Maximal Planarization Of Non-Planar Graphs

Suzan Khalil Yacob*

Received on: 27/11/2004
Accepted on: 20/6/2005

Abstract

This paper presents a MAXIMAL-PLANARIZE algorithm using EQUIVELANT-GRAPH procedure. The algorithm proceeds by embedding one or few edges at each stage, without creating nonplanarity of the resultant graph, and to construct a maximal planar subgraph G_p of G directly.

The present implementation shows that using two planarization algorithms is unnecessary because of their complexities. It runs in linear time to give a maximal planar subgraph and adds the maximum number of edges possible without creating nonplanarity[l] ,using only one simple and efficient algorithm.

Key words: Nonplanar graph, Hamiltonian circuit.

عمل المستوى الأعظم للمخططات اللامستوية

الخلاصة

هذا البحث يقدم خوارزمية المستوى الأعظم بأستعمال طريقة الدائرة العظمى. الخوارزمية تشرع باضافة بعض المركبات الخارجية في كل خطوة بشرط ان يكون المخطط الناتج مستوياً وذلك لتركيب المخطط الفرعي المستوي الأعظم G_p للمخطط G مباشرة.

المعالجة المقدمة توضح ان استعمال الخوارزميتين للحصول على المستوى الأعظم غير ضروري بسبب تعقد هاتين الخوارزميتين. تعمل في زمن خطي وتضيف اكبر عدد من الموصلات للحصول على التركيب للمخطط الفرعي المستوي الأعظم Gp للمخطط اللامستوي باستعمال خوارزمية واحدة فقط والتي تتميز ببساطتها وكفاءتها.

1. INTRODUCTION

A GRAPH is planar, if it can be drawn on a plane with no two edges crossing each other except at their end vertices. A subgraph G_p of a nonplanar graph G is a maximal planar subgraph of G if G_p is planar, and adding any edge to G_p result in a nonplanar subgraph of G. This process of removing a set of edges from G to obtain a maximal planar subgraph is known as maximal planarization of the nonplanar subgraph G.

On the other hand, maximal planarization of a planar subgraph G refers to the process of adding a maximal set of edges to G without causing nonplanarity.

Maximal planarization of a nonplanar graph is an important problem encountered in the automation design of printed circuit boards. If an electronic circuit cannot be wired on a single layer of a printed circuit board, then we need to determine the minimum number of layers necessary to wire the circuit. Since only a planar circuit can be wired on a single layer board, we would like to decompose the nonplanar circuit into a minimum number of maximal planar

* Dept. of Electrical Engineering, University of Technology, Baghdad-IRAQ.

https://doi.org/10.30684/etj.24.6.10
University of Technology-Iraq, Baghdad, Iraq/2412-0758

circuits. In general, for a nonplanar graph, neither the set of edges to be removed to maximally planarize it, nor the number of these edges is unique.

Determining the minimum number of edges whose removal from a nonplanar graph will yield a maximal planar subgraph is an NP-complete problem [1,5,6,7]

One of the earliest algorithms was proposed by Demoucron et al.as cited by ref[2]. Further improvement is presented by Rubin[3].

Recently, Jayakumar et al. [4] have proposed two planarization algorithm. These two algorithms are quite interesting because at each step of these algorithms ,as many edges as possible are added. The drawback is the complexity of such planarization algorithms and their requirement of long computation time.

This paper adopts MAXIMAL-PLANARIZE algorithm based on Demoucron planarity testing, the presented algorithm is an efficient, simple one. It is found necessary to generate and evaluate only a few exterior components at each stage, usually one to construct a maximal planar subgraph G_p which contains the maximum number of edges possible without creating nonplanarity of resultant graph. It runs in linear time, requires less computation time and includes larger number of edges than the two algorithms presented in [4].

2.DEMOUCRON, ALGRANGE, AND PERTUIST PLANARITY TESTING ALGORITHM

Consider a simple connected graph G=(V,E) with n=|V| vertices and m=|E| edges. The algorithm begins with a simple circuit G' in the graph G where G'=(V',E') adds one path at a time to build a mesh structure. An exterior component of G' in G is a maximal connected subgraph G"=(V" ,E") of G hence the endpoints of an exterior component G" of G' are the vertices $V' \cap V''$.

They noted that at each stage that:1) some of the exterior components can be embedded in any of two or more meshes; 2) others can be embedded in only one of the meshes; if the graph is nonplanar, 3)some cannot be embedded in any mesh. Consequently, if the latter case occurs, the graph may immediately be judged as nonplanar. If the second case occurs, such a component may be assigned to that mesh immediately.
But if every component has two or more possible meshes, then an arbitrary choice may be made for any one of them.

In Fig.1 each of the exterior components (G") 15,1237 and 123489 may be embedded inside or outside of the circuit (G') 123456. Component 15 is independent of the others, but an arbitrary choice for 1237 or 123489 will force the other to the opposite side of the circuit.
Now, it needs to be noticed that in the Demoucron et al. algorithm, not all the exterior components are embedded in the proper mesh because an arbitrary choice is made for some of them, hence this algorithm is good for deciding whether a given graph is planar or not but it fails for obtaining the maximal planar subgraph G_p of G.

3.THE EQUIVALENT-GRAPH PROCEDURE

The central concept of the EQUIVALENT-GRAPH procedure is stated in the following definitions.

Definition 1: Equivalent graph

It represents the equivalent graph of G which contains all the edges of G. Its vertices are labeled as their counterparts in G, but they are kept separate; i,e. there may be several vertices with the same label.

Definition 2: Maximal Subgraph G_c

the maximal cycle of a graph G containing the maximal vertices or very vertex of G, in other words G_c will represent a Hamiltonian cycle if G is a Hamiltonian graph.

Definition 3: Examining graph (G_e)

The graph G_e may represent a subgraph $G-G_c$ or a set of edges only, where its edges $E(G_e)=E(G)-E(G_c)$.
A chord of G_e is a path in G with endpoints in G_c but with no other vertices in G_c. If the chord has only one edge, then it is called simple, hence if G_c represents Hamiltonian cycle then the examining graph (G_e) contains only simple chord. The simple chords (edges) of G_e are classified according to their priorities in the list of the examined edges as follows:-

1. **Type S:** An edge is said to be type S if it is obtained by using the EQUIVALENT-GRAPH procedure. In the process of implementing the MAXIMAL-PLANARIZE algorithm, these edges of type S have priorities to be examined first, before the next type of edges.

2. **Type R:** An edge is said to be type R if it is not obtained from the EQUIVALENT-GRAPH procedure and it will be examined after examining all the edges of type S.

The Procedure

In this section a new and efficient procedure is presented to obtain the equivalent-graph which contains the maximal subgraph G_c plus the examining graph G_e.

Consider a simple connected graph $G=(V,E)$ with $n=|V|$ vertices and $m=|E|$ edges, with the assumption that every vertex in G should have a degree of at least three.

The procedure could be obtained by performing certain operations on the graph. These operations are:

(1) Determine the degree for each vertex of the graph G. Choose any vertex of minimum degree, to be called x1,and choose a neighbour vertex for x_1 which has the maximum degree corresponding to the other neighbour vertices of x_1.This vertex will be called x_k.

(2) Select an edge joining two distinct vertices $x_1$and x_k ($k\leq n$) such that x_1 and x_k denote the initial and terminal end points of the maximal circuit Gc, set i=1.

(3) Examine the neighbours of the vertex x_i according to the following conditions.

a. If a vertex(x_a) appears with degree one, this gives a notation that the vertex x_a is not contained in G_c ($x_a \notin G_c$), then x_a will represent exterior vertex.

b. Now, examining the remaining neighbour vertices such that this new vertex (x_{i+1}) will have the minimum degree according to the other neighbours to $xi(d(x_{i+1})\geq 2)$. Keeping in mind that if the vertex x_k is one of the vertices neighbouring x_i, is chosen if it is the only choice. Then stop.

c. If there exists more than one vertex with the same minimum degree, choose any one. The vertex or vertices which are not chosen will represent the end vertices of type S edges, which will be obtained by connecting every vertex of them with x_i.

(4) Delete all the edges incident to x_i to get the subgraph G_i, with vertex set $V(G_{i-1})$-x_i, hence the degree for each vertex which is a neighbour to vertex x_i will decrease by one.

(5) If the vertex x_k does not represent a neighbour vertex to x_i then go to step 8.

(6) Detect if the degree of vertex x_k is equal to one $(d(x_k)=1)$. Then choose its neighbour to the new x_k vertex(x_{kn}). Keeping in mind that the edge selected in step 2 will be joining a new edge with end vertices (x_k,x_{kn}). If $x_{kn}=x_{i+1}$ then stop.

(7) Delete the edge incident to vertex x_k, replace x_{kn} by x_k.

(8) Detect the neighbour vertices of the vertex x_k such that.

a. if they have degree larger than two then go to step 10.

b. if a vertex (x_a) appears with degree equal two then choose its other neighbour to be the new x_k vertex(x_{kn}). Keeping in mind that the edges selected in step 2 will be joining two new edges with end vertices (x_k,x_a) and (x_a,x_{kn}) . If $x_{kn}=x_{i+1}$ then stop.

c. If more than one vertex appears with degree equal two, then choose any one as in step b. Then the remaining vertices will represent the exterior vertices, delete the edges incident to them.

(9). Delete the edges incident to vertex x_a and x_k, and replace x_{kn} by x_k. If the degree of x_k is equal to one then go to step 6.If the degree of its neighbours vertex is equal to two then go to step 8.

(10). Replace x_{i+1} by xi and go to step 3.

Note:

In the following it is assumed that all the vertices of the maximal circuit are represented by horizontal segments, with the vertices numbered in an ascending order starting with the vertex which is given number 1.

Example 1

The above definitions and procedure may be further clarified by an example. The sample graph G consists of 10 vertices, and the connections between pairs of vertices are as in table No.1.

Table No.2 illustrates the EQUIVELENT-GRAPH Procedure. It is noted that:

A- vertex 3 has the minimum degree , vertex 9 is a neighbour vertex of vertex 3 with maximum degree.

B- the distinct vertices 3 and 9 are joining the selected edge.

C- the series of graphs G^i (i=1,2,...,7) represent the subgraph after deleting the edges whose initial vertex is x^i where x^i G_{i-1} .

D- the vertices assigned by star represent the neighbour of the vertex assigned by double stars represents the chosen vertex x^{i+1}. The vertices assigned by star and letter S represent the vertices which have degree equal to the degree of the chosen vertex x^{i+1}. Any vertex of these vertices with the vertex x^{i+1} will represent the end vertices of an edges of type S.

E- now, label the vertices of maximal circuit G_c in an ascending order, thus renumbering the vertices 3,4,5,8,10,1,2,6,7,9 by the numbers 1,2,3,...,10.

In Fig.2a The maximal circuit is 3-4-5-8-10-1-2-7-6-9-3. The bold edges represent the edges of type S. The broken edges represent the edges of type R.

Now, according to notion E the obtained equivalent graph is illustrated in Fig.2b

DISCUSSION OF THE PROCEDURE

The following is a discussion of the complexity of some stages of the procedure clarified further by relevant figures.

1. If a vertex appears in a specification as in step (3-a), this gives notion that the graph is not Hamiltonian.
2. In step (3-c) the edges of type S will play an important role in the algorithm because any one of them could be contained in the maximal subgraph. Hence they will have the priority to be examined first.
3. In step 6 a detection of the degree of x_k is very important because if it has a degree equal one, then it has only one neighbour vertex and if this neighbour vertex is chosen then it will have only one probability which is x_k. Then the procedure will stop before searching on the other vertices.

Algorithm 1: Let G be a simple connected graph then the EQUIVALENT-GRAPH procedure will be as follows. Test the graph, if it is Hamiltonian, then it constructs a Hamiltonian circuit. If not then it constructs a maximal circuit in a polynomial time.

Let G_c be a maximal circuit of a graph G of order $n \geq 3$ and that e is an edge of G_c, where G_c: $x_1,x_2,x_3,\ldots,x_k,x_1$ $(k \leq n)$ and $e = x_1\ x_k$, where x_1 denotes the vertex with the minimum degree with respect to the other vertices ,and x_k denotes the neighbour vertex of x_1 with the largest degree.

Let x_2 denote the vertex with the minimum degree $[d(x_2) \geq 2]$ with respect to other neighbour to x_1, (keeping in mind that the vertex x_k is chosen only if it is the only choice),then in the subgraph generated by $G(x_1)$-$\{x_1\}$ examine if the degree of vertex x_k is equal to one then choose its neighbour to the new x_k (x_{kn}). Hence the selected edge (e) is now joining a new edge (x_k, x_{kn}).

Fig.3a illustrates the basis for this decision. Suppose, to the contrary, that this decision is not defined, then according to the procedure, x_o is chosen, so is x_1 will be chosen ,then the subgraph generated by

$G(x_o)$-$\{x_1,x_2,x_o\}$ will never contain the vertex x_k, and a Hamiltonian (or maximal) circuit could not be obtained. Hence to overcome this problem this decision must be included in the procedure. Also, the neighbouring vertices of vertex x_k must be examined, where if a vertex of degree two is found then its other neighbour vertex will represent the new x_k (x_{kn}).

Fig.3b illustrates the essential use of this decision. According to the procedure, vertex x_m must be chosen ,so will x_o, and because it is of degree two then x_k is chosen because it is the only choice. But x_k is not the last choice. Now to overcome this problem, the above decision must be included in the procedure where accordingly x_m will become the x_{kn}, then after x_2 is chosen so will x_3 and the selected edge e will be joining the new path (x_k,x_o,x_m). Then, for k=n [x_1,x_2,x_3,...,x_k,x_1] is the Hamiltonian circuit while for k<n the maximal circuit is for the Hamiltonian subgraph (G-v); such that the removal of a number of vertices creates a graph with Hamiltonian circuit.

Next, an example illustrating algorithm 1 is presented. First one needs to determine whether the graph G of Fig.4 is Hamiltonian or not.

A sequence G_1,G_2,...(which is not unique) is shown. Since vertex (10) or the vertices (6 and 7) must be removed, graph G is not Hamiltonian (actually G is Peterson graph).

4.A NEW MAXIMAL PLANARIZATION ALGORITHM

In this section an efficient algorithm is presented to determine a maximal planar subgraph of a nonplanar graph G, based on the above procedure EQUIVALENT-GRAPH. The focus of maximal planarization is how to maximize the number of edges in the required graph.

On this basis attempts were made to add G_c as many edges as possible without affecting the planarity of the resultant graph.

Before giving the final form of the algorithm, a logical sequence of steps needs to be by definitions and examples.

Definition 4 : The I (Interior) and O (Exterior) Faces

Let G_c be a finite subgraph of G, then the infinite plane is divided into two faces. The interior is called the I face of the maximal circuit, and the exterior is called the O face of the maximal circuit G_c .

Definition 5: Initial Edge

Once the maximal circuit has been generated, every edge may be embedded in either faces. Then the initial edge is that which connects vertex 1 to A_1 . Where A_1 represents one of the neighbours (A_1, A_2 ,...,A_i)of vertex 1 such that $A_1 < A_2 ... < Ai$.

Definition 6: First, Second,..., lst edge of a vertex

Let the vertices be connected to vertex x by outward edges (type S or type R) of x be x_1,x_2,x_3,...,x_l such that $x_1 < x_2 < x_3 < ... < x_l$ then the first edge of x which will be examined first is (x,x_1), the second is the edge connecting x_2 to x and so on.

Definition 7: The One, and Only One Possible Embedding

If two edges A and B whose end vertices are numbered A_1,A_2 for B_1,B_2 for B such that these vertices $A_1,A_2,B_1,B_2 \in V(G_c)$ and $A_1<B_1<A_2<B_2$ or $A_1<B_2<A_2$ then if edge A is embedded in I face then edge B will be embedded in O face and vice and versa. Hence, if only this case occurs, the

possible embedding of the examined edge is only one.

Definition 8: Testing The Examining Graph

The chords which are only edges are examined first, by adding one edge at a time to build a mesh structure. It is noted that at each stage:

1. Some of the edges could be embedded in any of the two faces, in other words, $P(E,G_i)=2$, where P represents the number of possible embedding for the examined edge E in the plane subgraph G_i, where G_i is a sequences $G_1,G_2,\ldots,G_i$ of plane subgraphs of G such that $G_i \subset G_{i+1}$ for $(i \geq 1)$.
2. Other edges could be embedded only in one of the faces according to definition 7. It means that the number of possible embedding is one for the examined edge, hence $P(E,G_i)=1$;and if the graph is nonplanar.
3. Some cannot be embedded in any face, where $P(E,G_i)=\emptyset$. Consequently, if the third case occurs, the graph will be immediately judged as nonplanar. If the second case occurs, such an edge will be assigned to the face immediately. But if the first case occurs, this edge must be re-examined directly after examining the next edge in the list of edges, until the second case occurs, to embed the edge in the only one proper face.

   If G_c does not represent a Hamiltonian circuit then any exterior vertex (V_x) where $V_x \notin G_c$ will be tested according to its neighbours and it will be embedded in the mesh which contains the maximum number of its neighbours. Of course, this case rarely occurs in a connected nonplaner graph where there are enough edges to form a Hamiltonian circuit[5].

Example 2

After placing the maximal circuit vertices obtained from Example 1, the edges of the examined graph G_e will be examined in a systematic way starting with the initial vertex 1 and moving towards the end vertex. The procedure is as follows:

A. list the examined edges according to definition 6 keeping in mind that the edges of type S must be examined first.

B. add the Initial edge E_I, then by definition 5 the Initial edge is (1,7) to obtain the subgraph G_1

C. we then move to edge E_2. According to definition 8 part 1 this edge could not be embedded because $P(E,G_i)= 2$,then move to edge E_3 where $P(E_3,G_1)=2$ also.

Now move to edge E_4. According to definition 8 part 2 this edge will be embedded in face O because $P(E_4,G_1)=1$. Add this edge to obtain a plane subgraph called G_3.

E. now, re- examine the edges E_2 and E_3 of step C. We note that $P(E_2,G_2)=2$ also, but $P(E_3,G_2)=1$, hence embed this edge in the proper face which is the I face to obtain the plane subgraph G_3.

Similar consideration to the next edges will apply and the test ends when all such edges are considered. Fig.6 illustrates all these steps and determines $E_9=(2,6)$ and $E_{10}=(3,8)$ are the only sets of edges to be removed from G to maximal planarize it and the spanning planar subgraph G_{10} is shown in Fig.5

Now the maximal-planarization algorithm is presented which uses the procedure described so far.

Algorithm 2

1. Let G_c be a maximal cycle of G, such that G_c is embedded in the plane.
2. List the edges starting with the initial edge, the edges of type S, the edges of type R.
3. Embed the Initial edge inside the I face; set i=1.
4. Test the next edge for possible embeddings. If there are no edges left, go to Step 9.
5. If $P(E,G_i)=0$, then G is not planar, reassign this edge to be removed from G.
6. If $P(E,G_i)=1$, go to Step 8.
7. If $P(E,G_i)=2$, then this edge will be re-examined after obtaining the plane subgrsph G_{i+1},go to step 4.
8. Embed E in the proper face, replace i by i+1 and go to step 4.
9. If Gc is Hamiltonian, then stop.
10. List the exterior vertices, and examine them, if some of the outward edges could not be embedded, reassign them as to be removed from G.

Theorem :

Algorithm 2 determines a maximal planar subgraph Gp of G, if only the edges of type S are considered for testing before the edges of type R.

Proof :

Note that the edges of type S were obtained from the EQUIVALENT-GRAPH procedure. So it follows that any one of these edges may be contained in the maximal subgraph G_c where in this case they could not be deleted. This means that they must have priorities to be tested before the other edges. Let G be an equivalent-graph of the non planar graph. It is assumed that G_i is G-extendable ($i \geq 1$)and shown that G_{i+1} is G- extendable, since we can extend the embedding of Gi to a plane embedding of G. Select E and P, as in step No.6.If $P(E, G_i)=1$, then E is embedded in the proper face (I or O) then G_{i+1} is G-extendable. Otherwise, $P(E,G_i)=2$, then there are two possible embeddings. If E is embedded in I face, then, one again G_{i+1} is G-extendable.

Suppose, to the contrary, that E is drawn O face, further, assume that Ex is an edge of Gi in G that is embedded in I face whose vertices of Gi belong to the common boundary of E and Ex.Then E and Ex must be interchanged across this common boundary procedure- a new embedding of G in the plane in which E is embedded in the I face. Hence to overcome this permutation procedure must not be embedded by making an arbitrary choice in face I or O, until its testing for the possible embedding becomes one. Then E is embedded in the only one proper face.

6. SUMMARY AND CONCLUSION

This paper presents the MAXIMAL-PLANARIZE algorithm which constructs a maximal planar subgraph G_p of a nonplanar graph G. This algorithm is based on the Demoucrot, et. al planarity testing algorithm as cited by ref[2] and on the new EQUIVALENT-GRAPH procedure which is presented in this paper.

Procedure EQUIVALENT-GRAPH is used to construct a planar subgraph G_c and determine the type of the examining edges which play an important role in constructing the maximal planar graph.

The MAXIMAL-PLANARIZE algorithm is implemented in BASIC and tested on several nonplanar graphs. In Table No.3 the number of edges that needs to be removed is shown by this algorithm compared with the result from[4]. In addition, the two algorithms in [4] do not seem to lend themselves to easy modification resulting in such planarization algorithms.

We expect this algorithm to require on the average, less computation time since only one algorithm is needed and less number of removing edges, to construct a spanning planar subgraph of a given nonplanar graph.

REFERENCES:

1. Even, S."Graph algorithms", Computer Science Press, 1979.
2. Chartrand, G. & Lesniak, I." Graphs and Digraphs", Wardsworth & Brooks/cole, 1986.
3. Rubin, F."An Improved Algorithm for Testing The Planarity of a Graph", IEEE Trans. On computers Vol. C-24, No.2, pp.113-121, Feb.1975.
4. Jayakumar, R. Thulasirman, K. and Swamy, M.N.S." O (n^2) algorithms for Graph Planarization", IEEE. Trans. On Computer-Aided Design, Vol. 8,No.3, pp.257-267, March, 1989.
5. Thulasirman, K. and Swamy, M.N.S."Graphs, Networks, and Algorithms", WILEY Interscience, 1981.
6. Bang-Jensen, J. and Gutin, G. "Digraphs: Theory, Algorithms and Application." Springer-Verlag, London, 2000.
7. Kozen, C."The Design and Analysis of Algorithms", Springer-Verlag 1991.

Table-1- Example 1,a sample graph G consists of 10 vertices

1-2 1-4 1-6 1-7 1-10	2-1 2-3 2-6 2-8 2-9	3-2 3-4 3-9	4-1 4-3 4-5 4-8	5-4 5-6 5-8 5-9	6-1 6-2 6-5 6-7 6-9	7-1 7-6 7-9 7-10	8-2 8-4 8-5 8-10	9-2 9-3 9-5 9-6 9-7 9-10	10-1 10-7 10-8 10-9

Table-2- Illustrates the EQUIVALENT GRAPH procedure

V \ d(v)	G	G1	G2	G3	G4	G5	G6	G7
1	5	5*	4	4	4**	$3_{(X6)}$	0	
2	5*	4	4	4^{*s}	3	3**	$2_{(X7)}$	0
3	$3_{(X1)}$	0						
4	4**	$3_{(X2)}$	0					
5	4	4**	$3_{(X3)}$	0				
6	5	5	5*	4	4	4*	3 ** x_{kn}	$1_{xk(x8)}$
7	4	4	4	4	4^{*s}	3*s	$2_{(xa)}$	0
9	6^*_{xk}	5	5*	4	4^{*s}	3	3^{*s}	1
10	4	4	4	4**	$3_{(X5)}$	0		

stop
{ $x_8=x_k$ }

Table-3- Comparison between MAXIMAL-PLANARIZE algorithm and the result from [4]].

Graph	No. of Vertices	No.of Edges	No.of Edges removed by [4]	No. of edges removed By this procedure
G_1	10	22	3	2
G_2	10	35	18	12
G_3	20	60	24	13
G_4	30	95	37	21
G_5	40	125	38	29
G_6	50	150	43	33

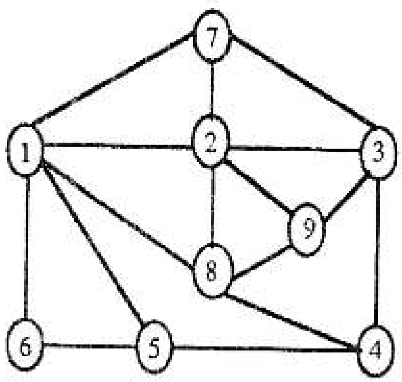
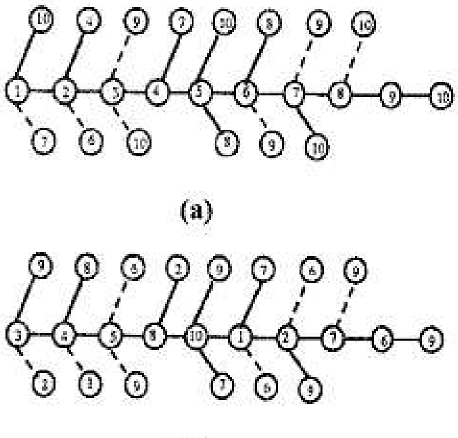


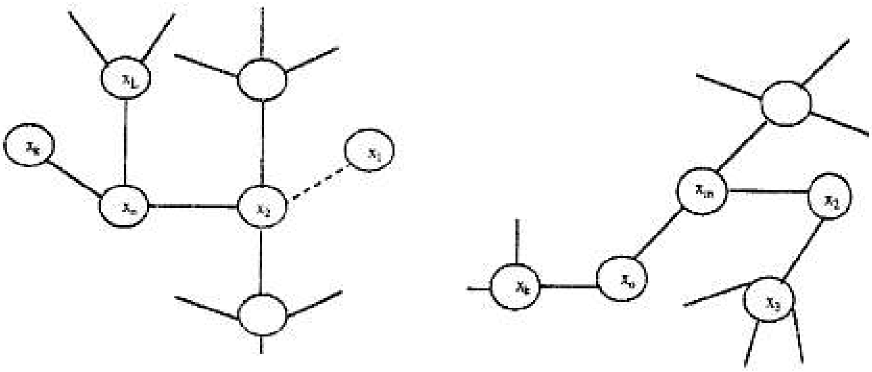
Fig. 1. Graph G



(a)

(b)

Fig. 2. Represent the equivalent graph



(a) (b)

Fig. 3. Illustrates algorithm 1

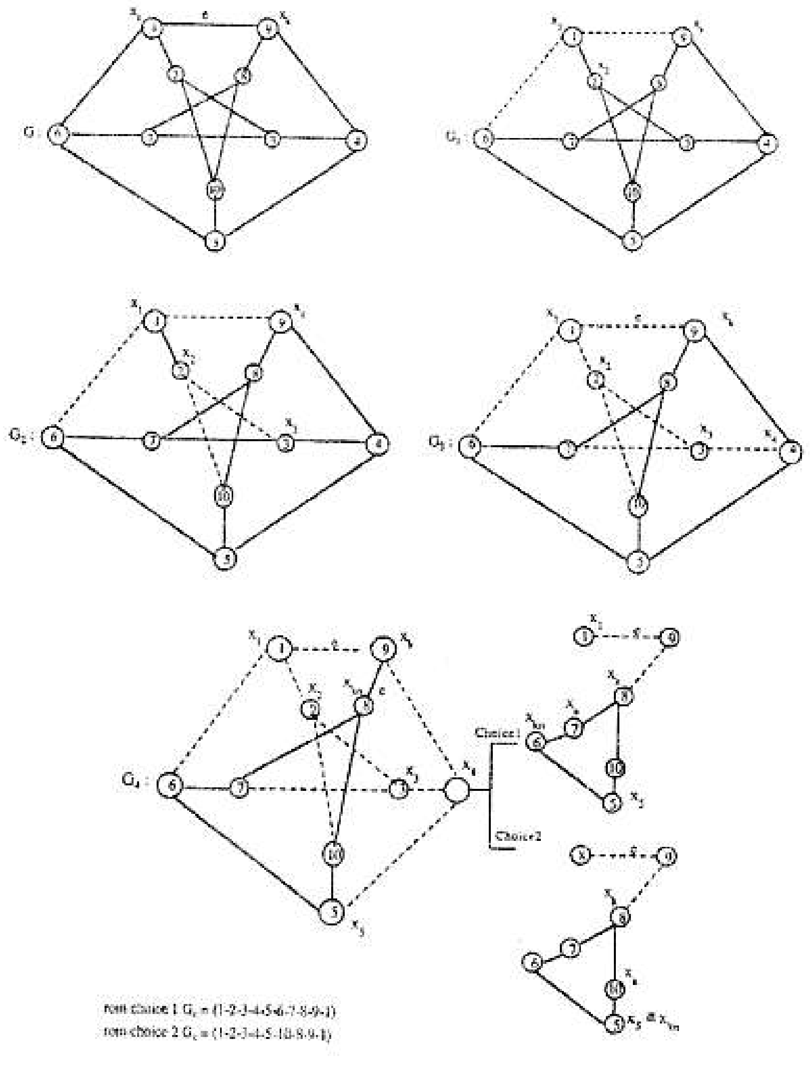
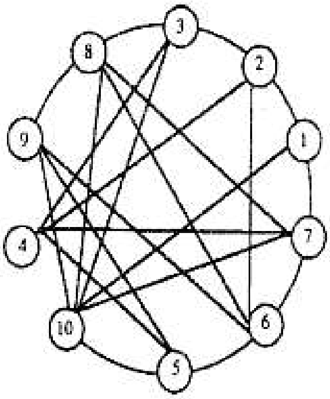
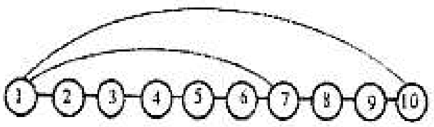


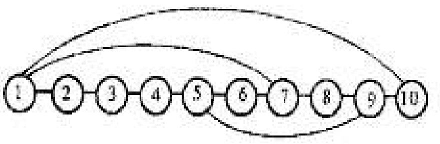
Fig. 4 . An example illustrating algorithm 1

G:


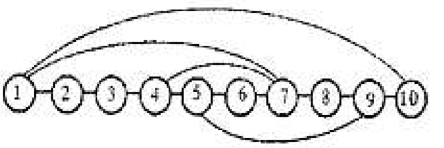
Initial Edge	E_1	(1,7)
The Edges of Type S	E_2	(2,4)
	E_3	(4,7)
	E_4	(5,9)
	E_5	(5,10)
	E_6	(6,9)
	E_7	(7,10)
	E_8	(8,10)
The Edges of Type R	E_9	(2,6)
	E_{10}	(3,8)
	E_{11}	(3,10)
	E_{12}	(6,8)

G_1:


$P(E_i , G_1) = 2$ [i =2,3]
$P(E_4 , G_1) = 1$ [O Face]

G_2:


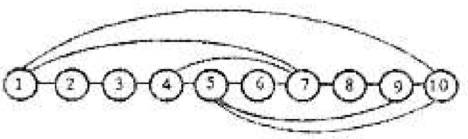
$P(E_2 , G_2) = 2$
$P(E_5 , G_2) = 1$ [1 Face]

G_3:


$P(E_2 , G_3) = 2$
$P(E_5 , G_3) = 1$ [O Face]

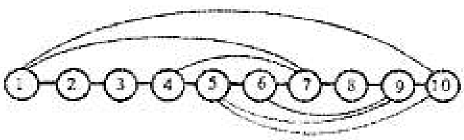
Continued on the next page

G_4:



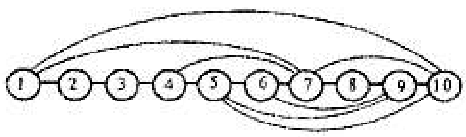
$P(E_2, G_4) = 2$
$P(E_6, G_4) = 1$ [O Face]

G_5:



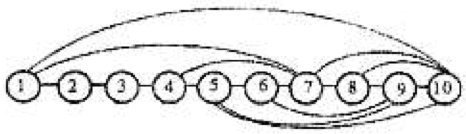
$P(E_2, G_5) = 2$
$P(E_7, G_5) = 1$ [I Face]

G_6:



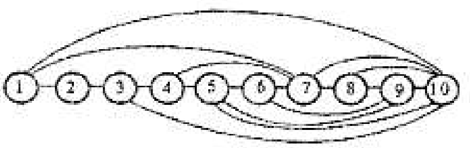
$P(E_2, G_6) = 2$
$P(E_8, G_6) = 1$ [I Face]

G_7:



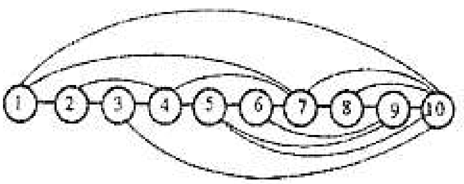
$P(E_2, G_7) = 2$
$P(E_i, G_7) = \varnothing$ [i=9,10]
$P(E_{11}, G_7) = 1$ [O Face]

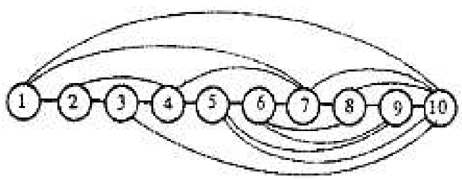
G_8:

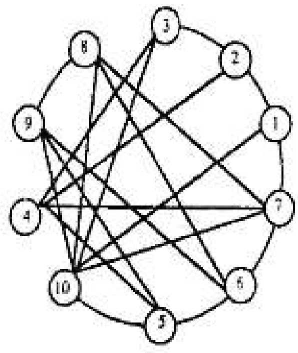


$P(E_2, G_8) = 1$ [I Face]

Continued on the next page

G_9:  $P(E_{12}, G_9) = 1$ [O Face]

G_{10}: 

G_p: 

$G_{10} = G_p$

G: non planar graph

$G_1, G_2, \ldots G_{10}$: represent logical sequences of steps to get G_p

G_p: maximal planar subgraph

Fig. 5. Example 2.