数理解析研究所講究録
第 397 巻 1980 年 156-178

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

by

Takao Nishizeki*

Takao Asano*

(*Faculty of Engineering, Tohoku University)

and

Takahiro Watanabe**

(**R & D Center, Toshiba Corporation)

Abstract 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. The hamiltonian walk problem in which one would like to find a hamiltonian walk of a given graph is a generalized hamiltonian cycle problem and is a modified traveling salesman problem, and is of course NP-complete. Employing the divide-and-conqure and greedy optimization techniques, we present a polynomial-time approximation algorithm with a constant worst-case bound for this problem. Our algorithm finds, in $O(p^2)$ time, a closed spanning walk of a given arbitrary maximal planar graph with p vertices, and the length of the obtained walk is always smaller than 3/2 times the length of a shortest one (i.e., a hamiltonian walk).

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 such 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 whether a given graph contains a hamiltonian cycle or not. 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 if we restrict ourselves to a class of (3-connected cubic) planar graphs. Hence the hamiltonian walk problem is also NP-complete even if we restrict ourselves to the same class. On the other hand the hamiltonian walk problem is a special case of the well known traveling salesman problem, restricted in a way that each edge of a given graph has a unit weight. Of course the traveling salesman problem is NP-complete [1]. It has been conjectured that there exist no polynomial-time exact algorithms for any of NP-complete problems. Consequently, attention has been given to developing algorithms that solve various NP-complete problems efficiently but only approximately [5]. We shall restrict our attention to approximation algorithms with a constant bound worst-case ratio.

Sahni and Gonzalez [12] have shown that if the triangle inequality is not satisfied, the problem of finding an approximation solution for the traveling salesman problem within any constant bound ratio of the optimum is as difficult as finding an exact solution. Christofides [4] has developed a polynomial-time algorithm with a worst-case bound of 3/2 for the problem in which the triangle inequality is satisfied. On the other hand, Hwang [11] has given a polynomial-time algorithm with the same worst-case bound for the rectilinear Steiner tree problem.

For the hamiltonian walk problem there exists a trivial approximation algorithm with a worst-case bound of 2: find a (spanning) tree of a given connected graph; and construct a closed spanning walk of the graph which traverses twice each edge of the tree; then the length of the walk is 2(p-1) if the graph has p vertices; clearly the length is smaller than twice the length of a shortest one (i.e. a hamiltonian walk).

In this paper we present a polynomial-time approximation algorithm with a worst-case bound of 3/2 for a restricted hamiltonian walk problem. Given a maximal planar graph with p vertices, our algorithm finds, in $O(p^2)$ time, a closed spanning walk of the graph whose length is smaller than 3/2 times the length of a shortest one. We will empoly, in our algorithm, two techniques: divide-and-conqure and greedy optimization. The algorithm is based on two early results : one is our previous work establishing that a maximal planar graph with p vertices always contains either a hamiltonian cycle or a closed spanning walk of length $\leq 3(p-3)/2$ [2]; the other is Whitney's establishing that every 4-connected maximal planar graph has a hamiltonian cycle [14]. 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 $v_0e_1v_1e_2 \dots e_kv_k$ whose terms alternately vertices and edges, such that the endvertices 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 $\ell(W)$. 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 of G. For a connected graph G, h(G) denotes 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 (≥ 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 a graph obtained from G by deleting all vertices in V'. A singleton set {v} is simply denoted by "v". 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, i.e., a sequence of edges and vertices, can be easily constructed from the multiset of edges. Note that this can be done by any algorithm for finding an eulerian walk of an eulerian graph. Refer to [1] or [9] for all undefined terms.

We next present some lemmas. Generalizing Whitney's result [14], Tutte has shown that every 4-connected planar graph has a hamiltonian cycle [13]. 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 an $O(p^2)$ time-algorithm for finding a hamiltonian cycle in a 4-connected maximal planar graph G with p vertices.

Proof. It is not difficult to implement a recursive algorithm for finding a hamiltonian cycle of G in $O(p^2)$ time, completely based on the inductive proof of Whitney [14] ensuring its existence. Q.E.D.

LEMMA 2. (a) Every maximal planar graph with ten or fewer vertices contains a hamiltonian cycle [3][14]. (b) Every nonhamiltonian maximal planar graph with 11 vertices has a hamiltonian walk of length 12. (Note 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.

One can easily develop an algorithm for determining whether a given graph with 11 or fewer vertices contains a hamiltonian cycle or not, and finding a hamiltonian walk in constant time. Let HAMILTON(G) be such an algorithm, which 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 of G, one can find a closed spanning walk W of G such that $\ell(W) \leq 2p - c$, 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 $2p - c$.

Q.E.D.

For a nonface triangle T of a maximal planar graph G, let $G_{TI} = (V_{TI}, E_{TI})$ denote the induced subgraph of G inside T, and $G_{TO} = (V_{TO}, E_{TO})$ 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_{TI} is the subgraph of G induced by the vertex set $\{x,y,z\} \cup U'(T)$, i.e., $G_{TI} = G - U''(T)$, and G_{TO} is the subgraph of G induced by the vertex set $\{x,y,z\} \cup U''(T)$, i.e., $G_{TO} = G - U'(T)$. Let $p_{TI} = |V_{TI}|$ and $p_{TO} = |V_{TO}|$. 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 $p_{TI} = 4$ or $p_{TO} = 4$ for each nonface triangle T of G, Algorithm LCYCLE finds a cycle C of length $\ell(C) \geq (p+9)/2$, 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 $\ell(W) \leq \max\{p, 3(p-3)/2\}$ of a given maximal planar graph G with p vertices. In the algorithm we will employ the divide-and-conqure 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_{TI} and G_{TO}, (ii) recursively call the algorithm with respect to G_{TI} and G_{TO}, and (iii) combine the closed spanning walks of G_{TI} and G_{TO} into a closed spanning walk of the whole graph G.

The Algol-like procedure HWALK depicted 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 3/2, 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 finds, in $O(p^2)$ time, a closed spanning walk W of G such that

$$\ell(W) \begin{cases} \leq 3(p-3)/2 & \text{if } p \geq 11; & (1\text{-a}) \\ = p & \text{otherwise.} & (1\text{-b}) \end{cases}$$

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 $\ell(W)$ satisfies (1). For the inductive step, we assume that the Algorithm correctly finds a closed spanning walk W satisfying (1) on any maximal planar graph with less than p (≥ 12) vertices. Let G be a maximal planar graph with p vertices. If G has no nonface triangle (i.e., G is 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 $\ell(C) \geq (p+9)/2$ (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

$\ell(W) \leq 2p - \ell(C) \leq 3(p-3)/2$. In the remaining case in which there exists a nonface triangle T such that $p_{TI}, p_{TO} \geq 5$, 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_I = HWALK(G_{TI}) of G_{TI} and W_O = HWALK(G_{TO}) of G_{TO} in line 8. Clearly the multiset $W=W_I+W_O$ returned in line 8 represents a closed spanning walk of G. We shall show $\ell(W) \leq 3(p-3)/2$. It can be shown that $\ell(W_I) \leq 3(p_{TI}-3)/2$ and $\ell(W_O) \leq 3(p_{TO}-3)/2$: if $p_{TI} \geq 11$, then by the inductive hypothesis $\ell(W_I) \leq 3(p_{TI}-3)/2$; otherwise, i.e., if p_{TI} = 9 or 10, then HWALK finds a hamiltonian cycle W_I, so $\ell(W_I) = p_{TI} \leq 3(p_{TI}-3)/2$; the proof for the case of G_{TO} is similar. Since $p = p_{TI} + p_{TO} - 3$, we have

$$\ell(W) = \ell(W_I) + \ell(W_O) \leq 3(p_{TI}-3)/2 + 3(p_{TO}-3)/2 = 3(p-3)/2.$$

If p_{TI} = 7 or 8 and $p_{TO} \geq 9$ (in line 9), then 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 $G_{TI} - \{z,x\}$ has a hamiltonian cycle, say C_I. Clearly $\ell(C_I) = p_{TI} - 1$ or $p_{TI} - 2$. Let W_O = HWALK(G_{TO}), i.e., a closed spanning walk of G_{TO} obtained by recursively calling HWALK for G_{TO}. Then since $9 \leq p_{TO} < p$, $\ell(W_O) \leq 3(p_{TO}-3)/2$ as shown above. Hence $W = C_I + W_O$ is a closed spanning walk of G and

$$\ell(W) \leq p_{TI} - 1 + 3(p_{TO}-3)/2 \leq 3(p-3)/2.$$

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. 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 endvertices 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_{TI} and p_{TO} 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_{TI} or G_{TO}. Once one finds all nonface triangles T of G together with p_{TI} and p_{TO} and determines the inclusion relation among them, one can update such information for G_{TI} and G_{TO} 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

$$T(p) \leq \max\{k_1p^2,\ T(p_{TI}) + T(p_{TO}) + k_2p,\ k_3 + T(p_{TO})\},$$

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

Q.E.D.

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 $p_{TI}=4$ or $p_{TO}=4$ for each nonface triangle T of G, Algorithm LCYCLE returns a cycle C with $\ell(C) \geq (p+9)/2$ in $O(p^2)$ time.

In order to find a required cycle in a given graph we will employ a kind of greedy algorithm : 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 $v_j = v_3$ where v_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 $C' = v_0v_1xv_2v_3\cdots v_0$ of G is longer than C.

(II) Fig. 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 Fig. 2(b) and (c). If k is odd, then clearly cycle C'

$$C' = v_0v_2v_4\cdots v_{k-1}v_{k+1}v_kv_{k-2}\cdots v_3v_1xv_{k+2}\cdots v_0$$

of G is longer than C. (See Fig. 2(b).) If k is even, then cycle C'

$$C' = v_0v_2v_4\cdots v_kv_{k+1}v_{k-1}\cdots v_3v_1xv_{k+2}\cdots v_0$$

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

The configuration illustrated in Fig. 2(a) is called <u>of type I</u>, and both in Fig. 2(b) and (c) together with symmetric ones are called <u>of type II</u>. 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,...,16}. Cycle C = 012...(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)9 7(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 $v_0v_1v_2\cdots v_{\ell(C)-1}v_0$ at any stage of the algorithm.

We next present the proof of Lemma 5

<u>Proof of Lemma 5</u>. 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 $\ell(C) \geq (p+9)/2$, and that (ii) if $p \geq 17$ and C is an arbitrary cycle of length $\ell(C) < (p+9)/2$ such that every vertex off C is 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 $\ell(C) \geq (p+9)/2$. Thus the

above (i) has been verified. Furthermore one can see that above (ii) 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 $p_{TI}, p_{TO} \geq 5$, 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 LCYCLE.

We next establish our claim on the time complexity of LCYCLE. If $p \leq 16$, one can find a cycle C of length $\ell(C) \geq (p+9)/2$ in constant time. Therefore line 1 of 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 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 clockwisely or counterclockwisely 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 an appropriate data structure 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 $\ell(C)$ 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 LCYCLE for a graph G satisfies $L(p) \leq O(p^2)$, so we have the Lemma. Q.E.D.

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).

REFERENCES

[1] A.V. AHO, J.E. HOPCROFT AND J.D. ULLMAN, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] T. ASANO, T. WATANABE AND T. NISHIZEKI, An upper bound on the length of a hamiltonian walk of a maximal planar graph, J. of Graph Theory, to appear.

[3] D. BARNETTE AND E.J. JUCOVIČ, Hamiltonian circuits on 3-polytopes, J. Combinatorial Theory, 2(1970), pp. 54-59.

[4] N. CHRISTOFIDES, Worst-case analysis of a new heuristic for the traveling salesman problem, Managing Sciences Research Report, 388, Carnegie-Mellon University (1976).

[5] M.R. GAREY AND D.S. JOHNSON, Approximation algorithm for combinatorial problems: An annoted bibliography, Algorithms and Complexity: Recent Results and New Directions, J.F. Traub. ed. Academic Press, New York, 1976, pp. 41-52.

[6] 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.

[7] S.E. GOODMAN AND S.T. HEDETNIEMI, On Hamiltonian walk in graphs, SIAM J. Comput., 3(1974), pp. 214-221.

[8] D. GOUYOU-BEAUCHAMPS, Un algorithme de recherche de circuit Hamiltonien dans les graphes 4-connexes planaires, Colloques Internationaux CNRS, No. 260 - Problèms Combinatoires et Thèorie des Graphes, ed. J.C. Bermond, J.C. Fournier, M. Las Vergnas and D. Scotteau, (1978), pp. 185-187.

[9] F. HARARY, Graph Theory, Addison-Wesley, Reading, MA, 1969.

[10] J. HOPCROFT AND R. TARJAN, Efficient planarity testing, J. Assoc. Comput. Mach., 21 (1974), pp. 549-568.

[11] F.K. HWANG, On Steiner minimal tree with rectilinear distance, SIAM, J. Appl. Math., 30(1976), pp. 104-114.

[12] S.K. SAHNI AND T. GONZALEZ, P-complete approximation problems, J. Assoc. Comput. Mach., 23(1976), pp. 555-565.

[13] W.T. TUTTE, A theorem on planar graphs, Trans. Amer. Math. Soc., 82(1956), pp. 99-116.

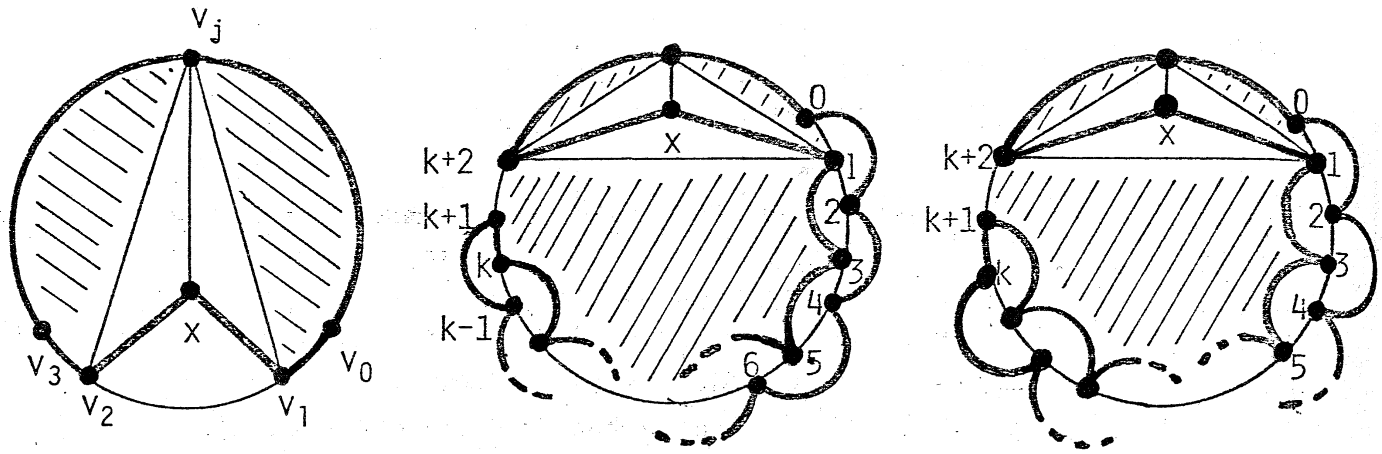
[14] H. WHITNEY, A theorem on graphs, Annals Math.,32(1931), pp. 378-390.

```
   procedure HWALK(G):
   begin
1     if p ≤ 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) ≥ (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 (x,y,z∈V) be a nonface triangle
                such that p_TI,p_TO ≥ 5;
               comment p = p_TI + p_TO - 3 ≥ 12;
7              wlg  assume that p_TI ≤ p_TO otherwise interchange
                roles of G_TI and G_TO in
                  begin
8                    if p_TO ≥ p_TI ≥ 9 then return
                      HWALK(G_TI) + HWALK(G_TO);
9                    if p_TI = 7 or 8 and p_TO ≥ 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;
```

Fig. 1   Algorithm HWALK.   (continued)

```
12          if p_TI = 7 or 8 and p_TO = 7 or 8 then
13             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
14                if either G_TO-{x,y}, G_TO-{y,z} or
                   G_TO-{z,x} contains a hamiltonian
                    cycle C_O then return HWALK(G_TI) + C_O
15                else return
                    HAMILTON(G_TI-x) + HAMILTON(G_TO-y);
16          if p_TI = 5 or 6 then
               begin
                  comment Either G_TI-{x,y}, G_TI-{y,z}
                   or G_TI-{z,x} is hamiltonian;
17                wlg G_TI-{x,y} is hamiltonian in
                     find a hamiltonian cycle C_I of
                      G_TI-{x,y};
18                return C_I + HWALK(G_TO)
               end
         end
      end
end
```

Fig. 1 Algorithm HWALK.



(a) Type I (b) Type II with odd k (c) Type II with even k

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.)

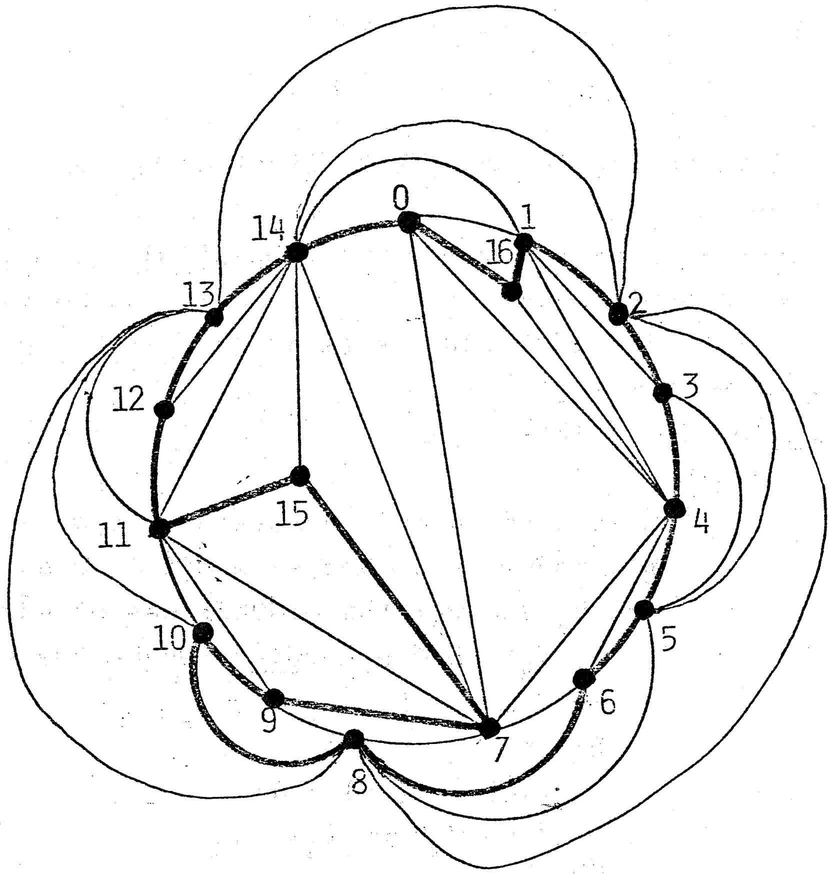


Fig. 3 An Illustrating Example.

```
   procedure LCYCLE(G):
1  if p ≤ 16 then return a cycle C of length ℓ(C) ≥ (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' ← G - V_3;
         comment G' is a 4-connected maximal planar graph;
4        C ← a hamiltonian cycle of G' which can be found
               by the algorithm in Lemma 1;
5        ℓ(C) ← p - |V_3|;
6        while ℓ(C) < (p+9)/2 do
            begin
7              if G contains a configuration of type I then
8                 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
9                      ℓ(C) ← ℓ(C) + 1;
10                     C ← v_0 v_1 x v_2 v_3 ... v_0
                    end
               else
                 begin
                  comment G contains a configuration of type II;
11                wlg assume that for some k ≥ 1 (i) (v_{i-1},v_{i+1})∈E
                   for each i, 1 ≤ i ≤ k, and (ii) vertex x off C
                    is adjacent to v_1 and v_{k+2} otherwise rename
                    vertices on C in
                    begin
12                     ℓ(C) ← ℓ(C) + 1;
13                     if k is odd then
                        C ← v_0 v_2 v_4 ... v_{k-1} v_{k+1} v_k ... v_3 v_1 x v_{k+2} ... v_0
                       else
14                      C ← v_0 v_2 v_4 ... v_k v_{k+1} v_{k-1} ... v_3 v_1 x v_{k+2} ... v_0
                    end
                 end
            end
      end
```

Fig. 4 Algorithm LCYCLE.