A LINEAR ALGORITHM FOR FINDING HAMILTONIAN CYCLES IN 4-CONNECTED MAXIMAL PLANAR GRAPHS

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

Shunji KIKUCHI
Research and Development Center, Toshiba Corporation, Kawasaki, Japan 210

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

Received 4 January 1982
Revised 9 July 1982

This paper describes a linear time algorithm to find a Hamiltonian cycle 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 cycle.

1. Introduction

The Hamiltonian cycle 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 [5,9]. Therefore, there seems to be no polynomial-time algorithm for the Hamiltonian cycle problem. However, for certain (nontrivial) classes of restricted graphs, there exist polynomial-time algorithms [3,4,6]. In fact, employing the proof technique used by Tutte [10], Gouyou-Beauchamps has given an O(n^3) time algorithm for finding a Hamiltonian cycle in a 4-connected planar graph G, where n is the number of vertices of G [6]. Although such a graph G always has a Hamiltonian cycle [10], it is not an easy matter to actually find a Hamiltonian cycle of G. However, for a little more restricted class of graphs, i.e., the class of 4-connected maximal planar graphs, we can construct an efficient algorithm. One can easily design an O(n^2) time algorithm to find a Hamiltonian cycle in a 4-connected maximal planar graph G with n vertices, entirely based on Whitney's proof of his theorem [11].

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 other techniques in the algorithm. The computational complexity of our algorithm is linear, thus optimal to within a constant factor.

0166-218X/84/$3.00 © 1984, Elsevier Science Publishers B.V. (North-Holland)
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., [2,7]. 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 with both \( v \) and \( w \). A walk of length \( k \) with end vertices \( v, w \) is a sequence \( v = v_0, v_1, v_2, \ldots, v_k = w \) such that \((v_{i-1}, v_i)\) is an edge for \( 1 \leq i \leq k \). If all the vertices \( v_0, v_1, v_2, \ldots, v_{k-1} \) are distinct, the walk is a path. If \( v = w \), the path is a cycle. A path is sometimes denoted by the vertex set. A cycle of length two (or three) is called a 2-cycle (or triangle). A path \( u_0, u_1, \ldots, u_j \) in the cycle \( R = u_0, u_1, u_2, \ldots, u_{k-1}, u_0 \) is called an arc of \( R \), and denoted by \( R[u_i, u_j] \) or \( R(u_{i-1} u_{j+1}) \). A chord of a cycle \( R = u_0, u_1, u_2, \ldots, u_{k-1}, u_0 \) is an edge \((u_i, u_j)\) of \( G \) such that \( |i - j| \neq 1 \pmod{k} \), that is, an edge joining non-consecutive vertices \( u_i \) and \( u_j \) on \( R \). A Hamiltonian cycle (path) of a graph \( G \) is a cycle (path) containing all vertices of \( G \). A graph \( G_i = (V_i, E_i) \) is a subgraph of a graph \( G = (V, E) \) if \( V_i \subseteq V \) and \( E_i \subseteq E \). If \( E_i = E \setminus \{ (v, w) \mid v, w \in V_i \} \), \( G_i \) is an induced subgraph of \( G \). The induced subgraph \( G_i \) is obtained from \( G \) by removing vertices in \( V - V_i \), and denoted by \( G_i = G - (V - V_i) \). A graph \( G \) is connected if every two vertices of \( G \) are joined by a path. The connected components of a graph \( G \) are its maximal connected subgraphs. A cut vertex 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 end vertices. A plane graph is a planar graph which has been 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 cycle 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 plane 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 \( G \). We refer to [1,2,7], for all undefined terms.

Next, introducing Whitney's condition, we describe Whitney's lemma necessary 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 \( G \).
Algorithm for finding Hamiltonian cycles 3

(W2) Either

(W2a) $R$ is divided into two arcs $R_1 = R[A, B] = a_0a_1 \cdots a_r$ and $R_2 = R[B, A] = b_0b_1 \cdots b_s$ ($a_0 = b_s = A$, $a_r = b_0 = B$), and there are no chords of $R$ joining any two vertices on $R_i$ ($1 \leq i \leq 2$), or

(W2b) $R$ is divided into three arcs $R_1 = R[A, B] = a_0a_1 \cdots a_r$, $R_2 = R[B, C] = b_0b_1 \cdots b_s$ and $R_3 = R[C, A] = c_0c_1 \cdots c_t$ 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 any 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 cycle of length two under our definition although it is not a cycle 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** [11]. 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 $B$.

![Fig. 1. A plane graph $G$ satisfying Conditions (W) and (X). $G$ satisfies the conditions (W1) and (W2b) (Fig. 1(a)), and also satisfies Condition (X) with proper labels (Fig. 1(b)).](image)
Based on the proof of Whitney's lemma, one can easily design an \(O(n^2)\) algorithm for finding a Hamiltonian cycle 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), which we shall replace with the following Condition (X3):

\[
\text{(X3) } R \text{ is divided into three arcs } R_1 = R[A, B] = a_0a_1 \cdots a_r, \quad R_2 = R[B, C] = b_0b_1 \cdots b_s, \quad \text{and } R_3 = R[C, A] = c_0c_1 \cdots c_t \text{ for some vertex } C \text{ on } R(B, A) (a_0 = c_t = A, a_r = b_s = B, b_s = c_0 = C), \text{ and there are no chords of } R \text{ joining any two vertices on } R_i (1 \leq i \leq 3). \text{ Moreover, there exists a chord of form } (b_{s-1}, c_k) (1 \leq k \leq t) \text{ or } (c_{t-1}, b_j) (0 \leq j \leq s-1).
\]

**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 \(R\) is incident with \(C\) (see Fig. 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 [8] 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 construct a Hamiltonian cycle by combining the Hamiltonian path and edge \((1, 2)\). Hence, it suffices to 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 need 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 conditions (A1)-(A5) below, then we say that \(G'\) satisfies Condition (A):

- **(A1)** \(G'\) is a connected spanning subgraph of \(G\).
- **(A2)** \(G'\) consists of \(g\) blocks \(G_1, G_2, \ldots, G_g \geq 2\) with \((g - 1)\) cut vertices \(x_1, x_2, \ldots, x_{g-1}\), such that each vertex \(x_f (1 \leq f \leq g - 1)\) belongs to exactly two blocks \(G_f\) and \(G_{f+1}\).
- **(A3)** Neither \(A\) nor \(B\) is a cut vertex of \(G'\).
Algorithm for finding Hamiltonian cycles

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

(A4) $A$ is a vertex of $G_1$ and $B$ is a vertex of $G_g$.

(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).

Our algorithm for finding a Hamiltonian path can now be obtained as follows: First, delete one or more edges from $G$ so that the resulting graph $G'$ may satisfy Condition (A); next, for each $G_f$ ($1 \leq f \leq g$), choose $C$ appropriately so that $(G_f, Q_f, x_f, x_{f-1})$ satisfies Condition (X); then, construct 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, construct a Hamiltonian path $H(G, A, B)$ by combining all $H(G_f, x_f, x_{f-1})$'s. In Fig. 3 below is the outline of the algorithm, written in Pidgin ALGOL [1].

```
procedure HPATH(G, R, A, B):
begin
    comment an outline of the algorithm to find a Hamiltonian path in a graph $G = (V, E)$ 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 $G'$ may satisfy Condition (A);
3      for each block $G_f$ of $G'$ do
4         begin
          comment $(G_f, Q_f, x_f, x_{f-1})$ satisfies Condition (W);
4         choose $C$ appropriately so that $(G_f, Q_f, x_f, x_{f-1})$ may satisfy Condition (X);
5         HPATH($G_f, Q_f, x_f, x_{f-1}$)
4         end
3   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 shall present in Section 4 a method to determine which edges are to be deleted). Via 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 $f$th left son of the root of $TR(G, R, A, B)$ and is the root of the execution tree $TR(G_f, R_f, x_f, x_{f-1})$, where $G_f$ is the $f$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 cut vertices 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 $V(F) \cap V(H) \subseteq \{A_F, B_F\} \cap \{A_H, B_H\}$. Since $G'$ satisfies Condition (A) we can observe the following:

**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 $x$ lies on the exterior face $R_H$ of $H$.

(R2) If $x$ is a cut vertex 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 the end vertices of the Hamiltonian path of $H$, that is $x = A_H$ or $x = B_H$.

(R3) If $x$ is an end vertex of the Hamiltonian path of $F$ (that is, $x = A_F$ or $x = B_F$), then $x$ is not a cut vertex of $F'$ and $x = A_H$ or $x = B_H$.

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 any two distinct vertices of the execution tree $TR(G, R, A, B)$. Then

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

$$CV(F, R_F, A_F, B_F) \cap CV(H, R_H, A_H, B_H) = \emptyset.$$  

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

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

(1)

for any $(G, R, A, B)$ satisfying Condition (X), where $K$ is a constant and $d(v)$ denotes the degree of the vertex $v$ of $G$. Via 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 must now 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 $R_1 = R[A, B] = a_0 a_1 \cdots a_r$ ($a_0 = A$, $a_r = B$), $R_2 = R[B, C] = b_0 b_1 \cdots b_s$ ($b_0 = B$, $b_s = C$), and $R_3 = R[C, A] = c_0 c_1 \cdots c_t$ ($c_0 = C$, $c_t = A$). If $(G, R, A, B)$ satisfies Condition (W2a), then $R_3$ is empty and $R_2 = R[B, A]$, that is, $b_s = A$. Since $(G, R, A, B)$ satisfies Condition (X), no chord of $R$ joins any two vertices on the same arc, and no chord joins vertex $C$ (if any) and a vertex of $R$. We thus 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 reasons why we introduced Condition (X). This fact, together with the method for finding a $Q$-chain (defined later) enables us to design a linear algorithm.

Throughout the rest of this paper, we shall use the following notation: Let

![Fig. 4. Three cases in the proof.](image)
\( CL(x, (y, z)) \) denote the set of edges incident with \( x \) from \((y, x)\) to \((z, x)\) in a clockwise sense, where \((y, x), (z, x) \notin CL(x, (y, z))\) and let

\[
CL(x, (y, z)) = CL(x, (y, z)) \cup \{(y, x)\},
\]

\[
CL(x, (y, z)) = CL(x, (y, z)) \cup \{(z, x)\}.
\]

Similarly, define \( CCL(x, (y, z)) \), \( CCL(x, [y, z]) \) and \( CCL(x, (y, z]) \) with respect to edges incident with \( x \) in a counter-clockwise sense.

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 to \( B \) among all chords of this type, that is, the cycle \( a_i a_{i+1} \ldots a_{r-1} B b_j \ldots b_j a_i \) has no chord. Now either,

- **Case 1(a)**, \( R \) has no chords other than \((a_i, b_j)\) joining \( b_j \) and a vertex on \( R_1 \), or
- **Case 1(b)**, \( R \) 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 1(a). Let \( p_0, p_1, p_2, \ldots, p_u \) \((p_0 = a_{i+1}, p_u = b_j)\) be a sequence of vertices adjacent to \( a_i \) such that each \((a_i, p_k)\) is the immediate 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

\[
E_{DEL} = CL(a_i, \{p_0, p_u\}), \quad \text{i.e.,} \quad E_{DEL} = \{(a_i, p_k) | 0 \leq k \leq u - 1\}.
\]

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 \( Q_{11} = a_0 a_1 \ldots a_i b_j, \ Q_{12} = b_j b_{j+1} \ldots b_s, \ Q_{13} = R_3, \) and \( Q_1 = Q_{11} Q_{12} Q_{13}, \) then \((G_1, Q_1, A, b_j)\)

![Case (1a)](image1)

Case (1a)

![Case (1b)](image2)

Case (1b)

Fig. 5. Case 1, where \((a_i, b_j)\) is the chord nearest to \( B \). \( G' = G - E_{DEL} \) consists of two blocks \( G_1 \) and \( G_2 \). \( Q_1 = Q_{11} Q_{12} Q_{13} \) and \( Q_2 = Q_{21} Q_{22} Q_{23} \) are the new exterior faces of \( G_1 \) and \( G_2 \), respectively.
Algorithm for finding Hamiltonian cycles

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 \( Q_{21} = b_0 b_1 \cdots b_j, Q_{22} = p_u p_{u-1} \cdots p_0, Q_{23} = a_{i+1} \cdots a_5 \) (if \( p_0 = a_5 = B \), 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). Via Remark 1, we can choose a vertex as \( C \) appropriately so that \((G_2, Q_2, B, b_j)\) may satisfy Condition (X). By the inductive hypothesis, \( G_1 \) has a 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 construct 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 1(b). In this case let \( p_0, p_1, p_2, \ldots, p_u \) \((p_0 = b_{j-1}, p_u = a_l)\) be a sequence of vertices adjacent to \( b_j \) such that each \((b_j, p_k)\) is the immediate counter-clockwise edge of \((b_j, p_{k-1})\) around \( b_j \). We delete all edges in \( E_{\text{DEL}} = \text{CCL}(b_j, [p_0, p_u]) = \{(b_j, p_k) \mid 0 \leq k \leq u - 1\} \). An argument similar to the one in Case 1(a) 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 1(a) with \( a_i = A \) and \( b_j = b_{j-1} \) or Case 1(b) with \( a_i = a_1 \) and \( b_j = b_1 = A \) (see Fig. 6). An argument similar to the one above works. Note that \( G_1 = K_2 \) satisfies Condition (X).

Case 3. Suppose that \((b_j, c_k)\) is the chord furthest from vertex \( C \) of \( R \). Note that \( b_j \) is one of \( b_1, \ldots, b_{j-1} \) and \( c_k \) is one of \( c_1, \ldots, c_{u-1} \). Let \( Q = q_0, q_1, q_2, \ldots, q_u \) \((q_0 = b_j, q_u = a_i)\) be a sequence of vertices which satisfies the following: Each \((q_i, q_{i+1})\) is the next edge of \((q_i, q_{i-1})\) in a counter-clockwise sense around \( q_i \) among all edges with an end vertex adjacent to a vertex on \( R_3 \), where \( q_{i+1} = b_{j-1} \). Such a sequence is called a \( Q \)-chain of \( R \). The existence of the \( Q \)-chain can be 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 cycle \( A a_1 \cdots a_{r-1} B b_1 \cdots b_j c_k \cdots c_{r-1} A \) adjacent to both \( b_j \) and \( c_k \). Thus \( q_1 \) always exists \((q_1 = x \) is possible). If \( q_1 = a_1 \), then \( q_u = q_1 \). Otherwise, let \( c_{k+1} \) 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 cycle \( A a_1 \cdots a_{r-1} B b_1 \cdots b_j q_1 c_{k+1} c_{k+1} \cdots c_{t-1} A \) adjacent to both \( q_1 \) and \( c_{k+1} \). Repeating this argument, we can prove that the \( Q \)-chain always exists since \( a_i \) is adjacent to \( A \). Note that no vertex of \( q_1, \ldots, q_{u-1} \) is on \( R \), and the cycle

![Fig. 6. Case 2.](image-url)
\[ D = q_a a_2 \cdots a_{r-1} B b_1 \cdots b_j q_1 \cdots q_u \] has no chord of form \((q_i, q_{i'}) (0 \leq i, i + 2 \leq i' \leq u)\). Let \(E_{\text{OUT}}(D)\) be the set of edges adjacent to \(q_1, q_2, \ldots, q_u\) outside \(D\), and let
\[ E_{\text{DEL}} = E_{\text{OUT}}(D) \cup \text{CL}(b_j, (c_k, q_1)), \] i.e.,
\[ E_{\text{DEL}} = \text{CL}(b_j, (c_k, q_1)) \cup \text{CL}(q_1, (q_0, q_2)) \cup \cdots \cup \text{CL}(q_{u-1}, (q_{u-2}, q_u)) \cup \text{CL}(q_u, (q_{u-1}, A)). \]

Let \(G' = G - E_{\text{DEL}}\), 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. Via 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 cut vertex 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 can construct 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.

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 end vertices \(x\) and \(y\), and four pointers \(c_1(e), c_2(e), cc_1(e)\) and \(cc_2(e)\), designating the edges immediate clockwise and counter-clockwise around the end vertices 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 immediate clockwise edge of \(cc(x)\) around \(x\). Furthermore, we need a representation of the exterior faces of

![Fig. 7. Q-chain and G'.](image-url)
Algorithm for finding Hamiltonian cycles

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 to $B$. The chords of $D13$ and $D23$ are arranged in furthest from $C$. $R(D12)$ stores the chord of $D12$ nearest to $B$. $R(D13)$ and $R(D23)$ store the

![Diagram](image)

**Vertex incidences**

<table>
<thead>
<tr>
<th>c</th>
<th>cc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$e_1$</td>
</tr>
<tr>
<td>2</td>
<td>$e_2$</td>
</tr>
<tr>
<td>3</td>
<td>$e_3$</td>
</tr>
<tr>
<td>4</td>
<td>$e_4$</td>
</tr>
<tr>
<td>5</td>
<td>$e_5$</td>
</tr>
<tr>
<td>6</td>
<td>$e_6$</td>
</tr>
<tr>
<td>7</td>
<td>$e_7$</td>
</tr>
<tr>
<td>8</td>
<td>$e_8$</td>
</tr>
</tbody>
</table>

**Edges, neighbors and next chords**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>$c_1$</th>
<th>$cc_1$</th>
<th>$c_2$</th>
<th>$cc_2$</th>
<th>NEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>1</td>
<td>2</td>
<td>$e_8$</td>
<td>$e_8$</td>
<td>$e_2$</td>
<td>$e_9$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>2</td>
<td>3</td>
<td>$e_9$</td>
<td>$e_1$</td>
<td>$e_3$</td>
<td>$e_{10}$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>3</td>
<td>4</td>
<td>$e_{11}$</td>
<td>$e_2$</td>
<td>$e_4$</td>
<td>$e_{12}$</td>
</tr>
<tr>
<td>$e_4$</td>
<td>4</td>
<td>5</td>
<td>$e_{12}$</td>
<td>$e_3$</td>
<td>$e_5$</td>
<td>$e_5$</td>
</tr>
<tr>
<td>$e_5$</td>
<td>5</td>
<td>6</td>
<td>$e_4$</td>
<td>$e_4$</td>
<td>$e_6$</td>
<td>$e_{12}$</td>
</tr>
<tr>
<td>$e_6$</td>
<td>6</td>
<td>7</td>
<td>$e_{13}$</td>
<td>$e_5$</td>
<td>$e_7$</td>
<td>$e_7$</td>
</tr>
<tr>
<td>$e_7$</td>
<td>7</td>
<td>8</td>
<td>$e_6$</td>
<td>$e_6$</td>
<td>$e_8$</td>
<td>$e_{13}$</td>
</tr>
<tr>
<td>$e_8$</td>
<td>8</td>
<td>1</td>
<td>$e_9$</td>
<td>$e_7$</td>
<td>$e_1$</td>
<td>$e_1$</td>
</tr>
<tr>
<td>$e_9$</td>
<td>2</td>
<td>8</td>
<td>$e_1$</td>
<td>$e_2$</td>
<td>$e_{10}$</td>
<td>$e_8$</td>
</tr>
<tr>
<td>$e_{10}$</td>
<td>3</td>
<td>8</td>
<td>$e_2$</td>
<td>$e_{11}$</td>
<td>$e_{13}$</td>
<td>$e_9$</td>
</tr>
<tr>
<td>$e_{11}$</td>
<td>3</td>
<td>6</td>
<td>$e_{10}$</td>
<td>$e_3$</td>
<td>$e_{12}$</td>
<td>$e_{13}$</td>
</tr>
<tr>
<td>$e_{12}$</td>
<td>4</td>
<td>6</td>
<td>$e_3$</td>
<td>$e_4$</td>
<td>$e_5$</td>
<td>$e_{11}$</td>
</tr>
<tr>
<td>$e_{13}$</td>
<td>6</td>
<td>8</td>
<td>$e_{11}$</td>
<td>$e_6$</td>
<td>$e_7$</td>
<td>$e_{10}$</td>
</tr>
</tbody>
</table>

**Face $R$ and its chords**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D12</th>
<th>D13</th>
<th>D23</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>$e_{12}$</td>
<td>$e_9$</td>
</tr>
</tbody>
</table>

Fig. 8. Representation of a plane graph $(G,R,A,B)$ satisfying Condition (X).
chords of $D_{13}$ and $D_{23}$ furthest from $C$, respectively. Array $\text{NEXT}(x)$ stores a next chord in the order above for each chord $x$ of $R$. Thus the chord of $D_{12}$ second nearest to $B$ is stored with $\text{NEXT}(R(D_{12}))$, 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 $c(v_j) = (y_j, u_{j+1})$ and $cc(v_j) = (u_{j-1}, y_j)$ for each vertex $v_j$ on the exterior face $R = u_0 v_1 \cdots u_{n-1} v_0$. Thus $R$ is represented as follows: $v_1$ is an end vertex of $c(v_0)$ different from $v_0$; $v_2$ is an end vertex of $c(v_1)$ different from $v_1$; and $v_0$ is an end vertex of $c(v_{n-1})$ different from $v_{n-1}$. Adjacency list $A(v)$ in a clockwise sense is represented as follows: the first vertex of $A(v)$ is an end vertex of $c(v)$ different from $v$; the second vertex of $A(v)$ is an end vertex of immediate clockwise edge of $c(v)$ around $v$ different from $v$; and the last vertex of $A(v)$ is an end vertex 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. 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
% $G = (V, E)$ is a plane graph with the exterior face $R$ satisfying Condition (X).
% $(G, R, A, B)$ is represented by the data structure described above. Let $R = R_1 R_2 R_3$, $R_1 = a_0, a_1, \ldots, a_r$ ($a_0 = A, a_r = B$), $R_2 = b_0, b_1, \ldots, b_l$ ($b_0 = B, b_l = C$), and $R_3 = c_0, c_1, \ldots, c_t$ ($c_0 = C, c_t = A$). If $G = K_2$, then $R$ is a 2-cycle. $R$ 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 $c_i$ and some vertex on $R_i$ or $b_{i-1}$ and some vertex on $R_i$. HPATH finds a Hamiltonian path in $G$ connecting $A$ and $B$, whose edges are marked by HPATH;
1 if $|V| = 2$
2 then a Hamiltonian path of $G$ from $A$ to $B$ is $(A, B)$ so mark $(A, B)$
3 else if $(R$ has a chord of form $(a_i, b_j)$, or $(a_i, c_k)$) or $(R$ has no chord)$\ then begin$
4 \text{begin comment Case 1 or Case 2;}
5 \text{let $E_{DEL}$ be the edge set defined in Case 1 (if Case 1) or Case 2 (if Case 2) in Section 4;}
6 \text{let $G' = G - E_{DEL};$}
7 \text{comment $G'$ satisfies Condition (A). $G'$ consists of two blocks $G_1$ and $G_2$. Let $z$ be the unique cut vertex of $G'$;}
8 \text{split $G'$ into $G_1 \ni A$ and $G_2 \ni B$ with respect to $z$;}
9 \text{for } f := 1 \text{ to } 2 \text{ do}$
10 \text{begin}
11 let $Q_f$ be the exterior face of $G_f$;
12 choose $Q_f(C)$ appropriately so that $(G_f, Q_f(A), Q_f(B))$ may satisfy Condition (X);
13 update the data structure so that it may represent $(G_f, Q_f(A), Q_f(B))$;
14 HPATH($G_f, Q_f(A), Q_f(B)$)
15 end
16 else begin comment Case 3. There is a chord of form $(b_j, c_k)$;
17 let $(b_j, c_k)$ be furthest from $C$;
18 let $Q = q_0, \ldots, q_u$ ($q_0 = b_j, q_u = a_1$) be the $Q$-chain of $R$ from $q_0$ to $q_u$;
19 let $E_{DEL}$ be the edge set defined in (2) in Section 4;
20 let $G' = G - E_{DEL};$
21 comment $G'$ satisfies Condition (A);
```
Algorithm for finding Hamiltonian cycles

let \( x_1, \ldots, x_{g-1} \) (\( x_{g-2} = c_k, x_{g-1} = b_j \)) be the sequence of cut vertices on \( R_2 \) or \( R_3 \) from \( A \) to \( B \) of \( G' \);
let \( G_f \) be the block of \( G' \) containing \( x_{g-1} \) and \( x_f \) (\( x_0 = A, x_g = B \));
for \( f = 1 \) to \( g - 2 \) do
  begin
    \( Q_f(A) := x_f, Q_f(B) := x_{f-1} \);
    if there is a vertex \( C_f \) in \( G_f \) joined to both \( q_i \) and \( q_{i+1} \) of the \( Q \)-chain in \( G \) then
      \( Q_f(C) := C_f \) else \( Q_f(C) := 0 \);
    split \( G_f \) from \( G' \) with respect to \( x_f \);
    comment \( G' := G' - (V(G_f) - x_f) \);
    let \( Q_f \) be the exterior face of \( G_f \);
    comment \((G_f, Q_f, Q_f(A), Q_f(B))\) satisfies Condition (W);
    choose \( Q_f(C) \) appropriately so that \((G_f, Q_f, Q_f(A), Q_f(B))\) may satisfy Condition (X);
    update the data structure so that it may represent \((G_f, Q_f, Q_f(A), Q_f(B))\);
    \( \text{HPATH}(G_f, Q_f, Q_f(A), Q_f(B)) \)
  end;
\( Q_{g-1}(A) := c_k ; Q_{g-1}(B) := b_j ; Q_{g-1}(C) := C ; Q_g(A) := B ; Q_g(B) := b_j \);
if \( a_1 = B \) then \( Q_g(C) := 0 \) else \( Q_g(C) := a_g \);
split \( G' \) into \( G_{g-1} \) and \( G_g \) with respect to \( b_j \);
for \( f := g - 1 \) to \( g \) do
  begin
    let \( Q_f \) be the exterior face of \( G_f \);
    comment \((G_f, Q_f, Q_f(A), Q_f(B))\) satisfies Condition (W);
    choose \( Q_f(C) \) appropriately so that \((G_f, Q_f, Q_f(A), Q_f(B))\) may satisfy Condition (X);
    update the data structure so that it may represent \((G_f, Q_f, Q_f(A), Q_f(B))\);
    \( \text{HPATH}(G_f, Q_f, Q_f(A), Q_f(B)) \)
  end;
end;

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

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

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

Lemma 4. If \((G, R, A, B)\) satisfies Condition (X), then \( \text{HPATH} \) 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 \( \text{HPATH}(G, R, A, B) \) for the graph \( G = (V, E) \). Let \( T'(G, R, A, B) \) denote the time spent by \( \text{HPATH}(G, R, A, B) \), exclusive of the time spent by its recursive calls. Let \( \text{EX}(G, R, A, B) \) 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 cut vertices. Clearly lines 1–3, 12, and 26–27 require constant time. Suppose Case 1 or 2 occurs. It can easily be shown that $G'$ can be obtained in $O(|E_{DEL}|)$ time by the data structure defined above. Thus lines 4 and 5 require $O(|E_{DEL}|)$ time. Lines 9 and 10 can be done by scanning adjacency lists $A(u)$, where $v$’s are the vertices newly appeared on the exterior face or the new cut vertices. Thus lines 6–10 require

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

time. Suppose next Case 3 occurs. Since $(q_-, q_0) = cc(q_0)$ 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 13 can be done in $O(\sum_{0 \leq i \leq u-1} d(q_i))$ time, that is, we can obtain $Q$-chain in $O(\sum_{0 \leq i \leq u-1} d(q_i))$. Similarly, lines 14–16 require scanning adjacency lists $A(q_i)$ for $1 \leq i \leq u-1$. Lines 23–24 and 31–32 can be done by scanning adjacency lists $A(v)$, where $v$’s are vertices newly appeared on the exterior face or the new cut vertices. Thus lines 17–24 and 28–32 require

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

time. Thus we have

$$T'(G,R,A,B) \leq O\left(\sum_{u \in EX(G,R,A,B)} d(u) + \sum_{v \in CV(G,R,A,B)} d(v)\right) + O(|E_{DEL}|) + O(1)$$

for any $(G,R,A,B)$ satisfying Condition (X). And, thus,

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

since

$$O(1) \leq O(|E_{DEL}|) \leq O \left(\sum_{u \in EX(G,R,A,B)} d(u) + \sum_{v \in CV(G,R,A,B)} d(v)\right).$$

This implies (1). Via Remarks 3–5, we have $T(G,R,A,B) \leq O(|V|)$.

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

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

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

We wish to thank Professor T. Nishizeki for valuable discussions and suggestions on this subject. 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). The first author was supported by The Sakkokai Foundation.

References