The Pagenumber of Genus g Graphs is O(g)

Lenwood Heath and Sorin Istrail

TR 90-21

THE PAGENUMBER OF

GENUS g GRAPHS IS O(g)

Lenwood S. Heath
Virginia Polytechnic Institute
& State University

Sorin Istrail
Wesleyan University

Abstract. In 1979, Bernhart and Kainen conjectured that graphs of fixed genus $g \geq 1$ have unbounded pagenumber. This paper proves that genus g graphs can be embedded in $O(g)$ pages, thus disproving the conjecture. An $\Omega(\sqrt{g})$ lower bound is also derived. The first algorithm in the literature for embedding an arbitrary graph in a book with a non-trivial upper bound on the number of pages is presented. First, the algorithm computes the genus g of a graph using the algorithm of Filotti, Miller, Reif (1979), which is polynomial-time for fixed genus. Second, it applies an optimal-time algorithm for obtaining an $O(g)$-page book embedding. We give separate book embedding algorithms for the cases of graphs embedded in orientable and nonorientable surfaces. An important aspect of the construction is a new decomposition theorem, of independent interest, for a graph embedded on a surface. Book embedding has application in several areas, two of which are directly related to the results obtained: fault-tolerant VLSI and complexity theory.

Categories and Subject Descriptors: F.2.2 **[Analysis of Algorithms and Problem Complexity]:** Nonnumerical Algorithms and Problems—*routing and layout;* G.2.2 **[Discrete Mathematics]:** Graph Theory—*graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Book embeddings, graph genus, surface embeddings, homotopy classes, planar-nonplanar decomposition

---

A preliminary announcement of this research appeared in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing.* ACM, New York, 1987, pp. 388-397.

This research was done while the first author was in the Mathematics Department at the Massachusetts Institute of Technology. Research supported by Air Force Contract AFSOR-86-0078.

Authors' present addresses: L. Heath, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061; S. Istrail, Department of Mathematics, Wesleyan University, Middletown, CT 06457.

1. Introduction

A (simple, undirected) *graph* $G = (V, E)$ consists of a finite set of vertices, V, and a set of edges, E; each edge is a two-element subset of V. Thus a graph has no loops or multiple edges. An edge $\{u, v\} \in E$ is denoted by (u, v); since G is undirected, $(u, v) = (v, u)$. A *book* consists of two parts: a *spine*, which is a line, and some number of *pages*, each of which is a half-plane having the spine as boundary. A *book embedding* of a graph consists of an ordering of the vertices of the graph along the spine, and an assignment of each edge to a single page, so that all the edges assigned to a page can be drawn in the page without crossings. The minimum number of pages in which a graph can be embedded is its *pagenumber*.

As an example, consider the complete bipartite graph G on the two sets of vertices $U = \{u_1, u_2, u_3\}$ and $V = \{v_1, v_2, v_3\}$. This graph is isomorphic to $K_{3,3}$, and hence nonplanar. By a result of Bernhart and Kainen [1], at least three pages are necessary for any nonplanar graph. Figure 1 illustrates one 3-page book embedding for G. The vertices are arranged on the spine in some order. The first page is the half-plane above the spine. The second page is the half-plane below the spine. The third page is indicated by dashed lines below the spine. There are 5 edges assigned to the first page, all drawn in that page so that no two edges cross. There are 3 edges assigned to the second page, again drawn so that no two edges cross; however, no one of the 3 edges could have been drawn in the first page without crossing one of the edges already there. There is one (dashed) edge (u_1, v_3) assigned to the third page; this edge could not be drawn in either of the first two pages without crossing an edge already there.

Recent interest in book embeddings has been motivated by VLSI design ([3], [20]), and by complexity theory ([6], [16], [19]). In the Diogenes methodology for the design of fault-tolerant arrays of VLSI processors [20], an array of identical processors is implemented using a book embedding; each page of the book embedding corresponds to a hardware "stack" in the implementation. Thus book embeddings of small pagenumber are related to reduced hardware complexity. The Diogenes methodology assumes as input an *arbitrary graph* for the array. However, there has been no (nontrivial) algorithm for embedding an arbitrary graph in a book. This paper presents the first efficient algorithm to embed an arbitrary graph in a book with a nontrivial upper bound on the pagenumber.

Noteworthy progress has been made for book embeddings of planar graphs. In 1979, Bernhart and Kainen [1] conjectured that the pagenumber of the class of planar graphs is unbounded. In 1984, Buss and Shor [2] disproved that conjecture by showing that any planar graph can be embedded in 9 pages. Also in 1984, using other techniques, Heath [12] improved that to 7 pages and developed book embedding algorithms for special classes of planar graphs [13]. In 1985, Istrail [15] showed that planar graphs require no more than 6 pages. In 1986, extending the techniques of [12], Yannakakis [22] proved that the pagenumber of planar graphs is exactly 4.

Bernhart and Kainen [1] proved that graphs with pagenumber 3 can have arbitrarily large genus. Looking for a result in the other direction, they conjectured that, by fixing the graph genus, the corresponding graphs will require an unbounded number of pages. They stated their conjecture in the very strong form of fixed genus 0 (i.e., planar graphs). That conjecture, of course, has been disproved. The more general conjecture, for fixed genus $g \geq 1$, has remained open since 1979.

The main result of this paper is the development of new book embedding algorithms for graphs embedded in a surface of genus g, where $g \geq 1$. We show that graphs of genus g have pagenumber $O(g)$. This disproves the conjecture of Bernhart and Kainen [1]. The previous upper bound was $O(\sqrt{gn})$ (combining [3] and [7]), where n is the number of vertices in the graph. We also obtain a lower bound on pagenumber of $\Omega(\sqrt{g})$. The method used in our algorithms is not a generalization of the techniques for embedding planar graphs in books. On the contrary, it relies on a new decomposition theorem for a graph of genus g, $g \geq 1$. This decomposition is of independent interest, and has potential applications distinct from book embeddings.

In this paper, we discuss graphs embedded in books and also graphs embedded in a surface of genus g. To avoid confusion, we use "layout" for a book embedding and "embedding" for a surface embedding. We follow the development of White [21] for graphs embedded in surfaces. An orientable surface of genus g is a sphere with g *handles;* a handle is a cylinder attached in an "oriented" way to the boundaries of two disks cut in the sphere. A nonorientable surface of genus g is a sphere with g *cross-caps;* a cross-cap is a cylinder attached in a "twisted" way to the boundaries of two disks cut in the sphere. A connected graph $G = (V, E)$ is embedded in a surface if it is drawn on the surface without crossing edges. The *genus* of G, $\gamma(G)$, is the minimum genus of an orientable surface into which G is embeddable. The *nonorientable genus* of G, $\tilde{\gamma}(G)$, is analogous for nonorientable surfaces. The connected components of the complement of an embedding of G are the *faces* of the embedding. The embedding is *2-cell* if every face is homeomorphic to an open disk. Any embedding of G in an orientable surface of genus $\gamma(G)$ is a 2-cell embedding (Youngs [23] or White [21], Theorem 6-11). The analogous statement for nonorientable surfaces is not necessarily true; however, we will assume that all given surface embeddings of graphs are 2-cell.

Six sections follow. Section 2 develops our new decomposition for graphs embedded in surfaces. Layout algorithms for graphs embedded in orientable and nonorientable surfaces are in sections 3 and 4, respectively. The analysis of the time complexity of the algorithms is deferred until section 5. Section 6 gives the short argument for an $\Omega(\sqrt{g})$ lower bound for the pagenumber of genus g graphs. We conclude in section 7 with two conjectures.

2. Decomposition Theorem

In this section, we develop a new decomposition for a connected, undirected graph $G = (V, E)$ embedded in a surface (orientable or nonorientable) of genus g. We let $n = |V|$ throughout. Our aim is to choose a planar subgraph $G_P = (V, E_P)$ of G (with a fixed planar embedding that is a subembedding of the genus g embedding) containing all vertices of V so that the following is true:

(A) the remaining edges of E attach to the *boundary vertices* (vertices on the exterior face) of G_P.

Let $E_N = E - E_P$. An edge $e \in E_N$ is *essentially nonplanar* with respect to G_P in the sense that e cannot be embedded in the plane with G_P without violating (A). As an example, consider the graph G of Figure 2(a). A choice for G_P is shown in Figure 2(b). The boundary vertices of G_P are v_1, v_2, v_3, v_4, v_5; the nonplanar edges are (v_1, v_4) and (v_3, v_5). (v_1, v_4) is essentially nonplanar because its addition to G_P would either remove v_5 (Figure 2(c)) or v_2 and v_3 (Figure 2(d)) from the boundary; in either case, the edge (v_3, v_5) would not be attached to the boundary of G_P as required by (A). (v_3, v_5) is similarly essentially nonplanar. (We follow certain conventions in our Figures. Thin lines represent planar edges; thicker lines represent nonplanar edges. Thin curves represent paths in the planar part. Shading represents the interior of the planar part.)

Before defining the decomposition, we need a representation for graphs embedded in surfaces due to Heffter [10] and Edmonds [4]. Let $G = (V, E)$ be a connected, undirected graph. For each $v \in V$, the *neighborhood* of v is $N(v) = \{u | (u, v) \in E\}$. A *rotation* of G is a set of $|V|$ cyclic permutations

$$R = \{\pi_v | v \in V \text{ and } \pi_v \text{ is a cyclic permutation of } N(v)\}.$$

If $H = (V_H, E_H)$ is a subgraph of G, define $N_H(v) = \{u | (u, v) \in E_H\}$. If π_v is a cyclic permutation of $N(v)$, then define $\pi_{v,H}$ to be the cyclic permutation of $N_H(v)$ that is consistent with the cyclic order of π_v. A rotation R of G induces the *subrotation* $R_H = \{\pi_{v,H} | v \in V_H\}$ of H.

Rotations represent surface embeddings. By Theorem 3.2.3 of [8], every rotation represents a unique 2-cell embedding of G in an orientable surface of some genus (not necessarily $\gamma(G)$). Conversely each embedding of G into an oriented surface is represented by a unique rotation (up to orientation-preserving equivalence). If G has a 2-cell embedding in an orientable surface, there is a rotation R of G representing the embedding. Each π_v is given by examining the edges $\{(u,v)|u \in N(v)\}$ on the surface in, say, clockwise order about v. (Remember, the edges of the embedding are noncrossing, so a clockwise visit of the edges out of v is well-defined.) For our purposes, the key property of this rotation is that it efficiently represents the boundary of any face of the embedding. Suppose $v_1, v_2, \ldots, v_k, v_1$ is the sequence of vertices encountered in traversing the boundary of a face in counterclockwise order. Then $v_{i+1} = \pi_{v_i}(v_{i-1}), 2 \leq i < k$, and $v_1 = \pi_{v_k}(v_{k-1})$. This representation allows the boundary of any face to be traversed in constant time per edge.

For a graph embedded in a nonorientable surface, a rotation is not quite sufficient to allow traversing the boundary of each face. In particular, the representation for each edge must include an *orientation* ([8], Section 3.2): whether the edge is orientation-preserving or orientation-reversing. In the case of a nonorientable surface, we take the definition (and representation) of rotation to include an orientation for each edge. Face traversal in constant time per edge is again possible.

Definition. A *planar-nonplanar decomposition* of $G = (V, E)$ is a triple (R, G_P, E_N), where R is a rotation of G representing a surface embedding, $G_P = (V, E_P)$ is a planar subgraph of G, and $E_N = E - E_P$, which satisfies these properties

(1) the subrotation R_{G_P} induces a planar embedding of G_P;

(2) there exists a face F_0 of the planar embedding such that each $e \in E_N$ is incident to two vertices on the boundary of F_0;

(3) E_P is maximal, i.e., no edge of E_N can be added to G_P without violating either property (1) or (2).

In Figure 3, G_P is represented by the interior of the large oval; the oval itself represents the boundary of G_P. $E_N = \{(u_i, v_i)|1 \leq i \leq 9\}$.

For definiteness, we take F_0 to be the exterior face of the planar embedding. (See Figure 4.) We can imagine traversing the boundary of F_0 in, say, clockwise order. Each vertex on the boundary is encountered at least once (multiple times if it is an articulation point of G_P); each edge on the boundary is encountered at least once (twice if it is a cut-edge of G_P). In any case, the traversal of the boundary defines a directed cycle (which is, in general, not simple). A directed subpath of this directed cycle is a *trace*. If the trace T consists of the sequence of vertices $v_1, v_2, \ldots, v_t$, then denote the trace by

$$T = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_t.$$

v_1 and v_t are the *endpoints* of T. The *inverse trace* to T is

$$T^{-1} = v_t \rightarrow v_{t-1} \rightarrow \ldots \rightarrow v_1,$$

i.e., the trace gotten by traversing T in the opposite direction.

For example, in Figure 4, the directed cycle for a clockwise traversal of the boundary of F_0 beginning at w_1 is

$$w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_5, w_4, w_2, w_1.$$

w_1, w_2, w_4, and w_5 are articulation points of G_P and occur twice in the directed cycle. (w_1, w_2) and (w_4, w_5) are cut edges of G_P and are traversed twice (once in each direction). A sample trace is

$$T = w_2 \rightarrow w_3 \rightarrow w_4 \rightarrow w_5 \rightarrow w_6$$

which has inverse

$$T^{-1} = w_6 \rightarrow w_5 \rightarrow w_4 \rightarrow w_3 \rightarrow w_2.$$

The following is not a trace of G_P

$$w_3 \rightarrow w_4 \rightarrow w_2$$

because w_2 does not follow w_4 in a clockwise (or counterclockwise) traversal of the boundary of G_P.

Given a planar-nonplanar decomposition (R, G_P, E_N) of G, our next aim is to define a partition of E_N into equivalence classes. We assume we have a 2-cell embedding of G into a surface such that R is a rotation consistent with the embedding. Suppose that $(u_1, v_1), (u_2, v_2) \in E_N$ are part of the boundary of the same face F of the embedding of G. (Notice that since $(u_1, v_1), (u_2, v_2) \notin E_P$, the face F is not a face of G_P.) Then (u_1, v_1) and (u_2, v_2) are *homotopic* (with respect to F) if

(1) (u_1, v_1) and (u_2, v_2) are the only edges of E_N on the boundary of F;

(2) there are traces $T_u = u_1 \rightarrow \ldots \rightarrow u_2$ and $T_v = v_1 \rightarrow \ldots \rightarrow v_2$ such that both T_u and T_v lie on the boundary of F.

If (u_1, v_1) and (u_2, v_2) are homotopic by this definition, then the entire boundary of F consists of (u_1, v_1), (u_2, v_2), T_u, and T_v. (The notion of homotopy in this paper is intimately related to the notion of homotopy in topology ([11], [18]) in the following sense. If we "shrink" the planar part to a point and take that point as the base of the homotopy, then two nonplanar edges are homotopic in our sense if and only if they are homotopic in the topology sense.) The relation $\equiv_h$ is defined to be the reflexive, symmetric, and transitive closure of the homotopy relation; $\equiv_h$ is an equivalence relation on E_N. Each equivalence class is a *homotopy class.* In Figure 3, the three homotopy classes are $\{(u_1,v_1),(u_2,v_2),(u_3,v_3)\}$, $\{(u_4,v_4),(u_5,v_5),(u_6,v_6)\}$, and $\{(u_7,v_7),(u_8,v_8),(u_9,v_9)\}$.

We translate the fact that $\equiv_h$ is transitive into the language of traces in the following Lemma.

Lemma 1. If C is a homotopy class, then the elements of C can be ordered

$$(u_1,v_1),\ldots,(u_k,v_k)$$

and two traces T_1 and T_2 defined such that

(1) for $1 \le i \le k-1$, (u_i,v_i) is homotopic to (u_{i+1},v_{i+1}) with corresponding traces T_{u_i} and T_{v_i};

(2) T_1 is the concatenation of

$$T_{u_1},T_{u_2},\ldots,T_{u_{k-1}}$$

and T_2 is the concatenation of

$$T_{v_1},T_{v_2},\ldots,T_{v_{k-1}}.$$

Note that u_i, u_{i+1} need not be distinct, and v_i, v_{i+1} need not be distinct. However, $u_i = u_{i+1}$, and $v_i = v_{i+1}$ is not allowed, as G may not have multiple edges.

As the boundary of the planar graph is traversed in a consistent direction, each of the two traces (defined by Lemma 1) of a homotopy class is encountered exactly once. If each trace is traversed in the same order, the homotopy class is *nonorientable.* If the traces are traversed in opposite orders, the homotopy class is *orientable.* In Figure 3, $\{(u_1,v_1),(u_2,v_2),(u_3,v_3)\}$ and $\{(u_7,v_7),(u_8,v_8),(u_9,v_9)\}$ are orientable homotopy classes, while $\{(u_4,v_4),(u_5,v_5),(u_6,v_6)\}$ is a nonorientable homotopy class.

Lemma 2. If G is 2-cell embedded in an orientable surface, then every homotopy class of any planar-nonplanar decomposition of G is orientable.

Proof: Say that a closed curve on an orientable surface is *collapsible* if it does not cross itself (but can touch itself) and is homotopic to a point. We use the fact that a collapsible closed curve has a consistently defined interior and exterior. Let G be embedded in an orientable surface S, and fix any planar-nonplanar decomposition of G.

To obtain a contradiction, suppose C is a nonorientable homotopy class with traces T_1 and T_2. If either trace consists of a single vertex, then C may also be taken to be an orientable homotopy class. Otherwise, by Lemma 1, there exists a subtrace $u_i \to u_{i+1}, u_i \ne u_{i+1}$ of T_1, and a subtrace

$v_j \to v_{j+1}, v_j \neq v_{j+1}$ of T_2, such that $(u_i, v_j), (u_{i+1}, v_{j+1}) \in C$, and such that either $(u_i, v_{j+1}) \in C$ is on a face with (u_i, v_j) and on a face with (u_{i+1}, v_{j+1}), or (u_i, v_j) and (u_{i+1}, v_{j+1}) are on the same face (see Figures 5 and 6). In the latter case, (u_i, v_{j+1}) may be added to C without changing the remainder of the planar-nonplanar decomposition; therefore, we may assume that the edge is present.

The cycle

$$\gamma_1 = (u_i, u_{i+1}), (u_{i+1}, v_{j+1}), (v_{j+1}, v_j), (v_j, u_i)$$

is a simple closed curve on the surface S. Since (u_i, u_{i+1}, v_{j+1}) is a face, and (u_i, v_j, v_{j+1}) is a face, γ_1 is collapsible. Let γ_2 be the boundary of G_P. It too is collapsible. If we traverse γ_2 on S clockwise, we visit the vertices $u_1, u_{i+1}, v_j, v_{j+1}$ in precisely this cyclic order. Since the interiors of γ_1 and γ_2 are disjoint, traversing γ_1 on S counterclockwise visits these vertices in the same order. But this is a contradiction—(u_{i+1}, v_j) is not an edge that belongs to γ_1. □

We now bound the number of homotopy classes as a function of the genus of the surface.

Lemma 3. If $G = (V, E)$ is 2-cell embedded in an orientable surface of genus $g, g \geq 1$, then any planar-nonplanar decomposition of G has at most $6g - 3$ homotopy classes.

Proof: Let (R, G_P, E_N) be a decomposition of G. Then $E_N \neq \emptyset$ (otherwise, G is planar). Draw a circle around the planar embedding of G_P (see Figure 7). The circle intersects each edge of E_N in exactly two points. Place a new vertex at each such intersection, and eliminate all of G_P. A new graph H is the result, with a 2-cell embedding having the same number of homotopy classes as G. The planar part of H is the circle; the nonplanar part consists of the nonplanar edges of G, as truncated. If two edges of H came from two homotopic, nonplanar edges of G, then these edges are incident to a face F of H bounded by a 4-cycle and are homotopic in H. We can contract H along the two planar edges incident to F, thus eliminating F, and obtaining a graph H' with a 2-cell embedding having the same number of homotopy classes. Continuing in this way, we reach a graph H^* which has the same number of homotopy classes as G and where each class contains a single edge. It is possible that H^* has only two vertices. In that case, G has only one homotopy class, and the result is trivially true. Thus we may assume that H^* has at least 4 vertices.

For the embedding of $H^* = (V^*, E^*)$, let $v = |V^*|$, $e = |E^*|$, $h =\#$ homotopy classes, and $f =\#$ faces. By Euler's identity for surfaces of genus g,

$$v - e + f = 2 - 2g.$$

H^* is regular trivalent, so $3v = 2e$. Since there is only one nonplanar edge in each homotopy class, $v = 2h$. The interior face of H^* has v incident edges. The remaining $f - 1$ faces have at least 6 incident edges each, since planar and nonplanar edges alternate and if a face had only 4 incident edges, the two nonplanar edges would be homotopic. By counting edges incident to faces, we have

$$6(f-1) + v \leq 2e.$$

From this inequality and the preceding equations, we obtain

$$6g - 3 \geq \frac{v}{2} = h. \qquad \square$$

We immediately obtain our Decomposition Theorem for orientable genus:

Theorem 4. Every graph of orientable genus g has a planar-nonplanar decomposition with $O(g)$ homotopy classes. In fact, any decomposition has $O(g)$ homotopy classes.

Lemma 3 has an analog for nonorientable surfaces:

Lemma 5. If $G = (V, E)$ is 2-cell embedded in a nonorientable surface of genus $g, g \geq 1$, then any planar-nonplanar decomposition of G has at most $\max(1, 3g - 3)$ homotopy classes.

Proof: Same as Lemma 3 except Euler's identity for the nonorientable case is

$$v - e + f = 2 - g. \qquad \square$$

This gives us the Decomposition Theorem for nonorientable genus:

Theorem 6. Every graph of nonorientable genus g has a planar-nonplanar decomposition with $O(g)$ homotopy classes. In fact, any decomposition has $O(g)$ homotopy classes.

3. Algorithm for Orientable Surfaces

Our algorithm (O-LAYOUT) for the layout of a connected graph embedded in an orientable surface has two stages. The first stage (DECOMPOSE) takes as input a graph G with a 2-cell embedding in a surface of genus g and produces as output a planar-nonplanar decomposition compatible with the surface embedding. The second stage (O-PAGES) takes as input a planar-nonplanar decomposition of G having only orientable homotopy classes and produces as output a book embedding of G in $O(g)$ pages. The first stage is the same for orientable and nonorientable surfaces. The second stage depends on the specific properties of the output of the first stage.

The first step of DECOMPOSE (see ALGORITHM DECOMPOSE) triangulates the surface embedding of G, obtaining a graph $G_T = (V_T, E_T)$. G_T is obtained by adding new vertices and edges to the embedding of G; we must be careful not to create any loops or multiple edges. All faces of the embedding of G_T are triangles. Consider any non-triangular face F of G; there are $O(n+g)$ such faces. Add a vertex v_F in the face F; this adds $O(n+g)$ vertices. Add an edge from v_F to each vertex on the boundary of F; this adds $O(n+g)$ edges. If some vertex v occurs multiple times on the boundary of F, this creates multiple edges. If this occurs, consider one such edge (v, v_F) and the two triangles it is in; let the triangles be (v_1, v, v_F) and (v_2, v, v_F). Then $v_1 \neq v_2$, for otherwise (v, v_1) and (v, v_2) would be multiple edges in G. Subdivide (v, v_F) by adding a new vertex v_s, and by replacing (v, v_F) by the path v, v_s, v_F. Triangulate the two faces by adding the edges (v_s, v_1) and (v_s, v_2). $O(n+g)$ edges are added. There are $O(n+g)$ edges in G_T.

One triangle is chosen as the initial planar part, and faces are added to the planar part incrementally, as possible. At any stage, $G_P = (V_P, E_P)$ represents the planar part constructed so far. E_N always represents $E_T - E_P$, those edges outside the current planar part. E_N can be partitioned into two parts. The first part consists of the essentially nonplanar edges, which have both endpoints in V_P (necessarily on the boundary of G_P) and which can never become edges of G_P. The second part consists of those edges that have at most one endpoint in V_P and that still have the potential to become edges in G_P.

Fix a clockwise orientation for the current boundary of G_P. If $v_i \rightarrow v_j \rightarrow v_k$ is a trace of G_P with no edge of E_N incident to v_j, then $(v_i, v_k) \in E_T$ is called a *safe edge* (see Figure 8). If $v_i \rightarrow v_j$ is a trace of G_P, $v_k \in V - V_P$, and (v_i, v_j, v_k) is a face of the embedding, then v_k is a *safe vertex* with respect to $v_i \rightarrow v_j$ (see Figure 9). Clearly, in a planar-nonplanar decomposition of G, there are no safe edges (adding a safe edge preserves planarity, contradicting the maximality of G_P), and there are no safe vertices ($V - V_P = \emptyset$). Normally, DECOMPOSE chooses safe vertices (steps (7-8)) and safe edges (steps (14-16)) to add to G_P. Whenever this is not possible, the algorithm choose some (unsafe) vertex adjacent to a boundary vertex of G_P. The algorithm "ages" the edges, vertices, and blocks (biconnected components) of G_P; those added later are newer, those added earlier are older. This aging is used explicitly in step (9) and in the discussion following. Step (9) chooses the newest vertex w' on the boundary of G_P that is adjacent to at least one vertex in $V - V_P$. Among

$G_T = (V, E_T) \leftarrow$ a surface triangulation of G
$G_P = (V_P, E_P) \leftarrow$ some face of G_T
while $V_P \neq V$ **do**
    if $\exists$ safe vertex v_k (with respect to $v_i \rightarrow v_j$)
        then (* add a safe vertex *)
        $V_P \leftarrow V_P \cup \{v_k\}$
        $E_P \leftarrow E_P \cup \{(v_i, v_k), (v_j, v_k)\}$
        else (* start a new block *)
        $w' \leftarrow$ newest vertex in V_P incident to a vertex in $V - V_P$
        $w \leftarrow$ vertex in $V - V_P$ adjacent to w' (* see text *)
        $V_P \leftarrow V_P \cup \{w\}$
        $E_P \leftarrow E_P \cup \{(w, w')\}$
    while $\exists$ safe edge $(v_i, v_k) \in E_N$ **do**
        $E_P \leftarrow E_P \cup \{(v_i, v_k)\}$ (* add a safe edge *)
        enddo
    enddo

ALGORITHM DECOMPOSE

the vertices adjacent to $w' \in V - V_P$, step (10) selects one in a manner described below.

Each time steps (9-12) are executed, it is not possible to extend any current block of G_P. Hence, a new block of G_P is started. The use of the newest w' possible in step (9) creates the blocks of G_P in a depth-first order; the new block (and all its descendants) will be completed before the current block is again examined. Consider an edge (x, y) on the boundary of the block completed just before step (9) is executed; let (x, y, z) be a face of G exterior to the planar embedding of G_P (there may be two choices for z, if the block consists only of the edge (x, y)). Then z is already on the boundary of G_P (since it is unsafe), and (x, z) and (y, z) are essentially nonplanar. At the completion of DECOMPOSE, these two edges will be homotopic; the face (x, y, z) will always be a witness to the homotopy relationship. At any step of DECOMPOSE, all those edges which are essentially nonplanar are already partitioned into homotopy classes (though it is conceivable that two classes could later merge).

Now we can describe the selection of w in step (10) (see Figure 10). Let (x, w') be the newest edge on the boundary of G_P that is incident to w'. Then there must be a triangle (x, w', z) exterior to G_P. Since z is unsafe, z is on the boundary of G_P, and (x, z) and (w', z) are essentially nonplanar. Examine the edges incident to w'—start with the edge (x, w') and sweep rotationally about w' in the direction of z (in Figure 10, this direction is clockwise). Let (w', w) be the first edge encountered such that $w \in V - V_P$, and let (w', y) be the last essentially nonplanar edge encountered before (w', w). Let y' be the next vertex adjacent to w' after encountering w; $(w', y') \in E_N$, for otherwise, w would be a safe vertex. (w', y, w) is a triangle. Once (w', w) is added to G_P (steps (11-12)), (w, y) becomes essentially nonplanar and will be homotopic to (w', y). Also, w is newer than y; thus the homotopy class will be extended by edges incident to y and never by edges incident to w (i.e., w, not y, will have the role of w' in future executions of steps (9-12)). If $y' \in V_P$ (i.e., (w', y') is already essentially nonplanar), then (w, y') also becomes essentially nonplanar and homotopic to (w', y'). w is newer than y'; thus the homotopy class of (w', y') will always be extended by edges incident to y', not to w.

The only time that a new homotopy class is created is when the addition of a vertex (steps (6-7) or steps (9-12)) causes one or more edges in E_N to become essentially nonplanar. (However, the addition of a new vertex that causes one or more edges in E_N to become nonplanar does not necessarily introduce a new homotopy class.) Such an edge (or edges) must be incident to v_k or w and to another vertex (or vertices) on the boundary of G_P. Note that several edges incident to v_k or w may become essentially nonplanar but not be homotopic.

We consider the case when steps (6-7) create a new homotopy class; the case when steps (9-12) create a new homotopy class is similar. Suppose that (v_k, z) is a new essentially nonplanar edge and that z is not in the current block. Then no other edge introduced by the addition of v_k is homotopic to (v_k, z); otherwise, v_k is a safe vertex for the block z is in and would have been added earlier to that block. Since z is older than v_k, this homotopy class will consist only of edges incident to z; in this case, z is said to be a *degenerate* trace for the homotopy class.

Suppose a new homotopy class receives more than one edge upon addition of v_k. Then the endpoints of these new edges must be in the same block as v_k. In this case, when a homotopy class has an edge with both endpoints in the same block, we call the homotopy class *nondegenerate,* even if all its edges are incident to the same point. Let $u_1 \to u_2 \to \ldots \to u_s$ and $v_1 \to v_2 \to \ldots \to v_t$ be the traces of a single homotopy class and part of the same block just before step (9) is executed. The only vertices of these traces that can be adjacent to vertices in $V - V_P$ are the endpoints. If the addition of w in step (10) is to extend this homotopy class, either $(u_1, w), (v_1, w) \in E$, or $(u_s, w), (v_t, w) \in E$. Without loss of generality, assume the second possibility. Also without loss of generality, assume v_t is newer than u_s. Then in step (10) we have $w' = v_t$, and the addition of w extends the trace to $v_1 \to v_2 \to \ldots \to v_t \to v_{t+1}$, where $w = v_{t+1}$. At any future time when the homotopy class is about to be extended at that end, the algorithm invariably chooses to extend the trace

$$v_1 \to v_2 \to \ldots \to v_t \to v_{t+1} \to \ldots$$

(because $v_{t+1}, \ldots$ are all newer than u_s). Similarly, the other end of the homotopy class may be extended beyond one of u_1 or v_1, but not both. We have

Lemma 7. Let $T_1 = u_1 \to u_2 \to \ldots \to u_s$ and $T_2 = v_1 \to v_2 \to \ldots \to v_t$ be the traces of a nondegenerate homotopy class at the completion of DECOMPOSE. Let x be the older of u_1 and v_1; let y be the older of u_s and v_t. Then all the trace vertices are in the same block except perhaps for vertices adjacent (by edges in the class) to x and y.

Call the edges of the homotopy class from x and y to vertices in a different block *exceptional edges.*

The second stage (O-PAGES) lays out the graph in a book of at most $18g - 5$ pages. By the algorithm of Yannakakis [22], a planar graph having a cycle as boundary can be laid out in 4 pages with the vertices on the boundary in cycle order. By an algorithm of Heath [13], all the 4 page layouts of the blocks of a planar graph can be combined in linear time into a 4 page layout for the entire graph, maintaining the cyclic order of vertices on the boundary of each block. Thus, G_P has such a 4 page layout. This establishes the vertex order and assigns pages to all edges in E_P. For each (orientable) homotopy class in E_P, allocate three pages. (In the degenerate case, one page suffices for the homotopy class). One page suffices for the edges between vertices in the same block. The exceptional edges are assigned to the two remaining pages. The resulting layout requires at most $4 + 3(6g - 3) = 18g - 5$ pages.

Theorem 8. The algorithm O-LAYOUT lays out any graph G that is 2-cell embedded in an orientable surface of genus $g, g \geq 1$, in $18g - 5$ pages. If G has n vertices, the running time of O-LAYOUT is $O(n + g)$, which is optimal.

4. Algorithm for Nonorientable Surfaces

Our algorithm (N-LAYOUT) for a graph embedded in a nonorientable surface also has two stages. The first stage (DECOMPOSE) is identical to the one for an orientable surface. Lemma 7 still holds. By the Lemma, it suffices (in the worst case) to consider only decompositions in which G_P has a cycle as boundary and has at least one nonorientable homotopy class. The difficulty introduced by a nonorientable homotopy class is that, to obtain a bounded number of pages, the direction of one of its traces must be the reverse of the direction in the orientable case.

In the development of the algorithm, we will not compute an exact upper bound on the number of pages used, as we did for O-LAYOUT. We content ourselves with showing that $O(1)$ pages suffice per homotopy class, plus $O(1)$ pages for G_P. In fact, we expect that the constants in our $O(g)$ can be improved somewhat.

Focus on the case in which G_P has a single block (i.e., it is bounded by a cycle), and $O(g)$ homotopy classes. Some of the homotopy classes will be non-orientable. Let C^N be one such non-orientable class. By Lemma 1, C^N has two traces T_1 and T_2 on the boundary of G_P; each trace is a simple, directed path (see Figure 11). To realize the edges of C^N in a single page, one trace should be in the reverse order of the second trace (with respect to the bounding cycle of G_P) in the layout. We choose one of T_1 and T_2 to be the one to be reversed. If one trace is *singular* (consists of a single vertex), then the homotopy class is degenerate, and we need not reverse either trace (i.e., the homotopy class is both nonorientable **and** orientable). Otherwise, arbitrarily choose either trace to be reversed. Call the chosen trace *reversed;* call the other trace *non-reversed.* Our intention is to lay out all non-reversed traces in a single consistent order and all reversed traces in the opposite order.

The boundary of G_P contains $O(g)$ traces induced by the homotopy classes of G. For each orientable homotopy class, both traces are non-reversed. For each non-orientable homotopy class, one trace is reversed and one is non-reversed. If adjacent traces are non-reversed, then we merge them into one non-reversed trace. Similarly, if two adjacent traces are reversed, then we merge them into one reversed trace. After all such merges are performed, the boundary of G_P is covered by non-singular traces that alternate reversed and non-reversed. Call the set of reversed traces T^R, and the set of non-reversed traces T^N. The number of traces is $O(g)$, as is the number of common vertices. (Two adjacent traces have one *common* vertex; all non-common vertices in a trace are *interior* to the trace.) We allot one page per common vertex for all edges incident to it, and do not consider the common vertex to be part of either trace.

The task remains to obtain the reversal of each trace of T^R. We need two techniques to accomplish the task. First, we describe a technique for the case where G_P has no chords of a certain type. (A *chord* in G_P is an interior edge of the planar embedding that is incident to two boundary vertices.) Then, we discuss the general case.

Suppose G_P has no chords from a non-reversed trace to a reversed trace (chords within T^R or within T^N are allowed). In Figure 12, traces T_1, T_2, and T_3 are representative reversed traces with common vertices u_1, u_i, v_1, v_j, and w_1, w_k, respectively. For each reversed trace, add an edge between its two common vertices in the exterior face of G_P; call such an added edge a *bypassing* edge. In Figure 13, bypassing edges (u_1, u_i), (v_1, v_j), and (w_1, w_k) have been added in the exterior face of G_P. By Heath [12], we may embed the resulting graph G'_P in seven pages. Furthermore, each trace in T^R is now at the second level of the planar graph and is hence in reverse order from the traces in T^N at the first level. In fact, each trace is on a **single cycle** at the second level. This cycle is laid out in cycle order by [12]; hence, in this case, all traces in T^R have been successfully reversed.

Now, consider the general case where there are chords from reversed to non-reversed traces. (In Figure 14, T_1 is non-reversed, T_2 is reversed, and the bypassing edge (v_1, v_j) has already been added at T_2; there are chords between T_1 and T_2). Let all chords of G_P having both endpoints in a single trace be temporarily removed. Since each trace is laid out in order, when these chords are restored, one page per trace will suffice for these edges. Now, all chords have endpoints in different traces. Consider a chord path (u, v, w), where u, v, and w are interior vertices in three different traces. Think of the following transformation involving only chords in such chord paths and the boundary of G_P. Shrink the interior of each trace to a single vertex; $O(g)$ vertices result. A chord between two vertices remains if originally there was a chord between the two traces. Any chord path will survive this transformation. Suppose $(\overline{u}, \overline{v}, \overline{w})$ is a transformed chord path. Then all the original chords that were compressed to obtain $(\overline{u}, \overline{v})$ share a single endpoint v in the middle trace; hence one page suffices for all these chords. Similarly, all the chords compressed to obtain $(\overline{v}, \overline{w})$ share the same endpoint v, and the same page suffices for these chords. If all these chords are eliminated, no chord paths remain in G_P. Because an outerplanar graph of k vertices can have at most $k-3$ chords, $O(g)$ pages suffice for all the eliminated chords.

The G_P that remains has no chord paths and no chords within a single trace. Without loss of generality, we may assume that the interior of G_P is triangulated (we may add a linear number of both vertices and edges to accomplish this).

Now consider two traces $T_1 \in T^N$, and $T_2 \in T^R$ with chords between them (Figure 14). Let (u', v') be the leftmost chord between T_1 and T_2, and let (u'', v'') be the rightmost chord between T_1 and T_2. Let u''' be the vertex following u'' in T_1 (u''' is in T_1, though it may be a common vertex). We "merge" the two subtraces $u' \rightarrow u''$ and $v' \rightarrow v''$ using the following Lemma.

Lemma 9. In the above described circumstances, there exists a path in G_P from u' to u''' such that the path contains

(1) all vertices in the subtrace of T_1 $u' \rightarrow u''$ in that order;

(2) all vertices in the subtrace of T_2 $v' \rightarrow v''$ in that order;

(3) no other vertices on the boundary of G_P.

Proof: We view any chords between T_1 and T_2 as obstacles to visibility. If there were no chords, the path would just be

$$u' \rightarrow u'', \ldots, v' \rightarrow v'', \ldots, u'''$$

where the dots represent subpaths through the interior of G_P.

Due to the chords, the path must shuffle back and forth between T_1 and T_2 as visibility allows. As an example, consider Figure 14. The first edge in the path cannot be along T_1 because there is a chord that obstructs visibility to v'. Therefore the path goes first from u' to v' and then follows T_2 until a vertex is reached from which the unfinished portion of T_1 is visible. At that point, the path returns to T_1 via a subpath through the interior of G_P (in some cases, this subpath will be a single chord) and follows T_1 for a while. This shuffling of the path back and forth between the two traces ultimately leads to the required path. □

If we continue the path around the boundary of G_P, using bypassing edges where available, we obtain a cycle that has $u' \rightarrow u''$ in nonreversed order and $v' \rightarrow v''$ in reversed order as needed. As we go around the boundary of G_P, we perform the same merging of subtraces whenever a non-reversed and a reversed trace have chords between them. Let C be the cycle obtained. C contains every vertex on the boundary of G_P, and every trace occurs in trace order in C (reversed traces in opposite order to non-reversed traces). C goes through the interior of G_P and defines two subgraphs of G_P. Let G_P^1 be C together with all of G_P interior to C, and let G_P^2 be C together with all of G_P exterior to C. Take C to be the boundary of both G_P^1 and G_P^2, and lay out each subgraph in seven pages according to the algorithm in [12]. The two layouts can be combined into a single layout for G_P. This combining is possible because [12] lays out the common cycle C of G_P^1 and G_P^2 in a consistent order. The combined layout may require 14 pages.

There is one more difficulty. It is possible that in the resulting layout the two traces of a single homotopy class may be intermingled by the "merging" operation. In the discussion above, this would occur if T_1 and T_2 were traces for the **same** homotopy class. We leave it to the reader to verify that two pages will suffice for such a homotopy class; one page will be for forward edges, the second page for backward edges.

This completes the algorithm for laying out a graph embedded in a nonorientable surface of genus g. The number of pages used was, at all points, either $O(g)$, or $O(1)$. Thus, we have the following Theorem.

Theorem 10. The algorithm N-LAYOUT lays out any graph G that is 2-cell embedded in a nonorientable surface of genus $g, g \geq 1$, in $O(g)$ pages. If G has n vertices, the running time of N-LAYOUT is $O(n+g)$, which is optimal.

5. Time Complexity

In the previous two sections, we have described three algorithms: DECOMPOSE, O-PAGES, and N-PAGES. We now show that each algorithm can be implemented in $O(n+g)$ time. After the first step of DECOMPOSE (the triangulation of the embedding of G), the graph for all remaining steps has size $\Theta(n+g)$. Hence, all three algorithms are linear in the size of the input graph for all graphs having a triangulated embedding. Our main assumption is that we have a data structure (essentially the rotation of the embedding) that allows us to traverse the boundary of any face in constant time per edge (called subsequently *constant time traversal*).

The triangulation step of DECOMPOSE was described earlier, where it was shown to produce a triangulation of size $O(n+g)$. Because of constant time traversal, adding edges can be done in constant time per original edge. Triangulation is accomplished in $O(n+g)$ time. The remaining steps of DECOMPOSE accomplish three operations: the adding of a safe vertex, the adding of a safe edge, and the creation of a new block. The set of potential safe vertices can be maintained in a linked list. A vertex is added to the list when an edge is added to G_P, and the vertex completes a triangular face with that edge. The list operations take constant time per vertex. Safe edges are found only after the addition of a safe vertex; each safe edge can be found in constant time due to constant time traversal. For creating new blocks, we keep the vertices on the boundary of G_P on a stack in order of age, and maintain for each edge the status of whether it is in G_P, essentially nonplanar, or still free. The search for edges of the form (w', w) is accomplished in constant time per edge. DECOMPOSE takes $O(n+g)$ time.

O-PAGES is the easiest of the three algorithms to analyze. The layout of the planar part is accomplished in $O(n)$ time [22]; this yields the ordering of the vertices on the spine and the page assignment for the planar edges. Three pages (at most) are assigned to the $O(g)$ homotopy classes. Each nonplanar edge is assigned a page based on its homotopy class in constant time per edge. Thus, O-PAGES (and hence O-LAYOUT) takes $O(n+g)$ time.

N-PAGES has a number of steps to analyze. It may break the planar part into a number of subgraphs, though no vertex is in more than two such subgraphs; thus, the algorithm of [12] lays out the planar part in $O(n)$ time and in a constant number of pages (by the merging of the layouts for the subgraphs). N-PAGES finds all traces associated with the $O(g)$ homotopy classes, selects one reversed trace for each nonorientable homotopy class, and merges adjacent reversed traces and adjacent nonreversed traces; this is accomplished in $O(n+g)$ time. Adding one edge to G_P for each reversed trace requires $O(g)$ time. Assigning edges adjacent to the $O(g)$ common vertices to one of $O(g)$ pages requires constant time per edge. Finding and classifying the chords of G_P requires $O(n)$ time. Assigning the chords within a trace to a page requires constant time per chord. Finding chord paths among three different traces and assigning these chords to pages takes constant time per chord. The merging of two subtraces is accomplished in time linear in the size of the subtraces. The net time complexity for N-PAGES (and hence N-LAYOUT) is $O(n+g)$.

6. Lower Bounds

A lower bound for genus g can be gotten from the family of complete graphs on n vertices, K_n. From Harary [9], the genus of K_n is

$$\gamma(K_n) = \left\lceil \frac{(n-3)(n-4)}{12} \right\rceil.$$

The pagenumber of K_n is $\Theta(n)$ [1]. Therefore, the number of pages required for the class of genus g graphs is $\Omega(\sqrt{g})$. We [14] conjectured that the pagenumber of the class is in fact $\Theta(\sqrt{g})$. This conjecture was based on several observations. First, for any graph G with pagenumber $page(G)$, Bernhart and Kainen [1] show the following for the chromatic number of G

$$\chi(G) \leq 2 \cdot page(G) + 2.$$

Therefore, $\chi(G) = O(page(G))$. On the other hand, if G is an arbitrary graph, we have the following inequalities

$$\chi(G) \leq \left\lceil \frac{7 + \sqrt{1 + 48\gamma(G)}}{2} \right\rceil$$
$$\chi(G) \leq \left\lceil \frac{7 + \sqrt{1 + 24\tilde{\gamma}(G)}}{2} \right\rceil.$$

by Heawood's formula [9]. Therefore, $\chi(G) = O\left(\sqrt{\gamma(G)}\right)$, and $\chi(G) = O\left(\sqrt{\tilde{\gamma}(G)}\right)$. Moreover, taking G to be K_n

$$\chi(K_n) = \Theta(page(K_n))$$
$$\chi(K_n) = \Theta\left(\sqrt{\gamma(K_n)}\right)$$
$$\chi(K_n) = \Theta\left(\sqrt{\tilde{\gamma}(K_n)}\right).$$

Using a nonconstructive argument, Malitz [17] settled our conjecture in the affirmative for graphs embedded in **orientable** surfaces.

7. Conclusions and Open Problems

We have shown algorithmically that a graph embedded in a surface (orientable or nonorientable) of genus $g, g \geq 1$, can be laid out in $O(g)$ pages. Our algorithms can be shown to run in optimal time $O(n+g)$. For graphs already embedded in a surface, our algorithms are an efficient means to obtain layouts with a guarantee on the number of pages used. The best algorithm known for embedding a graph G in an orientable surface of its genus $\gamma(G)$ requires time $O(n^{O(\gamma(G))})$ (Filotti, Miller and Reif [5]). We propose an approach to approximating $\gamma(G)$ that may be accomplished in better time. Suppose a planar-nonplanar decomposition for a graph can be constructed with $g = \Theta(\gamma(G))$ orientable homotopy classes. Then good layouts can be produced by our algorithms for arbitrary graphs (as used in the Diogenes methodology). In addition, such a planar-nonplanar decomposition describes an embedding of the graph in a surface of genus $\Theta(\gamma(G))$ (any rotation consistent with the planar part and with each homotopy class gives such an embedding). Any context that requires that a graph be embedded in a surface first can utilize this embedding (e.g., Gilbert, Hutchinson and

Tarjan [7]). This would be especially valuable if the g above is a good approximation (i.e., within a small constant factor or better) to the actual genus of the graph.

A lower bound on the number of pages required for the class of graphs embedded in an orientable or nonorientable surface of genus g is $\Omega(\sqrt{g})$. Using a nonconstructive argument and the planar-nonplanar decomposition in this paper, Malitz [17] has shown the matching upper bound of $O(\sqrt{g})$ pages for the class of graphs embedded in an orientable surface of genus g. We close with three conjectures.

Conjecture 1. $\Theta(\sqrt{g})$ pages are necessary and sufficient for the class of graphs embedded in a nonorientable surface of genus g.

Conjecture 2. There is a polynomial-time algorithm to lay out in $O(\sqrt{g})$ pages any graph embedded in an orientable surface of genus g (that is, there is a constructive proof of [17]).

Conjecture 3. The pagenumber of genus 1 (toroidal) graphs is 7 (matching the chromatic number as in the planar case); our algorithm gives a 13 page layout.

We note that our results depend only on the *number* of homotopy classes, not on their relative *structure*. The topology of genus g surfaces places restrictions on the order in which traces can occur. Further study of these restrictions may provide the information to answer our conjectures.

Acknowledgements

We wish to thank Tom Leighton and Arny Rosenberg for suggestions and encouragement. The second author wishes to thank Albert Meyer for providing the opportunity to visit MIT, thereby allowing this research to be pursued. We wish to thank Bill Reddy for initial topological suggestions and perspectives. We also thank the referees for a careful reading, for many helpful comments, for sharpening the proof of Lemma 2, and for improving the discussion of the algorithm N-LAYOUT.

References

[1] Bernhart, F., and Kainen, B. The book thickness of a graph. *Journal of Combinatorial Theory B* **27** (1979), 320-331.

[2] Buss, J. F., and Shor, P. W. On the pagenumber of planar graphs. *16th ACM Symposium on Theory of Computing* (1984), 98-100.

[3] Chung, F. R. K., Leighton, F. T., and Rosenberg, A. L. Embedding graphs in books: a layout problem with applications to VLSI design. *SIAM Journal on Algebraic and Discrete Methods* **8** (1987), 33-58.

[4] Edmonds, J. R. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society* **7** (1960), 646.

[5] Filotti, I. S., Miller, G. L., and Reif, J. On determining the genus of a graph in $O(v^{O(g)})$ steps. *11th ACM Symposium on Theory of Computing* (1979), 27-37.

[6] Galil, Z., Kannan, R., and Szemeredi, E. On nontrivial separators for k-page graphs and simulations by nondeterministic one-tape Turing machines. *18th ACM Symposium on Theory of Computing* (1986), 39-49.

[7] Gilbert, J. R., Hutchinson, J. P., and Tarjan, R. E. A separator theorem for graphs of bounded genus. *Journal of Algorithms* **5** (1984), 391-407.

[8] Gross, J. L., and Tucker, T. W. *Topological Graph Theory.* John Wiley & Sons, New York (1987).

[9] Harary, F. *Graph Theory.* Addison-Wesley Publishing Co., Reading, MA (1969).

[10] Heffter, L. Ueber das Problem der Nachbargebiete. *Mathematische Annalen* **38** (1891), 477-508.

[11] Henle, M. *A Combinatorial Introduction to Topology.* W. H. Freeman and Company, San Francisco (1979).

[12] Heath, L. S. Embedding planar graphs in seven pages. *25th IEEE Symposium on Foundations of Computer Science* (1984), 74-83.

[13] Heath, L. S. *Algorithms for Embedding Graphs in Books.* Dissertation, Department of Computer Science, University of North Carolina at Chapel Hill (1985).

[14] Heath, L. S., and Istrail, S. The pagenumber of genus g graphs is $O(g)$. *19th ACM Symposium on Theory of Computing* (1987), 388-397.

[15] Istrail, S. An algorithm for embedding planar graphs in six pages. Department of Mathematics, Wesleyan University, unpublished manuscript (1985).

[16] Kannan, R. Unraveling k-page graphs. *Information and Control* **66** (1985), 1-5.

[17] Malitz, S. M. Genus g graphs have pagenumber $O(\sqrt{g})$. *29th IEEE Symposium on Foundations of Computer Science* (1988), 458-468.

[18] Massey, W. S. *Algebraic Topology: An Introduction.* Harcourt, Brace & World, New York (1967).

[19] Paul, W. J., Pippenger, N., Szemeredi, E., and Trotter, W. T. On determinism versus nondeterminism and related problems. *24th IEEE Symposium on Foundations of Computer Science* (1983), 429-438.

[20] Rosenberg, A. L. The Diogenes approach to testable fault-tolerant arrays of processors. *IEEE Transactions on Computing* **C-32** (1983), 902-910.

[21] White, A. T. *Graphs, Groups and Surfaces.* Elsevier Science Publishers, Amsterdam (1984).

[22] Yannakakis, M. Four pages are necessary and sufficient for planar graphs. *18th ACM Symposium on Theory of Computing* (1986), 104-108.

[23] Youngs, J. W. T. Minimal imbeddings and the genus of a graph. *Journal of Mathematics and Mechanics* **12** (1963), 303-315.

List of Figures

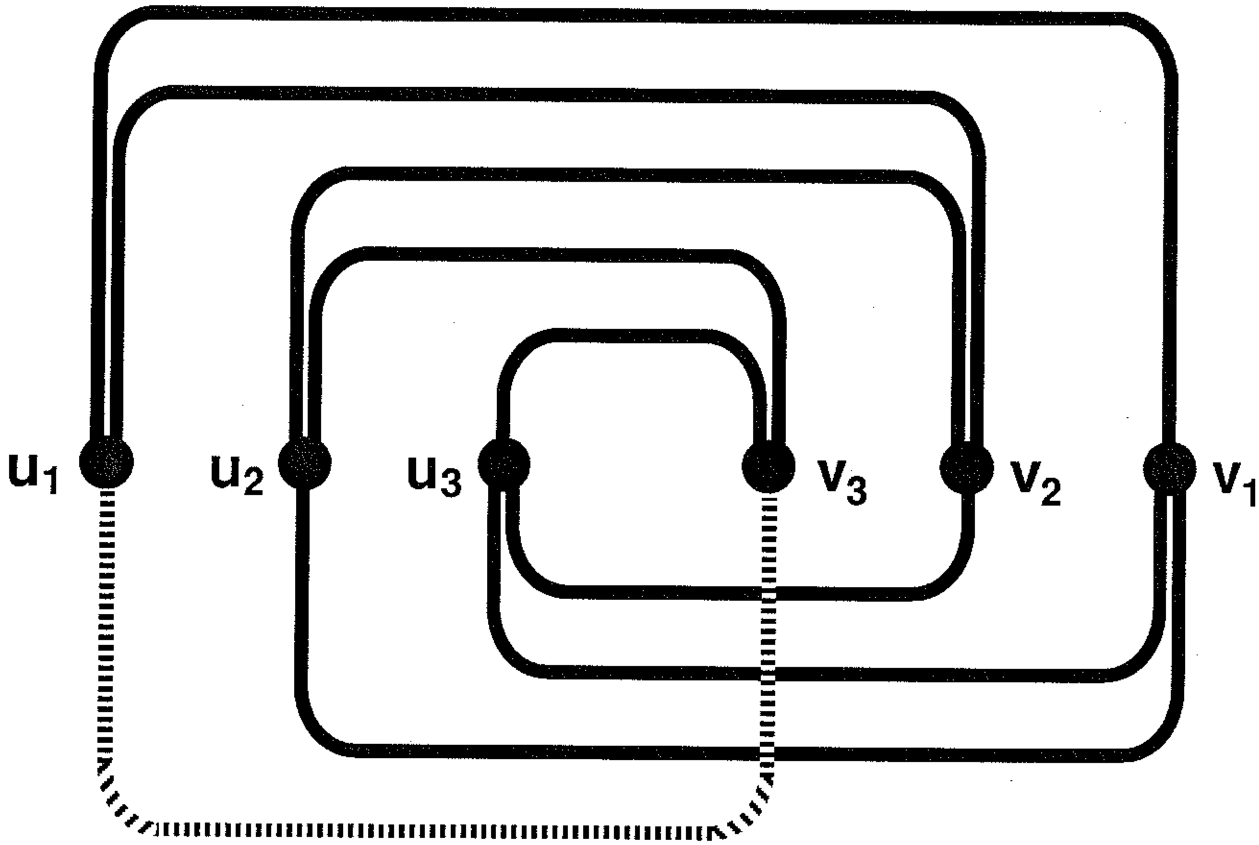

FIGURE 1

EXAMPLE OF BOOK EMBEDDING

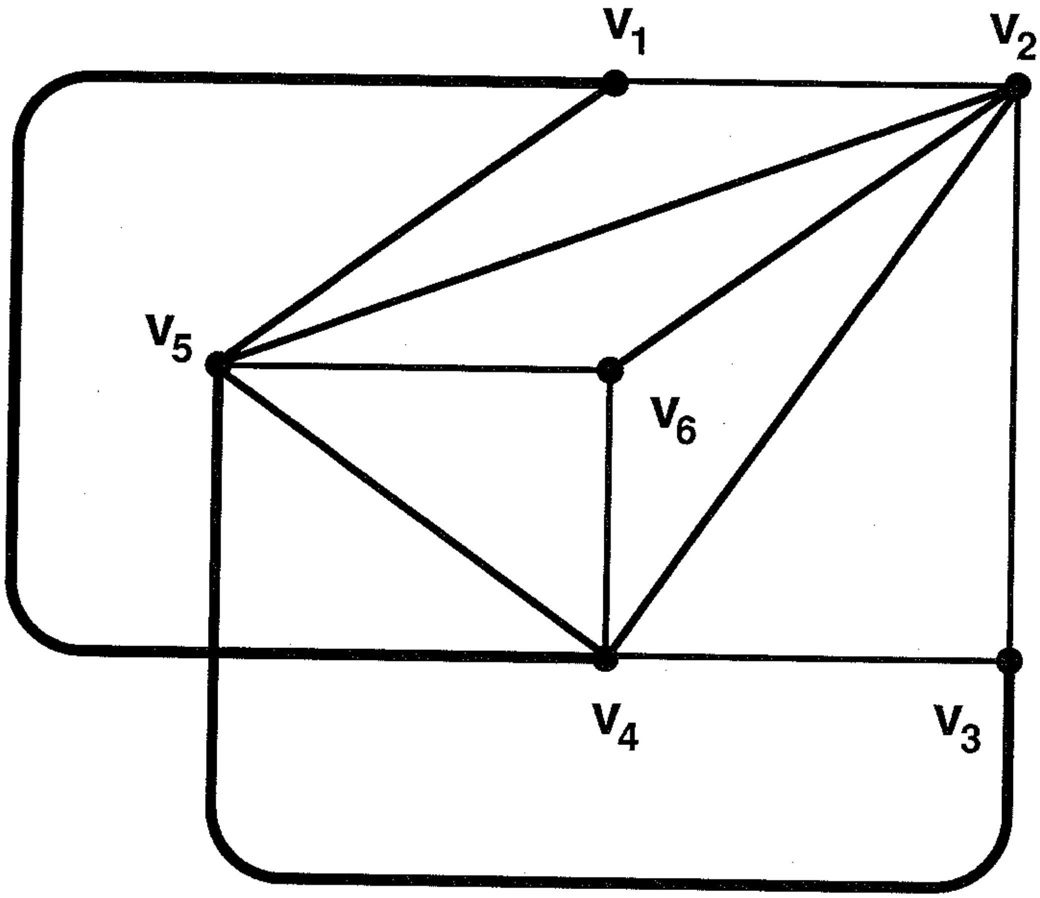

FIGURE 2(a)

NONPLANAR EDGES

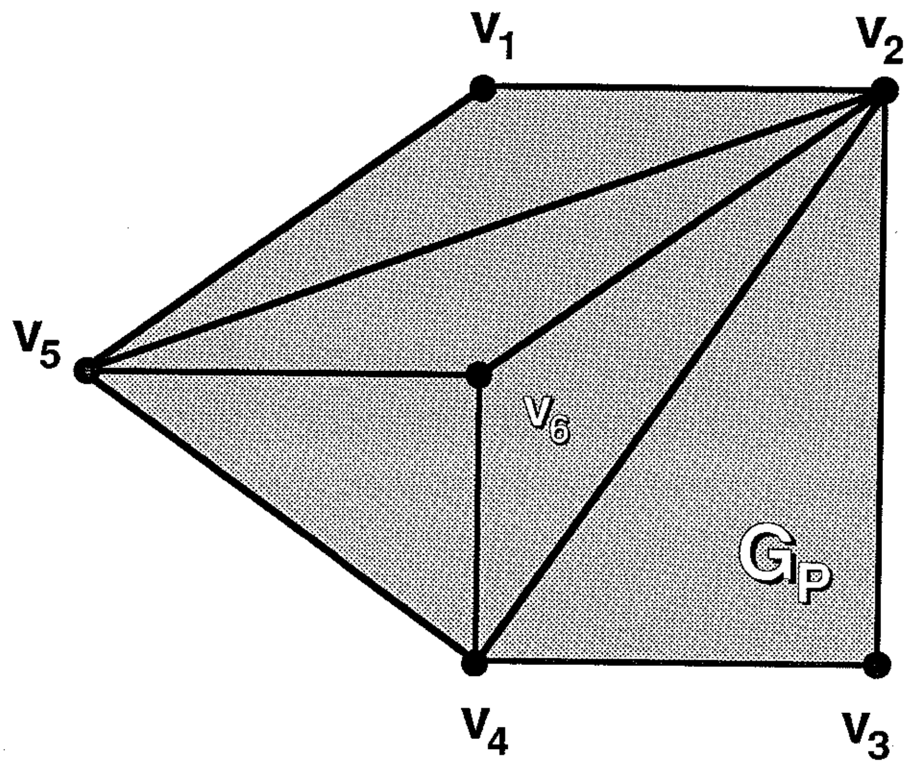

FIGURE 2(b)

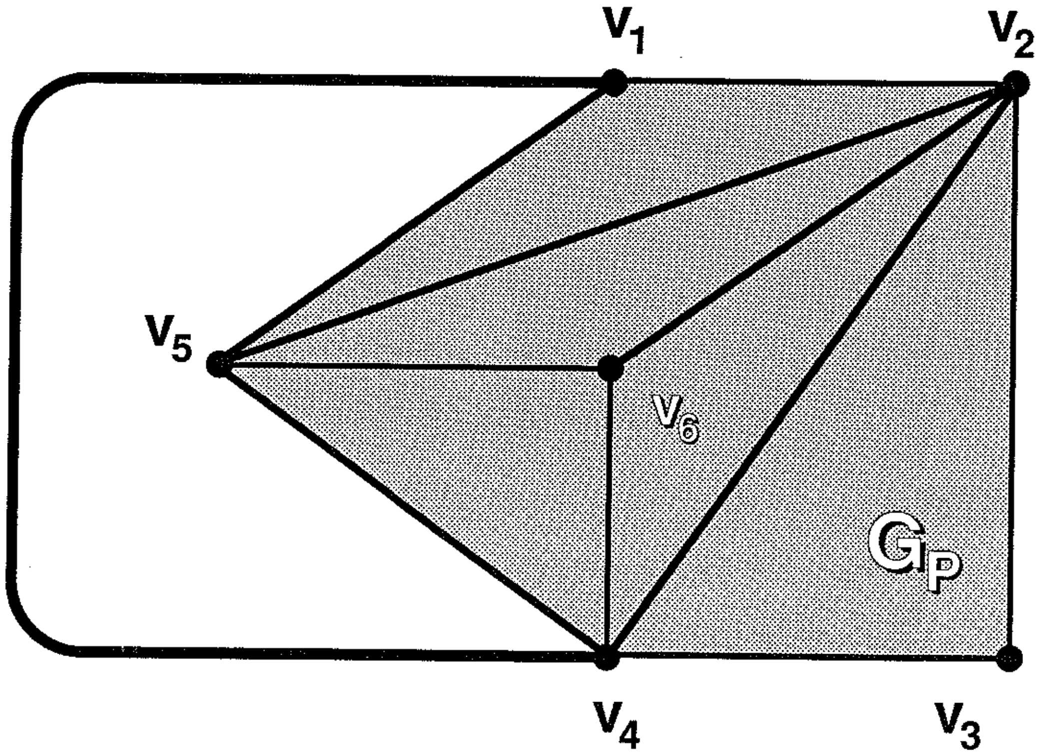

FIGURE 2(c)

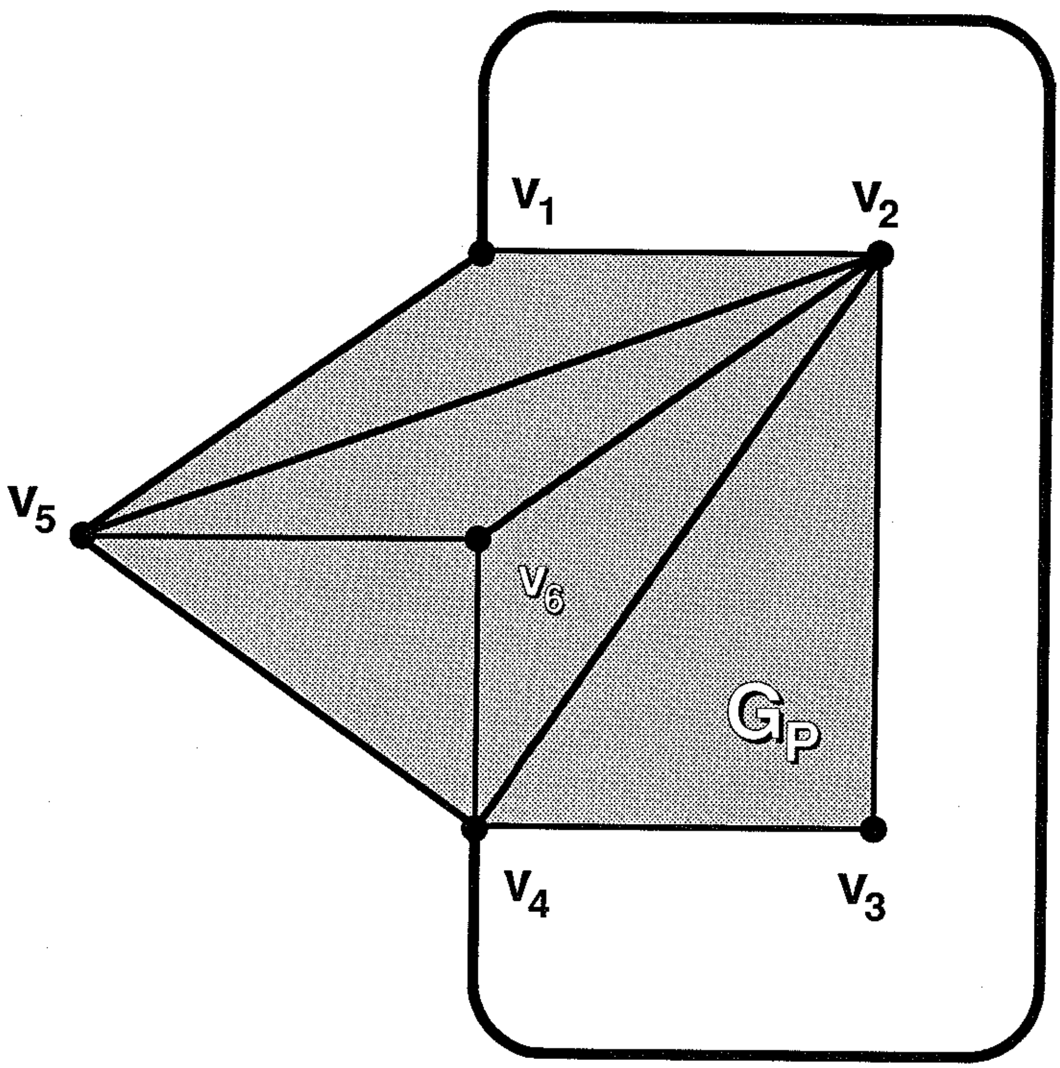

FIGURE 2(d)

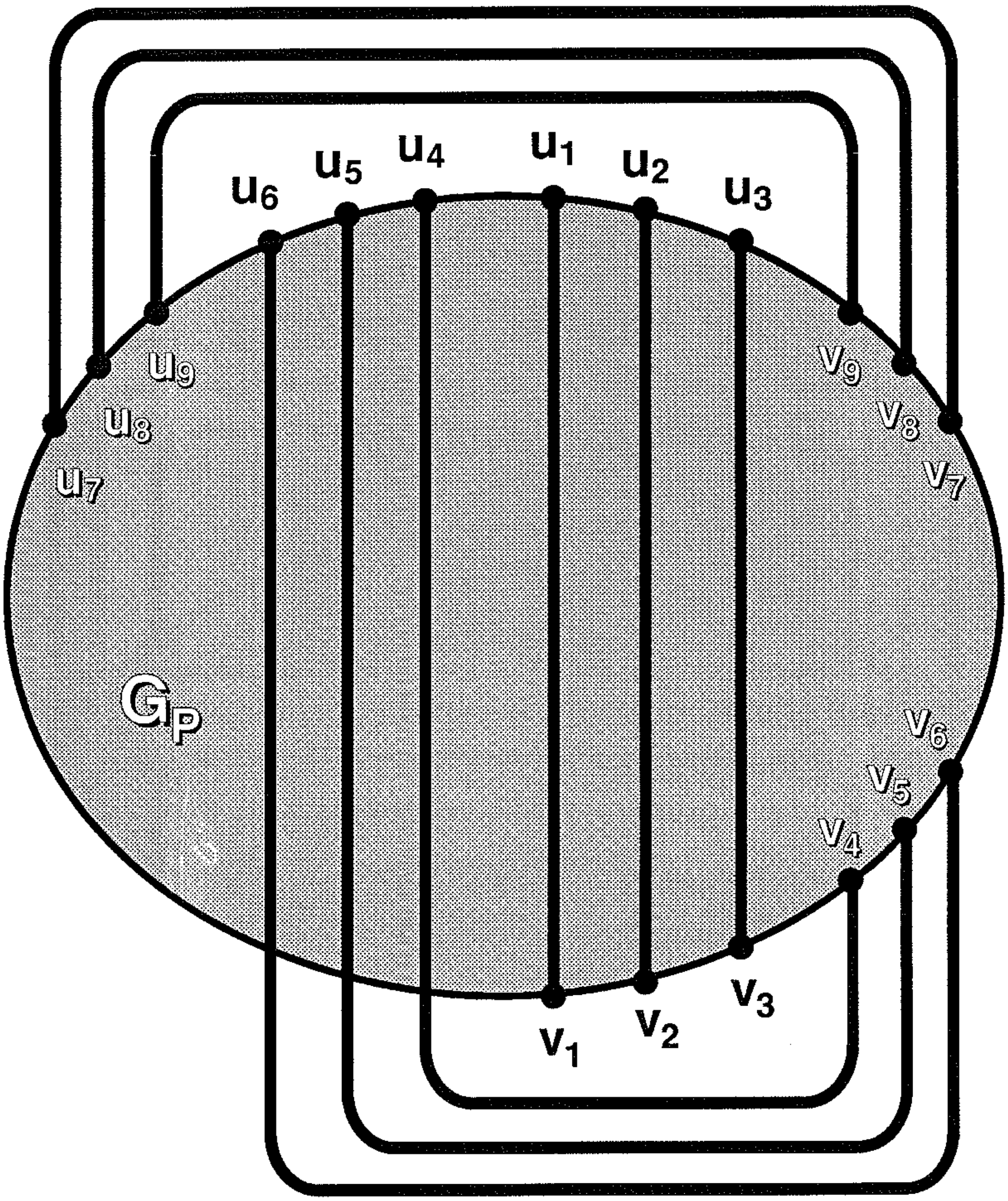

FIGURE 3

HOMOTOPY CLASSES

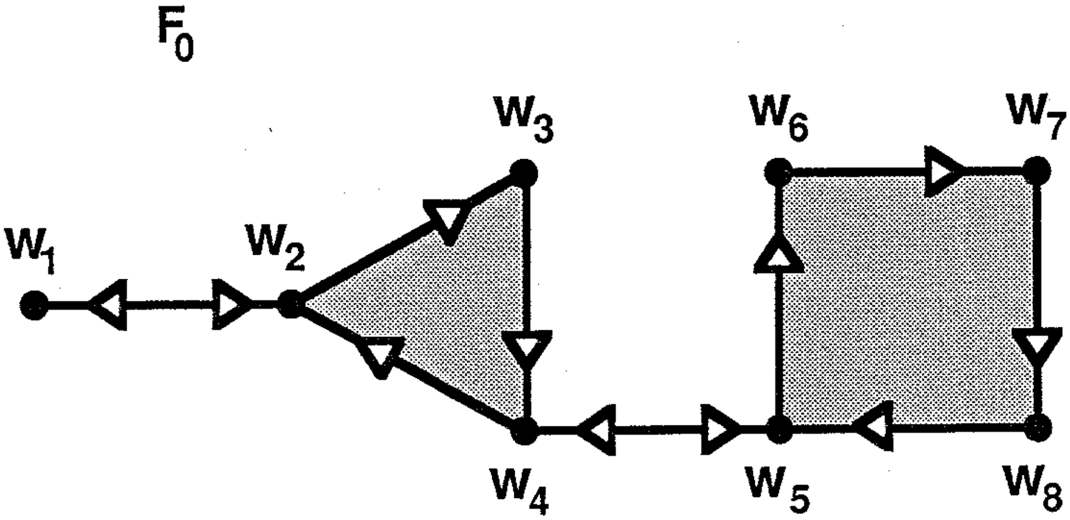

FIGURE 4

TRAVERSAL OF BOUNDARY OF PLANAR PART

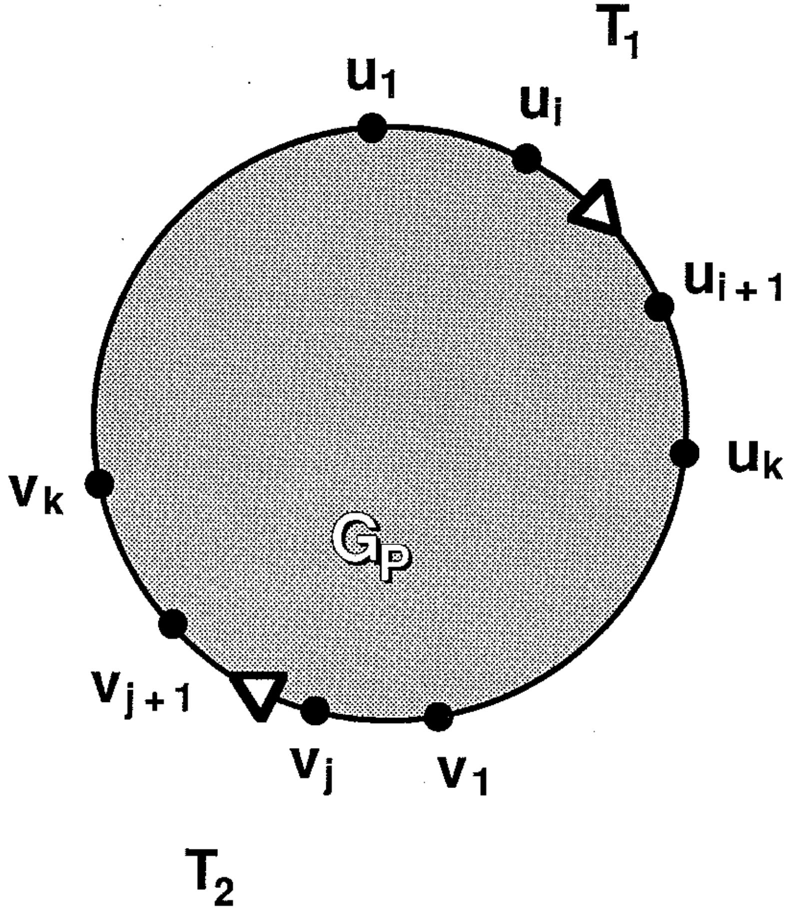

FIGURE 5

SUBTRACES of T_1 AND T_2

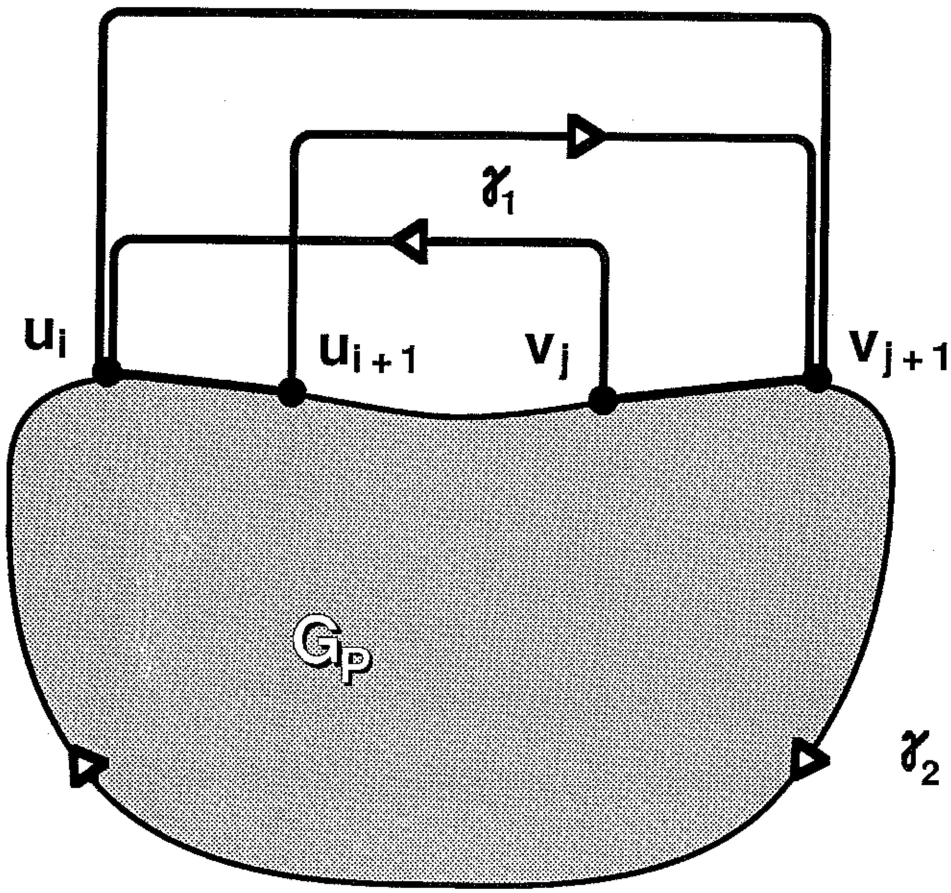

FIGURE 6

CURVES γ_1 AND γ_2

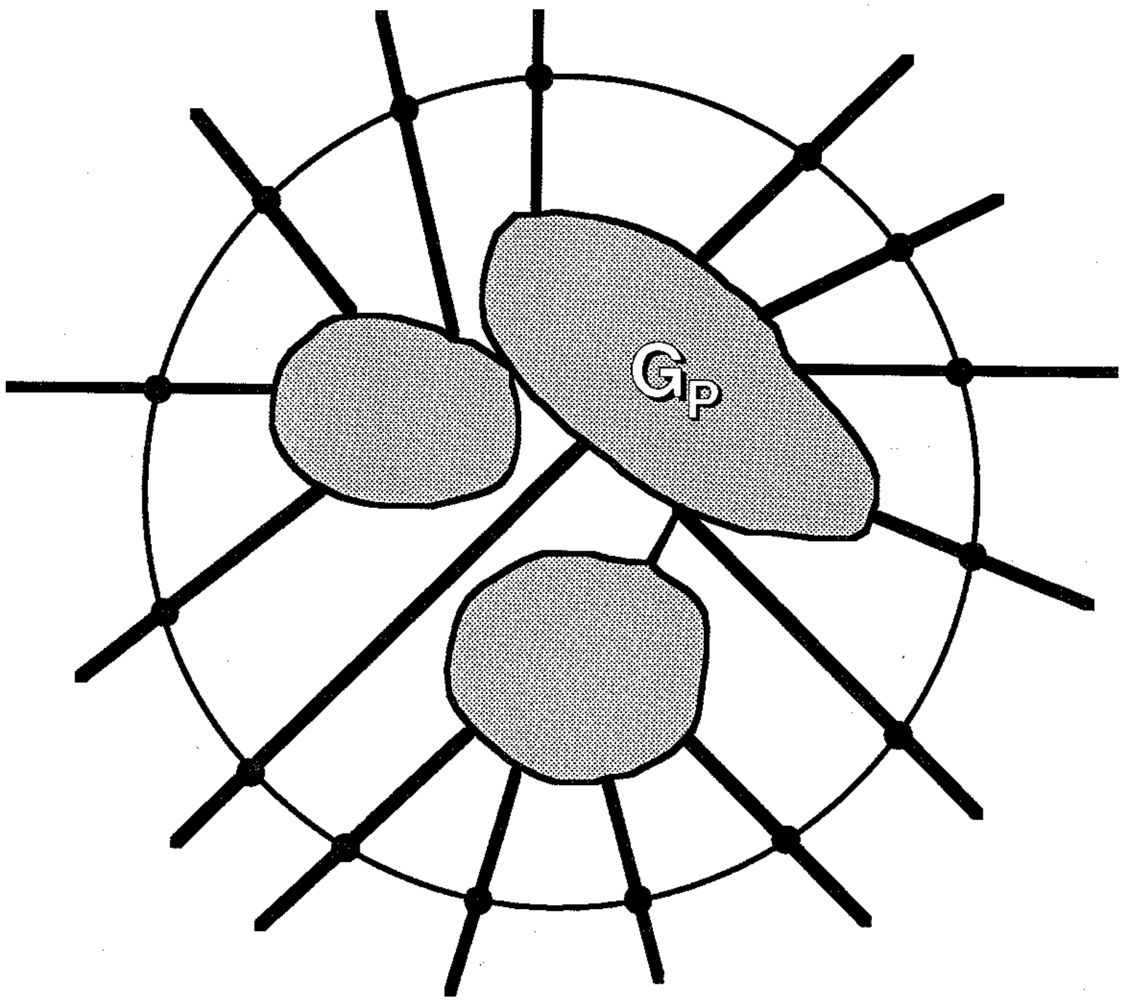

FIGURE 7

CONSTRUCTION FOR LEMMA 3

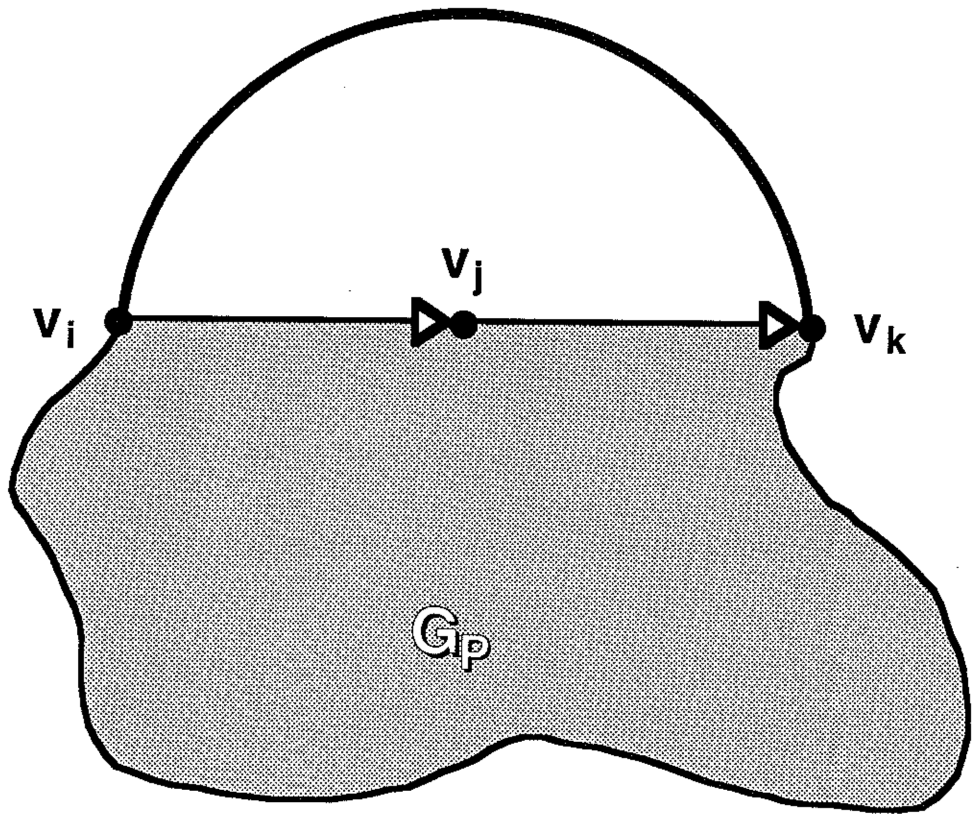

FIGURE 8

SAFE EDGE

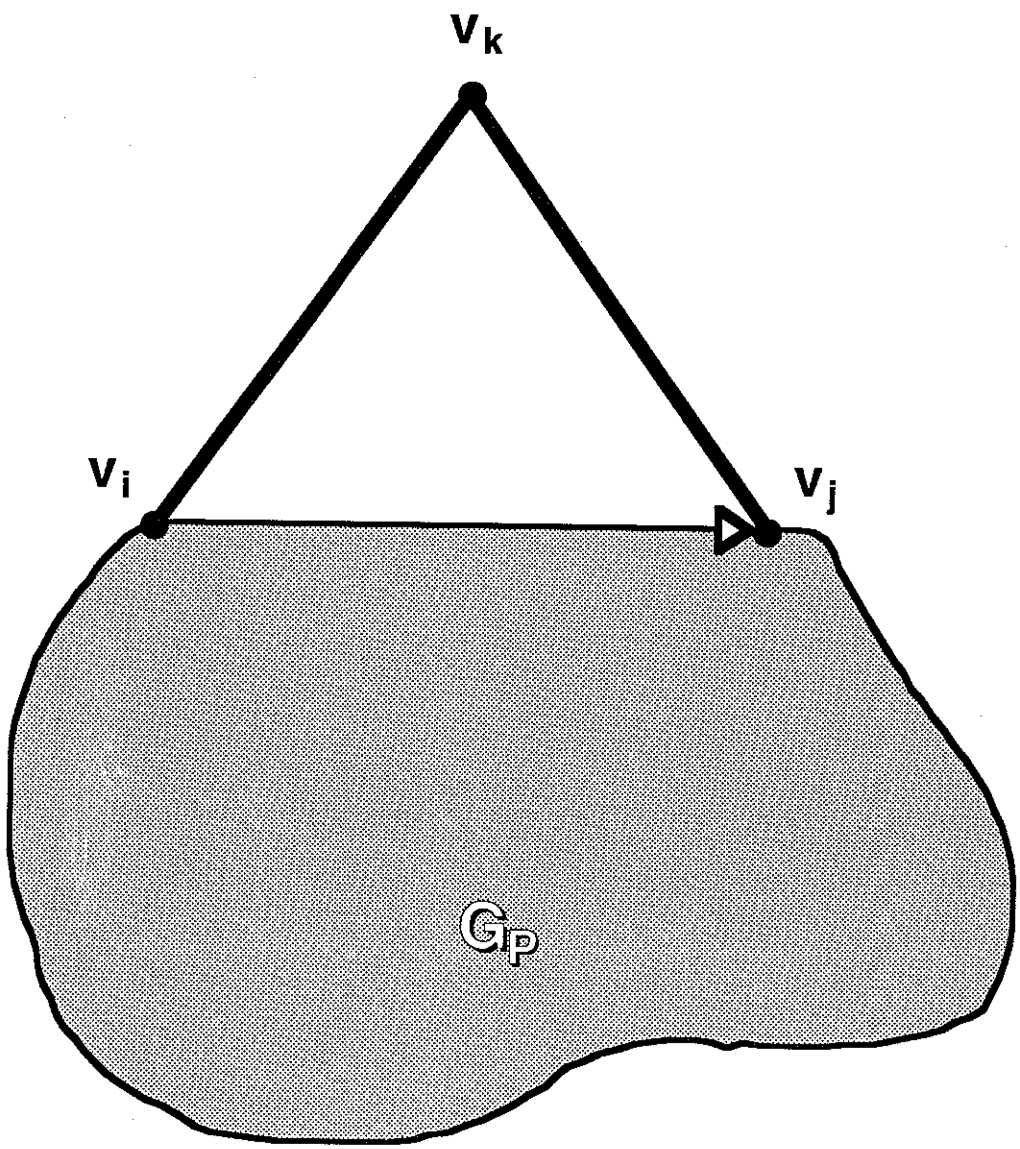

FIGURE 9

SAFE VERTEX

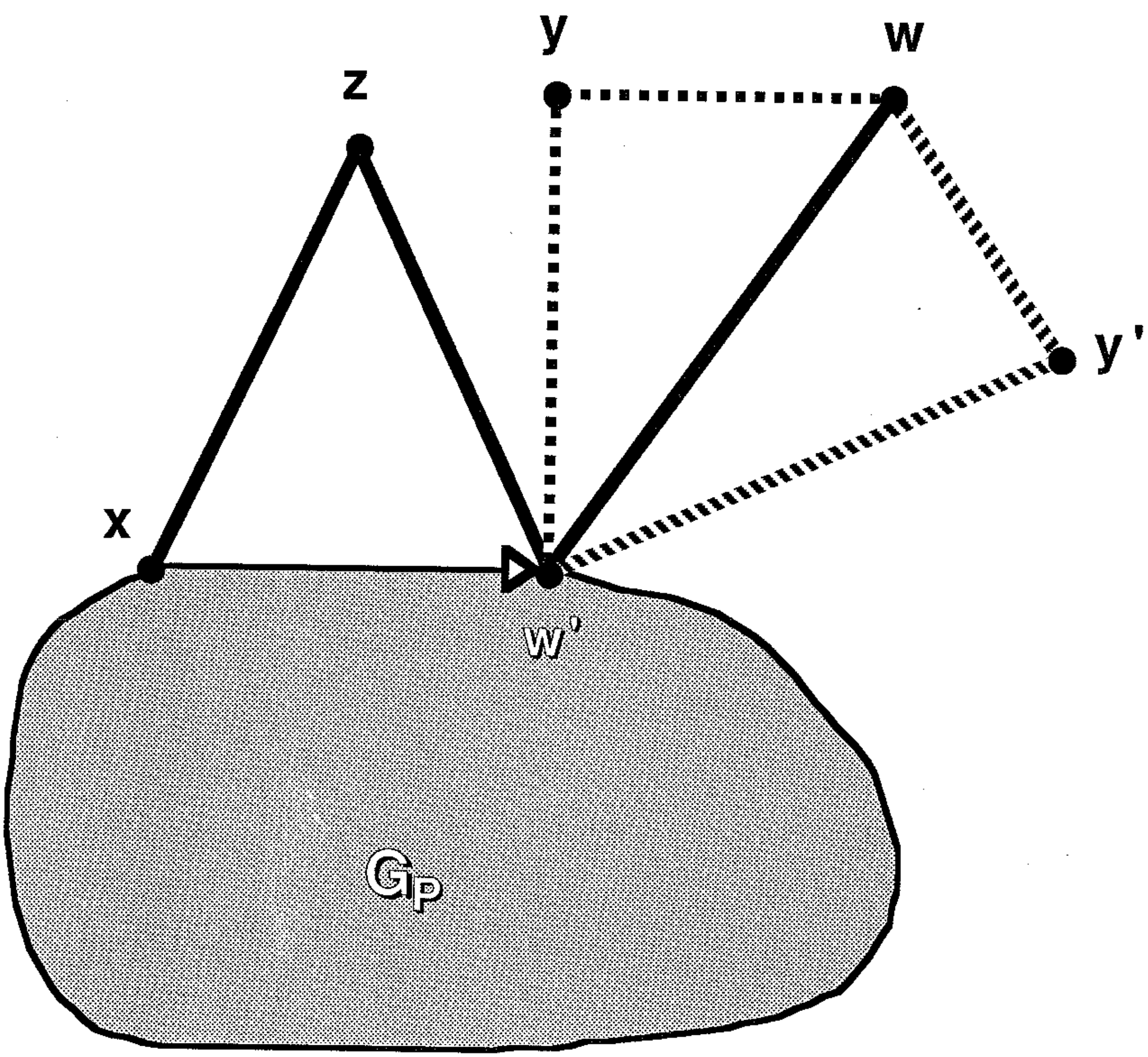

FIGURE 10

NEW BLOCK

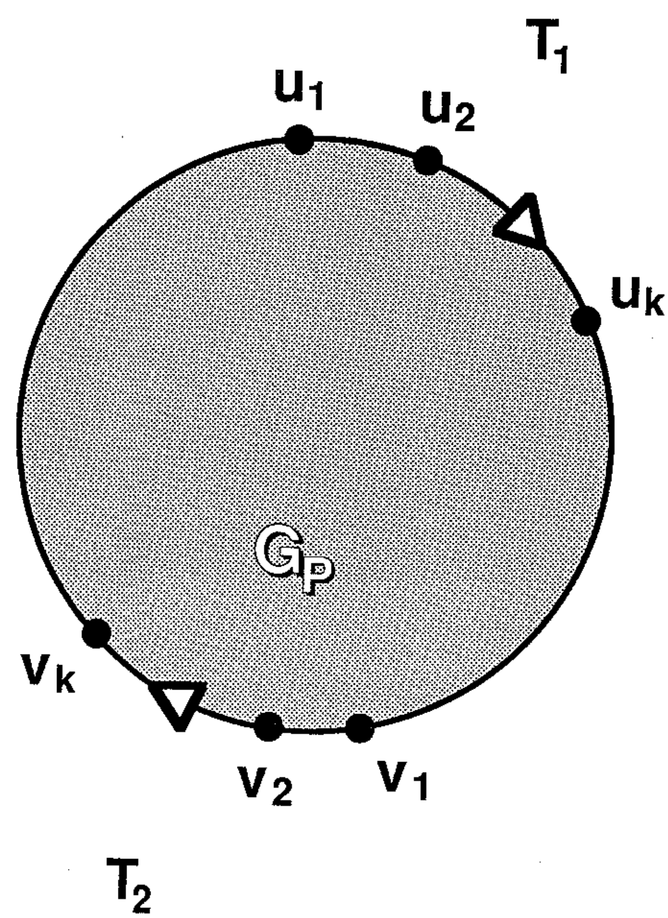

FIGURE 11

TRACES OF A NONORIENTABLE
HOMOTOPY CLASS

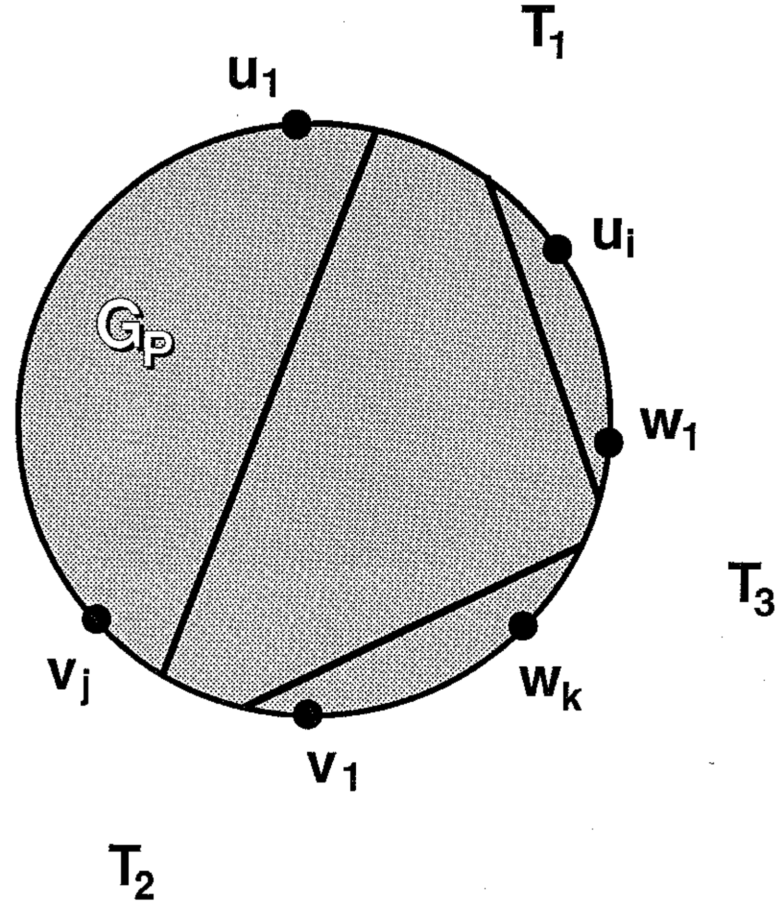

FIGURE 12

THREE REVERSED TRACES WITH CHORDS

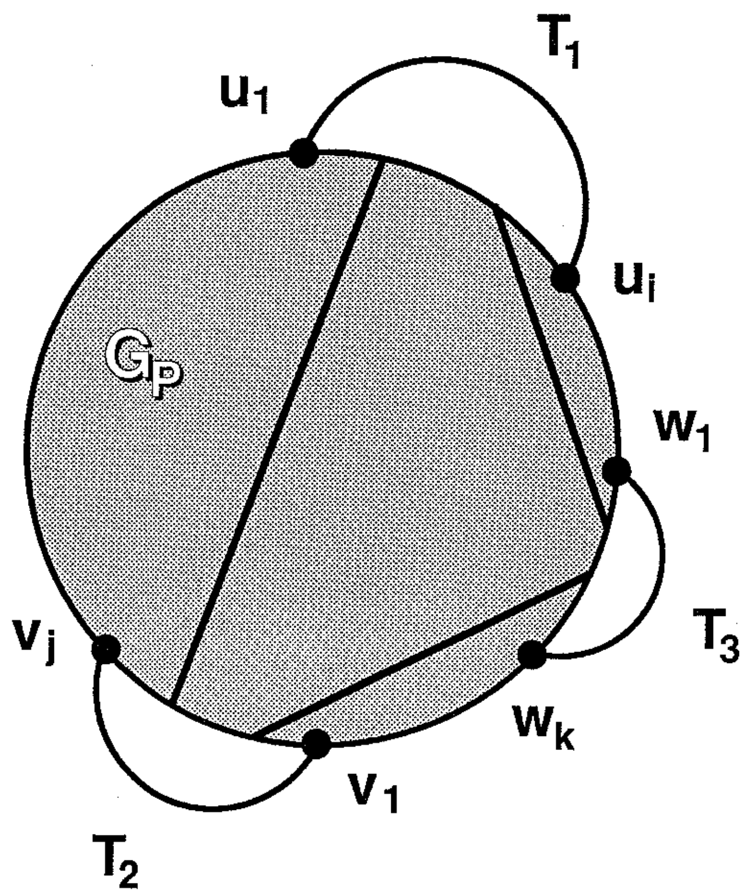

FIGURE 13

ADD EDGES TO REVERSED TRACES

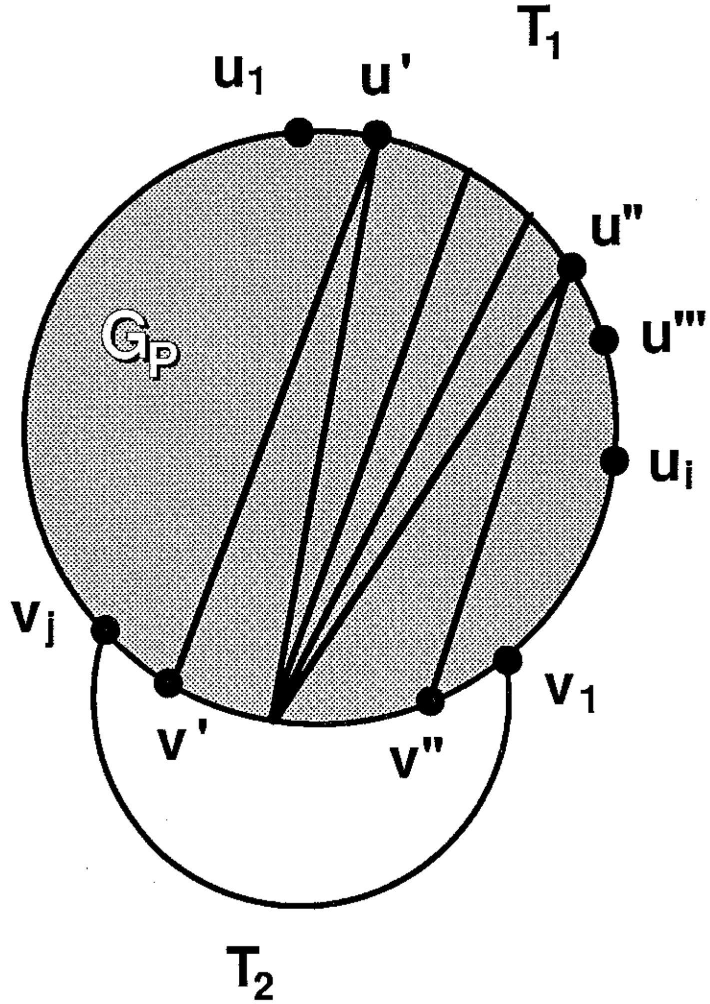

FIGURE 14

BUNDLE OF CHORDS BETWEEN
TWO TRACES