Carleton University
School of Computer Science
95.508 Computational Geometry

Embedding Planar Graphs on the Grid

A Review of the Paper by Walter Schnyder

Review By: Jon Harris
Student Number: 267 309
Submitted: December 14, 2002

Table of Contents

List of Figures

1. Introduction

Any given graph (with vertices and edges connecting these vertices) can be drawn in multiple ways if we require only that vertex adjacencies be preserved. The given graph can be drawn with straight line or curved edges, and intersecting or non intersecting edges.

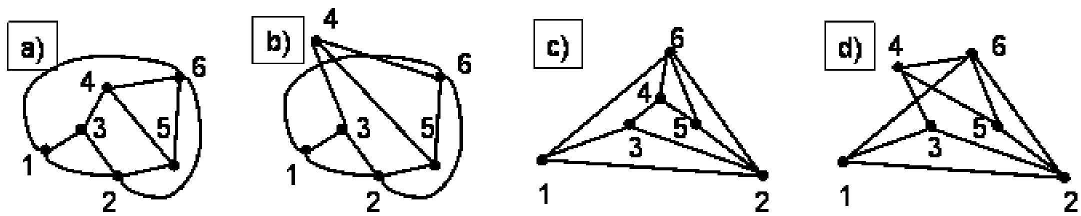

Figure 1 An illustration of different drawings of the same graph

Minimizing (or eliminating) the number of crossing edges in a drawing of a graph has important applications in the readability and visualization of graphs. As an example, consider VLSI[1] microchip layout and design. When designing a microchip, it is desirable to be able to visual its circuit path-ways in a clear manner (i.e. Minimize crossing edges).

A planar straight line embedding of a graph is its representation in a plane such that no two edges cross. Fary proved in 1948 that every planar graph has a straight line embedding [5]. We can divide the plane in which the planar graph is straight line embedded into cells, creating a grid structure. Smaller grid sizes are desirable for visualization purposes, especially when visual real estate is expensive (i.e. the area available on a computer screen).

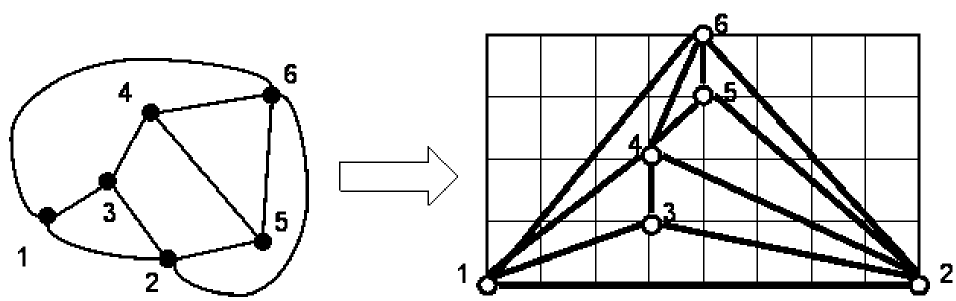

Figure 2 A Planar graph and its embedding on a grid

The purpose of Schnyder's paper is to give an algorithm that allows one to compute a straight line embedding of a planar graph in a grid. For a planar graph with n vertices, the algorithm will embed the graph in an n-2 by n-2 grid in linear time. This represents an improvement of the previously best known grid size of 2n-4 by n-2 achieved by the algorithm of De Fraysseix, Pach and Pollack [4].

[1] Very Large Scale Integration

2. Preliminaries

There are several requirements for the input graph to Schnyder's algorithm:

- The input graph must be **planar**. Several algorithms exist for testing a graph's planarity in time linear. Examples include the algorithm of Hopcroft and Tarjan [6], and also the PQ-Tree based algorithm of Booth and Lueker [1].

- An **embedding** must be provided of the input graph. More specifically, a *topological embedding* must be provided of the input graph such that the clockwise (counter-clockwise) order of the edges around each vertex must be available. Linear time algorithms exist for computing the topological embedding of a graph, such as the linear time algorithm of Chiba et al. which also uses PQ-Trees [2].

- A **triangulation** of the input graph is required. In other words, all faces of the input graph must have degree 3 (be bounded by 3 edges/vertices). Several linear time algorithms for triangulating a planar graph exist. Read provided a simple algorithm for triangulating a planar graph by checking all pairs of adjacent edges in clockwise (counter-clockwise) order around each vertex [7]. This algorithm was extended by de Fraysseix et al. to run in linear time (and space) [4].

- A **canonical ordering** of the vertices of the (triangulated) input graph is necessary to determine the order in which vertices are labeled during the *normal labeling* process of Schnyder's algorithm (normal labeling is described below).

  A *canonical ordering* of a maximal (triangulated) planar graph G embedded in the plane, with exterior vertices u, v, w is a labeling of all the vertices $v_1 = u$, $v_2 = v$, $v_3, \ldots v_{n-1}, v_n = w$ such that the following requirements are met:

  - The subgraph G_{k-1} of G induced by v_1, $v_2 = v$, $v_3, \ldots, v_{k-1}$ is bi-connected, and the boundary of its exterior face is a cycle C_{k-1} containing the edge {u,v}.
  - v_k is in the exterior face of G_{k-1}, and its neighbours in G_{k-1} form an (at least 2-element) subinterval of the path C_{k-1} – {u,v}.

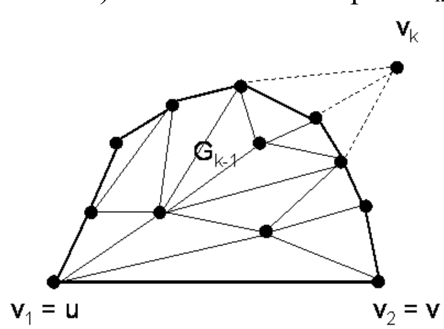

Figure 3 An Illustration of a Canonical Ordering

De Fraysseix et al. give a linear time algorithm for computing this ordering [4].

3. Schynder's Algorithm

As mentioned in the introduction, Schnyder's paper describes an algorithm for embedding planar graphs on a grid. The input graph G = (V, E) is assumed to satisfy all of the requirements described in the preliminaries section. Given that the input graph is planar, we have that |V| = |E| = O(n) by Euler's formula (|E| = 3|V|-6). In his paper, Schnyder describes an algorithm for obtaining the coordinates of every vertex v in V within an n-2 by n-2 grid.

The first phase of Schnyder's algorithm is the computation of a normal labeling of the input graph. A subsequent phase uses the normal labeling to obtain coordinates for each vertex (of the graph) on a grid. Both are summarized below.

3.1 Normal Labeling

The first phase of Schnyder's algorithm is the computation of a normal labeling of the triangular input graph.

3.1.1. Description

A *normal labeling* of a triangular Graph G is a labeling of all of the angles of its triangular faces such that the following conditions are satisfied:

a) For every triangle, there is one angle labeled with 1, second labeled with 2 and a third labeled with 3.
b) The angles of every triangle in counter-clockwise order are 1, 2 and then 3.
c) For every interior vertex, the labels of the angles of its adjacent triangles consist of (non-empty) intervals of 1's followed by 2's and then 3's.
d) All of the angles at an exterior vertex have the same label.
e) The three exterior vertices whose angles are all 1, 2, 3, appear in counter-clockwise in this order around the exterior triangle.

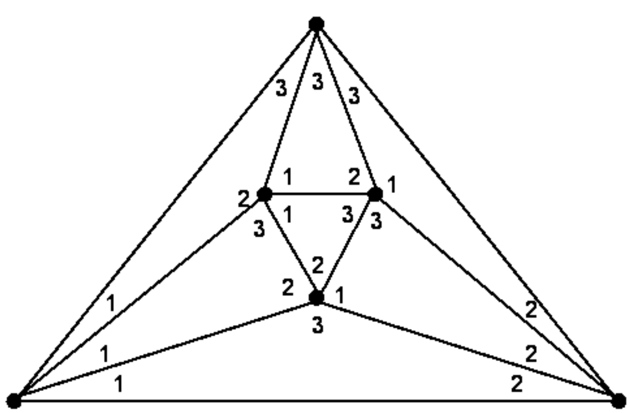

Figure 4 An illustration of the properties of a Normal Labeling

Given a triangular graph, a normal labeling can be defined as a labeling of each angle <(xy,xz) of the graph with a label k in {1, 2, 3} such that $x_k > y_k, z_k$. This labeling is guaranteed to be unique if we have a barycentric representation of the triangular graph (described below).

A *barycentric representation* of a Graph G is a one-to-one function on the vertices of G that maps each of the vertices v to three (barycentric) coordinates v_1, v_2, v_3 in R^3 which must exhibit the following properties for every vertex:

i) $v_1 + v_2 + v_3 = 1$
ii) For each edge {x, y} and each vertex z not in {x, y}, there is some k in {1, 2, 3} such that $x_k < z_k$ and $y_k < z_k$.

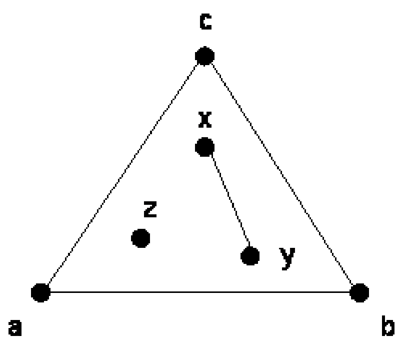

Figure 5 Convex combination

Conceptually, this means that for each vertex inside the exterior triangle bounded by vertices a, b and c, we can express their location within the exterior triangle as a *convex combination* of a, b and c.

For example, in the figure on the left we have $z = \frac{1}{2}a + \frac{1}{4}b + \frac{1}{4}c$ so (i) is satisfied with $z_1 = \frac{1}{2}$ and $z_2 = z_3 = \frac{1}{4}$.

Also, x and y are both further from a than z, so (ii) is satisfied (k = 1) for edge {x, y}.

Schnyder proves that all and only planar graphs have barycentric representations. He also proves that given a barycentric representation, then for any 3 non-collinear points a, b and c, the mapping of each vertex v in the graph to $v_1a + v_2b + v_3c$ is a straight line embedding of the graph in the plane spanned by a, b and c.

Another property of a normal labeling is that for every edge between two adjacent internal triangles, the four labels at its ends consist of a pair of angles labeled i, j (i != j) at one end, and a pair of angles both labeled k (k != i, j) at the other. (See figure 7 for some examples).

Schnyder uses this fact to introduce a labeling of the interior edges of the graph. Each interior edge is labeled with k, and directed towards the end where k appears twice.

The result of this edge labeling is three sets of edges with labels 1, 2 or 3 (referred to as T_1, T_2 and T_3 respectively). Schnyder refers to these three sets as a *realizer* of the triangular graph. The third property of normal labeling implies several properties of a realizer:

i) Each interior vertex has exactly one departing edge labeled with k for each k in {1, 2, 3}.
ii) The counter-clockwise order of the edges incident to each vertex is:

1. Departing with label 1.
2. Arriving with label 3.
3. Departing with label 2.
4. Arriving with label 1.
5. Departing with label 3.
6. Arriving with label 2.

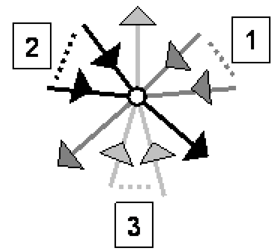

Figure 6 Realizer edge order

The fourth and fifth properties of normal labeling and the use of Euler's formula also tells us that each of the three exterior vertices have no arriving edges, and all of their departing edges have the same label. Thus the exterior vertices with edges labeled k can be regarded as the root or sink of T_k.

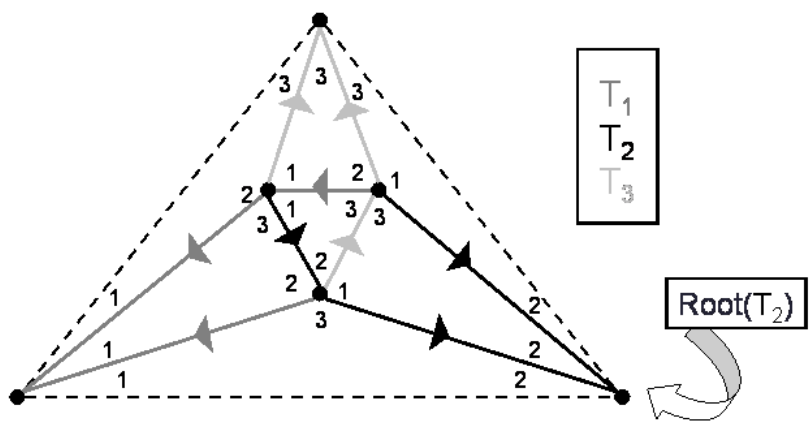

Figure 7 An illustration of a Realizer

3.1.2. Construction

Schnyder describes a method for constructing the normal labeling (and associated realizer) using a method call *edge contraction*. An interior edge {x, y} in the triangular graph is contractible if x and y have exactly two common neighbouring vertices. The result of an edge contraction is that the edge {x, y} is removed and an interior edge {x, z} is added to every neighbour z of y that is not already a neighbour of x.

Since an edge contraction on a triangular graph always returns another triangular graph with one fewer vertex, Schnyder indicates that it is possible to construct the normal labeling by reversing a sequence of edge contractions on the empty exterior triangle. This method also leads to an inductive proof that every triangular graph has a normal labeling. Using the *canonical ordering* for the vertices described in the preliminary section, it is possible to add each interior vertex incrementally such that it is initially connected to an exterior vertex. Interior vertices connected to an exterior vertex can be correctly labeled since we know that the interior edges at the exterior vertex must all have the same label and direction.

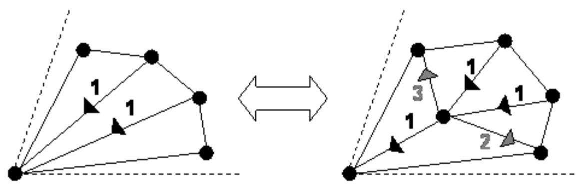

Figure 8 An illustration of edge contraction

Schnyder also sketches a proof that the incremental construction of the normal labeling via edge contraction will result in a realizer T_1, T_2, T_3 such that for any i in {1,2, 3}, Ti will be a tree with no cycles. He also shows that for any T_i, taking the union of the tree T_i with the reversed trees T_j (j != i) will result in a set with no directed cycles. (This fact will come in handy below).

3.2 Grid Coordinates

The second phase of Schnyder's algorithm is the determination of the grid coordinates for each interior vertex of the graph. This process relies on the normal labeling described in the previous section.

3.2.1. Vertex Regions

The normal labeling of the graph results in each interior vertex in the triangular graph having exactly one edge departing via T_i for i = {1, 2, 3}. Since these sets are disjoint (no interior edge can have two labels) and the union of the sets does not contain any cycles, the path from any interior vertex v to the root of each of the T_i results in a division of the graph into three regions for each vertex.

The Region R_i of an interior vertex v corresponds to the region of the triangular graph that is bounded by the directed paths from v to the roots of Tj, Tk {i != j != k}. (See figure 9).

Schnyder gives a simple proof based on the properties of normal labelings and realizers that shows that for any interior vertex u, and vertex v, if u is inside $R_i(v)$ then $R_i(u)$ is entirely contained in $R_i(v)$.

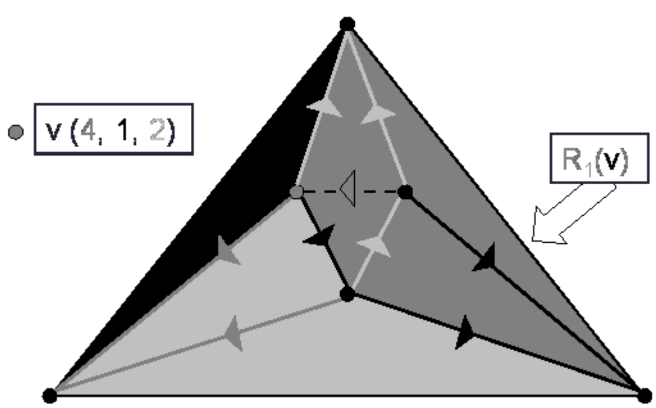

Figure 9 An illustration of the regions of a vertex

3.2.2. Counting Triangles in a Region

Schnyder shows that distinct coordinates for each vertex v of the triangular graph can be obtained by counting the number of triangles that occur in each region $R_i(v)$.

Figure 9 shows this process for a single vertex, and figure 10 shows the result of the process when applied to all vertices of a triangular graph.

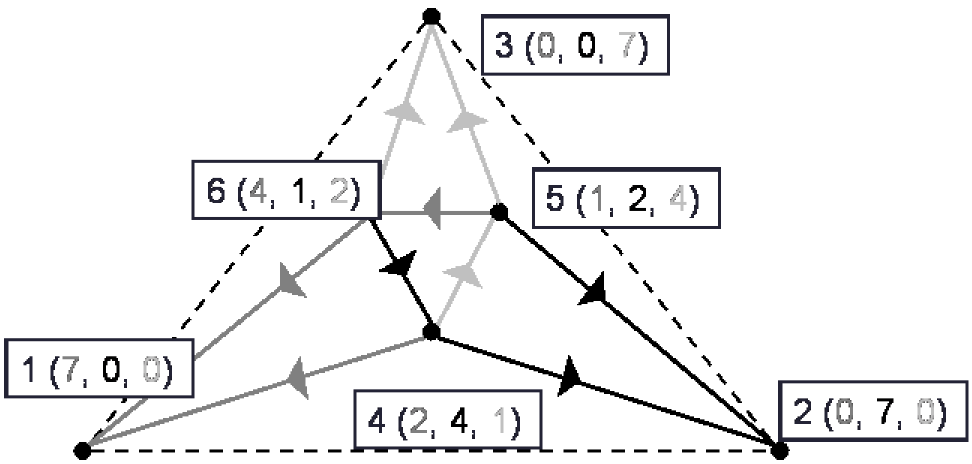

Figure 10 The computed coordinates of a planar graph

Since in a triangular graph with n vertices we have 3n-6 edges, we can apply Euler's formula to determine the total number of triangles in the graph.

$|V| - |E| + |F| = 2 \quad -> \quad n - (3n - 6) + |F| = 2 \quad -> \quad |F| = 2n - 4$

Subtracting one for the outer face/triangle tells us that the triangular graph has 2n – 5 interior triangles.

Schnyder shows that the grid coordinates for each vertex in a triangular graph will be in the range from 0 to 2n -5. Also the sum of the three coordinates will be exactly 2n – 5. This implies that dividing the coordinates of each vertex by 2n-5 will yield a barycentric representation of the triangular graph, and Schnyder provides a proof that this is indeed the case.

Schnyder goes on to show that using the three roots of the computed realizer (a, b, c), we can always obtain a straight line embedding of the triangular graph by choosing any three non-colinear positions for a, b and c.

3.2.3. Counting Vertices in a Region

To obtain a more compact coordinate system for the vertices of the triangular graph, Schnyder describes a scheme for counting the number of vertices that occur in each region R_i for every vertex.

The process is similar to that of counting the number of triangles in the regions of every vertex, except a weak barycentric representation of the triangular graph is obtained.

A *weak barycentric representation* of a triangular graph is similar to the barycentric representation mentioned previously, except the second requirement is relaxed.

The requirement:

For each edge {x, y} and each vertex z not in {x, y}, there is some k in {1, 2, 3} such that $x_k < z_k$ and $y_k < z_k$.

Is replaced with:

For each edge {x, y} and each vertex z not in {x, y}, there is some k in {1, 2, 3} such that $(x_k, x_{k+1}) <_{lex} (z_k, z_{k+1})$ and $(y_k, y_{k+1}) <_{lex} (z_k, z_{k+1})$.

Where $p_1 = (x_1, y_1)$ <lex $p_2 = (x_2, y_2)$ denotes that p_1 is lexicographically smaller than p_2. In other words, p_1 is smaller than p_2 if:

$x_1 < x_2$ or
$x_1 = x_2$ and $y_1 < y_2$.

This leads to grid coordinates for every vertex that are in the range from 0 to n-2.

Schnyder claims that it is possible to count the number of vertices that occur in each region in linear time by performing a constant number of traversals of the realizer. It can be assumed that a similar process would allow for the number of triangles in each region to also be computed in linear time.

3.2.4. Placing Vertices on the Grid

Once the coordinates for every vertex of the triangular graph have been computed, using either the triangle or vertex counting methods described above, it is a simple matter to derive the grid coordinates of each vertex on a 2n -5 by 2n-5 (or alternately n-2 by n-2) grid.

From the triangle counting method, we have the following barycentric representation of the triangular graph:

$v \rightarrow 1 / (2n - 5)\ (v_1, v_2, v_3)$ where $(v_1 + v_2 + v_3 = 2n - 5)$

We can choose the following grid coordinates for the exterior vertices:
$a = root(T_1) = (2n - 5, 0)$
$b = root(T_2) = (0, 2n - 5)$
$c = root(T_3) = (0, 0)$

Now using the mapping $v \rightarrow 1/ (2n -5)\ (v_1 a, v_2 b\ v_3 c)$ we can get the grid coordinates for each vertex v from the triangular graph. The choice of c = (0, 0) means that we can conveniently ignore the coordinate v_3 giving us 2 dimensional grid coordinates.

Figure 10 and 11 show the transition from the triangle counting coordinates to grid coordinates for a sample graph.

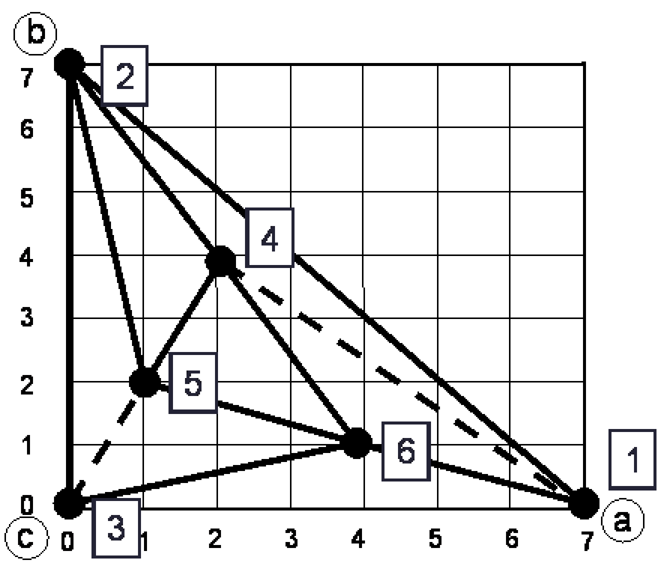

Figure 11 An embedding of a planar graph on the grid

When placing the graph on the grid, edges that were added during the triangulation of the input graph can be removed, giving a grid embedding of the original (possibly non-triangular) input graph.

4. Conclusion

This concludes the summary of Schnyder's algorithm. We have seen that by constructing a normal labeling of a triangular input graph, we can obtain the coordinates for each vertex on an 2n-5 by 2n-5 grid (alternately an n-2 by n-2 grid).

4.1 Discussion and Critique

Schnyder's paper was a pleasure to read. The description of the algorithm for embedding a planar graph on the grid was very well laid out and relatively easy to follow, particularly (according to me) in comparison to the algorithm presented by De Fraysseix and Pollack in [4] which was published at approximately the same time.

In my opinion there are a couple of areas for improvement in Schnyder's paper.

Some visual examples of the properties of barycentric representations would have been extremely helpful for relating them to the rest of the paper. Since Schnyder's paper begins with a description of barycentric representations, it would have been nice to have more of an explanation of their relevance than the brief sentence in the introduction.

Asides from the barycentric coordinate section, I found that the sections of the paper flowed very nicely from one to the next. I felt that this could have been augmented by having a running example of the algorithm on a sample graph that was updated to display the properties of each section.

I found it rather odd that not once in the paper was there a drawing of a planar graph on a grid. Perhaps such an illustration was deemed unimportant, but it did leave some doubts tickling the back of my mind about the aesthetics of the grid embeddings resulting from Schynder's algorithm. It is interesting to note that Chrobak and Payne [3] imply that grid embeddings achieved using Schnyder's algorithm are not very aesthetic, although they do not give any elaboration of this statement.

Schnyder's algorithm requires a triangular / maximal planar graph as input. This in turn implies other requirements which I have mentioned in the preliminaries section. Perhaps these requirements are immediately obvious to a veteran student of graph drawing and graph theory, but I feel that it would have been nice to have these requirements more clearly stated (possibly together) in some section of the paper.

Lastly, Schnyder refers the reader to a paper by Read [7] for a linear time algorithm for triangulating a planar graph. I have read the relevant section from Read's paper and tried the algorithm on several examples, but there were some missing details. A better, and I believe more accurate description of a linear triangulation algorithm is provided by De Fraysseix, Pach and Pollack in [4].

5. References

[1] K.S. Booth and G.S. Lueker,
Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms,
Journal of Computer and System Sciences, 13, pp. 335-379, 1976.

[2] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa,
A linear algorithm for embedding planar graphs using PQ-trees,
Journal of Computer and System Sciences, 30(1), pp. 54-76, 1985.

[3] M. Chrobak and T.H. Payne,
A Linear-time Algorithm for Drawing a Planar Graph on a Grid,
Technical Report UCR-CS-90-2, Dept. of Math. And Comp. Sc., Univ. California Riverside, 1990.

[4] H. de Fraysseix, J. Pach and R. Pollack,
How to draw a planar graph on a grid,
Combinatorica, 10, pp. 41-51, 1990.

[5] I. Fary
On Straight line representation of planar graphs,
Acta. Sc.i. Math. Szeged 11, pp. 229-233, 1948.

[6] J. Hopcroft and R. Tarjan
Efficient Planarity Testing
Journal of the Association for Computing Machinery, 21(4), pp. 549-568, 1974.

[7] R.C. Read,
A new method for drawing a planar graph given the cyclic order of the edges at each vertex,
Congressus Numerantium, 56, pp. 31-44, 1987.

[8] W. Schnyder,
Embedding Planar Graphs on the grid,
In *Proc. 1st ACM-SIAM Symposium on Discrete Algorithms*, pp. 138-148, 1990.