s to t that contains v.

The first property holds because of the way the numbers correspond with the directions of the edges. It is easy to see why the second property is true: if it was not, there would be either no path from s to v, or from v to t, thus s or t would not be the source or sink vertices respectively.

A *planar st-graph* is an st-graph that is planar and embedded with vertices s and t on the boundary of the external face. It is customary to visualize a planar st-graph as drawn upward in the plane (with s at the bottom and t at the top), as shown in Figure 3.3. All planar st-graphs, being acyclic, admit a topological ordering (numbering).

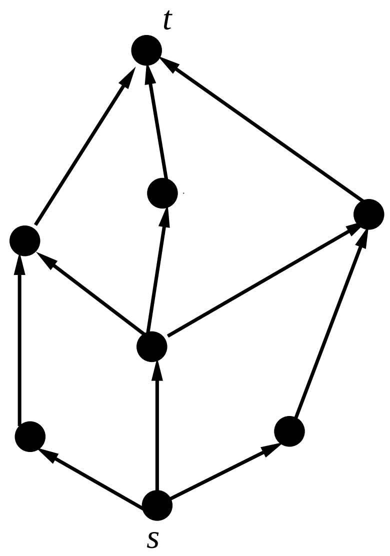


Figure 3.3: A planar st-graph.

Let now G be a planar st-graph and F be its set of faces. F contains two representatives of the external face: the left external face s^*, which is incident with the edges on the left boundary of G and the right external face t^*, which is incident with the edges on the right boundary of G. Additionally, for each $e = (u, v)$ we define $orig(e) = u$ and $dest(e) = v$. Also, we define $left(e)$ (respectively $right(e)$) to be the face to the left (respectively right) of e. Following, we give the definition of the dual graph G^* of a planar st-graph G. The *dual graph* G^* is a graph for which the following hold:

- The vertex set of G^* is the set F of faces of G including the faces s^* and t^* .

- For every edge $e \neq (s, t)$ of G, G^* has one edge $e^* = (f, g)$ where $f = left(e)$ and $g = right(e)$.

The dual graph of the graph depicted in Figure 3.3 can be seen in Figure 3.4. If we rotate G^* 90 degrees, we can see that G^* is also a planar st-graph.

Given a vertex v of a planar st-graph, the face separating the incoming from the outgoing edges in the clockwise direction is called $left(v)$, and the other separating face is called $right(v)$(see Figure 3.5).

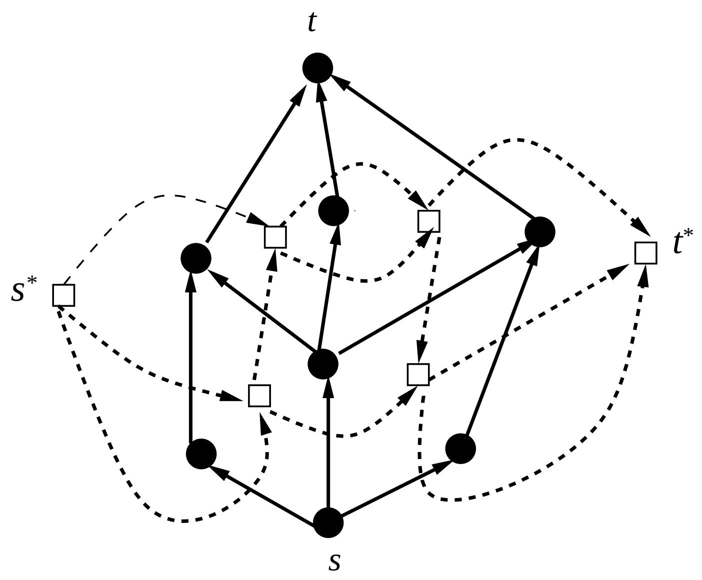


Figure 3.4: Constructing the dual graph G^* from planar st-graph.

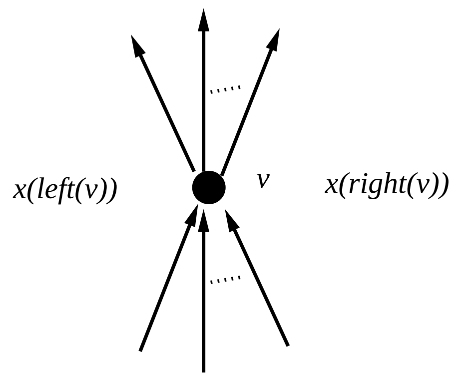


Figure 3.5: The left $(left(v))$ and right $(right(v))$ faces of a vertex v.

3.2.3 Bar Visibility Representations

In this section, we present the algorithm of Tamassia and Tollis [25] for bar visibility representations of planar graphs since some ideas of the algorithm will be used in our method.

We now formally present the algorithm for finding the visibility representation of planar graphs.

Algorithm VISIBILITY
Input : A planar st graph $G = (V, E)$.
Output: A visibility representation for G such that each vertex-segment and edge-segment has endpoints with integer coordinates.

1 Construct planar st-graph G^*.

2 Assign unit weights to the edges of G and compute st-numbering Y of G.

3 Assign unit weights to the edges of G^* and compute optimal topological numbering X of G^*.

4 For each vertex v, draw the vertex-segment $\tau(v)$ at y-coordinate $Y(v)$ and between x-coordinates $X(left(v))$ and $X(right(v) - 1)$.

5 For each edge e, draw the edge-segment $\tau(e)$ at x-coordinate $X(left(e))$ between y-coordinates $Y(orig(e))$ and $Y(dest(e))$.

Now we give an example of this algorithm. In Figure 3.6 is a planar st-graph G and its dual. At the top of Figure 3.7, we have the dual G^* of G and its optimal weighted topological numbering. This numbering provides the x-coordinates for the visibility representation. At the right of Figure 3.7, we have G and it's optimal topological numbering which provides the y-coordinates for the visibility representation. Vertices of G and G^* are represented by circle and squre respectively.

Tamassia and Tollis have given the following theorem [25]:

Theorem 3.2.1 *The algorithm VISIBILITY correctly computes a visibility representation of G.*

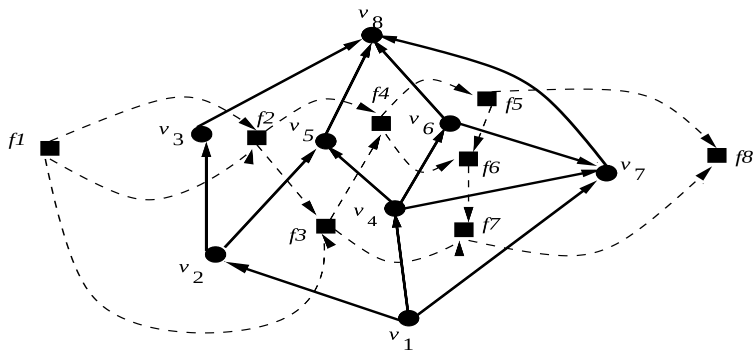


Figure 3.6: Graph G, its dual G^* and the numberings of G and G^*.

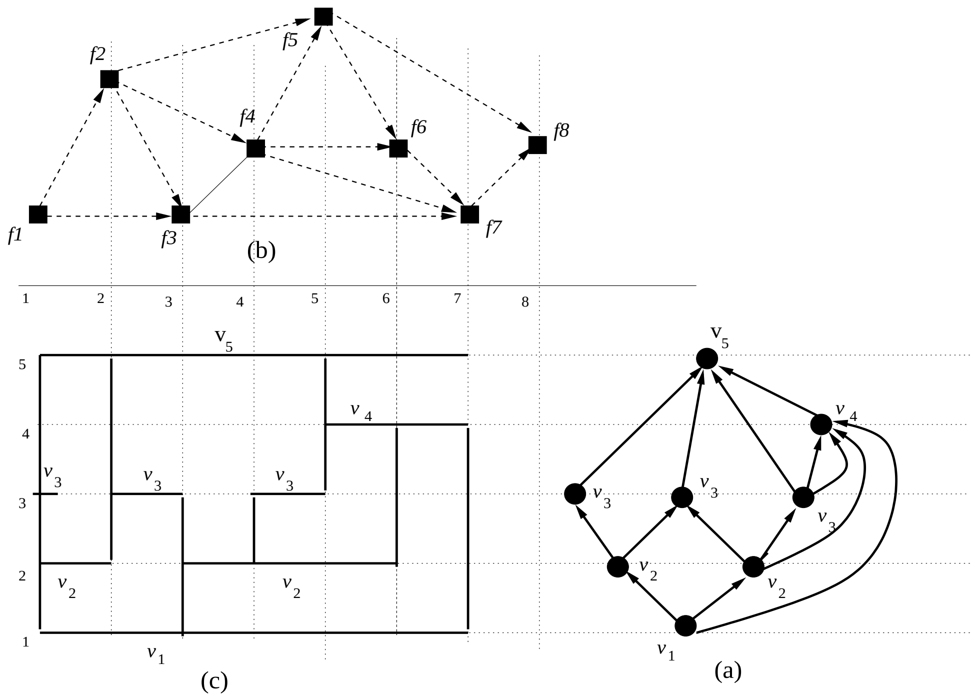


Figure 3.7: (a) Graph G, (b) dual graph G^* and (c) visibility representation of graph G.

The 0-visibility representation of a graph is also a 1-visibility representation of the graph. Since every planar graph has a 0-visibility representation, algorithms for finding 1-visibility representation of planar graphs are known. Thus the main idea is as follows: first obtain a planar graph by deleting some edges from the input non-planar graph, then obtain a visibilty drawing of the planar graph and finally placing the deleted edges give the bar 1-visibility representation.

If the input graph is a diagonal grid graph then a planar graph is obtained by deleting one edge from each grid cell. Then a source and a sink vertices will be added and a visibility representation of the resulting graph will be obtained using a technique based on st-numbering [17, 12]. The final 1-visibility representation will be obtained by suitably drawing a vertical line segment in the drawing for each deleted edge.

3.3 Bar 1-Visibility Representations

In this section we give an algorithm for obtaining bar 1-visibility representations of diagonal grid graph. This problem has an interesting correlation with the visibility representations of planar st-graph. If the input graph is a diagonal grid graph then a planar graph is obtained by deleting one edge from each grid cell. Using the numbering of diagonal grid graph and algorithm given by Tamassia and Tollis [25], we can get bar 1-visibility representations of diagonal grid graph.

3.3.1 Numbering of Diagonal Grid Graphs

We compute a special type of numbering of the undirected grid graph. We construct an st-graph from a diagonal grid graph adding one source vertex s and one single sink vertex t. If we apply a special type of st-numbering on an st-graph, we can see that the vertices are numbered from a vertex with a low number to a vertex with a higher number, to the edges. Grid are numbered sequentially from the origin at the bottom left. The following properties hold:

Let $G_{p,q}$ be a diagonal grid graph. Let i and j be the row and column number of the corresponding $p \times q$-grid graph. We assign number to each vertex $v_{i,j}$ where $1 \leq i \leq p$, $1 \leq j \leq q$ by $(i-1) * q + j$.

For example, using $p \times q$-grid where $p = 3$ and $q = 4$, we can generate the number of vertices as in Figure 3.8(b).

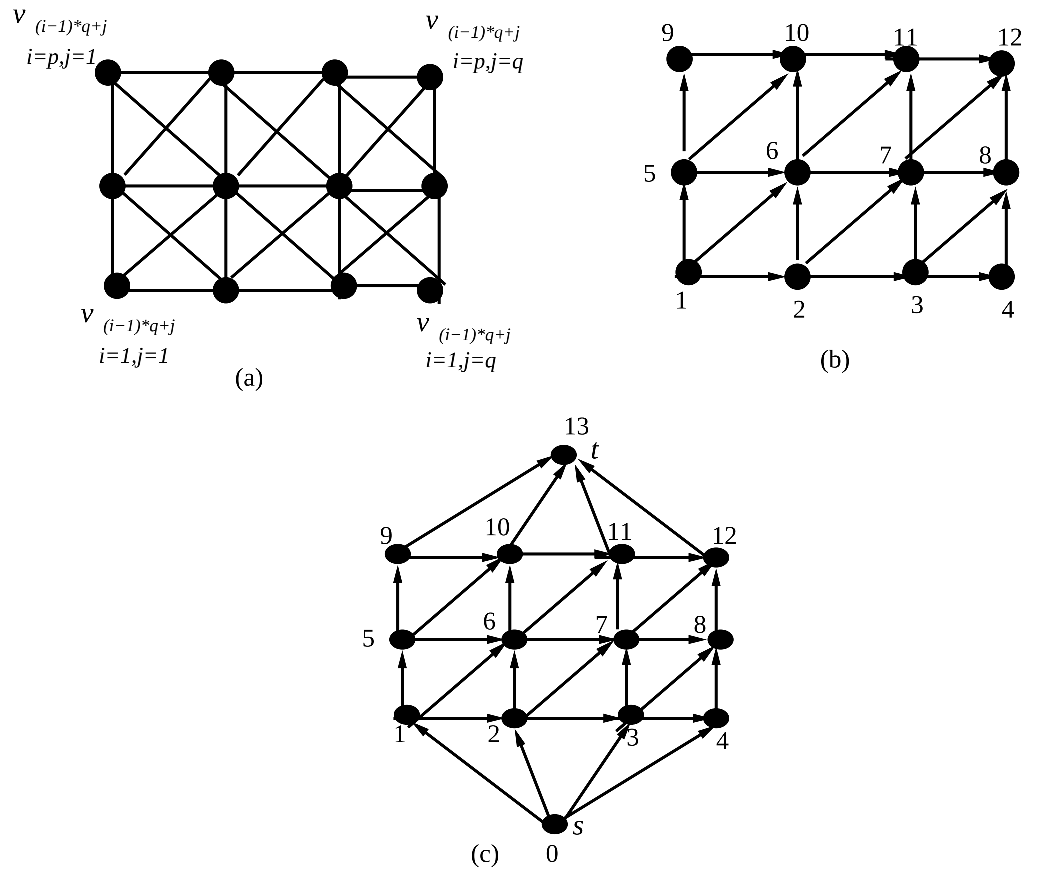


Figure 3.8: (a) Diagonal grid graph $G_{p,q}$, (b) numbering of vertices and (c) the augmented graph.

After generating the number of vertices in $p \times q$-grid graph, we also use this number in each vertices of diagonal grid graph $G_{p,q}$. Then delete one diagonal edge from each cell, we can convert to planar graph and orient the edges from lower numbered vertices to higher numbered vertices as in Figure 3.8(b). After that, we add source vertex s and sink vertex t and connect the edges to bottom vertices and top vertices of diagonal grid graph $G_{p,q}$ respectively. By orienting the edges from s to t, we get a planar st-digraph. This resulting st-digraph is called the augmented graph as in Figure 3.8(c).

We have the following lemma,

Lemma 3.3.1 *The augmented graph is a planar st-graph.*

Proof. Let $G_{p,q}$ be a diagonal grid graph. By deleting diagonal edges from this graph, we get $p \times q$-grid graph. This is a planar graph. After adding two vertices as source s and sink t and connect with top and bottom vertices of the $p \times q$-grid graph respectively, we can use special type of st-numbering for each vertex $v_{i,j}$ of the graph. Then if we direct the edges from s to t and assign number 0 to s and $n+1$ to t where n is the number of (p,q) vertex, we can get a directed graph which is called an augmented graph in Figure 3.8(c). In this graph, there is no edge crossing and we construct a digraph by orienting every edge from the lowest numbered vertex to the highest one. This graph is acyclic and has exactly one source s and one sink t. So we can say, the augmented graph is planar st-graph. $\mathcal{Q.E.D.}$

3.3.2 Bar 1-Visibility of Diagonal Grid Graphs

We present some definitions related to drawing algorithms.

Let $G_{p,q}$ is a diagonal grid graph. It consists some cells having diagonal edges. If we consider each cell then we can define its vertices and diagonal edges. In the Figure 3.9, *abcd* is a cell of a diagonal grid graph where a is the *Bottom-left* vertex; b is the *Bottom-right* vertex; c is the *up-right* vertex; d is the *up-left* vertex.

The edge between *Bottom-left* and *up-right* vertex in a cell is called the *Right diagonal edge*. The edge between *Bottom-right* and *up-left* vertex in a cell is called the *left diagonal edge*. In the Figure 3.9, ac is the *Right diagonal edge* and bd is the *left diagonal edge*.

We have the following theorem:

Theorem 3.3.2 *A diagonal grid graph $G_{p,q}$ admits a bar 1-visibility representation.*

Proof. We prove constructively that diagonal grid graph has bar 1-visibility representation. At first construct planar graph G' from diagonal grid graph $G_{p,q}$ by deleting left diagonal edge from each cell. After converting G' to an augmented graph, by Lemma 3.3.1, G' is a planar st-graph. By using algorithm given by Tamassia and Tollis, we can get visibility representation of the graph G'. Then delete the vertex segment of source s and sink t vertices and edges

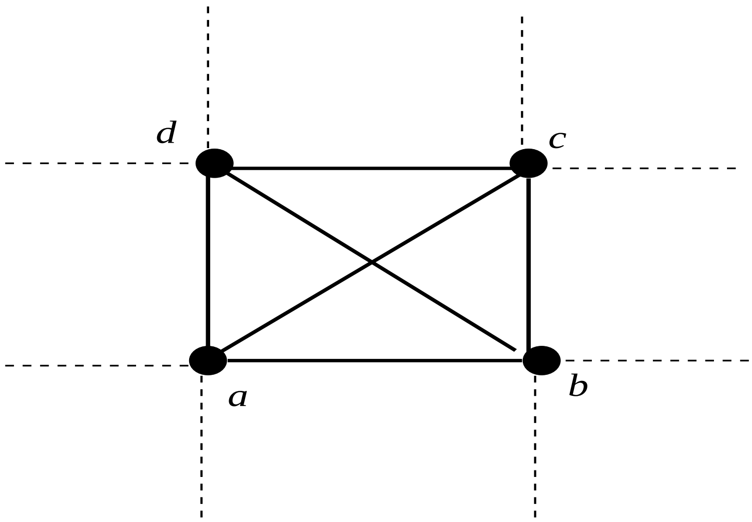


Figure 3.9: One cell of a diagonal grid graph.

which are adjacent to these vertices. In this visibility representation, we insert one column between every two grid in x-coordinate.

After that we extend each horizontal bar three right position at x-coordinate except horizontal segment corresponding to bottom and right vertices of diagonal grid graph. By extending these bars, the right diagonal edge in each cell crosses horizontal bar corresponding to up-left vertex in the representation. Then we extend horizontal bar corresponding to bottom-right vertex in each cell one left position. We can place the deleted left diagonal edges in the vertical segment which will be placed between starting point of the horizontal bar corresponding to bottom-right vertex and end point of the extended horizontal segment corresponding to up-left vertex in each cell. Since all edges including left diagonal edges can be placed at end point and startng point of the horizontal bars then only right diagonal edges always pass through one horizontal bar corresponding to the vertex. This maintains the property of bar 1-visibility representation. $\mathcal{Q}.\mathcal{E}.\mathcal{D}.$

The constructive proof of the theorem gives an algorithm. From the construction of the algorithm we can observe that any two vertex-segments are separated by a horizontal or vertical strip of at least unit width. In the representation constructed by the algorithm, we can observe that some edges have crossed one bar which maintains the properties of bar 1-visibility representations.

3.3.3 The Algorithm

We now formally present the algorithm for finding the bar 1-visibility representation of diagonal grid graphs. To construct bar 1-visibility representation we describe the shifting procedure of horizontal bar as follows:

We extend each horizontal bar three right position at x-coordinate except horizontal segment corresponding to bottom and right vertices of diagonal grid graph. By extendind these bars, the right diagonal edge in each cell crosses horizontal bar corresponding to up-left vertex in the representation. Then we extend horizontal bar corresponding to bottom-right vertex in each cell one left position. We can place the deleted left diagonal edges in the vertical segment which will be placed between starting point of the horizontal bar corresponding to bottom-right vertex and end point of the extended horizontal segment corresponding to up-left vertex in each cell. We call the procedure described above **Procedure Shift**.

Algorithm BAR 1-VISIBILITY

Input : A diagonal grid graph $G = (V, E)$.

Output: A bar 1-visibility representation for G such that each vertex-segment and edge-segment has endpoints with integer coordinates.

1 Construct planar graph G' from diagonal grid graph by deleting left diagonal edge from each cell.

2 Convert G' to an augmented graph.

6 Construct dual graph G^* of augmented graph G'.

7 Use steps 3 to 5 of **Algorithm VISIBILITY** [25]

8 Delete vertex-segment of source s and sink t vertices and edges which are adjacent to these vertices.

9 Insert one column between every two grid in x-coordinate.

10 Use **Procedure Shift**

We now give an illustrative example of this algorithm. In Figure 3.10 (a) shows a diagonal grid graph $G_{p,q}$, (b) shows planar graph G' by deleting left diagonal edge from each grid cell and (c) shows G' is an augmented graph. This numbering provides the y-coordinates for the visibility representation. In the Figure 3.11, we have the dual G^* of G' and its optimal topological numbering. This numbering provides the x-coordinates for the visibility representation. The Figure 3.12 shows visibility representation of the graph G' constructed by the algorithm given by Tamassia and Tollis [25]. In the visibility representation, after inserting one column between every two grid in x-coordinate, if we extend the bar representing vertices then we can place the deleted diagonal edges. Then one diagonal intersects one bar which maintains the properties of bar 1-visibility representation. The Figure 3.13 shows bar 1-visibility representation of the diagonal grid graph $G_{p,q}$ constructed by the algorithm.

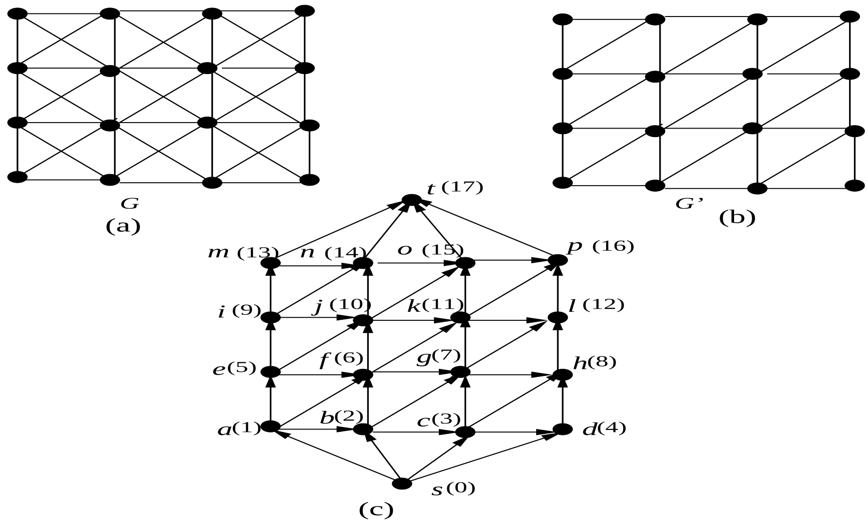


Figure 3.10: (a) A diagonal grid graph $G_{p,q}$, (b) planar graph G' and (c) G' is an augmented graph.

3.3.4 Area Compaction

To reduce the fabrication cost in VLSI design and also to visualize information in small area, area compaction of a drawing of a graph is a useful technique. To campact the representation area of bar 1-visibility representation, we use special type of topological numbering considering the vertices of each cell contains different number instead of numbering of the graph G'. In this numbering, we use the number of bottom-right in second cell at right position of G' for the up-left vertex in first cell. The number of the rest of the vertices in this row will be increased by one. The upper vertices will be numbered by same process.

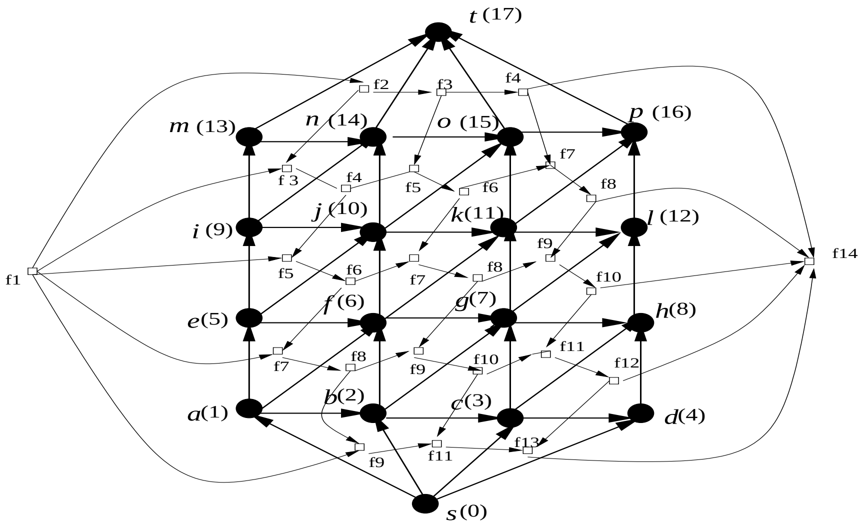
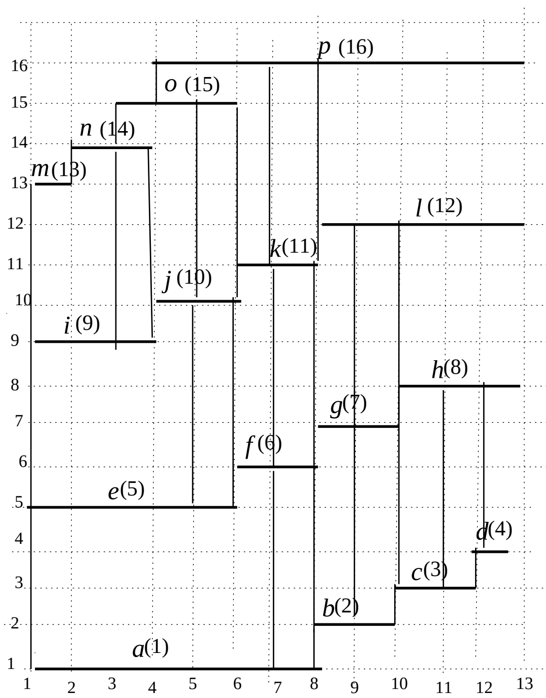


Figure 3.11: Dual graph $G^*$and the numberings of G^*.



G'

Figure 3.12: Visibility representation of graph G' constructed by the algorithm.

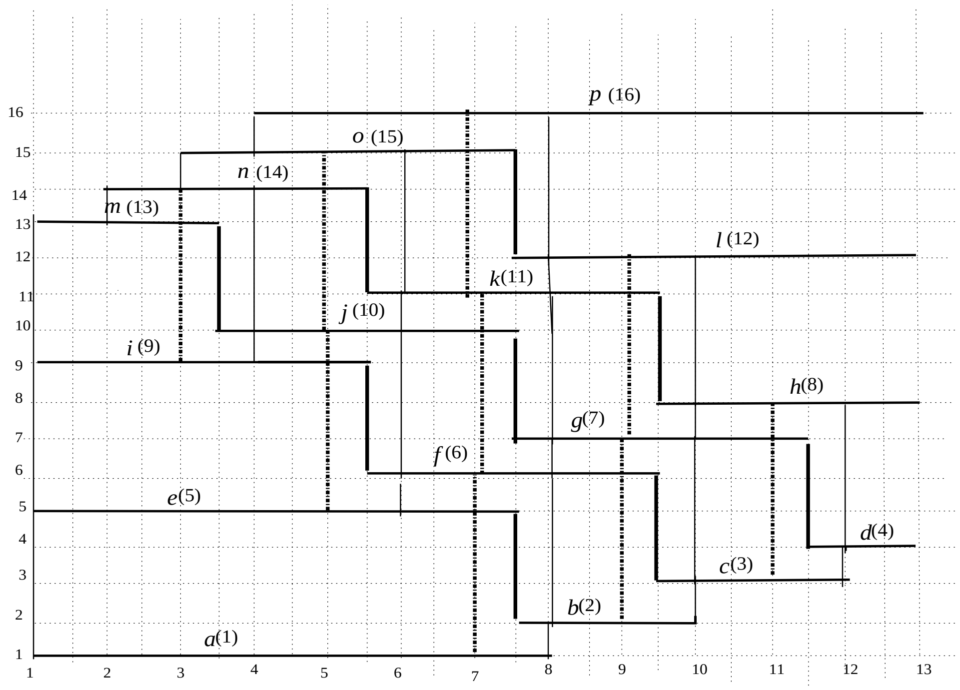


Figure 3.13: Bar 1-Visibility representation of diagonal grid graph $G_{p,q}$ constructed by the algorithm.

This labeling provides the y-coordinates. for the visibility representation. If we use this labeling then the height of the visibility representation is reduced from $p \times q$ to $q + 2p - 2$. In the Figure 3.14, (a) shows planar graph G' and (b) shows numbering of graph G'. After numbering the vertices of G', we can construct compact bar 1-visibility representation. The Figure 3.15 shows compact bar 1-visibility representation of diagonal grid graph $G_{p,q}$ constructed by the algorithm. We have the following theorem:

Theorem 3.3.3 *A diagonal grid graph $G_{p,q}$ has bar 1-visibility representation with height $q + 2p - 2$.*

Proof. A diagonal grid graph $G_{p,q}$ has bar 1-visibility representation. The fact that the special type of topological numbering considering the vertices of each cell contains different number of G' is obtainable in linear time. Let p and q represents the y-coordinates and x-coordinates in the grid respectively. If we use numbering of grid graph then the height of the output visibility representation by our algorithm is the number of vertex (p, q) which is equal to $p \times q$. When we compute special type of topological numbering then we can give the sequence

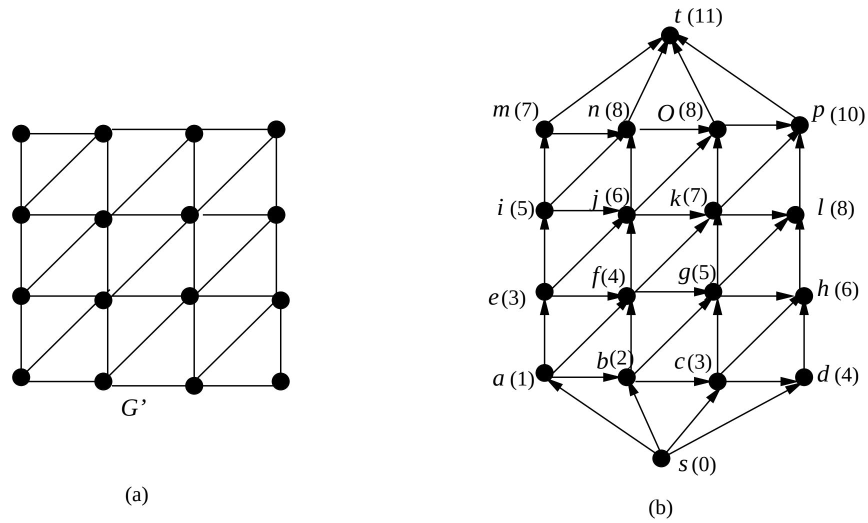


Figure 3.14: (a)A planar graph G' and (b) the numbering of G'.

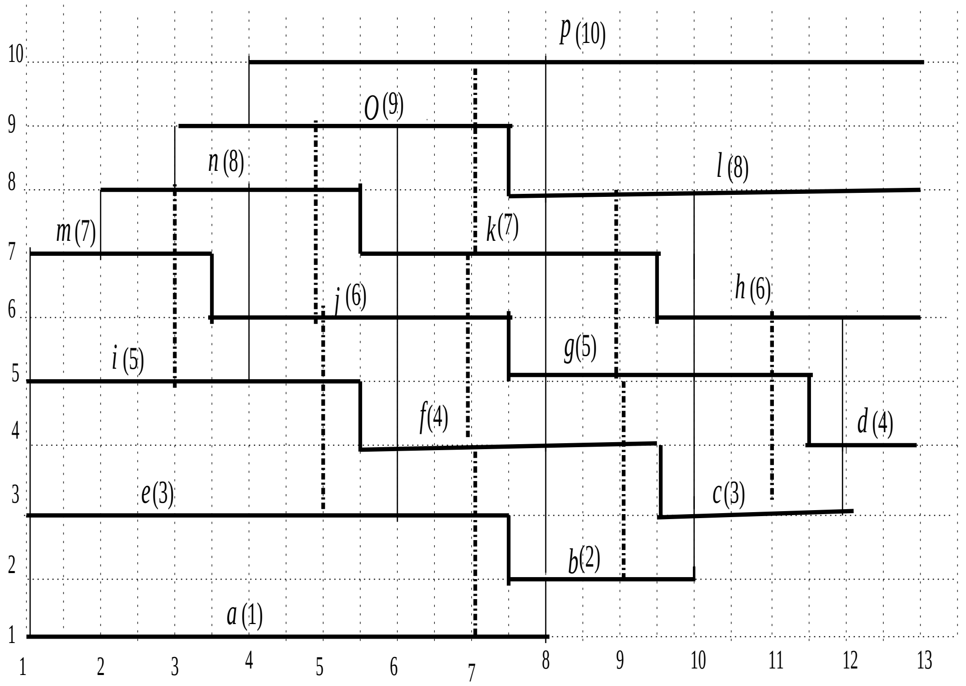


Figure 3.15: Compact bar 1-visibility representation of diagonal grid graph $G_{p,q}$.

$1, 3, 5 \cdots, n$ in the y-coordinate when x-coordinates is 1. Consequently when x-coordinates is q ,the sequence will be $q, q+2, q+4, q+6 \cdots, q+p+2$ in the y-coordinate . The sequence of x-coordinate will be 1,2,3,4 at $p = 1$; 3,4,5,6 at $p = 3$; 5,6,7,8 at $p = 5$; $\cdots$; $p, p+1, p+2 \cdots, p+q-1$ at $p = q$. From these sequences we can get the highest value $q + 2p - 2$ of the vertex (p, q) which gives the height of the output visibility representation by our algorithm . After inserting one column between every two grid in x-coordinate and using **Procedure Shift**, we can place deleted left diagonal edges of each cell of the diagonal grid graph $G_{p,q}$. In this way we can reduce the height of the visibility representation from $p \times q$ to $q + 2p - 2$. We can construct compact visibility representation.

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

3.3.5 Complexity of the Algorithm

The drawing we have proposed in this Algorithm to represent bar 1-visibility of a diagonal grid graph can be constructed in linear time. We have the following theorem:

Theorem 3.3.4 *Let $G_{p,q}$ be a diagonal grid graph. Then a bar 1-visibility representation of $G_{p,q}$ can be computed in linear time.*

Proof. Let $G_{p,q}$ be a diagonal grid graph. In step 1 and 2, we construct planar graph from diagonal grid graph by deleting left diagonal edge from each cell and adding two vertices souce and sink. After that we can make the augmented graph in linear time. Then we construct dual graph of the planar graph and compute optimal topological numbering of the dual graph. The visibility representation of the planar graph can be constructed in linear time. Then inserting column between each two grid in x-coordinate and shifting the vertex-segment corresponding vertices at x-coordinate, the deleted diagonal edges are placed in the representation in linear time. One vertical segment corresponding diagonal edge intersects one horizontal segment which maintains the properties of bar 1-visibility representation. Thus we can obtain a bar 1-visibility representation of diagonal grid graph in linear time. $\mathcal{Q}.\mathcal{E}.\mathcal{D}.$

3.4 Conclusion

In this chapter, we have explained important algorithm on visibility representation of planar graphs. The approach of this algorithm are very often followed by the researchers in this field. In addition, we have introduced a new class of graphs diagonal grid graphs as bar 1-visibility graphs. Finally, we have designed an algorithm to find bar 1-visibility representations of diagonal grid graphs. We have also modified the algorithm for finding visibility representations on a compact area.

Chapter 4

Diagonal Labeled Graphs

4.1 Introduction

In this chapter, we present, in brief, previously known algorithms on constrained visibility representation of planar graphs. Attentive readers should recall that in a *visibility representation* of a planar graph G, each vertex is drawn as a horizontal line segment and each edge is drawn as a vertical line segment where the vertical line segment representing an edge must connect the horizontal line segments representing the end vertices. In a *bar 1-visibility representation*, line segment corresponding to an edge intersects at most one bar which is not an end point of the edge [3]. In this chapter, we develop an algorithm for finding a bar 1-visibility representation of a newly introduced class of graphs called diagonal labeled graphs.

The rest of the chapter is organized as follows. We start with a brief review of some preliminary concepts of constrained visibility representation of planar graphs in Section 4.2. In Section 4.3, we cast a bird's eye view on the origin and development of constrained visibility representation of planar graphs. The rest of the sections are devoted to important algorithm in this field. In Section 4.4, we presents the constrained visibility representation of planar graphs and Section 4.5 depicts an algorithm for bar 1-visibility representations of diagonal labeled graph. Finally, we summarizes the result in Section 4.6.

4.2 Preliminaries

Most of the preliminary graph theoretic concepts have been covered in Chapter 2. However, for the sake continuity and clear conception, we will now recall some terminologies of constrained visibility representation of planar graphs and diagonal labeled graphs.

In some cases there is a need to give more emphasis to certain paths of a planar graph. These paths often called critical, should discriminate from all the other paths, on the visibility representation of the planar graph. Such a visibility representation could be used, for example, for having a quick inspection of critical paths on a workflow graph. Critical paths in that case they have the potential of delaying a project, thus they should be emphasized in a visibility representation. A way to emphasize certain paths in a visibility representation is to align their edges to the same horizontal coordinate. Such a visibility representation is called constrained visibility representation. This representation can be used as a starting point for obtaining orthogonal and polyline drawings with interesting properties

A *diagonal labeling* of a graph G is defined as an orientation if the graph G contains each edge crossing bounded by a quadrangle and vertices of lowest number and highest number are on a diagonal.

A graph which contains diagonal labeling is called *diagonal labeled graph*. For example, Figure 4.1(a) shows an input planar graph G and Figure 4.1(b) shows a diagonal labeling of G where the vertices 0 and 3, 2 and 6 , 0 and 4, 3 and 6 are placed along the diagonal of each quadrangle.

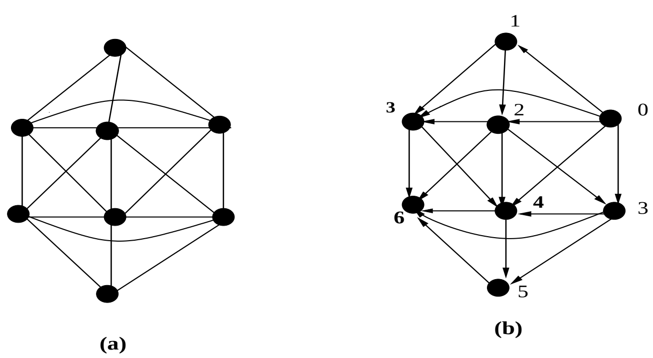


Figure 4.1: (a) A graph G and (b) a diagonal labeling of G.

If the input graph has a diagonal labeling and contains each edge crossing inside each quadrangle then bar 1-visibility representation of that graph is also found. At first, one diagonal which contains lowest and highest number in each quadrangle will be chosen. Then the planar graph is obtained by passing that diagonal within one vertex of the quadrangle and a visibility representation of the resulting graph will be obtained by using the technique of constrained visibility representation of a planar graph [5]. The edges which pass within the vertex will be drawn by keeping the properties of 1-visibility representation.

4.3 Historical Background

In this section, we will briefly review the origin of constrained visibility representation of planar graphs.

The idea of constrained visibility representation of planar graphs can be traced back earlier to work by Battista, Tamassia and Tollis [5]. They proposed such an algorithm, based on the algorithm of the visibility representation. Algorithm Constrained Visibility computes in $O(n)$ time a visibility representation of G with integer coordinates and $O(n^2)$ area, such that the edges of every path π in Π are vertically aligned.

4.4 Constrained Visibility Representation

In this section, we will review the algorithm of constrained visibility representation of planar graphs from [5].

It is easy to construct such an algorithm, based on the algorithm of the visibility representation. Let G be a planar st-graph with n vertices. The key idea is to construct a new planar st-graph G' that has an extra facet for each critical path. This can be done by duplicating each critical path. The visibility representation of that graph will have the edge segments of the left side of the boundary of each extra facet, vertically aligned. By removing the right copy of every edge of the duplicated path, and joining the copies of the duplicated vertices, we have each critical path aligned to one x-coordinate. The new facet for every critical path can be inserted directly to the dual of G, as a new vertex, at it will be shown, after the following definitions [5]:

Definition 4.4.1 *Two paths p_1 and p_2 of a planar st-graph G, are said to be non intersecting, if they do not share any edge, and do not cross at common vertices.*

This means that if the two paths p_1 and p_2 have a common vertice, which is the case in Figure 4.2, then two consecutive vertices, in the clockwise or anticlockwise order, must belong to the same path. For example in Figure 4.2, two paths are intersecting if e_1 and e_3 belong to the same path, while e_2 and e_4 belong to another.

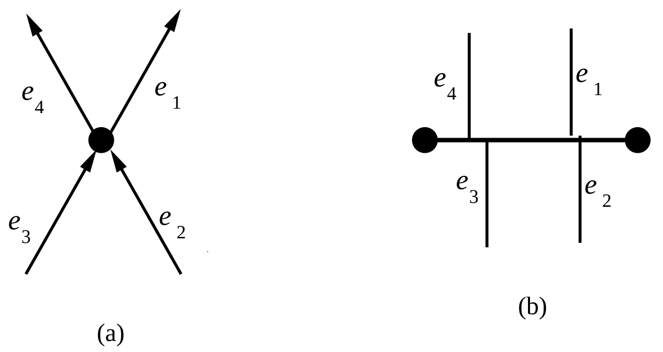


Figure 4.2: Two paths meet on an edge and its visibility representations.

Let G be a planar st-graph with n vertices. Two paths π_1 and π_2 of G are said to be non-intersecting if they are edge disjoint and do not cross at common vertices, i.e., there is no vertex v of G with edges e_1, e_2, e_3 and e_4 incident in this clockwise order around v, such that e_1 and e_3 are in π_1 and e_2 and e_4 are in π_2. Observe that any two vertex disjoint paths are also non-intersecting.

Given a collection Π of *non-intersecting paths* of G, we consider the problem of constructing a visibility representation of G such that for every path π of Π, the edges of π are vertically aligned. More formally, for any two edges e' and e'' of π the edge-segments $\Gamma(e')$ and $\Gamma(e'')$ have the same x-coordinate. Algorithm (constrained-visibility) takes as input G and Π, and constructs a constrained visibility representation Γ of G. In order to simplify the description of the algorithm, without loss of generality, we assume that the set Π of non-intersecting paths covers the edges of G. Otherwise, each edge originally not in Π, is inserted in Π as an independent path.

Algorithm Constrained-Visibility [5].

Input : Planar st-graph G with n vertices; set Π of non-intersecting paths covering the edges of G.
Output: A visibility representation Γ for G such that each vertex-segment and edge-segment has endpoints with integer coordinates and area $O(n^2)$.

1 Assign unit weights to the the edges of G and compute an optimal topological numbering Y of G such that $Y(s) = 0$.

2 Construct the graph G_π with vertex set $F \cup \Pi$ (recall that F is the set of faces of G and edge set $(f, \pi) \mid f = left(e)$ for some edge e of path $\pi \cup (\pi, g) \mid g = right(e)$ for some edge e of path π). Note that graph G_π is a planar st-graph.

3 Assign half-unit weights to the edges of G_π and compute an optimal topological numbering X of G_π such that $X(s^*) = -\frac{1}{2}$.

4 for each Path π in Π
for each edge e in π
draw $\Gamma(e)$ as the vertical segment with
$x(\Gamma(e)) = X(\pi)$;
$y_B(\Gamma(e)) = Y(orig(e))$;
$y_T(\Gamma(e)) = Y(dest(e))$;

5 for each Vertex v
draw $\Gamma(v)$ as the horizontal segment with
$y(\Gamma(v)) = Y(v)$;
$x_L(\Gamma(v)) = min_{v \in \pi} X(\pi)$;
$x_R(\Gamma(v)) = max_{v \in \pi} X(\pi)$;

To help the intuition of the reader, we observe that the computations performed by the algorithm are equivalent to the following construction. First, it modifies G by duplicating each path π in Π thus forming a new face for each path. This is equivalent to having a vertex of G_π (defined in Step 1) correspond to each path in Π. Second, it constructs a visibility representation for the modified graph such that the edge-segments of the left side of the boundary of each face are vertically aligned and two copies of an original vertex are horizontally aligned. Finally, it removes the right copy of every duplicated edge and joins the copies of the duplicated vertices. In Figure 4.3(a) shows planar st-graph G, topological numbering of G, and set Π of paths that cover the edges of G, where the paths with at least two edges are drawn with thick lines; 4.3(b) shows graph G_π and its topological numbering, where the square vertices represent faces of G and the pentagon vertices represent paths of Π. and Figure 4.4 shows constrained visibility representation of G.

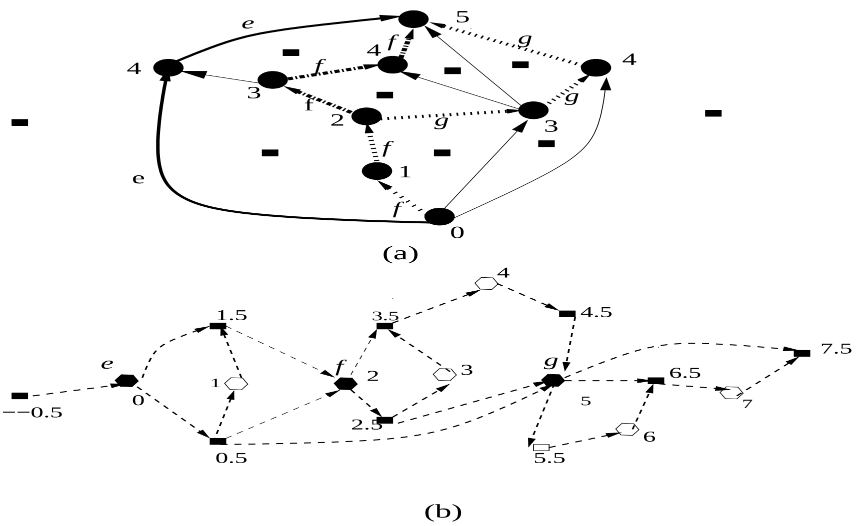


Figure 4.3: (a) Planar st-graph G and a topological numbering of G and (b) graph G_π and its topological numbering.

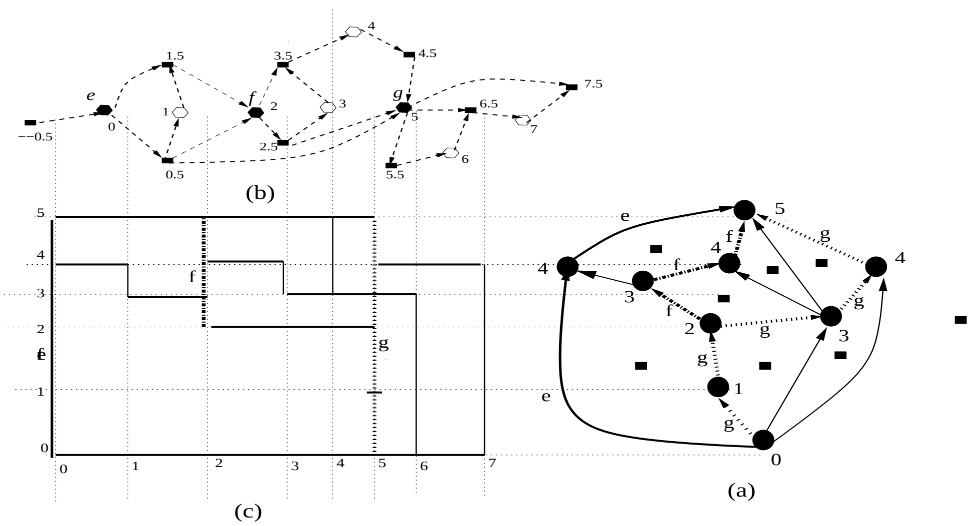


Figure 4.4: (a) A topological numbering of G, (b) graph G_π and its topological numbering and (c) constrained visibility representation of G.

Theorem 4.4.2 *[5] Let G be a planar st-graph with n vertices, and Π be a set of non-intersecting paths covering the edges of G. Algorithm constrained visibility computes in $O(n)$ time a visibility representation of G with integer coordinates and $O(n^2)$ area, such that the edges of every path π in Π are vertically aligned..*

4.5 Bar 1-Visibility Representations

In this section we give an algorithm for obtaining bar 1-Visibility Representations of diagonal labeled graph. This problem has an interesting correlation with the constrained visibility representations of planar *st*-graph. If the input graph has a diagonal labeling and contains each edge crossing inside each quadrangle then bar 1-visibility representation of that graph is also found. At first, one diagonal which contains lowest and highest number in each quadrangle will be chosen. Then the planar graph is obtained by passing that diagonal within one vertex of the quadrangle and a visibility representation of the resulting graph will be obtained by using the technique of constrained visibility representation of a planar graph [5]. The edges which pass within the vertex will be drawn by keeping the properties of 1-visibility representation. We have the following theorem:

Theorem 4.5.1 *The diagonal labeled graphs admit a bar 1-visibility representation.*

Proof. We prove constructively that the diagonal labeled graph has bar 1-visibility representation. A *diagonal labeling* of a graph G is an orientation if the graph G contains each edge crossing bounded by a quadrangle and vertices of lowest number and highest number are on a diagonal. At first we identify the diagonal edge which contains highest number and lowest number in each quadrangle. Then we convert the non-planar graph to planar graph G' by passing the diagonal edges through the vertex. After that we can Identify non intersecting paths π in Π in the graph G'. The diagonal edges which are passed through the vertex are identified as non intersecting paths. Because these satisfies the conditions which was described in constrained visibility representation [5]. The paths have no common edges and no crossing because if we pass all the diagonal edges through the vertex which is right side of the edge. But they can touch at vertices. These conditions are easily understood by the Figure 4.5.

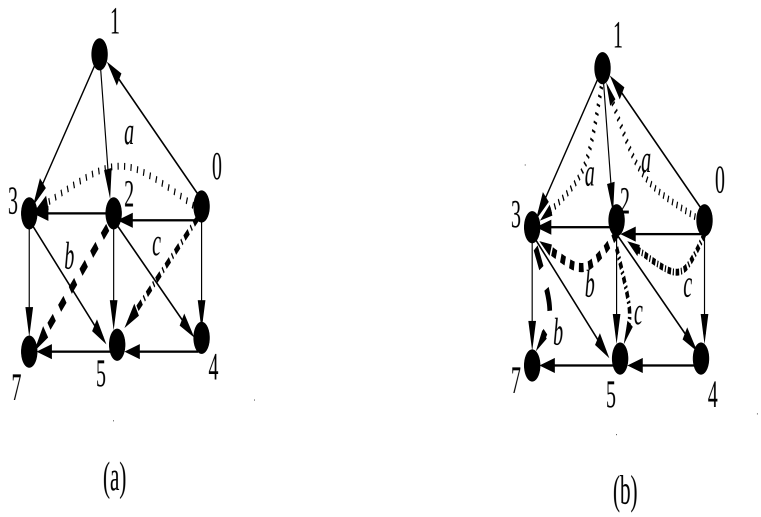


Figure 4.5: Non intersecting paths are a,b,c.

Then we assign half-unit weights to the edges of G_π and compute an optimal topological numbering X of G_π such that $X(s^*) = -\frac{1}{2}$. After that we can place each vertex and each edges by the algorithm of constrained visibility representation [5]. After placing each edges, we can see the diagonal edges which is identified as non intersecting path cross one vertex-segment which maintains the properties of bar 1-visibility representation. Thus we can prove the diagonal labeled graph admits a bar 1-visibility representation. $\mathcal{Q.E.D.}$

The constructive proof of the theorem gives an algorithm. From the construction of the algorithm, we can observe that any two vertex-segments are separated by a horizontal or vertical strip of at least unit width. In the representation constructed by the algorithm, we can observe that some edges have crossed one bar which maintains the properties of bar 1-visibility representations.

4.5.1 The Algorithm

We now formally present the algorithm for finding the bar 1-visibility representation of diagonal labeled graphs.

Algorithm BAR 1-VISIBILITY

Input : A diagonal labeled graph $G = (V, E)$.

Output: A bar 1-visibility representation Γ for G such that each vertex-segment and edge- segment has endpoints with integer coordinates.

1 In graph G, choose the diagonal edges which contain highest number and lowest number in each quadrangle.

2 Construct planar graph G' by passing the diagonal edges through the vertex.

3 Identify non intersecting paths π in Π in the graph in such a way–
- there is no common edges
- no crossings
- can touch at vertices.

5 Use **Algorithm Constrained-Visibility** [5]

We will make an example of this algorithm. In Figure 4.6 (a) Diagonal labeled graph G, (b) shows planar graph G' by passing the diagonal edges through the vertex and (c) shows optimal topological numbering of G_π In the Figure 4.7, shows bar 1-visibility representation Γ for G constructed by the algorithm .

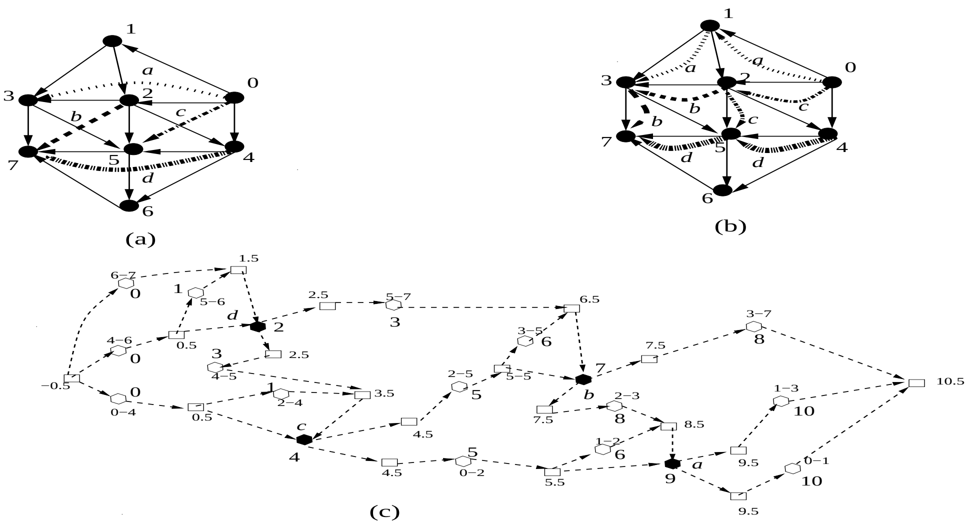


Figure 4.6: (a) Diagonal labeled graph G, (b) planar graph G' and (c) the numberings of G_π.

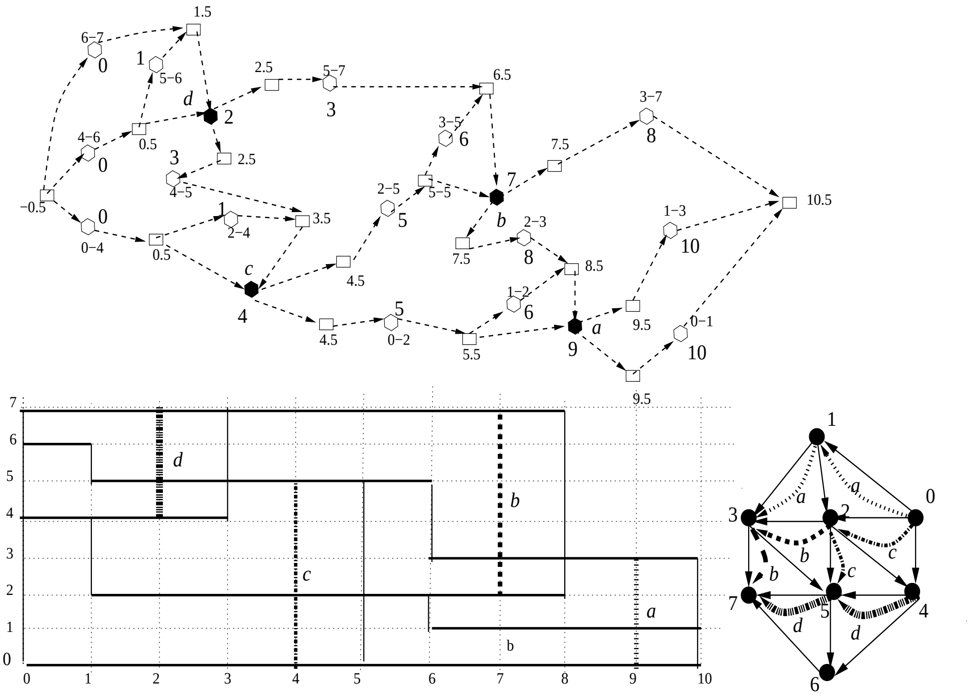


Figure 4.7: Bar 1-visibility representation Γ for G constructed by the algorithm.

4.5.2 Complexity of the Algorithm

The drawing we have proposed in this algorithm to represent bar 1-visibility of a diagonal labeled graph can be constructed in linear time. We have the following theorem:

Theorem 4.5.2 *Let G be a diagonal labeled Graph. Then a bar 1-visibility representation of G can be computed in linear time.*

Proof. Let G be a diagonal labeled graph. In step 1 and 2, we construct planar graph from diagonal labeled graph by passing the diagonal edges through the vertex and identify non intersecting paths π in Π in the graph in linear time. After that we Assign half-unit weights to the edges of G_π and compute an optimal topological numbering X of G_π such that $X(s^*) = -\frac{1}{2}$. It can be also done in linear time. The visibility representation of this planar graph can be constructed in linear time. After placing each edges, we can see the diagonal edges which is identified as non intersecting path cross one vertex-segment which maintains the properties of bar 1-visibility representation. Thus we can obtain a bar 1-visibility representation of diagonal labeled graph in linear time. $\mathcal{Q.E.D.}$

4.6 Conclusion

In this chapter, we have reviewed important algorithm on constrained visibility representations of planar graphs. The approach of this algorithm are very often followed by the researchers in this field. In addition, we have introduced a new class of graphs called diagonal labeled graphs as bar 1-visibility graphs. Finally, we have designed an algorithm to find a bar 1-visibility representation of a diagonal labeled graph in linear time.

Chapter 5

Conclusion

In this concluding chapter, we would like to review the concepts discussed in the earlier sections. Also, we give a summary of the remaining open problems in this field.

In this thesis, we have regarded the problem of computing algorithms on bar 1-visibility representations of non-planar graph such as diagonal grid graph and diagonal labeled graph. Given an undirected diagonal grid graph G, this thesis gives a drawing algorithm that gives bar 1-visibility representations with minimum area in linear time. In this thesis, we have addressed the problem of drawing visibility representation of non-planar graph. We have proved that some classes of non-planar graphs admit bar 1-visibility representation. Consequently, we focused on those non-planar graph which contains at most one edge crossing. If given the directed graphs having diagonal labeling where each edge crossing bounded by a quadrangle and vertices of lowest number and highest number are on a diagonal then this thesis also gives a linear time drawing algorithm for computing bar 1-visibility representation.

In Chapter 1, we presented a brief overview of the basic graph theoretic and drawing concepts. We discussed some of preliminaries of visibility representation of planar graph and further defined briefly bar visibility graph. Then we came into the point of bar k-visibility graph and consequently we discussed about bar 1-visibility graph and bar 1-visibility representation. Finally we regard the non-planar graphs as our problem domain and specify the problem of computing the bar 1-visibility representation.

In Chapter 2, we give some preliminary ideas on graph theory and algorithmic theory on st-numbering and optimal topological numbering. The chapter includes some concepts that may be outside of the visibility representation zone but helped in our research in this area.

Chapter 3 presents description of visibility representations of planar graph and our algorithm on bar 1-visibility representations of diagonal grid graph which is non-planar graph. This chapter gives the details of the previous results, including the related proofs and descriptions. Consequently, we focus on finding the area compaction of bar 1-visibility representation. The algorithm is linear-time algorithm that has been proved.

Finally, in Chapter 4, we present description of constrained visibility representations of planar graph . We have detailed our results on bar 1-visibility representations of diagonal labeled graph. Consequently, we focus on finding non planar graph admits bar 1-visibility representation. We illustrate in this chapter our linear-time drawing algorithm for bar 1-visibility representations of diagonal labeled graph.

There are some limitations of this work. Diagonal labeled graph has diagonal labeling. But in our work we can not generate diagonal labeling of the graph. We can not check whether all 1-planar graphs are bar 1-visibility graphs or not. Now we discuss some of the related open problems in this field.

- We introduce diagonal grid graphs and diagonal labeled graphs as bar 1-visibility graphs. Recently 1-planar graph has been studied in various ways [7, 6, 24]. So it would be interesting to characterize the graphs which admit bar 1-visibility representation.
- In [3], bar k-visibility representations have been introduced. Are there any drawing algorithm of bar k-visibility representation where $k = 1, 2, \cdots, k$?
- We have mentioned that the diagonal labeled graph admits bar 1-visibility representation. Here diagonal labeling of a graph G is given as an orientation if the graph G contains each edge crossing bounded by a quadrangle and vertices of lowest number and highest number are on a diagonal. So it would be interesting to investigate how diagonal labeling can be done.

References

[1] M. Chrobak and S. Nakano. Minimum-width Grid Drawings of Plane Graphs. *International Journal of Computational Geometry and Applications*, 10:29–54, 1998.

[2] F. J. Cobos, J. C. Dana, F. Hurtado, A. Marquez, and F. Mateos. On a Visibility Representation of Graphs. In *The Proceedings of Graph Drawing 95*, volume 1027 of *Lecture Notes in Computer Science*, pages 152–161. Springer-Verlag, 1995.

[3] A. M. Dean, W. Evans, E. Gethner, J. D. Laison, and M. Safari. Bar k-Visibility Graphs. *Journal of Graph Algorithms and Applications*, 11(1):45–59, 2007.

[4] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[5] G. Di Battista, R. Tamassia, and I. G. Tollis. Constrained Visibility Representation of Graphs. *Information Processing Letters*, 41(1):1–7, 1992.

[6] P. Eades and G. Liotta. Right Angle Crossing Graphs and 1-planarity. In *Appeared to GD 2011 proceedings of the symposium on Graph Drawing*, 2011.

[7] I. Fabrici and T. Madaras. The Structure of 1-Planar Graphs. *Journal on Discrete Mathematics*, 307(1):854–865, 2007.

[8] I. Fary. On Straight Line Representations of Planar Graphs. *Acta Sci. Math. Szeged.*, 11:229–233, 1948.

[9] S. Fleshner and M. Massow. Thickness of Bar 1-Visibility graphs. In *the Proceedings of Graph Drawing 06*, volume 4372 of *Lecture Notes in Computer Science*, pages 330–342. Springer-Verlag, 2007.

[10] S. Fleshner and M. Massow. Parameters of Bar k-Visibility Graphs. *Journal of Graph Algorithms and Applications*, 12(1):5–27, 2008.

[11] U. Fobmeier and M. Kaufmann G. Kant. 2-Visibility Drawing of Planar Graphs. In *GD '96 proceedings of the symposium on Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 155–168. Springer-Verlag, 1997.

[12] H. Fraysseix, J. Pach, and R. Pollack. How to Draw a Planar Graph on a Grid. *Combinatorica*, 10:41–51, 1990.

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

[14] P. Hutchinson. Arc and Circle Visibility Graphs. *Australasian J. Combinatorics*, 25:241–262, 2002.

[15] A. Lempel, S. Even, and I. Cederbaum. An Algorithm for Planarity Testing of Graphs. *P. Rosestiehl(ed.) Theory of Graphs: International Symposium July 1966*, pages 215–232, 1967.

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

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

[18] J. Otten and J. G. Van Wijk. Graph Representation in Interactive Layout Design. In *The Proceedings of IEEE Intternational Symposium On Circuits and Systems*, pages 914–918, 1978.

[19] B. Prosenjit, M. Dean, P. Hutchinson, and T. Shermar. On Rectangle Visibility Graphs. In *In Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 25–44. Springer-Verlag, 1997.

[20] 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 2 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 1997.

[21] W. Schnyder. Embedding Planar Graphs on the Grid. In *Proc. of the First ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148, 1990.

[22] P. Selvaraju and B. Pricilla. On Cordial Labeling: The Grid, Diagonal Grid, Structured Web Graphs. *International Journal of Algorithms, Computing and Mathematics*, 2(1):5–14, 2009.

[23] K. S. Stein. Convex Maps. *Proc. of American Mathematical Society*, 2(3):464–466, 1951.

[24] Y. Suzuki. Optimal 1-Planar Graphs which Triangulate Other Surfaces. *Journal on Discrete Mathematics*, 310(1):6–11, 2010.

[25] R. Tamassia and I. G Tollis. A Unified Approach to Visibility Representations of Planar Graphs. *Discrete and Computational geometry*, 1:321–341, 1986.

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

Index