

AN EFFICIENT ALGORITHM TO FIND A HAMILTONIAN CIRCUIT  
IN A 4-CONNECTED MAXIMAL PLANAR GRAPH

T. ASANO

Department of Mathematical Engineering and Instrumentation Physics  
Faculty of Engineering, University of Tokyo  
Bunkyo-ku, Tokyo, Japan 113

S. KIKUCHI and N. SAITO

Department of Electrical Communications  
Faculty of Engineering, Tohoku University,  
Sendai, Japan 980

**Abstract.** This paper describes an efficient algorithm to find a Hamiltonian circuit in an arbitrary 4-connected maximal planar graph. The algorithm is based on our simplified version of Whitney's proof of his theorem: every 4-connected maximal planar graph has a Hamiltonian circuit.

## 1. Introduction

The Hamiltonian circuit problem is one of the most popular NP-complete problems, and remains NP-complete even if we restrict ourselves to a class of (3-connected cubic) planar graphs [4,8]. Therefore, there seems to be no polynomial-time algorithm for the Hamiltonian circuit problem. However, for (nontrivial) certain classes of restricted graphs, there exist polynomial-time algorithms [2,3,5]. In fact, employing the proof technique used by Tutte [9], Gouyou-Beauchamps has given an  $O(n^3)$  time algorithm for finding a Hamiltonian circuit in a 4-connected planar graph  $G$ , where  $n$  is the number of vertices of  $G$  [5]. Although such a graph  $G$  always has a Hamiltonian circuit [9], it is not an easy matter to find actually a Hamiltonian circuit of  $G$ . However, for a little more restricted class of graphs, i.e., the class of 4-connected maximal planar graphs, there may be an efficient algorithm. One can easily design an  $O(n^2)$  time algorithm to find a Hamiltonian circuit in a 4-connected maximal planar graph  $G$  with  $n$  vertices, entirely based on Whitney's proof of his theorem [10].

In this paper, we present an efficient algorithm for the problem, based on our simplified version of Whitney's proof of his result. We employ "divide and conquer" and some more techniques in the algorithm. The computational complexity of our algorithm is linear, so optimal within a constant factor.

## 2. Preliminaries.

We first give some of the graph theoretic concepts needed to understand our algorithm. We use definitions similar to those found in any text on graph theory, e.g., [6]. A graph  $G=(V,E)$  consists of a set  $V$  of vertices and a set  $E$  of edges. Throughout this paper,  $n$  and  $m$  denote the number of vertices and edges of  $G$ , i.e.,  $n=|V|$  and  $m=|E|$ . Each edge is an unordered pair  $(v,w)$  of distinct vertices. If  $(v,w)$  is an edge,  $v$  and  $w$  are adjacent and  $(v,w)$  is incident to both  $v$  and  $w$ . A walk of length  $k$  with endvertices  $v,w$  is a sequence  $v=v_0, v_1, v_2, \dots, v_k=w$  such that  $(v_{i-1}, v_i)$  is an edge for  $1 \leq i \leq k$ . If all the vertices  $v_0, v_1, v_2, \dots, v_{k-1}$  are distinct, the walk is a path. If  $v=w$  the path is a circuit. A path is sometimes referred as the vertex set. A circuit of length two (or three) is called a 2-circuit (or triangle). A path  $v_i, v_{i+1}, \dots, v_j$  in the circuit  $R=v_0, v_1, v_2, \dots, v_{k-1}, v_0$  is called an arc of  $R$ , and denoted by  $R[v_i, v_j]$  or  $R(v_{i-1}, v_{j+1})$ . A chord of a circuit  $R=v_0, v_1, v_2, \dots, v_{k-1}, v_0$  is an edge  $(v_i, v_j)$  of  $G$  such that  $|i-j| \neq 1 \pmod{k}$ , that is, an edge joining nonconsecutive vertices  $v_i$  and  $v_j$  on  $R$ . A Hamiltonian circuit (path) of a graph  $G$  is a circuit (path) containing all vertices of  $G$ . A graph  $G_1=(V_1, E_1)$  is a subgraph of a graph  $G=(V, E)$  if  $V_1 \subseteq V$  and  $E_1 \subseteq E$ . If  $E_1 = E \cap \{(v,w) \mid v, w \in V_1\}$ ,  $G_1$  is an induced subgraph of  $G$ . The induced subgraph  $G_1$  is obtained from  $G$  by removing vertices in  $V-V_1$ , and denoted by  $G_1=G-(V-V_1)$ . A graph  $G$  is connected if any two vertices of  $G$  are joined by a path. The connected components of a graph  $G$  are its maximal connected subgraphs. A cutvertex of a graph is a vertex whose removal increases the number of connected components. A graph  $G$  is  $(k+1)$ -connected if the removal of any  $k$  or fewer vertices of  $G$  results in a connected graph. The blocks of a graph are its maximal 2-connected subgraphs. A graph is planar if it can be embedded in the plane so that its edges intersect only at their endvertices. A plane graph is a planar graph which is embedded in the plane. A plane graph divides the plane into connected regions called faces. The unbounded region is called the exterior face, and all the others are called interior faces. Each face of a 2-connected plane graph  $G$  is bounded by a curve corresponding to a circuit of  $G$ , called boundary of the face. We shall sometimes not distinguish between a face and a boundary. A maximal planar graph is a planar graph to which no edge can be added without losing planarity. Note that every face of a maximal planar graph  $G$  with  $n(\geq 3)$  vertices is a triangle. A triangle of a plane graph  $G$  is said to be a separating triangle if it is not a face of  $G$ . Refer to [1,6] for all undefined terms.

Next, introducing Whitney's condition, we describe Whitney's lemma used to establish his theorem.

Let  $G=(V,E)$  be a 2-connected plane graph with the exterior face  $R$ . Let  $A$  and  $B$  be two distinct vertices on  $R$ . If these  $G$ ,  $R$ ,  $A$  and  $B$  together satisfy the following conditions (W1) and (W2) (called Whitney's condition, or for short Condition (W)), then we say that  $(G, R, A, B)$  satisfies Condition (W):

(W1) All interior faces of  $G$  are triangles, and all triangles are faces of  $G$ ;

(W2) Either

(W2a)  $R$  is divided into two arcs  $R_1 = R[A, B] = a_0 a_1 \dots a_r$  and  $R_2 = R[B, A] = b_0 b_1 \dots b_s$

$(a_0 = b_s = A, a_r = b_0 = B)$ , and there are no chords of  $R$  joining two vertices on  $R_i$  ( $1 \leq i \leq 2$ ), or

- (W2b)  $R$  is divided into three arcs  $R_1 = R[A, B] = a_0 a_1 \dots a_r$ ,  $R_2 = R[B, C] = b_0 b_1 \dots b_s$  and  $R_3 = R[C, A] = c_0 c_1 \dots c_t$  for some vertex  $C$  on  $R(B, A)$  ( $a_0 = c_t = A, a_r = b_0 = B, b_s = c_0 = C$ ), and there are no chords of  $R$  joining two vertices on  $R_i$  ( $1 \leq i \leq 3$ ).

We sometimes say " $G$  satisfies Condition (W)" instead of " $(G, R, A, B)$  satisfies Condition (W)" if there is no confusion. It should be noted that the exterior face of  $K_2$  (the complete graph of two vertices) is a circuit of length two under our definition although it is not a circuit under Whitney's definition. Thus  $K_2$  satisfies Condition (W), since  $K_2$  has no interior faces. This observation can greatly simplify the proof of the following Whitney's lemma, from which his theorem immediately follows.

**Lemma 1.** [10] Let  $G$  be a 2-connected plane graph with the exterior face  $R$ , and let  $A$  and  $B$  be two distinct vertices of  $R$ . If  $(G, R, A, B)$  satisfies Condition (W), then  $G$  has a Hamiltonian path connecting  $A$  and  $B$ .

Based on the proof of Whitney's lemma, one can easily design an  $O(n^2)$  algorithm for finding a Hamiltonian circuit in a 4-connected maximal planar graph  $G$  with  $n$  vertices. In order to design a linear algorithm, we now introduce Condition (X), which is the same as Condition (W) except for Condition (W2b) above being replaced with the following Condition (X3):

- (X3)  $R$  is divided into three arcs  $R_1 = R[A, B] = a_0 a_1 \dots a_r$ ,  $R_2 = R[B, C] = b_0 b_1 \dots b_s$  and  $R_3 = R[C, A] = c_0 c_1 \dots c_t$  for some vertex  $C$  on  $R(B, A)$  ( $a_0 = c_t = A, a_r = b_0 = B, b_s = c_0 = C$ ), and there are no chords of  $R$  joining two vertices on  $R_i$  ( $1 \leq i \leq 3$ ). Moreover, there exists either chord  $(b_{s-1}, c_k)$  or  $(c_1, b_j)$ .

**Remark 1.** Whenever  $(G, R, A, B)$  satisfies Condition (W), we can choose some vertex as  $C$  so that  $G$  may satisfy Condition (X) (sometimes  $C$  may disappear) by scanning vertices on  $R_3$  from  $C$  to  $A$  or vertices on  $R_2$  from  $C$  to  $B$ . If  $(G, R, A, B)$  satisfies Condition (X), then it clearly satisfies Condition (W) and no chord of  $R$  joins  $C$  and other vertex (see Fig. 1).

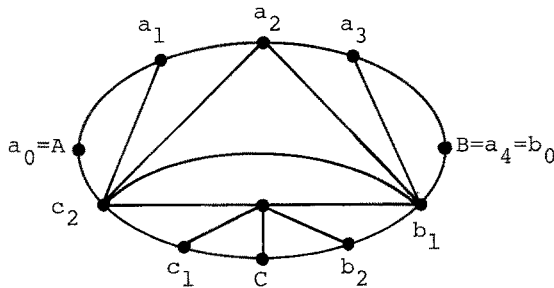


Fig. 1. A plane graph  $G$  satisfying Condition (W). If one take  $c_1$  or  $b_2$  as  $C$ , then  $G$  satisfies Condition (X), since there is a chord  $(c_2, b_1)$ .

We now obtain the following lemma.

**Lemma 2.** Let  $G$  be a 2-connected plane graph with the exterior face  $R$ , and let  $A$  and  $B$  be two distinct vertices of  $R$ . If  $(G, R, A, B)$  satisfies Condition (X), then  $G$  has a Hamiltonian path connecting  $A$  and  $B$ .

### 3. An outline of the algorithm.

This section sketches the idea behind our algorithm. We first apply the linear planar embedding algorithm [7] in order to embed a given planar graph in the plane. Thus we can assume that a 4-connected maximal plane graph  $G=(V,E)$  with the exterior face  $R=1,2,3,1$  is given, where 1, 2 and 3 are vertices of  $G$ . Clearly  $(G, R, 1, 2)$  satisfies Condition (X), so that  $G$  has a Hamiltonian path connecting vertices 1 and 2 by Lemma 2. Thus we can easily obtain a Hamiltonian circuit by combining the Hamiltonian path and edge  $(1,2)$ . So we may consider only the algorithm to find a Hamiltonian path connecting  $A$  and  $B$  for a graph  $G$  such that  $(G, R, A, B)$  satisfies Condition (X). We add one more definition.

Let  $G$  be a 2-connected plane graph such that  $(G, R, A, B)$  satisfies Condition (X). If a subgraph  $G'$  of  $G$  satisfies the following conditions (A1)–(A5), then we say that  $G'$  satisfies Condition (A):

- (A1)  $G'$  is a connected spanning subgraph of  $G$ ;
- (A2)  $G'$  consists of  $g$  blocks  $G_1, G_2, \dots, G_g$  ( $g \geq 2$ ) with  $(g-1)$  cutvertices  $x_1, x_2, \dots, x_{g-1}$ , where each  $x_f$  ( $1 \leq f \leq g-1$ ) belongs to exactly two blocks  $G_f$  and  $G_{f+1}$ ;
- (A3) Neither  $A$  nor  $B$  is a cutvertex of  $G'$ ;
- (A4)  $A$  is a vertex of  $G_1$  and  $B$  is a vertex of  $G_g$ ; and
- (A5) Let  $x_0=A$  and  $x_g=B$ , and let  $Q_f$  ( $1 \leq f \leq g$ ) be the exterior face of the plane subgraph  $G_f$  of  $G$ . Then each  $(G_f, Q_f, x_f, x_{f-1})$  ( $1 \leq f \leq g$ ) satisfies Condition (W) (see Fig. 2).

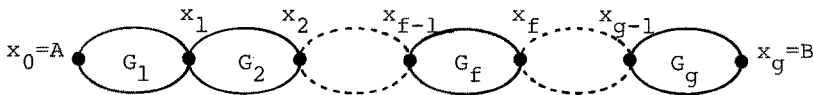


Fig. 2. Graph  $G'$  satisfying Condition (A), where each block  $G_f$  ( $1 \leq f \leq g$ ) satisfies Condition (W).

Our Hamiltonian path algorithm now can be outlined as follows:

First delete one or more edges from  $G$  so that the resulting graph  $G'$  satisfies Condition (A);

next for each  $G_f$  ( $1 \leq f \leq g$ ) choose  $C$  appropriately so that  $(G_f, Q_f, x_f, x_{f-1})$  may satisfy Condition (X); then obtain a Hamiltonian path  $H(G_f, x_f, x_{f-1})$  connecting  $x_f$  and  $x_{f-1}$  by recursively applying the algorithm for each  $G_f$  ( $1 \leq f \leq g$ ); and finally obtain a Hamiltonian path  $H(G, A, B)$  by combining all  $H(G_f, x_f, x_{f-1})$ 's. In Fig. 3 below is an outline of the algorithm written by Pidgin ALGOL [1].

procedure HPATH( $G, R, A, B$ ):

```

  begin comment an outline of the algorithm to find a Hamiltonian path in  $G=(V, E)$  connecting
  A and B such that  $(G, R, A, B)$  satisfies Condition (X). The set of edges marked by the
  procedure is a Hamiltonian path of  $G$  connecting A and B;
1  if  $|V|=2$  then  $(A, B)$  is a Hamiltonian path of  $G$  connecting A and B, so mark  $(A, B)$ 
  else begin
2    delete appropriate edges from  $G$  so that the resulting graph  $G'$  satisfies Condition (A);
3    for each block  $G_f$  of  $G'$  do
      begin comment  $(G_f, Q_f, x_f, x_{f-1})$  satisfies Condition (W);
      choose  $C$  appropriately so that  $(G_f, Q_f, x_f, x_{f-1})$  may satisfy Condition (X);
      HPATH( $G_f, Q_f, x_f, x_{f-1}$ )
      end
    end
  end;

```

Fig. 3 An outline of the algorithm.

Remark 2. We can always execute line 2 because  $(G, R, A, B)$  satisfies Condition (X) (we present in Section 4 a method to determine which edges are to be deleted). Noting Remark 1 we can always execute lines 3-5. Thus it is easy to show, by induction on the number of edges of  $G$ , that the algorithm correctly finds a Hamiltonian path of  $G$ , because  $G'$  satisfies Condition (A).

In order to make it easy to analyze the time complexity of the procedure HPATH, we define an execution tree.

An execution tree  $TR(G, R, A, B)$  for the procedure HPATH( $G, R, A, B$ ) is recursively defined as follows:

- (i)  $TR(G, R, A, B)$  is a rooted tree whose root is  $(G, R, A, B)$ ;
- (ii) If  $|V|=2$  then  $(G, R, A, B)$  has no sons. Otherwise the  $(G_f, R_f, x_f, x_{f-1})$  is the  $f^{\text{th}}$  left son of  $(G, R, A, B)$  and is the root of the execution tree  $TR(G_f, R_f, x_f, x_{f-1})$ , where  $G_f$  is the  $f^{\text{th}}$  block of  $G'$  obtained from  $G$  by the execution of line 2.

Let  $V(G)$  denote the vertex set of a graph  $G$ . Let  $EX(G, R, A, B)$  denote the set of vertices of  $G$  which newly appear on the exterior face of  $G'$ . Let  $CV(G, R, A, B)$  denote the set of vertices of  $G$  which newly become cutvertices of  $G'$ . Let  $(F, R_F, A_F, B_F)$  and  $(H, R_H, A_H, B_H)$  be two distinct vertices of the execution tree  $TR(G, R, A, B)$ . It is clear that if  $(H, R_H, A_H, B_H)$  is neither a descendant nor an ancestor of  $(F, R_F, A_F, B_F)$  in  $TR(G, R, A, B)$ , then  $V(F) \cap V(H) = \emptyset$ . Since  $G'$  satisfies Condition (A) we can observe the following fact.

Remark 3. Let  $(H, R_H, A_H, B_H)$  be a descendant of  $(F, R_F, A_F, B_F)$  in

$TR(G, R, A, B)$ , and let  $x$  be a vertex of both  $F$  and  $H$ . Then

- (R1) If  $x$  is on the exterior face  $R_F$  of  $F$  then  $x$  is on the exterior face  $R_H$  of  $H$ ;
- (R2) If  $x$  is a cutvertex of  $F'$  ( $F'$  is the graph satisfying Condition (A), which is obtained from  $F$  by the execution of line 2 in  $HPATH(F, R_F, A_F, B_F)$ ), then  $x$  is one of endvertices of the Hamiltonian path of  $H$ , that is,  $x=A_H$  or  $x=B_H$ ;
- (R3) If  $x$  is an endvertex of the Hamiltonian path of  $F$  (that is,  $x=A_F$  or  $x=B_F$ ), then  $x$  is not a cutvertex of  $F'$  and  $x=A_H$  or  $x=B_F$ .

By Remark 3 we have the following remarks.

**Remark 4.** Let  $(F, R_F, A_F, B_F)$  and  $(H, R_H, A_H, B_H)$  be two distinct vertices of the execution tree  $TR(G, R, A, B)$ . Then

$$EX(F, R_F, A_F, B_F) \cap EX(H, R_H, A_H, B_H) = \emptyset, \text{ and} \\ CV(F, R_F, A_F, B_F) \cap CV(H, R_H, A_H, B_H) = \emptyset.$$

**Remark 5.** Let  $T(G, R, A, B)$  denote the time spent by the  $HPATH(G, R, A, B)$  for the graph  $G=(V, E)$ . Let  $T'(G, R, A, B)$  denote the time spent by the  $HPATH(G, R, A, B)$ , exclusive of the time spent by its recursive calls. We claim

$$T'(G, R, A, B) \leq K \left( \sum_{v \in EX(G, R, A, B)} d(v) + \sum_{v \in CV(G, R, A, B)} d(v) \right) \quad (1)$$

for any  $(G, R, A, B)$  satisfying Condition (X), where  $K$  is constant and  $d(v)$  denotes the degree of vertex  $v$  of  $G$ . Noting Remark 4 and the fact that  $G$  is planar we obtain

$$T'(G, R, A, B) \leq K \left( \sum_{v \in V} d(v) + \sum_{v \in V} d(v) \right) \leq 4K|E| \leq 12K|V|.$$

Thus (1) implies that the algorithm is linear. We shall verify (1) in Section 5.

#### 4. Proof of Lemma 2.

In this section we give the proof of Lemma 2 which is a simplified version of Whitney's proof of Lemma 1. Since the proof is constructive, we can easily design an algorithm based on the proof.

We proceed to prove Lemma 2 by induction on the number of edges of  $G=(V, E)$ . Let  $m=|E|$ . The claim is obviously true if  $m=1$  (that is,  $G=K_2$ ). For the inductive step, we assume that the claim is true for all graphs with at most  $m-1$  edges ( $m \geq 2$ ). We should show that the claim is true for any 2-connected plane graph with  $m$  edges.

Let  $G$  be a 2-connected plane graph with  $m$  edges. We consider the exterior face  $R$  of  $G$  as a sequence of vertices on  $R$  ordered in a clockwise sense, and denote three arcs of  $R$  by  $R_1 = R[A, B] = a_0 a_1 \dots a_r$  ( $a_0 = A, a_r = B$ ),  $R_2 = R[B, C] = b_0 b_1 \dots b_s$  ( $b_0 = B, b_s = C$ ), and  $R_3 = R[C, A] = c_0 c_1 \dots c_t$  ( $c_0 = C, c_t = A$ ). If  $(G, R, A, B)$  satisfies Condition (W2a), then  $R_3$  is empty and  $R_2$

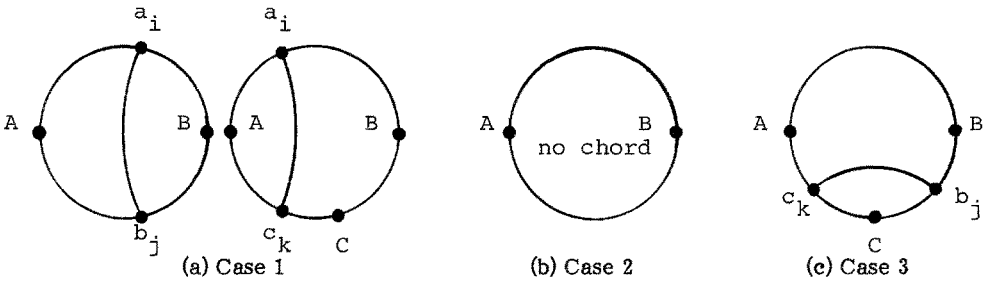


Fig. 4 Three cases in the proof.

$=R[B,A]$ , that is,  $b_s=A$ . Since  $(G,R,A,B)$  satisfies Condition (X), no chord of  $R$  joins two vertices on the same arc, and no chord joins vertex  $C$  (if any) and a vertex of  $R$ . We have three cases according to the types of the chords of  $R$  (see Fig. 4).

Case 1.  $R$  has a chord of form  $(a_i, b_j)$  ( $0 \leq i \leq r-1$ ,  $1 \leq j \leq s-1$ ) or  $(a_i, c_k)$  ( $1 \leq i \leq r$ ,  $1 \leq k \leq t-1$ ).

Case 2.  $R$  has no chords. (In this case  $G$  satisfies Condition (W2a).)

Case 3 (the remaining case).  $R$  has no chords of form  $(a_i, b_j)$  or  $(a_i, c_k)$ , but has a chord of form  $(b_j, c_k)$  ( $1 \leq j \leq s-1$ ,  $1 \leq k \leq t-1$ ).

Note that vertex  $C$  of  $R$  disappears in Case 2 since  $(G,R,A,B)$  satisfies Condition (X). This is one of the reason why we introduce Condition (X). This fact together with technique finding a  $Q$ -chain (defined later) enables us to design a linear algorithm.

Case 1. We can assume without loss of generality that  $R$  has a chord of form  $(a_i, b_j)$ . If  $R$  has a chord of form  $(a_i, c_k)$  it suffices to interchange the roles of  $A$  and  $B$  and of  $c_k$  and  $b_j$ . Suppose that  $(a_i, b_j)$  is the chord nearest  $B$  among all chords of this form, that is, the circuit  $a_i a_{i+1} \dots a_{r-1} B b_1 \dots b_j a_i$  has no chord. Now either,

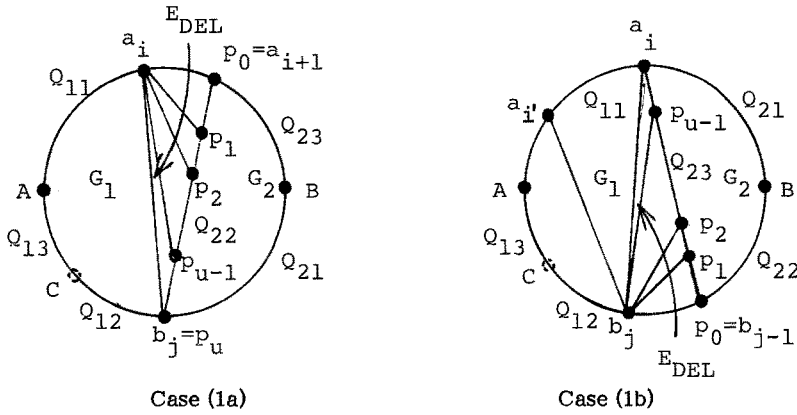
Case (1a),  $R$  has no chords other than  $(a_i, b_j)$  joining  $b_j$  and a vertex on  $R_1$ , or

Case (1b), otherwise, that is,  $R$  has a chord other than  $(a_i, b_j)$  joining  $b_j$  and a vertex on  $R_1$ . (See Fig. 5.)

We first consider Case (1a). Let  $p_0, p_1, p_2, \dots, p_u$  ( $p_0=a_{i+1}$ ,  $p_u=b_j$ ) be a sequence of vertices adjacent to  $a_i$  such that each  $(a_i, p_k)$  is the immediately clockwise edge of  $(a_i, p_{k-1})$  around  $a_i$ . Since  $(G,R,A,B)$  satisfies Condition (W1), all  $(p_k, p_{k+1})$  ( $0 \leq k \leq u-1$ ) are edges of  $G$ , and there are no edges of form  $(p_k, p_{k'})$  ( $0 \leq k, k+2 \leq k' \leq u$ ). Let

$$E_{DEL} = \{(a_i, p_k) \mid 0 \leq k \leq u-1\}.$$

Delete all edges in  $E_{DEL}$  from  $G$ , and let  $G'$  be the resulting graph, i.e.,  $G'=G-E_{DEL}$ . Then  $G'$  consists of two blocks  $G_1$  and  $G_2$ , one of which contains  $A$  and the other  $B$ . Both  $G_1$  and  $G_2$  have fewer edges than  $G$ . Moreover, let  $Q_{11}=a_0 a_1 \dots a_i b_j$ ,  $Q_{12}=b_j b_{j+1} \dots b_s$ ,  $Q_{13}=R_3$ , and  $Q_1=Q_{11} Q_{12} Q_{13}$ , then  $(G_1, Q_1, A, b_j)$  satisfies Condition (X), where  $Q_{11}$ ,  $Q_{12}$  and  $Q_{13}$  are three arcs (possibly  $Q_{13}$  is empty) concerned in this case. Similarly let  $Q_{21}=b_0 b_1 \dots b_j$ ,  $Q_{22}=p_u p_{u-1} \dots p_0$ ,  $Q_{23}=a_{i+1} \dots a_s$  (if  $p_0=a_r=B$  then  $Q_{23}$  is empty), and  $Q_2=Q_{21} Q_{22} Q_{23}$ . Then  $(G_2, Q_2, B, b_j)$  satisfies Condition (W). Thus the resulting graph  $G'$  of  $G$  satisfies Condition (A). Noting Remark 1, we can choose a vertex as  $C$  appropriately so that  $(G_2, Q_2, B, b_j)$  satisfies Condition (X). By the inductive hypothesis  $G_1$  has a

Fig. 5 Case 1, where  $(a_i, b_j)$  is the chord nearest B.

Hamiltonian path  $H(G_1, A, b_j)$  connecting A and  $b_j$  and  $G_2$  has a Hamiltonian path  $H(G_2, B, b_j)$  connecting B and  $b_j$ . Thus we obtain a Hamiltonian path  $H(G, A, B)$  of  $G$  connecting A and B by combining  $H(G_1, A, b_j)$  and  $H(G_2, B, b_j)$ .

Next consider Case (1b). In this case let  $p_0, p_1, p_2, \dots, p_u$  ( $p_0 = b_{j-1}$ ,  $p_u = a_i$ ) be a sequence of vertices adjacent to  $b_j$  such that each  $(b_j, p_k)$  is the immediately counter-clockwise edge of  $(b_j, p_{k-1})$  around  $b_j$ . We delete all edges in  $E_{DEL} = \{(b_j, p_k) \mid 0 \leq k \leq u-1\}$ . An argument similar to one in Case (1a) shows that  $G$  has a Hamiltonian path  $H(G, A, B)$  connecting A and B (see Fig. 5).

Case 2. This case can be considered to be a special case of either Case (1a) with  $a_i = A$  and  $b_j = b_{s-1}$  or Case (1b) with  $a_i = a_1$  and  $b_j = b_1 = A$  (see Fig. 6). An argument similar to one above works. Note that  $G_1 = K_2$  satisfies Condition (X).

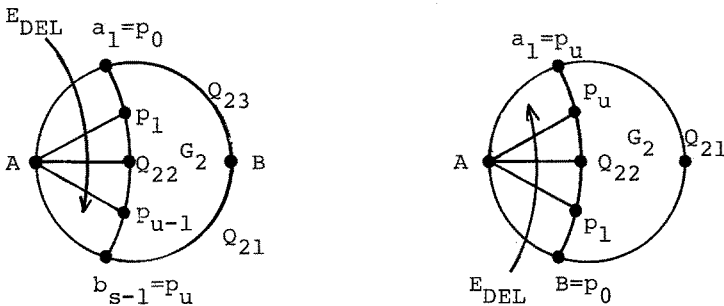
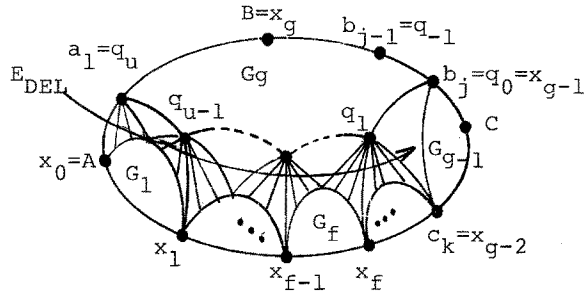


Fig. 6 Case 2.

Case 3. Suppose that  $(b_j, c_k)$  is the chord furthest from vertex C of  $R$ . Note that  $b_j$  is one of  $b_1, \dots, b_{s-1}$  and  $c_k$  is one of  $c_1, \dots, c_{t-1}$ . Let  $Q = q_0, q_1, q_2, \dots, q_u$  ( $q_0 = b_j$ ,  $q_u = a_1$ ) be a sequence of vertices which satisfies the following: Each  $(q_i, q_{i+1})$  ( $0 \leq i \leq u-1$ ) is the next edge of  $(q_i, q_{i-1})$  in a counter-clockwise sense around  $q_i$  among all edges with an endvertex adjacent to a vertex on



Fig. 7 Q-chain and  $G'$  satisfying Condition (A).

$R_3$ , where  $q_{-1} = b_{j-1}$ . Such a sequence is called a Q-chain of  $R$ . The existence of the Q-chain is verified as follows. Since  $(b_j, c_k)$  is the chord furthest from  $C$ , and every interior face is triangle, there exists a vertex  $x$  inside the circuit  $A a_1 \dots a_{r-1} B b_1 \dots b_j c_k \dots c_{t-1} A$  adjacent to both  $b_j$  and  $c_k$ . Thus  $q_1$  always exists (possible  $q_1 = x$ ). If  $q_1 = a_1$  then  $q_u = q_1$ . Otherwise, let  $c_{k1}$  be the vertex furthest from  $C$  among all vertices on  $R_3$  adjacent to  $q_1$ . Similarly we can easily show that  $q_2$  exists, since there is a vertex inside the circuit  $A a_1 \dots a_{r-1} B b_1 \dots b_j q_1 c_{k1} c_{k1+1} \dots c_{t-1} A$  adjacent to both  $q_1$  and  $c_{k1}$ . Repeating this argument, we can prove that the Q-chain always exists since  $a_1$  is adjacent to  $A$ . Note that any vertex of  $q_1, \dots, q_{u-1}$  is not on  $R$ , and the circuit  $D = q_u a_2 \dots a_{r-1} B b_1 \dots b_j q_1 \dots q_u$  has no chord of form  $(q_i, q_{i'})$  ( $0 \leq i, i+2 \leq i' \leq u$ ). Let  $E_{OUT}(D)$  be the set of edges adjacent to  $q_1, q_2, \dots$ , or  $q_u$  outside  $D$ , and let

$$E_{DEL} = E_{OUT}(D) \cup CL(q_0, c_k, q_1) \quad (2)$$

where  $CL(q_0, c_k, q_1)$  denotes the set of edges incident to  $q_0$ , from  $c_k$  to  $q_1$  clockwise. Let  $G' = G - E_{DEL}$ , then  $G'$  consists of  $g (\geq 3)$  blocks, and satisfies Condition (A) (see Fig. 7). Each block  $G_f$  of  $G'$  has at most  $m-1$  edges. Noting Remark 1, we can choose a vertex as  $C$  of  $Q_f$  so that  $(G_f, Q_f, x_f, x_{f-1})$  may satisfy Condition (X), where  $Q_f$  ( $1 \leq f \leq g$ ) is the exterior face of  $G_f$  and  $x_f$  ( $1 \leq f \leq g-1$ ) is the cutvertex of  $G'$  belonging to both  $G_f$  and  $G_{f+1}$  ( $x_0 = A, x_g = B$ ). By the inductive hypothesis, each  $G_f$  ( $1 \leq f \leq g$ ) has a Hamiltonian path  $H(G_f, x_f, x_{f-1})$  connecting  $x_f$  and  $x_{f-1}$ . Thus we obtain a Hamiltonian path  $H(G, A, B)$  of  $G$  connecting  $A$  and  $B$  by combining all  $H(G_f, x_f, x_{f-1})$ 's.

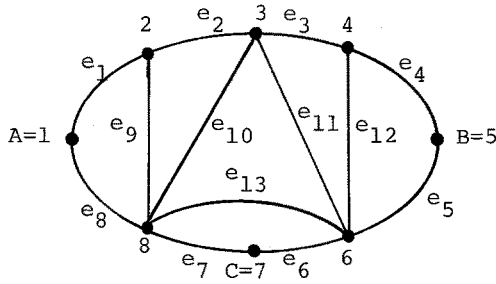
This completes the proof.

Q.E.D.

##### 5. The Hamiltonian path algorithm.

The proof of Lemma 2 leads to an algorithm for finding a Hamiltonian path in a graph satisfying Condition (X). To make the algorithm efficient, we need a good representation of a plane graph. (We assume that a given graph satisfying Condition (X) is already embedded in the

plane by a linear planar embedding algorithm [8].) For this purpose we use a list structure whose elements correspond to the edges of the graph. Stored with each edge  $e=(x,y)$  are its endpoints  $x$  and  $y$ , and four pointers  $c_1(e)$ ,  $c_2(e)$ ,  $cc_1(e)$  and  $cc_2(e)$ , designating the edges immediately clockwise and counter-clockwise around the endpoints of the edges. Stored with each vertex  $x$  are two edges  $c(x)$  and  $cc(x)$  incident to  $x$  which indicate the starting edge and the final edge of the adjacency list  $A(x)$ , where  $c(x)$  is the immediately clockwise edge of  $cc(x)$  around  $x$ . Furthermore, we need a representation of the exterior faces of blocks of a graph. For this purpose we use another list structure together with an array. Each exterior face  $R$  of a simple block  $G$  has pointers, designating  $A, B$ , and  $C$  of  $R$ , chords of  $R$  and so on. Pointers  $R(A)$ ,  $R(B)$  and  $R(C)$  represent current vertices  $A, B$  and  $C$  of the exterior face  $R$ . The set of chords of  $R$  are partitioned into three classes: the set  $D12$  of chords of form  $(a_i, b_j)$ ; the set  $D13$  of chords of form  $(a_i, c_k)$ ; and the set  $D23$  of chords of form  $(b_j, c_k)$ . The chords of  $D12$  are arranged in nearest order to  $B$ . The chords of  $D13$  and  $D23$  are arranged in furthest order from  $C$ .  $R(D12)$  stores the chord of  $D12$  nearest  $B$ .  $R(D13)$  and  $R(D23)$  store the



Vertex incidences

	c	cc
1	e <sub>1</sub>	e <sub>8</sub>
2	e <sub>2</sub>	e <sub>1</sub>
3	e <sub>3</sub>	e <sub>2</sub>
4	e <sub>4</sub>	e <sub>3</sub>
5	e <sub>5</sub>	e <sub>4</sub>
6	e <sub>6</sub>	e <sub>5</sub>
7	e <sub>7</sub>	e <sub>6</sub>
8	e <sub>8</sub>	e <sub>7</sub>

Edges, neighbors and next chords

	1	2	c <sub>1</sub>	cc <sub>1</sub>	c <sub>2</sub>	cc <sub>2</sub>	NEXT
e <sub>1</sub>	1	2	e <sub>8</sub>	e <sub>8</sub>	e <sub>2</sub>	e <sub>9</sub>	0
e <sub>2</sub>	2	3	e <sub>9</sub>	e <sub>1</sub>	e <sub>3</sub>	e <sub>10</sub>	0
e <sub>3</sub>	3	4	e <sub>11</sub>	e <sub>2</sub>	e <sub>4</sub>	e <sub>12</sub>	0
e <sub>4</sub>	4	5	e <sub>12</sub>	e <sub>3</sub>	e <sub>5</sub>	e <sub>5</sub>	0
e <sub>5</sub>	5	6	e <sub>4</sub>	e <sub>4</sub>	e <sub>6</sub>	e <sub>12</sub>	0
e <sub>6</sub>	6	7	e <sub>13</sub>	e <sub>5</sub>	e <sub>7</sub>	e <sub>7</sub>	0
e <sub>7</sub>	7	8	e <sub>6</sub>	e <sub>6</sub>	e <sub>8</sub>	e <sub>13</sub>	0
e <sub>8</sub>	8	1	e <sub>9</sub>	e <sub>7</sub>	e <sub>1</sub>	e <sub>1</sub>	0
e <sub>9</sub>	2	8	e <sub>1</sub>	e <sub>2</sub>	e <sub>10</sub>	e <sub>8</sub>	e <sub>10</sub>
e <sub>10</sub>	3	8	e <sub>2</sub>	e <sub>11</sub>	e <sub>13</sub>	e <sub>9</sub>	0
e <sub>11</sub>	3	6	e <sub>10</sub>	e <sub>3</sub>	e <sub>12</sub>	e <sub>13</sub>	0
e <sub>12</sub>	4	6	e <sub>3</sub>	e <sub>4</sub>	e <sub>5</sub>	e <sub>11</sub>	e <sub>11</sub>
e <sub>13</sub>	6	8	e <sub>11</sub>	e <sub>6</sub>	e <sub>7</sub>	e <sub>10</sub>	0

Face R and its chords

	A	B	C	D12	D13	D23
R	1	5	7	e <sub>12</sub>	e <sub>9</sub>	e <sub>13</sub>

Fig. 8 Representation of a plane graph  $(G, R, A, B)$  satisfying Condition (X).

chords of D13 and D23 furthest from C, respectively. Array NEXT(x) stores a next chord in the order above for each chord x of R. Thus the chord of D12 second nearest B is stored with NEXT(R(D12)), and so on. Stored with each vertex are flags, indicating whether or not the vertex is on R or whether or not it is adjacent to some vertex on R. Fig. 8 illustrates such a data structure. Moreover, we set  $c(v_i) = (v_i, v_{i+1})$  and  $cc(v_i) = (v_{i-1}, v_i)$  for each vertex  $v_i$  on the exterior face  $R = v_0 v_1 \dots v_{k-1} v_0$ . Thus R is represented as follows:  $v_1$  is an endvertex of  $c(v_0)$  different from  $v_0$ ;  $v_2$  is an endvertex of  $c(v_1)$  different from  $v_1$ ; and  $v_0$  is an endvertex of  $c(v_{k-1})$  different from  $v_{k-1}$ . Adjacency list A(v) in a clockwise sense is represented as follows: the first vertex of A(v) is an endvertex of  $c(v)$  different from v; the second vertex of A(v) is an endvertex of immediately clockwise edge of  $c(v)$  around v different from v; and the last vertex of A(v) is an endvertex of  $cc(v)$  different from v. Counter-clockwise adjacency list is similarly obtained. Thus we can consider that our data structure contains adjacency lists.

Now we are ready to present the algorithm. In Fig. 9 below is the algorithm to find a Hamiltonian path in a 2-connected plane graph G such that (G,R,A,B) satisfies Condition (X).

procedure HPATH(G,R,A,B):

```

  begin comment G=(V,E) is a plane graph with the exterior face R satisfying Condition (X).
  Let  $R=R_1R_2R_3$ ,  $R_1=a_0a_1\dots a_r$  ( $a_0=A$ ,  $a_r=B$ ),  $R_2=b_0b_1\dots b_s$  ( $b_0=B$ ,  $b_s=C$ ), and  $R_3=c_0c_1\dots c_t$  ( $c_0=C$ ,  $c_t=A$ ). If  $G=K_2$  then R is a 2-circuit. R has no chords which join vertices on  $R_i$  ( $i=1,2,3$ ). If  $R_3$  is not empty, then R has a chord joining either  $c_1$  and some vertex on  $R_2$  or  $b_{s-1}$  and some vertex on  $R_3$ . HPATH(G,R,A,B) finds a Hamiltonian path in G connecting A and B, whose edges are marked by HPATH;
1  if  $|V|=2$ 
2  then (A,B) is a Hamiltonian path of G from A to B so mark (A,B)
3  else if (R has a chord of forms  $(a_i, b_j)$ , or  $(a_i, c_k)$ ) or (R has no chord) then
4    begin comment Case 1 or Case 2;
5    if R has a chord of form  $(a_i, b_j)$ , or  $(a_i, c_k)$ 
6    then begin comment Case 1;
7      wlg we can assume that R has a chord of form  $(a_i, b_j)$  otherwise
      interchange the roles of A and B and of  $b_j$  and  $c_k$  in
      begin
8      let  $(a_i, b_j)$  be the edge nearest B among such edges;
9      if R has a chord other than  $(a_i, b_j)$  joining  $b_j$  and a vertex on  $R_1$  then
      begin comment Case 1(b).  $Q_1$  and  $Q_2$  correspond to the
      new exterior faces;
       $x:=b_j$ ;  $y:=b_{j-1}$ ;  $z:=a_i$ ; CLOCK:=false;  $Q_2(A):=z$ ;  $Q_2(B):=B$ 
      end
      else begin comment Case 1(a);
       $x:=a_i$ ;  $y:=a_{i+1}$ ;  $z:=b_j$ ; CLOCK:=true;  $Q_2(A):=B$ ;  $Q_2(B):=z$ 
      end
      end
      end
      begin comment Case 2. R has no chord;
10     if  $b_1 \neq A$  then
11       begin
       $x:=A$ ;  $y:=a_1$ ;  $z:=b_{s-1}$ ; CLOCK:=true;  $Q_2(A):=B$ ;  $Q_2(B):=z$ 
      end

```

```

12      else begin comment b1=A;
           x:=A; y:=B; z:=a1; CLOCK:=false; Q2(A):=z; Q2(B):=B
           end
           end;
13      Q1(A):=A; Q1(B):=z; Q1(C):=C;
14      if y=B then Q2(C):=0 else Q2(C):=y;
15      if CLOCK then EDEL:= $\overline{CL}(x,y,z)$  else EDEL:= $\overline{CCL}(x,y,z)$ ;
           comment EDEL= $\overline{CL}(x,y,z)$  ( $\overline{E_{DEL}}=\overline{CCL}(x,y,z)$ ) is the list of edges incident
           to x from (y,x) to (z,x) in a (counter-)clockwise sense, where (y,x)  $\in$  EDEL and
           (z,x)  $\notin$  EDEL;
16      let G'=G-EDEL;
           comment G' satisfies Condition (A). G' consists of two blocks G1 and G2
           and z is the unique cutvertex of G';
17      split G' into G1 A and G2 B with respect to z;
18      for f:=1 to 2 do
           begin
19         let Qf be the exterior faces of Gf;
20         choose Qf(C) appropriately so that (Gf,Qf,Qf(A),Qf(B)) may
           satisfy Condition (X);
21         HPATH(Gf,Qf,Qf(A),Qf(B))
           end
           end
else begin
comment Case 3. There is a chord of form (bj,ck);
22 let (bj,ck) be furthest from C;
23 let Q=q0,...,qu (q0=bj, qu=a1) be the Q-chain of R from q0 to qu;
24 let EDEL be the edge set defined in (2) in Section 4;
25 let G'=G-EDEL;
           comment G' satisfies Condition (A);
26 let x1,...,xg-1 (xg-2=ck, xg-1=bj) be the sequence of
           cutvertices of G' on R2 or R3 from A to B;
27 let Gf be a block of G' containing xf-1 and xf (x0=A, xg=B);
28 for f=1 to g-2 do
           begin
29      Qf(A):=xf; Qf(B):=xf-1;
30      if there is a vertex Cf in Gf adjacent to both qi and qi+1 of the
           Q-chain in G then Qf(C)=Cf else Qf(C)=0;
31      split Gf from G' with respect to xf;
           comment Gf'=Gf-(V(Gf)-xf);
32      let Qf be the exterior face of Gf;
           comment (Gf,Qf,Qf(A),Qf(B)) satisfies Condition (W);
33      choose Qf(C) appropriately so that (Gf,Qf,Qf(A),Qf(B)) may satisfy
           Condition (X);
34      HPATH(Gf,Qf,Qf(A),Qf(B))
           end;
35      Qg-1(A):=ck; Qg-1(B):=bj; Qg-1(C):=C; Qg(A):=B; Qg(B):=bj;
36      if a1=B then Qg(C):=0 else Qg(C):=a1;
37      split G' into Gg-1 and Gg with respect to bj;
38      for f:=g-1 to g do
           begin
39         let Qf be the exterior face of Gf;
           comment (Gf,Qf,Qf(A),Qf(B)) satisfies Condition (W);
40         choose Qf(C) appropriately so that (Gf,Qf,Qf(A),Qf(B)) may satisfy
           Condition (X);
41         HPATH(Gf,Qf,Qf(A),Qf(B))
           end
           end
end;

```

Fig. 9 The algorithm for finding a Hamiltonian path.

We now verify the correctness and the time complexity of the algorithm.

**Lemma 3.** If  $(G, R, A, B)$  satisfies Condition (X), then HPATH correctly finds a Hamiltonian path connecting vertices A and B in G.

**Proof.** Note that HPATH finds an edge set  $E_{DEL}$  in G whose removal results in the graph  $G'$  satisfying Condition (A). Thus, the correctness of HPATH can be proved by the induction on the number of edges of a graph. Q.E.D.

**Lemma 4.** If  $(G, R, A, B)$  satisfies Condition (X), then HPATH requires  $O(|V|)$  time to find a Hamiltonian path connecting A and B in  $G=(V, E)$ .

**Proof.** We show that the algorithm requires  $O(|V|)$  time with the data structure described above. We first establish (1). Let  $T(G, R, A, B)$  denote the time spent by the HPATH( $G, R, A, B$ ) for the graph  $G=(V, E)$ . Let  $T'(G, R, A, B)$  denote the time spent by the HPATH( $G, R, A, B$ ), exclusive of the time spent by its recursive calls. Clearly lines 1-14, 22, and 35-36 require constant time. Let  $EX(G, R, A, B)$  denote the set of vertices of G which newly appear on the exterior face of  $G'$ . Let  $CV(G, R, A, B)$  denote the set of vertices of G which newly become cutvertices. Suppose Case 1 or 2 occurs. Since  $(x, y) = c(x)$  or  $cc(x)$  we can obtain  $G'$  in  $O(|E_{DEL}|)$  time by scanning the adjacency list  $A(x)$  from  $c(x)$  or  $cc(x)$  successively. Thus lines 15-17 requires  $O(|E_{DEL}|)$  time. Line 20 can be done by scanning adjacency lists  $A(v)$ , where  $v$ 's are the vertices newly appeared on the exterior face or the new cutvertices. Thus lines 19-20 requires

$$O\left(\sum_{v \in EX(G, R, A, B)} d(v) + \sum_{v \in CV(G, R, A, B)} d(v)\right)$$

time. Suppose next Case 3 occurs. Since  $(q_{-1}, q_0) = cc(q_0)$  we can find  $q_1$  in  $O(d(q_0))$  time by scanning adjacency list  $A(q_0)$  in a counter-clockwise sense. Thus line 23 can be done in  $O(\sum_{1 \leq i \leq u-1} d(q_i))$  time, that is, we can obtain Q-chain in  $O(\sum_{1 \leq i \leq u-1} d(q_i))$ . Similarly lines 24-27 require scanning adjacency lists  $A(q_i)$  for  $1 \leq i \leq u-1$ . Lines 33 and 40 can be done by scanning adjacency lists  $A(v)$ , where  $v$ 's are vertices newly appeared on the exterior face or the new cutvertices, that is,  $v \in (EX(G_1, Q_1, Q_1(A), Q_1(B)) - R) \cup \{Q_1(A), Q_1(B)\}$ . Thus lines 29-33, 37 and 39-40 require

$$O\left(\sum_{v \in EX(G, R, A, B)} d(v) + \sum_{v \in CV(G, R, A, B)} d(v)\right)$$

time. Thus we obtain

$$T'(G, R, A, B)$$

$$\leq O\left(\sum_{v \in EX(G, R, A, B)} d(v) + \sum_{v \in CV(G, R, A, B)} d(v)\right)$$

for any  $(G, R, A, B)$  satisfying Condition (X). This implies (1). Noting Remarks 3-5 we have  $T(G, R, A, B) \leq O(|V|)$ . Q.E.D.

Thus by Lemmas 3 and 4 we obtain the following theorems.

**Theorem 1.** If  $G=(V, E)$  is 2-connected plane graph and  $(G, R, A, B)$  satisfies the condition(X), then HPATH finds a Hamiltonian path joining A and B in G in  $O(|V|)$  time.

Theorem 2. There exists a linear time algorithm for finding Hamiltonian circuits in 4-connected maximal planar graphs.

#### Acknowledgements.

We wish to thank Professor T. Nishizeki for valuable discussions and suggestions on this subjects. This research was supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant: YSE(A) 575215 (1980).

#### References

- [ 1 ] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.
- [ 2 ] R. E. Bixby and D. Wang, An algorithm for finding hamiltonian circuits in certain graphs, Mathematical Programming Study, 8(1978), pp. 35-49.
- [ 3 ] J. A. Bondy and V. Chvátal, A method in graph theory, Discrete Math., 15(1976), pp. 111-135.
- [ 4 ] M. R. Garey, D. S. Johnson and R. E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, SIAM J. Comput., 5(1976), pp. 704-714.
- [ 5 ] D. Gouyou-Beauchamps, Un algorithme de recherche de circuit Hamiltonien dans les graphes 4-connexes planaires, Colloques Internationaux CNRS, No. 260 - Problèmes Combinatoires et Theorie des Graphes, ed, J.C. Bermond, J.C. Fournier, M. Las Vergnas and D. Scotteau, (1978), pp. 185-187.
- [ 6 ] F. Harary, Graph Theory, Addison-Wesley, Reading, Mass., 1969.
- [ 7 ] J. E. Hopcroft and R. E. Tarjan, Efficient planarity testing, J. Assoc. Comput. Mach., 21(1974), pp. 549-568.
- [ 8 ] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller and J. W. Thatcher, eds, Complexity of Computer Computations, Plenum Press, New York, (1972) pp. 85-104.
- [ 9 ] W. T. Tutte, A theorem on planar graphs, Trans. Amer. Math. Soc., 82(1956), pp. 99-116.
- [ 10 ] H. Whitney, A theorem on graphs, Annals Math., 32(1931), pp. 378-390.