
Journal of Mathematical Modelling and Algorithms **1:** 257–268, 2002.

An Efficient Algorithm for Embedding Nonplanar Graphs in Planes

A. KAVEH and H. RAHAMI
Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran. e-mail: alikaveh@iust.ac.ir

(Received: 13 August 2001; accepted in final form: 14 October 2002)

Abstract. An efficient multi-step algorithm is presented for embedding a nonplanar graph on the least possible number of planes. In the first step, the graph is drawn on a plane with the smallest possible number of crossings. Then the appropriate crossing edges are selected and moved onto the second plane. This process is repeated introducing subsequent planes until no crossing remains.

Mathematics Subject Classifications (2000): 05Cxx, 68R10, 05C10.

Key words: thickness, graph, decomposition, embedding, crossings.

1. Introduction

Planarity is important if shorting is to be avoided on a circuit board. An interesting related question is to determine how many different circuit boards with crossing-free drawings are needed to print a given nonplanar network.

The minimum crossing problem has applications in printed circuit board layout, large-scale integration circuit routing, automated graph drawing, and calculating the degree of static indeterminacy of structures [1–3]. A similar problem arises in the force method of structural analysis where a cycle basis is constructed using regional cycles after embedding the graph on a set of planes with some interface properties, see Henderson [4] and Kaveh [5].

A planar graph can always be embedded on a single plane (disk). However, a nonplanar graph can be embedded on τ disks. The minimum value of τ is defined as the *thickness* of a graph G denoted by $t(G)$. Manfield [6] showed that determining the thickness of a graph is NP-hard, and therefore the focus of this paper will be on a heuristic algorithm for embedding a general graph on the least possible number of planes.

An efficient multi-step algorithm is developed for embedding a nonplanar graph on the least possible number of planes. In the first step, the graph is drawn on a plane with the smallest possible number of crossings. Then the appropriate crossing edges are selected and moved onto the second plane. This process is repeated intro-

ducing subsequent planes until no crossing remains. The efficiency of the algorithm is illustrated using different types of graphs presented in this paper.

Here, three relevant theorems are stated and their proofs can be found in standard textbooks on the theory of graphs, e.g., [7]:

THEOREM 1. *If G is a graph with N $(N > 2)$ nodes and M edges, then*

$$t(G) \geqslant \left\lceil \frac{M}{3N-6} \right\rceil, \tag{1}$$

and $t(G) \geqslant \lceil M/(2N-4) \rceil$ if G has no triangle.

THEOREM 2. *The thickness of the complete graph K_N is*

$$t(K_N) = \left\lfloor \frac{N+7}{6} \right\rfloor, \quad \text{except that } t(K_9) = t(K_{10}) = 3. \tag{2}$$

THEOREM 3. *The thickness of the complete bipartite graph $K_{M,N}$ is*

$$t(G) = \left\lceil \frac{MN}{2(M+N-2)} \right\rceil, \tag{3a}$$

except possibly if M and N are both odd, $M \leqslant N$ and there is an integer k satisfying

$$N = \left\lfloor \frac{2k(M-2)}{M-2k} \right\rfloor. \tag{3b}$$

For other relevant theorems and an excellent survey on the thickness of graphs, the reader may refer to Alekseev and Goncakov [8] and Mutzel *et al.* [9]. Standard definitions of the theory of graphs may be found in any textbook on the theory of graphs or in [10].

2. Algorithm for Embedding a Graph on Planes

The main idea employed in this paper comprises of drawing a given graph on a plane with the least possible crossings, and then the edges necessary for the elimination of the crossings are selected. Clearly, it is ideal to have a minimum number of such edges. In fact the reverse is also applicable, i.e. in order to reduce the crossings, the number of such out-of-plane edges should be minimized.

Once the planar part of the graph containing a maximal number of edges is identified, the deleted edges are transferred onto the second plane, and the second planar subgraph is obtained. This process is repeated until the number of crossings becomes zero and no new plane is required. In the process of embedding, it is important to minimize the number of edges which should be deleted. Heuristically, the minimum number of crossings will require less such edges to be moved to the subsequent planes.

In a good drawing, a crossing is produced by two edges, and the question is to identify the edge to be moved to the next plane. The problem of selecting the edges to be moved and the layout of the nodes in each plane are the main issues investigated in this paper.

In [11] it is shown that, for some special node layouts, drawings with the least possible crossings can be obtained. The algorithm for such a drawing is as follows:

First select a layout for the given nodes of a graph by positioning them on a straight line and then construct the corresponding adjacency matrix. Define a territory for each nonzero entry of this matrix. The relationship between each edge and the edges in its territory form a virtual graph G_v. Construct the corresponding Laplacian matrix of G_v, and calculate the maximal eigenvalue $\lambda_{\max}$ of this matrix. Order the nodes according to the corresponding eigenvector $\mathbf{v}_{\max}$, and separate its positive and negative entries. We use for the nodes corresponding to the negative and positive values -1 and $+1$. This operation is equivalent to drawing the edges at the top and bottom of the straight line containing the nodes. The graph obtained in this way has a small number of crossings for the selected layout.

Suppose for the given graph, the layout of the nodes is chosen as in Figure 1(a). This is not the best layout and it is taken only as an illustrative example. For this graph, the smallest number of crossings is found to be 10 ([11]). The challenge is to find the least number of edges that should be removed in order to reduce the crossings to zero.

The symmetric matrix $\mathbf{A}^*$ is defined for a graph with nodes on a straight line (Figure 1(a)). This matrix contains $+1$ and -1 entries, similar to the adjacency matrix $\mathbf{A}$ of a general graph. For this example, the matrix $\mathbf{A}^*$ is formed as

$$\mathbf{A}^* = \begin{array}{c} \\ 5 \\ 6 \\ 3 \\ 4 \\ 8 \\ 7 \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccccc} 5 & 6 & 3 & 4 & 8 & 7 & 2 & 1 \end{array} \\ \begin{bmatrix} & 1 & 0 & -1 & 0 & -1 & 0 & -1 \\ & & 1 & -1 & -1 & 0 & 1 & 1 \\ & & & 1 & 0 & -1 & 1 & 1 \\ & & & & 1 & 0 & 1 & 1 \\ & & & & & 1 & 1 & 1 \\ & & & & & & 0 & 0 \\ & & & & & & & 1 \\ & & & & & & & \end{bmatrix} \end{array}. \tag{4}$$

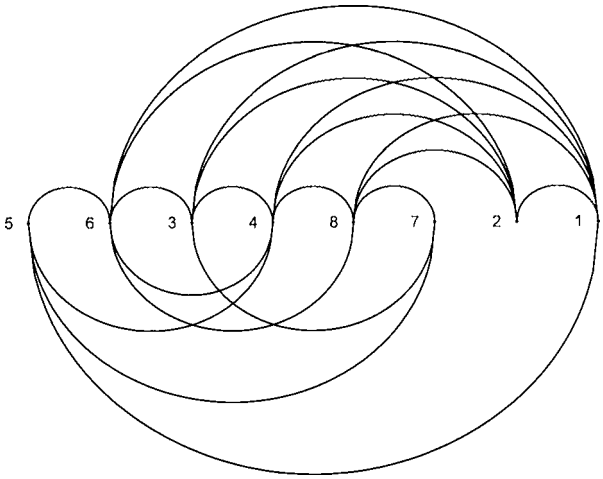
Here, a *territory* is defined for each entry of the matrix $\mathbf{A}^*$. The territory for the entry (i, j) is the union of the left and right parts, as

$$\mathrm{Ter}(i, j) = \mathbf{A}^*(1 : i-1, i+1 : j-1) \cup \mathbf{A}^*(i+1 : j-1, j+1 : \mathrm{end}). \tag{5}$$

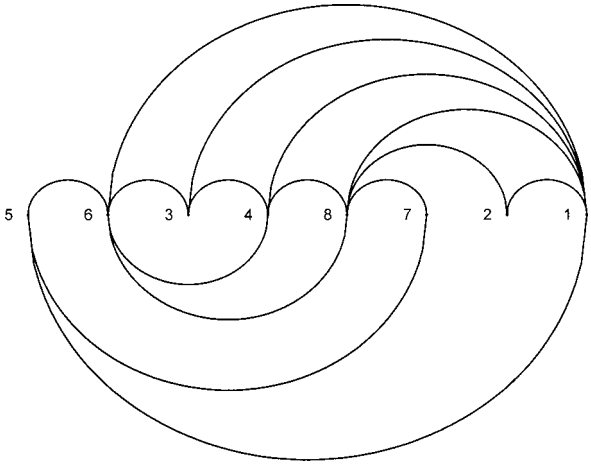
As an example, the territory for the entry (2,5) of $\mathbf{A}^*$ corresponding to the edge 6–8 is the union of the left and right parts as follows:

$$\mathrm{Ter}(2, 5) = \mathbf{A}^*(1 : 1, 3 : 4) \cup \mathbf{A}^*(3 : 4, 6 : 8).$$

This territory is shaded on $\mathbf{A}^*$. The use of row and column headings helps us to identify the corresponding edge in the graph.



(a) Number of edges: 20; number of crossings: 10



(b) Page 1; number of edges: 15

Figure 1. A graph with an arbitrary layout and its subgraph on the first plane.

For the entry (2,5) with -1 value, the left part of the territory contains only one entry with -1 corresponding to the edge 5–4, i.e. the edges 5–4 and 6–8 cross each other. However, the edge 6–8 has two crossings and the second crossing is generated by the edge 3–7 belonging to the right part of the territory. It should be mentioned that, in a problem, if the sum of the crossings is important, then only the left part of the territory can be used, [11]. However, here it is crucial to know the number of crossings for each individual edge of the graph. Thus both left and right parts of the territory are employed.

As a second example, consider the entry (2,7) of $\mathbf{A}^*$, as marked in the following matrix. The territory of this entry with $+1$ value, in the left part has no crossing (i.e. there is no $+1$ in the left part of the territory) and in the right part it contains three crossings.

$$\mathbf{A}^* = \begin{array}{c} \\ 5 \\ 6 \\ 3 \\ 4 \\ 8 \\ 7 \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccccc} 5 & 6 & 3 & 4 & 8 & 7 & 2 & 1 \end{array} \\ \left[\begin{array}{cccccccc} \square & 1 & 0 & -1 & 0 & -1 & 0 & -1 \\ & \square & 1 & -1 & -1 & 0 & 1 & 1 \\ & & \square & 1 & 0 & -1 & 1 & 1 \\ & & & \square & 1 & 0 & 1 & 1 \\ & & & & \square & 1 & 1 & 1 \\ & & & & & \square & 0 & 0 \\ & & & & & & \square & 1 \\ & & & & & & & \square \end{array}\right] \end{array}. \tag{6}$$

Now consider Figure 1(a) once again. The edge 6–2 has three crossings which can also be observed from the right territory of the entry (2,5). Using $\mathbf{A}^*$, the number of crossings can be found for each edge. Based on this data, the *crossing matrix* $\mathbf{C}$ for edges can be constructed. This matrix for the considered example is obtained as

$$\mathbf{C} = \begin{array}{c} \\ 5 \\ 6 \\ 3 \\ 4 \\ 8 \\ 7 \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccccc} 5 & 6 & 3 & 4 & 8 & 7 & 2 & 1 \end{array} \\ \left[\begin{array}{cccccccc} \square & & & 2(1) & & & & \\ & \square & & 1(2) & 2(3) & & 3(5) & \\ & & \square & & & 3(4) & 2(6) & 1(8) \\ & & & \square & & & 1(7) & 2(9) \\ & & & & \square & & & 3(10) \\ & & & & & \square & & \\ & & & & & & \square & \\ & & & & & & & \square \end{array}\right] \end{array}. \tag{7}$$

It is obvious that the sum of the entries (only for the upper triangle) is twice the number of crossings, since each crossing appears in the territory of two entries of $\mathbf{A}^*$. In the present example, this number is 20 corresponding to 10 crossings. Now each number in $\mathbf{C}$ is the representative of the number of crossings, therefore for an efficient embedding, one should delete the edges having the highest crossings. When only one edge with this property exists, it can be deleted as suggested. However, if many edges with maximum crossings exist, as in the above example (three edges each with three crossings), then further consideration is required. Although the deletion of any of these edges reduces the crossings by three, however, the effect of the removal of any one such edge on subsequent steps of the algorithm is different.

As an example, the territory of the entry (2,7) of $\mathbf{A}^*$ shows that the corresponding edge 2–6 crosses the edges 3–1, 4–1 and 8–1. The numbers of crossings for these edges, from the matrix $\mathbf{C}$, are 1, 2 and 3, respectively, i.e. the edge 6–2 crosses

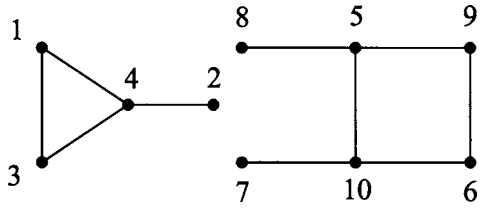


Figure 2. An artificial graph associated with **C**.

three edges with one, two and three crossings. Similarly, the edge 3–7 has crossings with three edges having two, one and two crossings, and the edge 8–1 has crossings with three edges with three, two and one crossings. It is a reasonable heuristic to delete the edge having the least number of maximum crossings. Therefore, the edge 3–7 should be deleted, since the crossing number '3', corresponds to none of the edges crossing this edge.

Now we associate an artificial graph with the matrix **C** as follows: Suppose the nonzero entries of **C** receive numbers $1, \ldots, n$. Associate one node with each number and connect two nodes if the corresponding numbers have the same sign in the considered territory of the matrix $\mathbf{A}^*$. For the previous example, with the numbers $1, 2, \ldots, 10$ associated to the entries of **C** (shown in parenthesis in matrix **C**), the graph of Figure 2 is formed.

Such a graph may be connected or disconnected. The degree of a typical node of this graph represents the number of crossings of the corresponding edge in **C**. For each node, the degrees of neighbouring nodes are the number of crossings of edges having crossings with the considered edge in the main graph.

The idea of deleting edges for the formation of a maximal planar subgraph can be viewed as selecting the minimum number of co-cycles of the graph which span (cover) all its edges. Here we start with a node of maximal degree and among the nodes of equal degrees, priority is given to the node with minimum number of nodes of maximal degree in its neighbourhood. For example, for the nodes 4, 5 and 10 having equal degrees, priority is given to the node 4 compared to the nodes 5 and 10. Since every elimination reduces one degree from the adjacent nodes, it is better if this reduction corresponds to the nodes with lower degrees so that some higher degrees remain available for further elimination in subsequent steps. This might be interpreted as selecting a node adjacent to the nodes having a lower degree sum among the equivalent nodes. However, this is not true as shown in the hypothetical example for three edges (corresponding to nodes A, B and C) with equal degrees. The degrees of the adjacent nodes in the artificial graph for these nodes are shown in Table I.

At this stage, one may choose the node B for the elimination. However, since the maximum degree among the neighbouring nodes is 9, and 9 does not exist among the adjacent nodes of C, hence C should be chosen for elimination, despite the fact that the sum of valencies for the neighbouring nodes of C is more than those of A and B. The next selection is not necessarily from A or B, since with

Table I. Properties for the three nodes of a hypothetical graph

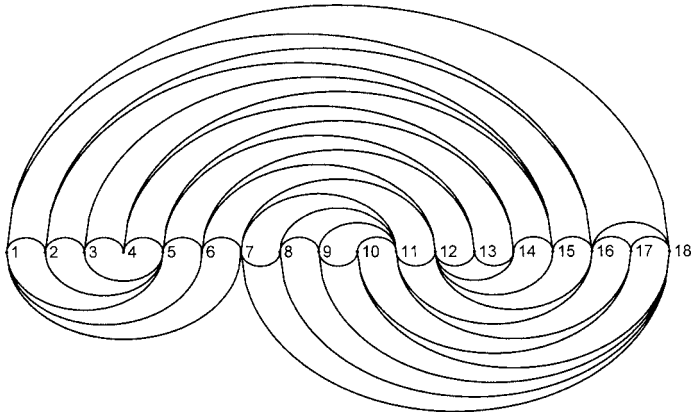
Nodes	Degrees of the adjacent nodes	Sum of the degrees of the neighbouring nodes
A	7 7 1 9 4	28
B	1 9 1 9 1	21
C	8 8 8 8 8	40

each elimination the degrees of the other nodes may be changed in a way that the selection does not necessarily correspond to the comparison of A or B. The simplest rule for selecting a node from the nodes of equal degrees, is to consider the degrees of the adjacent nodes and place them in a descending order. As an example, for nodes A, B and C, the corresponding numbers are 97741, 99111 and 88888, respectively. The least of these virtual numbers, i.e. 88888 corresponds the node C, and therefore this node should be eliminated at this stage. As mentioned before, the entries of the matrix **C** are updated after each elimination, and this makes the next selection unpredictable beforehand. The above process should be continued until all the entries of the matrix **C** changes to zero. This is the end of the first stage, where a drawing with zero crossing on the first plane with the maximum number of possible edges is produced.

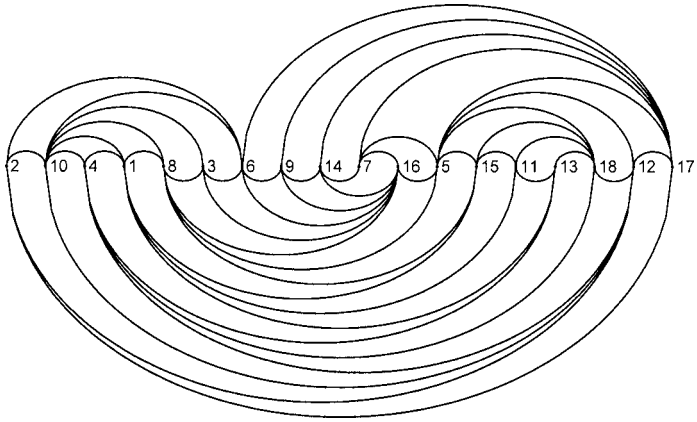
As already mentioned, apart from the importance of selecting the edges for elimination, the layout of the nodes at the beginning of each stage is also important. This is because, usually, a small number of crossings results in a planar graph with higher number of edges. For the present example, the selected layout was arbitrarily chosen and was not necessarily the best possible one. However, the number of edges left on the plane was 15, as shown in Figure 1(b). The remaining five edges were drawn on the second plane with no crossing and therefore the thickness of this graph is at most equal to 2.

It has been observed that for each layout, the method of drawing with the least crossings and eliminating the least number of edges is a rule-based problem. However, the layout for nodes requiring the alteration of their position does not obey a simple rule. The Genetic Algorithm may be used to overcome the problem for the minimum crossings. Alternatively, an ordering algorithm may be used leading to a banded matrix for the adjacency matrix **A** ([11]).

In this article a simplified version of King's algorithm [12], originally designed for the profile reduction of sparse matrices is employed. In King's algorithm, the starting node corresponds to a node of minimum degree while, in the present approach, no such requirement is needed. In the process of applying King's algorithm, different nodes may have the same priorities and therefore various cases should be examined. In the present algorithm, control of all such cases is not needed, as an optimal layout does not correspond to a single ordering of the nodes.



(a) $\mathbf{K}_{18}$ Page 1; number of edges: 48



(b) $\mathbf{K}_{18}$ Page 2; number of edges: 48

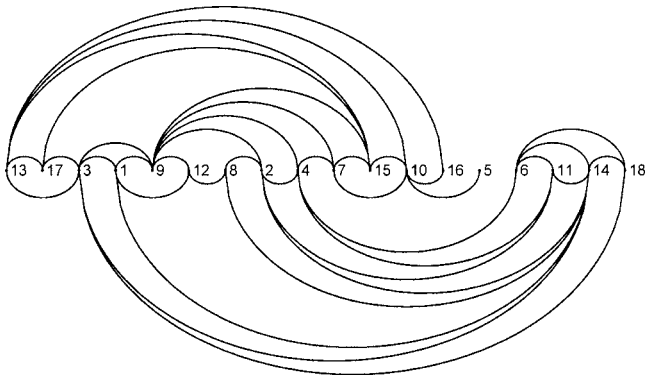
Figure 3. Embedding of K_{18}.

The present algorithm, being a direct graph-theoretical approach, requires far less storage and operations compared to that of the genetic algorithm.

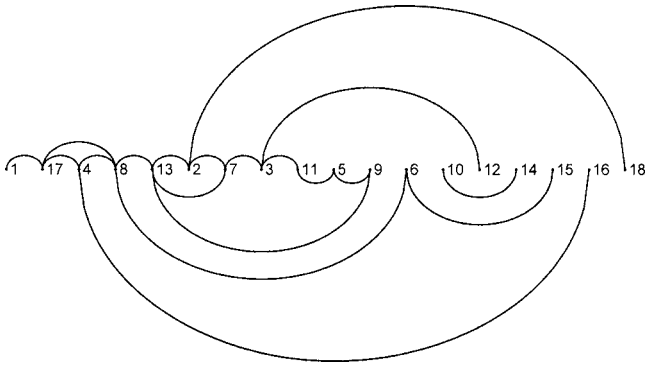
3. Examples

The algorithm is implemented on a PC Pentium IV and many graphs of different types are examined. In the following some results are presented.

EXAMPLE 1. The complete graph K_{18} is embedded on four planes as illustrated in Figures 3(a–d), where the number of edges in each plane are provided. The computer time required has been 1.64 sec.



(c) $\mathbf{K}_{18}$ Page 3; number of edges: 38



(d) $\mathbf{K}_{18}$ Page 4; number of edges: 19

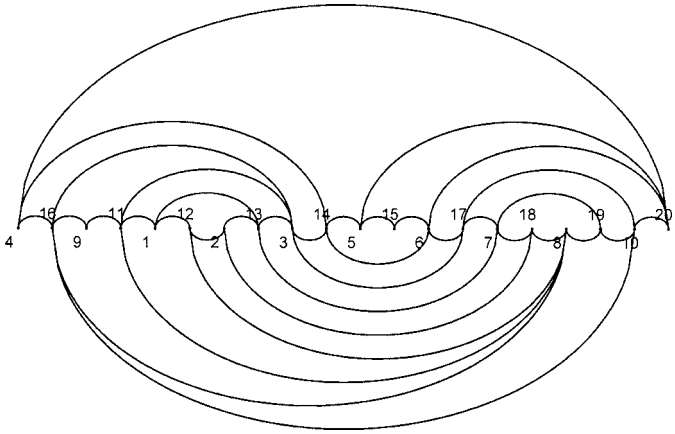
Figure 3. Embedding of K_{18} (continued).

EXAMPLE 2. The complete graph K_{24} is embedded on five planes. The number of edges of the embedded graph on plane 1 to plane 5 were 66, 66, 62, 51 and 31, respectively. The computer time required was 8.64 sec.

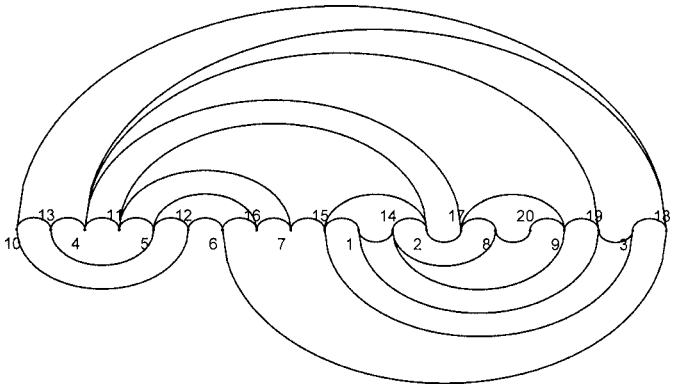
From Theorem 2 stated in the introduction, we have $t(K_{18}) = 4$ and $t(K_{24}) = 5$. It can be seen that in both examples, the present method leads to embeddings corresponding to the thickness of the considered complete graphs.

EXAMPLE 3. The complete bipartite graph $K_{10,10}$ is embedded on three planes as illustrated in Figures 4(a–c), where the number of edges in each plane are provided. The computer time required has been 0.66 sec.

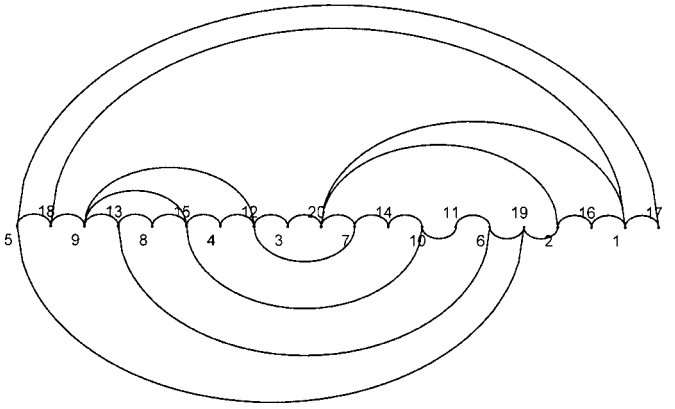
EXAMPLE 4. The complete graph $\mathrm{K}_{20,20}$ is embedded on six planes. The number of edges of the embedded graph on plane 1 to plane 6 were 76, 76, 75, 74, 66 and 33, respectively. The computer time required was 30.48 seconds.



(a) $\mathbf{K_{10,10}}$ Page 1; number of edges: 36



(b) $\mathbf{K_{10,10}}$ Page 2; number of edges: 35



(c) $\mathbf{K_{10,10}}$ Page 3; number of edges: 29

Figure 4. Embedding of $K_{10,10}$.

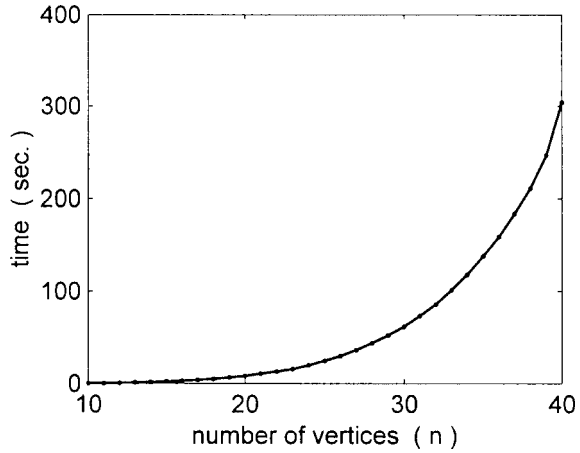


Figure 5. CPU-time for complete graphs versus the number of nodes.

ADDITIONAL EXAMPLES. Two general graphs containing 50 nodes with 149 edges, and 100 nodes with 296 edges are randomly generated and examined. The computational time for embedding these graphs has been 2.47 and 28.21 seconds, respectively.

4. Concluding Remarks

An upper bound to the thickness of a graph is obtained with a series of repeated operations, drawings with the least crossings, and by the elimination of the minimal number of crossings. The reason behind the selection of the examples in this article from the complete graph and complete bipartite graph viewpoint, is that the thickness of the graphs is known. Completeness of the graphs does not reduce the generality of the problem, as the graph becomes noncomplete from the second step onward.

In order to show the efficiency of the algorithm, the complete graphs K_{10} to K_{40} are embedded on planes and the computation time for constructing the maximal subgraph of the first plane is plotted against the number of nodes as shown in Figure 5.

References

1. Cimikowski, R. and Shope, P.: A neural-network algorithm for a graph layout problem, *IEEE Trans. Neural Networks* **7** (1996), 341–345.
2. Tamasia, R., Di Battista, G. and Batini, C.: Automatic graph drawing and readability of diagrams, *IEEE Trans. Sys. Man. Cybern.* **SMC-18** (1988), 61–79.
3. Kaveh, A.: Space structures and crossing number of their graphs, *Mech. Struct. Mach.* **21** (1993), 151–166.
4. Henderson, J. C. de C.: Topological aspects of structural analysis, *Aircraft Engng.* **32** (1960), 137–141.

5. Kaveh, A.: *Structural Mechanics: Graph and Matrix Methods*, 2nd edn, Research Studies Press, (Wiley), London, 1995.
6. Mansfield, A.: Determining the thickness of graphs is NP-hard, *Math. Proc. Cambridge Philos. Soc.* **93** (1983), 9–23.
7. Beineke, L. W.: *Topology*, In: L. W. Beineke and R. J. Wilson (eds), *Graph Connections*, Oxford Science Publ., 1996, pp. 155–173.
8. Alekseev, V. B. and Goncakov, V. S.: The thickness of an arbitrary complete graph, *Math. Sbornik* **30** (1976), 187–202.
9. Mutzel, P., Odenthal T. and Scharbrodt, M.: The thickness of graphs; a survey, *Graphs Combinatorics* **14** (1998), 59–73.
10. Kaveh, A.: *Optimal Structural Analysis*, Research Studies Press (Wiley), London, 1997.
11. Kaveh, A. and Rahami, H.: Planar drawing of space structures using algebraic graph theory, *Asian J. Civil Engng.* **3** (2002), 20–32.
12. King, I. P.: An automatic reordering scheme for simultaneous equations derived from network systems, *Internat. J. Numer. Meth. Engng.* **2** (1970), 523–533.