AN APPROXIMATION ALGORITHM FOR THE HAMILTONIAN WALK PROBLEM ON MAXIMAL PLANAR GRAPHS

Takao NISHIZEKI, Takao ASANO* and Takahiro WATANABE**

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

Received 24 November 1980
Revised 30 October 1981

A hamiltonian walk of a graph is a shortest closed walk that passes through every vertex at least once, and the length is the total number of traversed edges. The hamiltonian walk problem in which one would like to find a hamiltonian walk of a given graph is NP-complete. The problem is a generalized hamiltonian cycle problem and is a special case of the traveling salesman problem. Employing the techniques of divide-and-conquer and augmentation, we present an approximation algorithm for the problem on maximal planar graphs. The algorithm finds, in $O(p^2)$ time, a closed spanning walk of a given arbitrary maximal planar graph, and the length of the obtained walk is at most $\frac{3}{2}(p - 3)$ if the graph has $p$ ($\geq 9$) vertices. Hence the worst-case bound is $\frac{3}{2}$.

1. Introduction

A hamiltonian walk of a graph is a shortest closed walk that passes through every vertex at least once, and the length of a hamiltonian walk is the total number of edges traversed by the walk [7]. The hamiltonian walk problem in which one would like to find a hamiltonian walk of a given graph would arise in situations where it is necessary to periodically traverse a network or data structure in a way as to visit all vertices and minimize the length of the traversal.

The hamiltonian walk problem is a generalization of the hamiltonian cycle problem in which one would like to determine if a given graph contains a hamiltonian cycle. It is well known that the hamiltonian cycle problem is NP-complete. Furthermore Garey, Johnson and Tarjan [6] have shown that the hamiltonian cycle problem is NP-complete even when restricted to 3-connected, cubic, planar graphs. Hence the hamiltonian walk problem is also NP-complete even when restricted to the same class. On the other hand, Christfides [4] has developed a polynomial-time approximation algorithm with a worst-case bound of $\frac{1}{2}$ for the traveling salesman problem in which the distance between vertices satisfies the triangle inequality. The

* Current address: Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo, Bunkyo-ku, Tokyo, Japan 113.
** Current address: Research and Development Center, Toshiba Corporation, Kawasaki, Japan 210.
hamiltonian walk problem is a special case of the traveling salesman problem, in which the distances are the shortest path lengths in a graph. Therefore we can use the algorithm of Christofides to find in $O(p^3)$ time a closed spanning walk of a graph whose length is smaller than $\frac{1}{2}$ times the length of a shortest one, where $p$ denotes the number of vertices in a graph.

In this paper we present a new approximation algorithm for the hamiltonian walk problem on maximal planar graphs. Our algorithm is more efficient than the method employing the Christofides' algorithm, and is completely different from it. Given a maximal planar graph with $p$ ($\geq 9$) vertices, our algorithm finds, in $O(p^2)$ time, a closed spanning walk of the graph whose length is at most $\frac{3}{2}(p - 3)$. Hence it has a worst-case bound of $\frac{3}{2}$ which is same as his. However it should be noted that the Christofides' method does not always produce a closed spanning walk of length at most $\frac{3}{2}(p - 3)$ if the given graph is not hamiltonian, since in this case the shortest closed walk has length greater than $p$). We will employ, in our algorithm, two techniques: divide-and-conquer and augmentation. The algorithm is based on two known results: one is our previous result that a maximal planar graph with $p$ vertices always contains either a hamiltonian cycle or a closed spanning walk of length at most $\frac{3}{2}(p - 3)$ [2]; the other is Whitney's result that every 4-connected, maximal planar graph has a hamiltonian cycle [12]. We conjecture that the hamiltonian walk problem remains NP-complete even if we restrict ourselves to the class of maximal planar graphs.

2. Terminology and basic results

We proceed to some basic definitions. An (undirected simple) graph $G = (V, E)$ consists of a set $V$ of vertices and a set $E$ of edges. Throughout this paper $p$ denotes the number of vertices of $G$, i.e., $p = |V|$. A walk of length $k$ of $G$ is a sequence $v_0e_1v_1e_2\cdots e_kv_k$, whose term are alternately, vertices and edges, such that the end-vertices of edge $e_i$ are $v_{i-1}$ and $v_i$, for each $1 \leq i \leq k$. The length of a walk $W$ is denoted by $l(W)$. The walk $W$ is a closed spanning walk of $G$ if $v_0 = v_k$ and every vertex of $G$ appears in the sequence at least once. A hamiltonian walk of $G$ is a closed spanning walk of minimum length. For a connected graph $G$, let $h(G)$ denote the length of a hamiltonian walk of $G$. Clearly $p \leq h(G) \leq 2(p - 1)$. A cycle is a closed walk whose vertices are all distinct. A hamiltonian cycle of $G$ is a closed spanning walk of length $p$, i.e., a cycle that passes through every vertex of $G$ exactly once. A graph is hamiltonian if it contains a hamiltonian cycle. A maximal planar graph is a planar graph to which no edge can be added without losing planarity. Note that every maximal planar graph $G$ is connected and every face of $G$ is a triangle. A triangle of a maximal planar graph is called a nonface triangle if it is not a boundary of a face. A maximal planar graph with $p$ ($\geq 5$) vertices has no nonface triangles if and only if it is 4-connected. For a graph $G = (V, E)$ and a subset $V'$ of $V$, $G - V'$ denotes the graph obtained from $G$ by deleting all vertices in $V'$. A singleton
set \{u\} is simply denoted by 'u'. A multiset is a set with a function mapping the elements of the set into the positive integers, to indicate that an element may appear more than once. We sometimes represent a walk by the multiset of edges traversed by it. A walk \(W\), i.e., a sequence of edges and vertices, can be easily constructed from the multiset of edges of a walk \(W\), (possibly \(W' \neq W\)). Note that this can be done by any algorithm for finding an eulerian walk of an eulerian graph. We refer to [1] or [9] for all undefined terms.

We next present several lemmas. Generalizing Whitney's result [12], Tutte has shown that every 4-connected planar graph has a hamiltonian cycle [11]. Employing the proof technique used by Tutte, Gouyou-Beauchamps has given an \(O(p^3)\) algorithm for finding a hamiltonian cycle in a 4-connected planar graph \(G\) [8]. If \(G\) is maximal planar, we can improve the time-complexity as follows.

**Lemma 1.** There is a \(O(p^2)\) time-algorithm for finding a hamiltonian cycle in a 4-connected maximal planar graph \(G\) with \(p\) vertices.

**Remark.** It is not difficult to implement a recursive algorithm for finding a hamiltonian cycle of a 4-connected maximal planar graph \(G\) in \(O(p^2)\) time, completely based on the inductive proof of Whitney [12] ensuring its existence. On the other hand, Asano, Kikuchi and Saito have recently obtained an \(O(p)\) algorithm for the same purpose. (T. Asano, S. Kikuchi and N. Saito, An efficient algorithm to find a Hamiltonian circuit in a 4-connected maximal planar graph, in: N. Saito and T. Nishizeki, eds., Graph Theory and Algorithms, Lecture Notes in Computer Science 108 (Springer-Verlag, Berlin, 1981) 182-195.)

**Lemma 2.**

(a) Every maximal planar graph with ten or fewer vertices contains a hamiltonian cycle [3,12].

(b) Every nonhamiltonian, maximal planar graph with 11 vertices has a hamiltonian walk of length 12. (It is implicit in [3] that every such graph is isomorphic to a certain graph depicted in Fig. 1 of [2].)

**Lemma 3.** [2] Let \(xyz\) be any (triangular) face of a maximal planar graph \(G = (V,E)\) with \(p\) vertices, where \(x, y, z \in V\).

(a) If \(p = 5\) or \(6\), then at least one of the three graphs \(G - \{x,y\}\), \(G - \{y,z\}\) and \(G - \{z,x\}\) contains a hamiltonian cycle.

(b) If \(p = 7\) or \(8\), then (i) at least one of \(G - \{x,y\}\), \(G - \{y,z\}\) and \(G - \{z,x\}\) contains a hamiltonian cycle, or (ii) \(G - x\), \(G - y\) and \(G - z\) all have hamiltonian cycles.

Using the dynamic programming, one can obtain an \(O(p \cdot 2^p)\) algorithm for determining if a given graph contains a hamiltonian cycle. The algorithm requires a constant time if \(p\) is a constant. Let HAMILTON\((G)\) be such an algorithm which determines if a given planar graph with 11 or fewer vertices contains a hamiltonian cycle, and moreover finds a hamiltonian walk in constant time. This algorithm will be used in the next section.
Lemma 4. Given a connected graph \( G = (V, E) \) with \( p \) vertices and given a cycle \( C \) of length \( c \) in \( G \), one can find a closed spanning walk \( W \) of \( G \) such that \( l(W) \leq 2p - c \), in \( O(|E|) \) time.

Proof. Contract all the vertices on \( C \) into one vertex, and find a (spanning) tree of the obtained graph. If \( C \) is the set of edges of the cycle \( C \) and \( T \) the set of edges of the tree, then the multiset \( W = C + T + T \) is a closed spanning walk of \( G \) which traverses twice each edge of the tree and once each edge of \( C \). Clearly the length of \( W \) is \( 2p - c \).

For a nonface triangle \( T \) of a maximal plane graph \( G \), let \( G_{T_1} = (V_{T_1}, E_{T_1}) \) denote the induced subgraph of \( G \) inside \( T \), and \( G_{T_0} = (V_{T_0}, E_{T_0}) \) the induced subgraph of \( G \) outside \( T \). Specifically if \( T = xyz \) \((x, y, z \in V)\), \( U'(T) \) is the set of vertices lying inside \( T \), and \( U''(T) \) is the set of vertices outside \( T \), then \( G_{T_1} \) is the subgraph of \( G \) induced by the vertex set \( \{x, y, z\} \cup U'(T) \), i.e., \( G_{T_1} = G - U''(T) \), and \( G_{T_0} \) is the subgraph of \( G \) induced by the vertex set \( \{x, y, z\} \cup U''(T) \), i.e., \( G_{T_0} = G - U'(T) \). Let \( p_{T_1} = |V_{T_1}| \) and \( p_{T_0} = |V_{T_0}| \). The following lemma plays a crucial role in the design of our algorithm. The precise description of Algorithm LCYCLE and the proof of Lemma 5 will appear in Section 4.

Lemma 5. For a maximal planar graph \( G \) with \( p \) \((\geq 11)\) vertices such that either \( p_{T_1} = 4 \) or \( p_{T_0} = 4 \) for each nonface triangle \( T \) of \( G \), Algorithm LCYCLE finds a cycle \( C \) of length \( l(C) \geq \frac{1}{2}(p + 9) \), in \( O(p^2) \) time.

3. Approximation algorithm HWALK

In this section we present a polynomial-time algorithm for finding a closed spanning walk \( W \) with \( l(W) \leq \max\{p, \frac{1}{2}(p - 3)\} \) of a given maximal planar graph \( G \) with \( p \) vertices. In the algorithm we will employ the divide-and-conquer technique: if a given maximal planar graph \( G \) has a nonface triangle \( T \) satisfying a certain condition, then (i) divide \( G \) into two smaller maximal planar graphs \( G_{T_1} \) and \( G_{T_0} \), (ii) recursively call the algorithm with respect to \( G_{T_1} \) and \( G_{T_0} \), and (iii) combine the closed spanning walks of \( G_{T_1} \) and \( G_{T_0} \) into a closed spanning walk of the whole graph \( G \).

The Algol-like procedure HWALK depicted in Fig. 1 takes as input a maximal planar graph and returns a closed spanning walk of the graph represented by a multiset of edges.

We can show that HWALK is a polynomial-time algorithm with a worst-case bound of \( \frac{1}{2} \), establishing the following theorem. Remember that \( h(G) \geq p \) for every connected graph \( G \).

Theorem 1. For a maximal planar graph \( G \) with \( p \) vertices, Algorithm HWALK
procedure HWALK(G):
begin
1 if p \leq 11 then return a hamiltonian walk W of G which can be found by the Algorithm HAMILTON;
2 if G has no nonface triangle (i.e., G is 4-connected) then return a hamiltonian cycle W of G which can be found by the algorithm in Lemma 1
else
3 if either p_{TI} = 4 or p_{TO} = 4 for every nonface triangle T of G then
begin
4 find, in G, a cycle C of length l(C) \geq (p+9)/2 by Algorithm LCYCLE (Lemma 5);
5 return a closed spanning walk constructed from C by the algorithm in Lemma 4
end
else
begin
6 let T = xyz \in \{x,y,z \in V\} be a nonface triangle such that p_{TI}p_{TO} \geq 5;
7 wlg assume that p_{TI} \leq p_{TO} otherwise interchange roles of G_{TI} and G_{TO} in
begin
8 if p_{TO} \geq p_{TI} \geq 9 then return HWALK(G_{TI}) + HWALK(G_{TO});
9 if p_{TI} = 7 or 8 and p_{TO} \geq 9 then
begin
comment By Lemma 3(b) at least one of G_{TI}-{x}, G_{TI}-{y}, G_{TI}-{z}, G_{TI}-{x,y}, G_{TI}-{y,z} and G_{TI}-{z,x} is hamiltonian;
10 find a hamiltonian cycle C_I of one of the six graphs in the above comment;
11 return C_I + HWALK(G_{TO})
end;
12 if p_{TI} = 7 or 8 and p_{TO} = 7 or 8 then
if either G_{TI}-{x,y}, G_{TI}-{y,z} or G_{TI}-{z,x} contains a hamiltonian cycle C_I then return C_I + HWALK(G_{TO})
else
if either G_{TO}-{x,y}, G_{TO}-{y,z} or G_{TO}-{z,x} contains a hamiltonian cycle C_0 then return HWALK(G_{TI}) + C_0
end

Fig. 1. Algorithm HWALK.
finds, in $O(p^2)$ time, a closed spanning walk $W$ of $G$ such that

$$l(W) \begin{cases} \frac{3}{2}(p-3) & \text{if } p \geq 11; \\ p & \text{otherwise.} \end{cases}$$

(1a) (1b)

**Proof.** We first prove correctness by induction on the number $p$ of vertices of $G$. If $p \leq 11$, then the algorithm finds a hamiltonian walk $W$ in line 1, and Lemma 2 implies that $l(W)$ satisfies (1). For the inductive step, we assume that the Algorithm correctly finds a closed spanning walk $W$ satisfying (1) in any maximal planar graph with less than $p$ ($\geq 12$) vertices. Let $G$ be a maximal planar graph with $p$ vertices. If $G$ has no nonface triangle (i.e., $G$ is a 4-connected), then the Algorithm returns in line 2 a hamiltonian cycle $W$ (by the algorithm in Lemma 1) which clearly satisfies (1). If either $p_{TI} = 4$ or $p_{TO} = 4$ for each nonface triangle $T$ of $G$, then the Algorithm $LCYCLE$ called in line 4 finds a cycle $C$ of $G$ such that $l(C) \leq \frac{1}{2}(p + 9)$ (Lemma 5), and the Algorithm $HWALK$ returns in line 5 a closed spanning walk $W$ which is constructed from $C$ of $G$. By Lemma 4 we have that

$$l(W) \leq 2p - l(C) \leq \frac{3}{2}(p - 3).$$

In the remaining case in which there exists a nonface triangle $T$ such that $p_{TT}, p_{TO} \geq 5$, we can assume without loss of generality that $p_{TI} \leq p_{TO}$: otherwise interchange roles of $G_{TI}$ and $G_{TO}$. Note that both $G_{TI}$ and $G_{TO}$ are maximal planar graphs with less than $p$ vertices. If $p_{TO} \geq p_{TI} \geq 9$, then recursively calling itself the Algorithm finds closed spanning walks $W_1 = HWALK(G_{TI})$ of $G_{TI}$ and $W_O = HWALK(G_{TO})$ of $G_{TO}$ in line 8. Clearly the multiset $W = W_1 + W_O$ returned in line 8 represents a closed spanning walk of $G$. We shall show $l(W) \leq \frac{1}{2}(p - 3)$. It can be shown that

$$l(W_1) \leq \frac{1}{2}(p_{TI} - 3) \quad \text{and} \quad l(W_O) \leq \frac{1}{2}(p_{TO} - 3).$$

If $p_{TI} \geq 11$, then by the inductive hypothesis $l(W_1) \leq \frac{1}{2}(p_{TI} - 3)$; otherwise, i.e., if
If \( p_{T1} = 9 \) or 10, then HWALK finds a hamiltonian cycle \( W_1 \), so \( l(W_1) = p_{T1} \leq \frac{3}{5}(p_{T1} - 3) \). The proof for the case of \( G_{T0} \) is similar. Since \( p = p_{T1} + p_{T0} - 3 \), we have

\[
l(W) = l(W_1) + l(W_0) \leq \frac{3}{5}(p_{T1} - 3) + \frac{3}{5}(p_{T0} - 3) = \frac{3}{5}(p - 3).
\]

If \( p_{T1} = 7 \) or 8 and \( p_{T0} \geq 9 \) (in line 9), then by Lemma 3(b) at least one of \( G_{T1} \setminus x, G_{T1} \setminus y, G_{T1} \setminus z, G_{T1} \setminus \{x, y\}, G_{T1} \setminus \{y, z\} \) and \( G_{T1} \setminus \{z, x\} \) has a hamiltonian cycle, say \( C_1 \). Clearly \( l(C_1) = p_{T1} - 1 \) or \( p_{T1} - 2 \). Let \( W_0 = HWALK(G_{T0}) \), i.e., a closed spanning walk of \( G_{T0} \) obtained by recursively calling HWALK for \( G_{T0} \). Then since \( 9 \leq p_{T0} < p \), \( l(W_0) \leq \frac{3}{5}(p_{T0} - 3) \) as shown above. Hence \( W = C_1 + W_0 \) is a closed spanning walk of \( G \) and

\[
l(W) \leq p_{T1} - 1 + \frac{3}{5}(p_{T0} - 3) \leq \frac{3}{5}(p - 3).
\]

Using Lemma 3 and the inductive hypothesis we can easily establish the correctness for the remaining cases.

We next prove that the total amount of time spent by HWALK is at most \( O(p^2) \). Algorithm HAMILTON used in line 1 etc. determines whether a given graph with 11 or fewer vertices is hamiltonian or not and returns a hamiltonian walk (or cycle), both in constant time \( (\) since \( p \leq 11 \). By Lemma 1 the algorithm used in line 2 requires \( O(p^2) \) time. By Lemma 5 LCYCLE called in line 4 requires \( O(p^2) \) time, and by Lemma 4 a closed spanning walk of \( G \) can be constructed, in \( O(p) \) time, from a cycle found by LCYCLE. Note that \( O(|E|) = O(p) \) since \( G \) is planar. It shall be noted that if a maximal planar graph \( G \) contains a vertex \( w \) such that both end-vertices \( u \) and \( v \) of an edge \( e = (u, v) \) are adjacent to \( w \) and the triangle \( uvw \) is not a face, then \( uvw \) is a nonface triangle of \( G \). Using this fact, one can determine, in \( O(p) \) time, whether \( G \) contains a nonface triangle with \( e \) as a boundary edge. Since \( O(|E|) = O(p) \), one can find all nonface triangles of \( G \) in \( O(p^2) \) time. It can be easily shown by induction on \( p \) that every maximal planar graph with \( p \) vertices contains at most \( p - 4 \) nonface triangles. Hence one can determine all \( p_{T1} \) and \( p_{T0} \) for all nonface triangles \( T \) of \( G \) in \( O(p^2) \) time. Moreover one can determine the inclusion relation among all nonface triangles of \( G \). The relation is represented by a rooted tree \( R \) such that

(i) the root of \( R \) corresponds to the exterior face triangle of \( G \);

(ii) each vertex of \( R \) except the root corresponds to a nonface triangle of \( G \); and

(iii) a directed edge joins vertex \( x \) to vertex \( y \) in \( R \) if and only if the nonface triangle of \( G \) corresponding to \( y \) is an outmost triangle contained in the triangle corresponding to \( x \).

If \( T \) is a nonface triangle of \( G \), every nonface triangle except \( T \) is also a nonface triangle of \( G_{T1} \) or \( G_{T0} \). Once one finds all nonface triangles \( T \) of \( G \) together with \( p_{T1} \) and \( p_{T0} \) and determines the inclusion relation among them, one can update such information for \( G_{T1} \) and \( G_{T0} \) in \( O(p) \) time. Hence it is not difficult to implement HWALK so that the time \( T(p) \) spent for a graph with \( p \) vertices satisfies

\[
T(p) \leq \max\{k_1 p^2, T(p_{T1}) + T(p_{T0}) + k_2 p, k_3 + T(p_{T0})\},
\]
where $k_1$, $k_2$ and $k_3$ are constants. Noting that $p = p_{T_1} + p_{T_2} - 3$, and solving the above equation, we have that $T(p) \leq O(p^2)$, establishing Theorem 1.

4. Algorithm LCYCLE

In this section we describe a polynomial-time algorithm LCYCLE and prove Lemma 5. Given a maximal planar graph $G$ with $p \geq 11$ vertices such that either $p_{T_1} = 4$ or $p_{T_2} = 4$ for each nonface triangle $T$ of $G$, Algorithm LCYCLE returns a cycle $C$ with $l(C) \geq \frac{1}{2}(p + 9)$ in $O(p^2)$ time.

In order to find a required cycle in a given graph we will employ a kind of augmentation: whenever the graph contains a vertex $x$ off a currently obtained cycle $C$ satisfying a certain requirement, some edges of $C$ are replaced by appropriate edges off $C$ so that $x$ becomes a vertex on the newly obtained cycle and the length of the cycle increases by one. Consider the configurations depicted in Fig. 2, where $C$ is written as $C = v_0v_1v_2\cdots v_0$.

(I) Fig. 2(a) shows a configuration in which $G$ has a vertex $x$ off $C$ which is adjacent to the endvertices $v_1$ and $v_2$ of an edge $(v_1, v_2)$ on $C$. It shall be noted that probably $u_j = u_3$ where $u_j$ is the third vertex to which $x$ is adjacent. (It will be known that every vertex off $C$ is of degree 3.) Clearly cycle $C' = v_0v_1xu_2u_3\cdots u_0$ of $G$ is longer than $C$.

(II) Figs. 2(b) and (c) show configurations in which for some integer $k \geq 1$,
(i) $(v_{i-1}, v_{i+1}) \in E$ for each $i$, $1 \leq i \leq k$, and
(ii) a vertex $x$ off $C$ is adjacent to $v_1$ and $v_{k+2}$.

For simplicity vertex $v_i$ is indicated by '$i$' in Figs. 2(b) and (c). If $k$ is odd, then clearly cycle

$$C' = v_0v_2v_4\cdots v_{k-1}v_k+1v_kv_{k-2}\cdots v_3v_1xv_k+2\cdots v_0$$

Fig. 2. Configurations of type I and II. (An old cycle is drawn by lines on a circle, and a new cycle is drawn by thick lines.)
of $G$ is longer than $C$. (See Fig. 2(b).) If $k$ is even, then cycle

$$C' = v_0v_1v_2\cdots v_{k+1}v_{k-1}\cdots v_4v_1xv_{k+2}\cdots v_0$$

of $G$ is longer than $C$. (See Fig. 2(c).)

The configuration illustrated in Fig. 2(a) is called of type I, and both in Figs. 2(b) and (c) together with symmetric ones are called of type II. Note that a configuration of type I can be regarded as a special case of type II with $k = 0$.

For an illustration we depict in Fig. 3 a maximal planar graph $G = (V, E)$ with $V = \{0, 1, 2, \ldots, 16\}$. Cycle $C = 012\cdots(14)0$ is drawn by lines on a circle. Vertices 15 and 16 off $C$ are of degree 3. $G$ contains a configuration of type I with respect to vertex 16: $(16, 0), (16, 1) \in E$. $G$ also contains a configuration of type II with respect to vertex 15: $(6, 8), (7, 9), (8, 10), (15, 7), (15, 11) \in E (k = 3)$. The new cycle

$$C' = 0(16)1234568(10)97(15)(11)(12)(13)(14)0$$

longer than $C$ is drawn by thick lines.

Algorithm LCYCLE is depicted in Fig. 4. We assume that cycle $C$ is written generically as $v_0v_1v_2\cdots v_{l(C)}v_0$ at any stage of the algorithm.

We next present the proof of Lemma 5.

**Proof of Lemma 5.** In order to prove the correctness of Algorithm LCYCLE, it is sufficient to show that (i) if $p \leq 16$, then $G$ contains a cycle $C$ of length $l(C) \leq \frac{1}{2}(p + 9)$, and that (ii) if $p \geq 17$ and $C$ is an arbitrary cycle of length $l(C) < \frac{1}{2}(p + 9)$.

Fig. 3. An illustrating example.
procedure LCYCLE(G):
1 if p < 16 then return a cycle C of length \( \ell(C) \geq \frac{(p+9)}{2} \)
   found by any reasonable algorithm
else
   begin
2 let \( V_3 \) be the set of all vertices of degree 3;
3 \( G' \leftarrow G - V_3; \)
   comment \( G' \) is a 4-connected maximal planar graph;
4 C \leftarrow \) a hamiltonian cycle of \( G' \) which can be found
   by the algorithm in Lemma 1;
5 \( \ell(C) \leftarrow p - |V_3|; \)
6 while \( \ell(C) < \frac{(p+9)}{2} \) do
   begin
7 if \( G \) contains a configuration of type I then
   wlg assume that vertex \( x \) off \( C \) is adjacent
   to \( v_1 \) and \( v_2 \), the endvertices of edge
   \( (v_1, v_2) \) on \( C \) otherwise rename the vertices
   on \( C \) in
   begin
8 \( \ell(C) \leftarrow \ell(C) + 1; \)
9 \( C \leftarrow v_0v_1v_3v_2...v_0 \)
   end
   else
   begin
   comment \( G \) contains a configuration of type II;
   wlg assume that for some \( k \geq 1 \) (i) \( (v_{i-1}, v_{i+1}) \in E \)
   for each \( i \), \( 1 \leq i \leq k \), and (ii) vertex \( x \) off \( C \)
   is adjacent to \( v_1 \) and \( v_{k+2} \) otherwise rename
   vertices on \( C \) in
   begin
10 \( \ell(C) \leftarrow \ell(C) + 1; \)
11 if \( k \) is odd then
12 \( C \leftarrow v_0v_2v_4...v_{k-2}v_{k+1}v_k...v_3v_1v_{k+2}...v_0 \)
   else
13 \( C \leftarrow v_0v_2v_4...v_kv_{k+1}v_{k-1}...v_3v_1v_{k+2}...v_0 \)
   end
   end
   end
   end

Fig. 4. Algorithm LCYCLE.

such that every vertex off \( C \) has degree 3, then \( G \) contains a configuration of type I
or II. In Section 4 of our previous paper [2] we showed via a lengthy argument that
every maximal planar graph satisfying the requirement of Lemma 5 contains a cycle
\( C \) of length \( l(C) \geq \frac{1}{4}(p + 9) \). Thus (i) above has been verified. Furthermore one can
see that (ii) above is implicit in the arguments in Stages 1 and 2 of Section 4 of [2]. It
shall be noted that the graph \( G' = G - V_3 \) in line 3 is 4-connected since \( G' \) contains no
nonface triangles: otherwise \( G \) would contain a nonface triangle \( T \) with \( p_{T_B} p_{T_O} \geq 5 \), contradicting the assumption of \( G \). Thus every vertex off \( C \) is of degree 3 for the cycle \( C \) obtained in line 4. Once a vertex of \( G \) is inserted into \( C \), it will be never deleted from \( C \) in the algorithm. Consequently, every vertex off \( C \) is of degree 3 at any stage of \( \text{LCYCLE} \).

We next establish our claim on the time complexity of \( \text{LCYCLE} \). If \( p \leq 16 \), one can find a cycle \( C \) of length \( l(C) \geq \frac{1}{4}(p + 9) \) in constant time. Therefore line 1 of \( \text{LCYCLE} \) requires \( O(1) \) time. Clearly lines 2 and 3 can be executed in \( O(p) \) time. In line 4 we use the algorithm of Lemma 1 which is the most time consuming part of \( \text{LCYCLE} \) and requires \( O(p^2) \) time. Clearly line 5 requires \( O(1) \) time. If edge \( e = (v_1, v_2) \) is on \( C \), \( (v_1, x) \) is an edge which is incident to \( v_1 \) and is clockwise or counterclockwise next to \( e \) in the plane embedding of \( G \), and \( x \) is off \( C \), then \( v_2 \) is adjacent to \( x \), i.e., \( G \) contains a configuration of type I. Using the doubly linked adjacency lists for representing the plane embedding of \( G \) so that an edge embedded next to a given edge can be directly accessed, one can determine in \( O(1) \) time whether both endvertices of a given edge on \( C \) are adjacent to a vertex off \( C \). Checking this for each edge on \( C \), one can determine in \( O(p) \) time whether \( G \) contains a configuration of type I. Note that one can embed a planar graph \( G \) on the plane in \( O(p) \) time [10]. Similarly one can determine in \( O(p) \) time whether \( G \) contains a configuration of type II. Each time a configuration of type I or II is found, cycle \( C \) is replaced by a longer one in lines 10, 13 and 14. Each replacement requires \( O(p) \) time. Clearly each execution of line 9 or 12 requires \( O(1) \) time. Thus every execution of the loop of lines 7–14 requires \( O(p) \) time. Since \( l(C) \) increases by one after every execution of the loop, the loop is executed at most \( p \) times. Hence the total amount \( L(p) \) of time spent by \( \text{LCYCLE} \) for a graph \( G \) satisfies \( L(p) \leq O(p^2) \), so we have the lemma.

Acknowledgements

We wish to thank Professor N. Saito for valuable discussions and suggestions on the subjects. This work was supported in part by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant: Cooperative Research (A) 435013(1979), EYS 475235(1979) and EYS 475259(1979).

Note added in proof

Recently the conjecture mentioned at the end of Section 1 has been proved in: V. Chvátal, Hamiltonian cycles, Tech. Rep. 81-38, School of Compt. Sci., McGill Univ., Montreal, Canada (1981).

References


