An Efficient Algorithm for the Genus Problem with Explicit Construction of Forbidden Subgraphs

Hristo Djidjev+ and John Reif*

0. Abstract

We give an algorithm for imbedding a graph G of n vertices onto an oriented surface of minimum genus g. If $g > 0$ then we also construct a forbidden subgraph of G which is homeomorphic to a graph of size $\exp(O(g)!)$ which cannot be imbedded on a surface of genus g-1. Our algorithm takes sequential time $\exp(O(g)!)n^{O(1)}$. Since $\exp(O(g)!) = \exp(\exp(O(g\log(g))))$, our algorithm is polynomial time for genus $g=O(\log\log(n)/\log\log\log(n))$. A simple parallel implementation of our algorithm takes parallel time $(\log n)^{O(1)}+O(g)!$ using $\exp(O(g)!)n^{O(1)}$ processors. We give also the smallest known upper bound, namely $\exp(O(g)!)$, on the number $F(g)$ of homeomorphic distinct forbidden subgraphs for graph imbeddings onto a surface of genus g.

Previous algorithms for imbedding a graph onto a surface of genus g had the following sequential time bounds: $n^{O(n)}$ for the naive algorithm, $n^{O(g)}$ for the algorithm of [Filotti, Miller, Reif,79], and $f(g)n^2$ by the algorithm of [Robertson and Seymour,86], where $f(g)$ is a function only of g. The celebrated work of [Robertson and Seymour,86] also gave the first known finite bound for $F(g)$. However their proof spanned many papers and was highly nonconstructive; $f(g)$ and $F(g)$ were bounded by some (large) tower of exponents of g.

Our work provides a distinct constructive approach giving considerably improved bounds for $f(g)$ and $F(g)$ and vastly simplified proofs. In particular, we use a "bootstrap" technique that uses a discovered forbidden subgraph for given genus $g'<g$ to aid us in determination of genus g'+1 imbeddings. It seems likely that our techniques can be extended to many other problems on graphs with bounded tree width.

1 Introduction

1.1 Preliminary Definitions of Imbeddings

Topological Imbeddings

Throughout this paper, we only consider *surfaces* which are orientable 2-manifolds. For our purposes, such surfaces can be uniquely characterized by their *genus* g. Informally, a closed surface of genus g consists of a sphere with the addition of g handles. For example, the sphere has genus 0 and the torus has genus 1. The plane also has genus 0 but is not closed. A *topological imbedding* of an undirected graph $G = (V, E)$ is a mapping of G onto a surface S of genus g (this is also called a *2-cell imbedding*; see [White, 1973]), where each edge is associated with a simple segment on the surface S, where the vertices of the edge are at the two distinct endpoints of the segment, and where no two such edges intersect except at endpoints in the case of common vertices. The *faces* of the imbedding will be defined to be the boundaries of the connected regions obtained by deleting the imbedding of G from the surface. Euler's equation gives n-m+f = 2c-2g, where m, n, f, c, g are the numbers of edges, vertices, faces, connected components, and genus of the imbedding, respectively.

Combinatorial Imbeddings

The topological definition of graph imbedding given above presents difficulties to computer algorithms and their proofs. An alternative (but equivalent) definition will better serve our purpose. Given an undirected graph $G = (V,E)$, let n, m denote the number of vertices and edges, respectively and let its *size* $|G| = n+m$. We will represent an imbedding of graph G in a compact way of size $|G|$ by use of a combinatorial definition of graph imbeddings that is attributed to [Edmonds,60]. Let $D(G)$ be a directed graph derived from G by substituting in place of each undirected edge $\{u,v\}$, a pair of directed edges (u,v) and its reverse $(u,v)^R = (v,u)$. A *combinatorial graph imbedding* $I(G)$ of (undirected connected) graph G is an assignment of a cyclic ordering to the set of the directed edges departing each vertex. **(See Figure 1.1)** The *faces* of this combinatorial imbedding will be defined to be the orbits of a certain permutation of directed edges; this permutation orders (w,v) immediately before (v,u) iff the combinatorial imbedding orders (v,u) immediately before (v,w) in the cyclic order around vertex v. **(See Figure 1.2)** The *genus* of a combinatorial imbedding is defined as $g = (m-n-f)/2+c$ by the Euler formula using the numbers of (undirected) edges m, vertices n, faces f, and connected components c. Edmonds (see also [White,73]) showed that combinatorial imbeddings are equivalent to topological imbeddings. The advantage of combinatorial imbeddings is not just that they can be represented in size $|G|$ in a random access computer. An important additional advantage is that definitions and proofs about such imbeddings can be made entirely combinatorial. For example, given a directed simple cycle $C = (v_0,v_1,...,v_k= v_0)$ of G and an edge $\{v_i,x\}$ where x is not in C but v_i is in C, we define $\{v_i,x\}$ to be imbedded *inside C* (and otherwise *outside C*) if (in the cyclic order defined by $I(G)$ on the directed edges departing vertex v_i) directed

+Institute of Mathematics, Bulgarian Academy of Sciences, Sofia, Bulgaria and School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6. **Supported by Bulgarian Academy of Sciences grant for travel and visit to CS Dept, Duke University in Dec, 1988.**

***Computer Science Dept, Duke University, Durham, NC 27706.** **Supported by DARPA/ISTO Contract N00014-88-K-0458, NASA/CESDIS Contract 550-63 NAS 5-30428 URSA, AFOSR-87-0386.**

edge (v_i,x) appears after directed edge (v_i,v_{i+1}) and before directed edge (v_i,v_{i-1}). (see **Figure 1.3**) As another example, given a directed simple path $p = (v_0,v_1,...,v_k)$ of G and an edge $\{v_i,x\}$ where x is not in p but v_i is in p, we define $\{v_i,x\}$ to be imbedded on *side-1* of p (and otherwise *side-2* of p) if (in the cyclic order defined by $I(G)$ on directed edges departing vertex v_i) directed edge (v_i,x) appears after directed edge (v_i,v_{i+1}) and before directed edge (v_i,v_{i-1}). Hereafter, we will simply use the term ***imbedding*** to denote a combinatorial imbedding.

We will let "$\leq$" denote the subgraph relation. Let H be a subgraph of G and let $I(H)$ be an imbedding of H of genus g. $I(H')$ is an *imbedding extension* of $I(H)$ with regard to G if $H \leq H' \leq G$, $I(H')$ is also an imbedding of H' of genus g and furthermore the orientation of edges around vertices of $I(H')$ is consistent with the orientation of those edges around vertices as given in $I(H)$. If FS is a subgraph of G-H, then *imbedding FS onto I(H)* means finding an imbedding extension $I(H+FS)$ of $I(H)$.

1.2. The Complexity of Some Previous Algorithms for Graph Genus

The *genus* of graph G which will be denoted by genus(G) is the minimum $g \geq 0$ s.t. G can be imbedded onto a surface of genus g. Using purely combinatorial techniques, [Miller,85] has shown that the genus of a graph is the sum of the genus numbers of its biconnected components. He also showed that minimal genus imbeddings of any biconnected subgraphs can be easily combined in time $O(|G|)$, where $|G|$ denotes the number of vertices and edges of G, to get a minimal genus imbedding of G. Hereafter, we assume without loss of generality that the graph is biconnected.

The *genus problem* is: given a graph G, determine the genus g of G. The genus problem is very difficult for g growing as a function of $|G|$. An enumerative algorithm of [Edmonds,60] gave a $|G|^{O(|G|)}$ algorithm for the genus of G. [Reif,78] first showed that the problem of extending a given graph imbedding is NP complete, and recently [Thomassen,89] showed that given a graph imbedding of genus g, the problem of testing if there is an imbedding of genus $< g$ is NP complete. This implies the problem of testing if a graph has genus g is NP complete, and therefore there does not exist a polynomial algorithm for finding the genus of the graph unless P=NP.

Nevertheless, the problem of imbedding graphs of unbounded size onto fixed surfaces of low genus g may be efficiently solved. Let a ***PT algorithm*** be a planarity testing algorithm taking sequential time $O(|G|)$, e.g. that of [Hopcroft and Tarjan,74]. [Klein and Reif,1987] gave the first efficient parallel planarity algorithm with $O(\log^2 n)$ time and n processors, where n is the number of vertices of G. Recently, [Ramachandran and Reif,89] gave a $O(\log n)$ time parallel algorithm for graph planarity with work nearly $O(n)$. As another example, [Filotti,80] gave an $n^{O(1)}$ time algorithm for testing if a graph G has genus 1, i.e. can be imbedded onto a torus.

Graphs of bounded genus appear naturally in various applications - for example, in VLSI layout via bounded book thickness imbeddings. Many difficult graph problems can be solved in polynomial time in the case of graphs of bounded genus; for example [Miller,83] showed that for bounded genus graphs, the isomorphism problem can be solved in polynomial time. [Djidjev,85] gave a linear time algorithm for finding small separators of graphs of bounded genus.

[Fillotti, Miller, Reif,79] showed that given a graph G, its genus g and imbedding of G of genus g can be computed in time $|G|^{O(g)}$. This gave the first polynomial time bound for the genus problem with fixed genus g.

1.3. Forbidden Subgraphs

A key aspect of our algorithm, used to aid us in the construction of higher genus imbeddings, is the discovery of certain forbidden subgraphs of imbeddings of lower genus, as defined here.

A path in graph G will be called a *2-path*, if each of its (non-endpoint) vertices is incident to no more than two edges of G. A 2-path p of G will be called a *maximal 2-path*, if no other 2-path of G contains p. We will define the *branchsize* $[G]$ of G to be the number of maximal 2-paths of G. If $p=(v_1,...,v_k)$ is a maximal 2-path of F, then $(v_k,...,v_1)$ is also a maximal 2-path of F and will be denoted by p^R. The ***homeomorphic contraction*** of G is obtained by substituting an edge for each maximal 2-path in G. (**See Figure 1.4**). Note that the branchsize $[G]$ is the number of edges of the homeomorphic contraction of G. Two graphs are *homeomorphic* if their homeomorphic contractions are isomorphic. (**See Figure 1.5**).

We define graph FS to be a *forbidden$_g$ subgraph* of graph G if FS is a minimal subgraph with genus $> g$ (i.e., deletion of an edge or vertex of FS results in a graph of genus at most g). [Kuratowski,30] showed that the forbidden$_0$ subgraphs are all homeomorphic to K_5 or $K_{3,3}$. A number of researchers have independently observed that, if G is nonplanar, *PT* algorithms can be extended to find in time $O(|G|)$ a maximal (in specific context) planar subgraph of a nonplanar graph G, and also in time $O(|G|)$, given the maximal planar subgraph, to find a forbidden$_0$ subgraph of G homeomorphic to K_5 or $K_{3,3}$. We will call such an extension a ***PT-FS algorithm.*** Recently [Khuller,Mitchell,Vazirani,89] gave an $O(\log^2 n)$ time and $O(n)$ processor parallel *PT-FS* algorithm, using the techniques of [Klein and Reif,88].

1.4. The Work of Robertson and Seymour

In a celebrated series of papers on graph minors, [Robertson and Seymour,I-VIII] proved that for each genus $g \geq 0$, there is a finite number $F(g)$ of homeomorphic distinct forbidden$_g$ subgraphs and furthermore, they showed that given a graph G of genus $> g$, in time $f(g)n^2$ there can be found a forbidden$_g$ subgraph of G. In their original work , the upper bounds $f(g)$ and $F(g)$ were nonelementary functions of g (in fact $f(g)$ and $F(g)$ were originally not explicitly known and instead were computed by a procedure involving a sequence of towers of towers of iterated exponents). More recent unpublished work of Robertson and Seymour has brought bounds on $f(g)$ and $F(g)$ to a bounded but large number of repeated exponents.

These results in graph theory where a major breakthrough - in our opinion the greatest breakthrough in graph algorithms in at least the last decade. However, the extreme dependence on the parameter g made their results difficult to apply in practice even for very small g (say $g > 3$). Also, because of the extreme complexity of their proof (encompassing many papers), their work has not been properly understood by the theoretical computer science community.

1.5. Our Contribution

The key contribution of our paper is to derive considerably improved bounds on both $f(g)$ and $F(g)$, namely of $\exp(O(g)!) = \exp(\exp(O(g\log(g))))$. Our algorithm is polynomial time for genus $g = O(\log\log(n)/\log\log\log(n))$. Our results are proved essentially independently of the work and extensive papers of Robertson and Seymour. We use none of their results but do use minors extensively; we use a "bootstrap" technique that uses a discovered forbidden subgraph for given genus $g'<g$ to aid us in determination of genus $g'+1$ imbeddings. We also use constraint graphs related to those of the parallel planarity paper of [Ramachandran and Reif,89]. Aside from our improved results and more constructive approach, the further impact of our work is that we provide an independent verification and vastly simpler proofs (using new techniques) of the basic results of Robertson and Seymour in a particular fundamental area of application, namely graphs of bounded genus. It seems likely that our new techniques can be extended to many other problems on graphs with bounded tree width.

2. General Description of Our Imbedding Algorithm

2.1. Definition of Bridges

We will require a few more graph definitions concerning subgraphs. Given a graph G and a subgraph H of G, let G-H consist of the subgraph obtained by deleting from G all edges of H and deleting every vertex of H incident only to edges of H. Note that G-H may have vertices in common with H; these are called the *attachment vertices* of G-H. A *bridge* B of H in G is a subgraph of G-H induced from a maximal set of edges for which between any pair of edges there is a path in G-H avoiding attachment vertices (**See Figure 2.1**). Furthermore B is a *bridge* of F, where F is a face of $I(H)$, if all attachment vertices of B are on F. By $[G\text{-}H]$ we denote the sum of the branchsizes of the bridges of G-H. The edges of bridges adjacent to attachment vertices are called *attachment edges*. If G_1 and G_2 are graphs, we define the graph $G_1+G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$.

Fix a graph G and a subgraph H with a given imbedding $I(H)$ of genus g. If FS is a subgraph of G-H and FS can not be imbedded onto $I(H)$, then FS is a *forbidden$_g$ subgraph of G with respect to $I(H)$*. Let a *forbidden$_g$ subgraph FS* for (non-imbedded) H in G be a minimal subgraph of G s.t. genus$(FS+H)$ > genus(H) (see also Section 1.4).

2.2. Skeletal Imbeddings

Given an imbedding $I(G)$ of genus g of a biconnected graph G, a *skeletal subimbedding $I(H)$* of $I(G)$ is an imbedding $I(H)$ of genus g (consistent with imbedding $I(G)$) of a minimal biconnected subgraph H of G, such that if we delete any edge of $I(H)$ then the resulting imbedded graph is either now a genus $< g$ imbedding or no longer biconnected.

Let T be any spanning tree of G. A skeletal subimbedding $I(H)$ of $I(G)$ can be found by a 2 step process:

(1) Repeatedly delete from G nontree edges incident to two faces in $I(G)$ (not necessarily preserving biconnectivity) until there is only one face remaining in the resulting graph. By Euler's formula there will be only $2g$ remaining nontree edges. The $2g$ basis cycles defined by these remaining nontree edges define a subimbedding of genus g with 1 face.

(2) Next, we need add to this subimbedding at most $2g$ further basis cycles and then we repeatedly delete dangling (incident to vertices of degree 0) tree edges until the resulting imbedded subgraph $I(H)$ becomes biconnected.

Thus finding a skeletal subimbedding $I(H)$ of $I(G)$ requires nondeterministic choice of at most $O(g)$ nontree edges, and H consists of at most $O(g)$ basis cycles. The key point is that H has at most $O(g!)=g^{O(g)}$ distinct imbeddings of genus g, whereas G may have $[G]!$ distinct imbeddings of genus g.

Given an (unimbedded) biconnected graph F and a spanning tree T of F, let $SI_g(F,T)$ be the set of all possible (with respect to T) skeletal subimbeddings $I(H)$ for all possible imbeddings $I(F)$ of genus g. We can construct $SI_g(F,T)$ in deterministic time $([F]g)^{O(g)}$ (rather than inefficiently enumerating through all imbeddings of F of genus g) by simply enumerating all possible choices of the $O(g)$ basis cycles used to construct the skeletal subimbeddings and further enumerating through all possible imbeddings of these $O(g)$ basis cycles, and then verifying whether each resulting subimbedding is skeletal.

2.3 Outline of our Imbedding Algorithm

Note: "guess" and "choose" are to be executed by sequentially iterating over all possibilities.

Algorithm 2.1

Input biconnected graph G

Output the genus of G and an imbedding of G on a surface of minimum genus

Call procedure *PT-FS* on G. ***If*** G is planar ***then output*** "genus(G) = 0" and the planar imbedding ***halt;***

else let F_1 be the forbidden$_0$ subgraph of G and ***let*** $g := 1$

While $g \le (n^2-n)/2$ ***do***

Comment: F_g denotes the current forbidden$_{g-1}$ subgraph of G.

Comment: U_g denotes a set of subgraphs of G used to augment F_g to F_{g+1} if genus$(G)>g$.

Comment: Assume as loop invariants: genus$(F_g) = g$; F_g is biconnected; $[F_g] \le \exp(O(g)!)$.

1. Ug := empty set {};
2. Construct some spanning tree T of Fg
3. Construct the set of skeletal subimbeddings $SI_g(F_g,T)$ by Algorithm of Section 2.2
4. For each skeletal subimbedding I(Hg) $\in$ *SIg (Fg,T)* ***do***

Comment: there are $\le$ $([F_g]g)^{O(g)}$ skeletal subimbeddings of F_g

4.1. Construct all bridges $B_1,\ldots,B_b$ of G-Hg.

4.2. ***If*** for some i, $1 \le i \le b$, there is no planar imbedding of $B_i' = B_i$ - attachments(B_i) ***then let*** FS := forbidden$_0$ subgraph of B_i' of branchsize $O(1)$. To insure biconnectivity, add to FS two simple paths to the attachments of B_i ; ***goto*** 4.6 ; ***else do***

4.3. Find by the algorithms of Section 3 an imbedding $I(H''_g)$ of a subgraph H''_g of G, $H_g \le H''_g \le G$, such that H''_g-H_g consists of $O(g)$ maximal 2-paths and also find a minimal set FS of $\le O(g)$ maximal 2-paths of G-H''_g that cannot be imbedded onto $I(H''_g)$ without increasing the genus.

4.4 ***If*** FS= {} ***then do***

Attempt to imbed the bridges of G-H''_g onto $I(H_g)$ using the reduction of Section 4 to 2-satisfiability and the algorithm of Appendix A for 2-satisfiability

if successful ***then do***

output "genus$(G)=g$" and the resulting genus g imbedding $I(G)$ of G; ***halt od***

4.5. ***else do*** Using Algorithm 5.1 of Section 5, find a set of $O(1)$ maximal 2-paths in G-H''_g that cannot be all imbedded onto $I(H''_g)$. Let FS be the subgraph of G consisting of those paths and H''_g-H_g ***od fi***

4.6. $Ug := Ug \cup \{FS\}$ ***od***

5. Using Algorithm 6.1 of Section 6, construct from U_g and F_g a biconnected subgraph $F_{g'}$ of G of branchsize $[F_g]^{O(g)}$ such that genus$(F_{g'}) = g' > g$
6. $g := g'$

od

2.4. Organization of the Paper

We have just given some preliminary definitions of graph imbeddings and an outline of our imbedding algorithm. Section 3 describes a reduction of the original embedding problem to certain restricted imbedding problems (namely, the weakly quasiplanar extension imbedding problem).

The *2-satisfiability problem* is to determine satisfiability of a boolean CNF formula with at most 2 literals per clause. Section 4 gives a reduction from this restricted graph imbedding extension problem to 2-satisfiability. While there are known linear time(and NC) solutions to the 2-satisfiability problem, for completeness we give a particularly simple linear time algorithm in Appendix A, namely a direct. reduction to digraph reachability.

Section 5 gives a reduction of the problem of finding forbidden$_g$ subgraphs of imbedded graphs to finding forbidden$_0$ subgraphs. Section 6 gives an algorithm for defining a forbidden$_g$ subgraph for F_g of G of branchsize $[F_g]^{O(g)}$. Section 7 concludes the paper with bounds on the size of $[F_g]$ thus giving bounds on the total time of our imbedding algorithm as well as bounds on the size and number of forbidden$_g$ subgraphs.

3. Reduction to Planarity Testing and 2-Constrained Imbedding Problems

3.1. Quasiplanar and Non-Splittable Extensions

First let us introduce some additional definitions and notations. Let H be a biconnected graph with imbedding $I(H)$ on a surface of genus g. If $g>0$ then some vertices may appear more than once on a single face. Let an *internal vertex* (*edge*) of $I(H)$ be a vertex(edge, respectively) of H which appears at least twice in a single face F of $I(H)$ (See **Figure 3.1**). If a maximal 2-path in $I(H)$ contains one internal edge, then all its vertices are internal. Such path will be called *a maximal internal 2-path.* For any path p let $\hat{p}$ denote the set of all vertices of p except the endpoints of p. From the definitions we get

Observation 3.1. A maximal 2-path p in $I(H)$ is a maximal internal 2-path iff $\hat{p}$ belongs to exactly one face of $I(H)$.

Observation 3.2. Let p be a maximal internal 2-path in $I(H)$ and let v be an endpoint of p. Then the degree of v in $I(H)$ is at least 3.

Note that Observation 3.2 would have not been true if H was not biconnected.

If not mentioned otherwise, in this section H will denote a subgraph of G of genus g and $I(H)$ will denote a genus g imbedding of H. We will define an imbedded internal 2-path p of $I(H)$ (note p may not be a 2-path of G due to attachment edges in in G-H) to be *constrained* (with respect to a specified side *side-1* or *side-2*) if in any imbedding extension of $I(H)$, we require all attachment edges (in G-H) to p to be imbedded to the specified side of the ordered p (see Section 1.2 for definition of sides). Otherwise, p is *non-constrained.*In the following, it may be that certain internal 2-paths of I(H) are so constrained (this will occur during our algorithm).

Definition 3.1. Let k be an integer, p be a 2-path of G-H with endpoints in H and g be the genus of $I(H)$. If there do not exist more than k imbeddings of p onto $I(H)$, then p is called a *k-constrained path* with respect to $I(H)$. If B is a bridge of G-H and there do not exist more than k imbeddings of B onto $I(H)$, then B is called a *k-constrained bridge* with respect to $I(H)$.

For example, if G is planar and triconnected and H is biconnected, then any bridge of G-H will be 2-constrained. As is well known, planarity testing requires linear time [HT]. More generally, if all bridges of G -H are 2-constrained, Section 4 and Appendix A show that an imbedding (if it exists) of G extending imbedding $I(H)$ can be found in polynomial time.

Definition 3.3. $I(H')$ is a *weakly quasiplanar extension* (WQPE) of $I(H)$ regarding G if:

1) $I(H')$ is an extension of $I(H)$ regarding G and
2) all bridges of G-H' are 2-constrained.

For comparison, in [Filotti, Miller, Reif, 1979] the stronger notion of a *quasiplanar extension* (QPE) was used, where $I(H')$ is a QPE of $I(H)$ regarding G if no face of $I(H')$ has any internal vertex. We shall prove in this section that given an imbedding $I(H_g)$ of H_g of genus g, a possible WQPE of $I(H_g)$ regarding G can be found in $|G|[H_g]O((g^3)!)$ time.

The advantage of having a WQPE (or QPE) of $I(H_g)$ is that the corresponding extension problem can be reduced to solving a 2-satisfiability problem(see Section 4). Appendix A shows that this 2-satisfiability problem can be solved in polynomial time.

Let F be a face of $I(H)$. The *repetition number* $\mathrm{rep}_F(v)$ of an internal vertex v of F is the number of times v appears on F. Let $A(F)$ denote the set of all non-constrained internal vertices v of F such that $\mathrm{rep}_F(v)\geq 3$ and such that there is an attachment edge of G-H containing v.

Define $S(F) = \sum_{v\in A(F)} (\mathrm{rep}_F(v)-2)$.

Let $S(I(H))$ denote the sum of $S(F)$ over all faces F of $I(H)$. Then by Observation 3.2 it follows that we can obtain an upper bound on the number of maximal internal 2-paths if we can estimate $S(I(H))$. By the next lemma we provide an upper bound on $S(I(H))$.

Lemma 3.1. $S(I(H)) \leq 4g$.

Proof: The Euler characteristic of H is m-n-f = $2g$-2 where m,n, f are the number of edges, vertices, and faces of $I(H)$. Build a spanning tree of H. Suppose that there exists a nontree edge e incident on two different faces of $I(H)$. Then e is a non-internal edge. Consider a maximal 2-path p in H containing e. From Observation 3.1 $\hat{p}$ will contain no internal vertices. Remove $\hat{p}$ and its attachment edges and merge the faces. Since the genus is the same, the Euler characteristic remains invariant. Repeat until no such edge e exists. The resulting graph H^* has 1 face and Euler characteristic $2g$-2. Therefore

(3.1) $\quad m_1-n_1-1=2g-2,$

where n_1 and m_1 are the number of vertices and edges of H^*. As obviously the deletion of the vertices of $\hat{p}$ (which are not internal) from the corresponding graph does not modify the current value of $S(I(H))$, then $S(I(H)) = S(I(H^*))$. We will show that the minimum vertex degree of H^* is at least 2 whence $S(I(H^*)) \leq 2(m_1-n_1)$ will follow.

If $g = 0$, then it is easy to show that H^* is a single vertex, and so $S(I(H)) = 0$.

Otherwise, assume $g > 0$. As H is biconnected, then the vertex degree of H is at least 2. Let w be an endpoint of p. Then w can not be incident to only one edge of H. Furthermore, since p is maximal and g > 0, then w is incident to more than two edges of H. Then the removal of $\hat{p}$ from H does not create vertices of degree one or zero. Thus the vertex degree of H-p is at least 2. By induction, the minimum vertex degree of H^* is at least 2. By the definition of $S(I(H^*))$ and (3.1) we get

$S(I(H^*))\leq 2(m_1-n_1) \leq 4g$. **Q.E.D.**

Corollary 3.1. The number of all maximal internal 2-paths of $I(H)$ is $O(g)$.

Recently [Bender and Richmond,1990] have obtained a similar result to Corollary 3.1 giving the exact worst-case bound of $6g$-3 on the number of all maximal internal 2-paths of $I(H)$ ($g\geq 1$).

Note that an imbedding $I(H)$ is a WQPE iff $S(I(H)) = 0$. Thus our goal is to decrease $S(I(H))$.

Let $F = (q_1,q,q_2,q^*)$ be a face of $I(H)$ where q and q^* are maximal internal 2-paths (not necessarily reverses of each other). Let p be a path in G-H with endpoints on F. If an imbedding of p in F is such that the two endpoints of p are on q_1 and q_2 respectively, then the imbedding will be called a *splitting imbedding* of p and p will be called a *splitting imbedded path.* (**See Figure 3.2.**) We define p as a *splitting path* for F, if some imbedding of p is a splitting imbedding.

Lemma 3.2. Any splitting imbedding of a path of G-H onto $I(H)$ decreases the number $S(I(H))$ defined in Lemma 3.1 or splits a face F into 2 faces F_1 and F_2 such that $S(F)=S(F_1)+S(F_2)$ and $S(F_1)>0$, $S(F_2)>0$.

Proof. See [Filotti, Miller, Reif,79].

Definition 3.3. Let H be a subgraph of G and $I(H)$ be an imbedding of H. A *non-splittable extension* (NSE) of $I(H)$ regarding G is an imbedding $I(H')$ of a graph H', where $H \leq H' \leq G$ such that

1) $I(H')$ is an extension of $I(H)$;
2) No splitting path of G-H' exists.

To find a WQPE of $I(H_g)$ regarding G, where H_g is a subgraph of G and $I(H_g)$ is an imbedding of H_g of genus g, we first construct an NSE $I(H'_g)$ of $I(H_g)$ regarding G and then we find a WQPE of $I(H'_g)$ regarding G.

Theorem 3.1. Let $I(H_g)$ be an imbedding of a graph H_g of genus g. Then a NSE $I(H'_g)$ of $I(H_g)$ regarding G such that $[H'_g\text{-}H_g]=O(g)$ can be constructed in $|G|[H_g]2^{O(g)}$ time.

Proof. We apply a simple procedure that repeatedly ($O(g)$ times) chooses any splitting path and guesses an imbedding of the path. In particular, apply the following procedure to $I(H_g)$.

Algorithm 3.1

Input: biconnected graph G, a subgraph H_g of G and a genus g imbedding $I(H_g)$

Output: $I(H'_g)$, an NSE of $I(H_g)$ regarding G

Initially let $H'_g := H_g$ and $I(H'_g) := I(H_g)$.

repeat do

1. Pick any splitting path p in $G\text{-}H'_g$

If no such splitting path p exists, ***then*** output $I(H'_g)$; ***halt else***

2.$H'_g := H'_g+p$.

3. Guess (by sequencing through all possibilities) an imbedding of p and add it to $I(H'_g)$.

fi od

Denote $h=[H_g]$ and $s=S(I(H_g))$. To estimate the number of all possible ways to imbed p in Step 3 we notice that, since both endpoints of p are of degree at most g in any face of H'_g, there are no more than g possible choices of imbedding each end-edge of p in any face of H'_g, giving totally at most g^2 possible imbeddings. As p is splitting path and by Corollary 3.1 the number of faces that contain an internal path is $O(g)$ then the number of all possible imbeddings of p in $I(H'_g)$ is $O(g^3)$.

By Lemma 3.2, each Step 3 decreases the current value of $S(I(H'_g))$ by at least 1 or will not change $S(I(H'_g))$ but will increase the number of faces F with $S(F)>0$. All executions of Step 1 will require $O(|G|[H_g])$ time. Then, by the above arguments, the maximum time $T(h,s)$ required by Steps 2 and 3 of Algorithm 3.1 to make any imbedding with parameters $h = [H_g]$ and $s= S(I(H'_g))$ an NSE can be upper bounded by

$T(h,s) \leq O(g^3)T(h,s\text{-}1)+ O(|G|\, h) \leq O(|G|\, h)O(g^3)^s$.

Thus the total sequential time required by Algorithm 3.1 is $|G|[H_g]g^{O(g)}$. **Q.E.D.**

For notational simplicity, fix $H' = H'_g$. To simplify reasoning about an imbedding extension problem G, $I(H')$ we define a *representative* imbedding extension problem $R(G)$, $I(R(H'))$ as follows:

$R(H')$ is derived from H' by collapsing together (into a single vertex) on each maximal 2-path p, all the consecutive vertices $\hat{p}$ of degree 2 with respect to H'. (Note the result is *not* quite the homeomorphic contraction of H' since maximal 2-paths of length > 1 are collapsed to 2-paths of length 2, rather than 1).

For each bridge B of $G\text{-}H'$, $R(B)$ is a tree of depth 1(which may have multiple edges) derived from B by:

(i)collapsing together any attachment vertices of B to H' which were collapsed together in $R(H')$,

(ii)collapsing all non-attachment vertices of B into a single root vertex,

(iii)for each group of resulting multiple edges, delete all but 4g multiple edges in the group.

$R(G)$ is the union of $R(H')$ and $R(B)$ for all bridges B of $G\text{-}H'$. An imbedding $I(G)$ extending $I(H')$ induces a corresponding *consistent* imbedding $I(R(G))$ of the representative graph.

Two paths p, p' of R(G)-R(H) with the same attachment vertices are said to be *imbedded side-by-side* if they induce a new face consisting only of p and p'. Two simple bridges are *parallel* if they have the same attachment vertices. A bridge B of $G\text{-}H'$ is *simple* if $R(B)$ is a path with at most 2 edges, and otherwise *non-simple*.

Consider two simple (but not necessarily parallel) bridges B, B' of $G\text{-}H'$. These bridges are defined to be *similar* if $R(B)$, $R(B')$ have the same attachment vertices (note however that B, B' may not have the same attachment vertices). Furthermore, similar B, B' are said to be ***imbedded side-by-side*** if the imbedding $I(B)$, $I(B')$ induces an imbedding in the representative graph where the paths $I(R(B))$, $I(R(B'))$ are imbedded side-by-side. A ***bridge path*** is a path in a bridge between two of its attachments. (The above definitions naturally extend also to apply to sets of bridges and bridge paths in $G\text{-}H'$.)

Lemma 3.3. If G has an imbedding extension of $I(H')$ with the same genus, then G has an imbedding extension of $I(H')$ where each group of simple parallel bridges are imbedded side-by-side.

proof. Fix an imbedding $I(G)$ extending $I(H')$. Note that deleting any simple bridge B of $G\text{-}H'$ does not change the Euler characteristic. Also, inserting the simple bridge B side-by-side to another parallel bridge B' does not change the Euler characteristic. Thus the Lemma follows. **Q.E.D.**

This lemma implies that for the purpose of finding an imbedding, we can delete all but one of multiple parallel bridges.

Lemma 3.4. Let M be the set of bridges of $G\text{-}H'$ which are not 2-constrained. If G has an imbedding extension of $I(H')$ with the same genus, then in that imbedding $I(G)$, $O(g)$ upper bounds the total number of edges of $I(R(M))$, not counting multiple edges imbedded side-to-side.

proof. (Note this lemma also follows from the results of [Filotti, Miller, and Reif,79], [Reif,79].) Consider the bridges to be imbedded onto a give face F of $I(H')$. The imbedding of a non-simple bridge B onto F, where $R(B)$ has $k > 2$ edges, decreases $S(F)$ by at least k-2, since B is attached to at least k-2 vertices of F whose repetition number will decrease when B is imbedded. Furthermore, at most $4S(F)$ simple distinct non-similar bridges can be imbedded into F before $S(F)$ decreases by at least one half. Thus at total of at most $8S(F)$ simple non-similar bridges and at most $S(F)$ non-simple bridges can be imbedded into F. Hence the total number of bridges in M that can be imbedded onto F is $S(F)$. $S(I(H'))$ is by definition the sum of $S(F)$ for each face of H'. Thus by Lemma 3.1, M has at most $O(S(I(H')))\leq O(S(I(H)))= O(g)$ edges. **Q. E. D.**

Define a bridge or bridge path of $G\text{-}H'$ to be *internal* if it is incident to at least one internal vertex of $I(H')$.

Consider an imbedding I(G) extending I(H'). A bridge path p in I(G-H') is *extremal* if p is imbedded the first or last among all similar bridge paths imbedded side-by-side with p.

Lemma 3.4 implies:

Lemma 3.5 The number of distinct groups of similar internal 2-paths is $O(g)$, and furthermore, the number of extremal bridge paths in a genus g extension is $O(g)$. Thus, the number of extremal bridge paths in a toroidal (i.e. genus 1) imbedding extension is O(1).

We will use M to find an imbedding extension $I(H'')$ such that the remaining non-imbedded bridges of $G\text{-}H''$ are (1) 2-constrained or (2) are simple and have at most one attachment vertex of degree > 2 in H''. Our remaining problem now is to determine how exactly to imbed these latter bridges. For this, we will use a reduction to the toroidal imbedding problem.

Theorem 3.2. [Filotti, Miller, Reif,79], [Robertson and Seymour,86] Given a graph of n vertices, we can determine in polynomial time either a toroidal imbedding of G or a forbidden subgraph consisting of O(1) maximal 2-paths.

A *toroidal pair* of paths of $I(H'')$ is a pair, (p_1,p_2), of maximal internal 2-paths such that there exists a face $\{p_1\, p^{(1)}\, p_2\, p^{(2)}\, (p_1)^R\, p^{(3)}\, (p_2)^R\, p^{(4)}\}$ of H''_g (any of the paths $p^{(i)}$, i=1,...,4, and p_2 might be of length 0), where paths $\hat{p}^{(i)}$, i=1,2,3, contain no internal vertices (**Figure 3.3**). Let P_{tor} denote the set of all bridge paths q of $G\text{-}H''$ joining two internal vertices in H'' (where at least one of these internal vertices has degree 2, and with at least three non-splitting imbeddings of q in $I(H'')$).

Lemma 3.6. If $q \in P_{tor}$ then the endpoints of q belong to a toroidal pair of paths.

Proof: Let v_1 and v_2 be the endpoints of q. Then at least one of v_1 and v_2 are of degree 2 and both belong to two maximal internal 2-paths p_1 and p_2. There are two copies of v_1 (in p_1 or $(p_1)^R$) and of v_2 (in p_2 or $(p_2)^R$) which gives four possible imbeddings of q. W.l.o.g. assume that the face containing p_1 and p_2 be $\{p_1\, p^{(1)}\, p_2\, p^{(2)}\, (p_1)^R\, p^{(3)}\, (p_2)^R\, p^{(4)}\}$ of H'' , where paths $\hat{p}_i$, i=1,2,3, contain no internal vertices. If all $\hat{p}_i$, i=1,2,3,4, contain no internal vertices, then the lemma obviously holds. Assume that $\hat{p}^{(4)}$ contains an internal vertex. We will show that $\hat{p}^{(i)}$, i=1,2,3, contain no internal vertex. Indeed, if $\hat{p}^{(1)}$ contains an internal vertex, then the two imbeddings of q with v_1 in p_1 are splitting; if $\hat{p}^{(2)}$ contains an internal vertex, then the imbedding of q with v_1 in p_1 and v_2 in p_2 and the imbedding with v_1 in $(p_1)^R$ and v_2 in $(p_2)^R$ are splitting; and if $\hat{p}^{(3)}$ contains an internal vertex, then the two imbeddings of q with v_2 in $(p_2)^R$ are splitting. Therefore $\hat{p}^{(i)}$, i=1,2,3, contain no internal vertex. **Q.E.D.**

Lemma 3.7. The total number of distinct toroidal pairs is $O(g)$

Proof: Since $S(I(H''))$ = $O(g)$, it suffices to prove:

Claim. If (q_1',q_2') and (q_1'',q_2'') are toroidal pairs of paths, then either $\{q_1',q_2'\}=\{q_1'',q_2''\}$, or q_i' and q_j'' share no internal edge for all $i,j \in \{1,2\}$.

Assume that two paths q_i' and q_j'' (say q_1' and q_1'') share an internal edge. As q_1' and q_1'' are maximal 2-paths, then $q_1'=q_1''$. Assume that $q_2' \neq q_2''$. Then by the definition of a toroidal pair of paths there exist faces F_1 and F_2 such that $F_1=\{q_1'\, p^{(1)}\, q_2'\, p^{(2)}\, (q_1')^R\, p^{(3)}\, (q_2')^R\, p^{(4)}\}$ and $F_2=\{q_1''\, q^{(1)}\, q_2''\, q^{(2)}\, (q_1'')^R\, q^{(3)}\, (q_2'')^R\, q^{(4)}\}$, where paths $\hat{p}^{(i)}$ and $\hat{q}^{(i)}$, i=1,2,3, contain no internal vertices. As $q_1'=q_1''$ and each internal path belongs to a single face, then $F_1 = F_2 = \{q^{(4)}\, (q_2'')^R\, q^{(3)}\, (q_1'')^R\, q^{(2)}\, q_2''\, q^{(1)}\, q_1''=q_1'\, p^{(1)}\, q_2'\, p^{(2)}\, (q_1')^R\, p^{(3)}\, (q_2')^R\, p^{(4)}\}$. This is a contradiction since $(q_1')^R$ belongs to a single face and $(q_1')^R = (q_1'')^R$. **Q.E.D.**

Given a toroidal pair (p_1,p_2) of paths, let $H_{p_1p_2}$ be the subgraph of G induced by the bridges attached to p_1 or p_2 . Fix a representative imbedding I(R(M)), where M be the set of bridges of G-H' which are not 2-constrained. To constrain an imbedding extension $I(H''_g+H_{p_1p_2})$ of $H''_g+H_{p_1p_2}$ in $I(H''_g)$ to be consistent with $I(R(M))$, we can introduce and imbed new auxiliary edges between endpoint vertices of p_1,p_2. Lemma 3.6 implies that O(1) such edges will suffice. These edges will force the imbedding extension to be consistent with $I(R(M))$. Then apply the results of [Filotti, Miller, Reif,79], [Robertson and Seymour,86] (Alternatively, a simpler toroidal imbedding algorithm can also be used here.) in the case of genus 1, giving

Lemma 3.8. In polynomial time we can construct an imbedding extension $I(H''_g+H_{p_1p_2})$ of $H''_g+H_{p_1p_2}$ in $I(H''_g)$ consistent with $I(R(M))$ if this is possible, or otherwise, a minimal nonempty set FS of O(1) maximal 2-paths of $H_{p_1p_2}$ that can not be all imbedded onto $I(H''_g)$ consistent with $I(R(M))$ without increasing the genus.

Next we present an algorithm that extends an NSE $I(H'_g)$ of H'_g to a WQPE $I(H''_g)$ of a graph H''_g , $H'_g \leq H''_g \leq G$, $[H''_g - H'_g]=O(g)$, and then either extends the WQPE to an imbedding of G, if possible, or finds a minimal set R of bridges of G-H''_g that cannot be all imbedded onto $I(H''_g)$ without increasing the genus.

Algorithm 3.2

Input: A non-splittable imbedding $I(H'_g)$ of H'_g in G

Output: A subgraph H''_g of G, an imbedding $I(H''_g)$ of H''_g of genus g, such that $[H''_g - H'_g]=O(g)$, and either an imbedding of G of genus g, or a minimal nonempty set FS of O(g) maximal 2-paths of G-H''_g that can not be all imbedded onto $I(H''_g)$ without increasing the genus.

1. Construct the representative graph imbedding problem $R(G)$, $I(R(H'_g))$ as described above.
2. Let M be the bridges of G-H' which are not 2-constrained.
3. Guess(by sequential enumeration) an imbedding $I(R(M))$ onto $I(H)$ where we delete from the imbedding all multiple edges imbedded side-by-side.

Comment: Note by Lemma 3.4, there are at most O(g)! distinct possibilities for these guesses.

4. ***For*** each non-simple bridge B of G-H'_g ***do*** Construct a minimal subgraph B' of B such that $I(R(B'))$ = $I(R(B))$. Delete from B' any edges to attachment vertices of degree < 3 in H'. Repeatedly delete any resulting dangling edges(with a degree 1 vertex) not connected to attachment vertices. ***If*** the resulting B'' consists of more than one maximal path ***then*** imbed B'' in $I(H'_g)$ consistent to the chosen imbedding of $I(R(B))$ ***fi od***

Comment: B' is implicit from the construction of $R(B)$ and is a tree except its leaves may be repeated attachment vertices. B'' is biconnected and contains paths only to attachment vertices of degree > 2 in H'. By Lemma 3.4, $[B''] = O(g)$.

Comment: The resulting problem $G, I(H''_g)$ is WQPE, except there may be simple bridges which are not 2-constrained and which are either (1)parallel or (2)have at most one attachment vertex of degree > 2 in H''_g.

5. ***For*** each toroidal pair (p_1,p_2) of paths ***do***

5.1. Attempt to construct an extension imbedding $I(H''_g+H_{p_1p_2})$ of $H''_g+H_{p_1p_2}$ in $I(H''_g)$ consistent with $R(I(M))$, where $H_{p_1p_2}$ is the subgraph of G induced by the bridges attached to p_1 or p_2

Comment: Apply Lemma 3.8.

5.2 ***If*** such an extension exists ***then do*** Find the extremal set ES of 2-paths defined from this toroidal extension imbedding; $H''_g=H''_g+ES$; $I(H''_g)=I(H''_g+ES)$ ***od***

Comment: By Lemma 3.5, $[ES] = O(1)$.

Else do let FS' be a minimal nonempty set of O(1) maximal 2-paths of $H_{p_1p_2}$ that can not be all imbedded onto $I(H''_g)$ consistent with $R(I(M))$ without increasing the genus ; $FS:=FS \cup \{FS'\}$ ***od fi od***

6. ***for*** each maximal group of parallel simple bridges ***do*** temporarily delete(these will be reinserted in step 8) all but one parallel bridge ***od***

Comment: The resulting partial imbeddding is WQPE.

7. To imbed the rest of G apply the algorithm from Section 4 and Appendix A which will either solve the arising 2-satisfiability problem or find a minimal set FS of bridges of G-H''_g that cannot be imbedded onto $I(H''_g)$ without increasing the genus.
8. Finally, reinsert the parallel bridges deleted in step 6 and imbed them side-by-side with the remaining bridge to which they are parallel.

Comment: This is justified by Lemma 3.3

The above Algorithm gives:

Theorem 3.3. Given a non-splittable imbedding $I(H'_g)$ of H'_g in G, Algorithm 3.2 finds in $O(g)!\, n^{O(1)}$ time a subgraph H''_g of G such that $H'_g \leq H''_g \leq G$ *and* $[H''_g - H'_g]=O(g)$, and either an imbedding of G of genus g, or an imbedding $I(H''_g)$ of H''_g of genus g and a minimal nonempty set FS of O(g) maximal 2-paths of G-H''_g that can not be all imbedded onto $I(H''_g)$ without increasing the genus.

4. WQPE Imbeddings: The Reduction to 2-Satisfiability

Let H be a subgraph of G with a fixed imbedding $I(H)$ and let B be a bridge of G-H. We call B *incident* to face F of $I(H)$ if all attachment vertices of B (with respect to subgraph H) are in F. Note that B may be incident to more than one face. Since we are concerned with WQPE imbeddings, we can assume throughout this section that all bridges are 2-constrained. Let bridges B, B' *interlace in* F (with respect to given imbedding of B and B' in F) if both these bridges are incident to F and furthermore either (i) F contains distinct vertices ordered u, u', v, v' where u, v are in B and u', v' are in B' (see **Figure 4.1**) or (ii) both B and B' are attached to the same 3 distinct vertices of F. Recall the definition of inside and outside of an imbedded directed cycle given in Section 1.2.

Proposition 4.1 If each of B, B' can be separately imbedded inside a face F, then B, B' can be simultaneously imbedded inside F iff B, B' do not interlace.

Let $I(H)$ be a weakly quasiplanar imbedding regarding G. Suppose that any bridge of G-H can be added to $I(H)$. We shall investigate the problem of determining when *all* bridges of G-H can be added to $I(H)$. We are going to show that if all bridges cannot be added to $I(H)$, we can choose a suitable small subset of bridges, *a* $forbidden_g$ *set of bridges*, that cannot be added to $I(H)$.

The *2-satisfiability problem* is to determine satisfiability of a boolean CNF formula with at most 2 literals per clause. While there are known linear time solutions to this problem, for completeness we give a particularly simple linear time algorithm in Appendix A.

First we define a *2-constraint formula* $K = K(I(H)|G)$ as follows: For every bridge B of G-H, and every face F that B is incident to, construct literals $v_{B,F}$ and $\tilde{v}_{B,F}$. Literal $\tilde{v}_{B,F}$ is called the *complement* of $v_{B,F}$ and vice versa. We add clauses ($v_{B,F}$ **or** $\tilde{v}_{B,F}$) and ((**not** $v_{B,F}$) **or** (**not** $\tilde{v}_{B,F}$)). If B can be imbedded in two different faces F and F', then we add a clause ($v_{B,F}$ **or** $v_{B,F'}$) and a clause ($\tilde{v}_{B,F}$ **or** $\tilde{v}_{B,F'}$). In order to introduce a uniform notation for all cases we will call the inside of F *side-1 of* F and the outside of F *side-2 of* F. If two different imbeddings of B in F exist we choose arbitrarily one of these to be called *side-1 of F imbedding*, to correspond to $v_{B,F}$, and the other, to be called *side-2 of F imbedding*, to correspond to $\tilde{v}_{B,F}$. $v_{B,F}$ will be called *fixed* literal if it can be imbedded only on one side of F (side-1 or side-2). Moreover it is said to be *fixed to true* if its attachment vertices are non-constrained. If its attachment vertices are constrained, it is fixed to ***true*** if the constraint implied the attachment is to be imbedded on side-1 of F (i.e. the internal path is directed in the same (opposite) direction as the directed face F and the constraint is to imbed the attachment to the right (left, respectively) of the internal path), or otherwise ***false***. (see **Figures 4.2 a and b**). If $v_{B,F}$ is fixed to a boolean value, then its complement $\tilde{v}_{B,F}$ is fixed to the complement boolean value. To complete the definition of K, for all interlacing bridges B and B' define a clause ($\tilde{v}_{B,F}$ **or** $\tilde{v}_{B',F}$) of K, where B and B' are incident to face F.

Definition 4.1. The 2-constraint formula $K = K(I(H)|G)$ is said to be *satisfiable*, if there exists an assignment of ***true*** and ***false*** to the literals of K such that each clause of K is satisfied.

From these definitions we easily obtain:

Lemma 4.1. $I(H)$ is extendible to G iff any bridge of G-H can be added to $I(H)$ and the 2-constraint formula $K = K(I(H)|G)$ is satisfiable.

Proof. => Assume that K is not satisfiable. Then one of the Conditions (1) and (2) from Theorem A.1 is not satisfied. Suppose that Condition (1) is not satisfied. Then for some similarly fixed literals $v_{B,F}$ and $v_{B',F'}$ there exists an implication path p in K between literals $v_{B,F}$ and $\tilde{v}_{B',F'}$. W.l.o.g. we assume that the literals $v_{B,F}$ and $v_{B',F'}$ are fixed to value **true** and p does not contain other fixed literals or their complements. Let $p=(v_{B,F}=v_{B_0,F_0}, v_{B_1,F_1}, \ldots, v_{B_{j-1},F_{j-1}}, \tilde{v}_{B',F'})$. Suppose that we try to construct an extension of H by imbedding the bridges $B_0, B_1, \ldots, B_j$. As $v_{B,F}$ is fixed, then B is incident only to F and should be imbedded on side-1 of F in any extension of $I(H)$ in G. If $I(H)$ is extendable to G, then in any such extension B_0 is imbedded on side-1 of F_0, B_1 is imbedded on side-1 of $F_1, \ldots, B_{j-1}$ is imbedded on side-1 of F_{j-1}, B' is imbedded side-2 of F'. But the latter is impossible as $v_{B',F'}$ is fixed to value **true** and therefore B' can be imbedded only on side-1 of F'.

Suppose Condition (2) of Theorem A.1 is not satisfied. Then there exists an implication path $p=(v_{B_0,F_0}, v_{B_1,F_1}, \ldots, v_{B_j,F_j} = \tilde{v}_{B_0,F_0})$ in K. Suppose that we try to construct an extension of H by imbedding the bridges $B_0, B_1, \ldots, B_j$. Then B_0 has to be imbedded either on side-1 or side-2 of F_0. Assume that B_0 is imbedded on side-1 of F_0. Then, by the definitions of K and p, B_1 has to be imbedded on side-1 of F_1, B_2 has to be imbedded on side-1 of F_2, ..., and hence B_j has to be imbedded on side-1 of F_j. So B_0 has to be imbedded on side-2 of F_0. Then $I(H)$ is not extendable to G, a contradiction. Assume that B_0 is imbedded on side-2 of F_0. Then B_j has to be imbedded on side-1 of F_j, B_{j-1} has to be imbedded on side-1 of F_{j-1}, ... , B_0 has to be imbedded on side-1 of F_0, a contradiction with the assumption.

Thus in all cases the assumption that K is not satisfiable leads to a contradiction that $I(H)$ is extendable to G.

<= Assume that K is satisfiable. Assign **true** and **false** values to the literals of K by the algorithm of Appendix A. Intuitively, a **true** (**false**, respectively) value to $v_{B,F}$ will correspond to an imbedding of B on side-1 (on side-2, respectively) of F. The algorithm constructs a value assignment of a satisfiable 2-constraint formula $K = K(I(H)|G)$ in $O(|G|)$ time. This gives the body of the following algorithm:

Algorithm 4.1

Input $I(H)$, a weakly quasiplanar imbedding regarding G

Output value assignment on literals of a satisfiable 2-constraint formula $K=K(I(H)|G)$

We have assumed above that K is satisfiable and the literals of K have values assigned by the algorithm of Appendix A. We imbed the bridges of G according to these values: if $v_{B,F}$ is **true** (**false**, respectively) then imbed B on side-1 (side-2) of F respectively. This gives a simple procedure for extending our imbedding of H_g to include the bridges of G-H_g. We show that the resulting imbedding extension still has genus g. By Observation 4.1, 1) any bridge incident to only one side of a face should be embedded on that side. Then the value assignment obtained from Algorithm 4.1 allows each bridge to be embedded into its corresponding face provided that there is no conflict with the other bridges. For the sake of contradiction, suppose that bridges B and B' are assigned to be imbedded on side-1 of the same face F, but cannot in fact be simultaneously imbedded on side-1 of F. Then $v_{B,F}$ and $v_{B',F}$ have a **true** value and B and B' interlace. From our construction of the 2-constraint formula, there exists a clause in K of form ($\tilde{v}_{B,F}$ **or** $\tilde{v}_{B',F}$). This clause is not satisfied as both $\tilde{v}_{B,F}$ and $\tilde{v}_{B',F}$ have a **false** value. This contradiction shows that we have defined a valid imbedding of G extending H_g. **Q.E.D.**

By the definition of K and Proposition 4.1, it follows that

Observation 4.1. The values assigned by Algorithm 4.1 satisfy the following properties:

1) All fixed literals receive their defined value;

2) For any literal $v_{B,F}$ in K, the value of $\tilde{v}_{B,F}$ is equal to the negation of the value of $v_{B,F}$.

Assume that K is satisfiable and the literals of K have values assigned by Algorithm 4.1. We imbed the bridges of $G\text{-}H$ according to these values. This gives a simple procedure for extending our imbedding of H_g to include the bridges of $G\text{-}H_g$. In Section 5 we prove that the resulting imbedding extension still has genus g.

The results of this section can be used as a basis of the following algorithm for finding a forbidden$_g$ set C of bridges for $K(I(H)|G)$, if $I(H)$ is not extendible to G. Otherwise the algorithm extends $I(H)$ to an imbedding of G of the same genus.

Algorithm 4.2

Input: $I(H)$, a weakly quasiplanar imbedding regarding G

Output: Imbedding of G of the same genus as $I(H)$, if $I(H)$ is extendible to G, or a path p in K whose vertices correspond to a forbidden$_g$ set C of bridges

1. Construct the 2-constraint formula $K = K(I(H)|G)$.
2. Apply the algorithm of Appendix A

Step 1 can be implemented in $O(|V(G)|^2)$ time. The time required by the connectivity algorithm required by Appendix A is $O(|K|)$, if breadth-first search is used.

5 Finding Forbidden$_g$ Subgraphs using a Planarity Algorithm

For a given WQPE $I(H_g)$ we would like to find some subgraph FS of G with branchsize $[FS]=O(1)$ that cannot be added to $I(H_g)$, if the genus of H_g is less than the genus of G. For this end we could apply a known planarity testing algorithm [Hopcroft and Tarjan, 74] so to test in linear time whether $I(H_g)$ can be extended to an imbedding of the entire graph. When the extension is not possible, a modification of the algorithm will find a forbidden$_g$ subgraph of branchsize $O(1)$. In order to skip the lengthy description of that modified algorithm, we shall show here (using a somewhat less efficient, but polynomial time construction) that a forbidden$_g$ subgraph of $I(H_g)$ of branchsize $O(1)$ in $H_g + B_1+...+B_j$ (where $B_1,...,B_j$ are the bridges) can be found as well by applying the forbidden$_0$ subgraph algorithm $PT\text{-}FS$ as described below on an appropriate modification of $H_g + B_1+...+B_j$. Let $p = (v_{B_0,F_0}, v_{B_1,F_1},..., v_{B_j,F_j})$ be an implication path in $K(H_g|G)$ found by Algorithm 4.2. Notice that bridges B_{i-1} and B_i are interlacing for $1 \le i \le j$.

Lemma 5.1. There exists a face F of $I(H_g)$ and a minimal subpath $p' = (v_{B_{j'},F_{j'}}, ..., v_{B_{j''},F_{j''}})$ of p such that

(i) bridges B_{i-1} and B_i, $j'+1<i<j''$, are interlacing in F, and

(ii) bridges $B_{j'},...,B_{j''}$ cannot all be added to $I(H_g)$.

Proof: Consider the following cases (a) and (b).

(a) p is not a cycle. Suppose that p' is a subpath of p of minimum length that satisfies Condition (ii) of the lemma. We shall prove that p' contains no fixed literals except for its endpoints. As p is not a cycle and p is an implication path then either v_{B_0,F_0} and $\tilde{v}_{B_j,F_j}$ or $\tilde{v}_{B_0,F_0}$ and v_{B_j,F_j} are literals fixed to **true**. Let v_{B_0,F_0} and $\tilde{v}_{B_j,F_j}$ be the literals fixed to **true**. Suppose that p' contains a literal v_{B_i,F_i} fixed to **true** different from v_{B_0,F_0}. Then the path $(v_{B_i,F_i},..., \tilde{v}_{B_j,F_j})$ satisfies Condition (ii) of the lemma and is shorter than p'. A similar contradiction will arise in the case where p' contains a complement of a literal fixed to true different from $\tilde{v}_{B_0,F_0}$. Then p' contains no fixed literals except for its endpoints. Therefore there exists a face F such that any bridge B_i, $j'<i<j''$ is attached to F and thus bridges B_{i-1} and B_i, $j'+1<i<j''$, are interlacing in F.

(b) p is a cycle. Suppose that p contains a fixed (say fixed to **true**) literal v_{B_i,F_i} (otherwise the lemma follows directly). Then we can transform p into an implication path p^* of K, $p^* = (v_{B_i,F_i}, v_{B_{i+1},F_{i+1}},..., v_{B_j,F_j} = \tilde{v}_{B_0,F_0}, \tilde{v}_{B_1,F_1},..., \tilde{v}_{B_i,F_i})$. As the endpoints of p^* are fixed literals we can apply the proof from Case (a) to p^*. **Q.E.D.**

According to Lemma 5.1 there exists an implication path p' and a face F of $I(H_g)$ such that the bridge corresponding to each literal from p' is attached to F. We shall apply the $PT\text{-}FS$ algorithm on the subgraph J of H_g to find a forbidden$_g$ subgraph FS for $I(H_g)$ of branchsize $O(1)$, where J consists of F plus all bridges of p'.

Case 1. Let p' be a cycle. Then bridges $B_{j'}$ and $B_{j''}$ also interlace in F. Thus J is nonplanar and the $PT\text{-}FS$ algorithm yields a forbidden subgraph of $O(1)$ size.

Case 2. Let p' be not a cycle. Then its endpoints $v_{B_{j'},F_{j'}}$ and $v_{B_{j''},F_{j''}}$ will be fixed vertices. Construct an edge e (corresponding to H_g) that interlaces with p'. Define a graph $J'=J+e$ to which the proof of Case 1) can be applied. Remove e from the resulting forbidden$_g$ subgraph for $I(H_g)$ of branchsize $O(1)$.

To summarize the results of this section we state the following:

Theorem 5.1. If the genus of G is greater than g, then a forbidden$_g$ subgraph of branchsize $O(1)$ for extending a WQPE $I(H_g)$ in G can be found in $|G|^{O(1)}$ time.

6. Construction of a branchsize $[F_g]^{O(g)}$ forbidden$_g$ subgraph for G

Let $I_1, I_2, ..., I_s$ be the finite list of all possible distinct skeletal subimbeddings $I(H_g)$ in $SI_g(F_g,T)$. In the previous sections we showed how to construct a branchsize $O(g)$ forbidden$_g$ subgraph of G for extending $I(H_g)$, for each $I_j=I(H_g)$. Define U_g to be the union of these forbidden$_g$ graphs for I_j, $j = 1, ...,s$. Then any possible imbedding of U_g will clearly contain a forbidden$_g$ subgraph of G. The branchsize of U_g defined in this way could be in the worst case of the same order as the number of possible skeletal subimbeddings $I(H_g)$ in $SI_g(F_g,T)$. By the results of Section 2.2, this number is $([F_g]g)^{O(g)} \le [F_g]^{O(g)}$ since $[F_g]$ is at least g.

Lemma 6.1 Given the union U_g of the forbidden$_g$ subgraphs for all possible WQPE imbeddings $I_j(H_g)$ of H_g, for all possible skeletal subimbeddings H_g in $SI_g(F_g,T)$, then in time $[F_g]^{O(g)}$ we can construct a genus $g' > g$ biconnected subgraph $F_{g'}$ of G of branchsize $[F_g]^{O(g)}$.

This suffices for our theoretical results. It is possible in practice, however, to significantly reduce the maximum possible branchsize of a forbidden$_g$ subgraph of G by use of an equivalence relationship on elements of U_g, as shown below.

By the results of Section 5, the forbidden$_g$ subgraph for each I_j has branchsize $O(1)$ and so joins $O(1)$ vertices. Two subgraphs FS and FS' from U_g are ***equivalent*** if they satisfy the conditions:

(i) FS and FS' are homeomorphic, and

(ii) $(FS+F_g)$ and $(FS'+F_g)$ are homeomorphic.

Lemma 6.2 If FS, FS' are distinct but equivalent elements of U_g, then $U_g\text{-}FS'$ remains a set of forbidden$_g$ subgraphs of G.

Note by the algorithms of Section 3 and 4, that each subsequent F_{g+1} consists of F_g unioned with at most O(g) distinct 2-paths. Hence:

Lemma 6.3 From any fixed forbidden$_g$ subgraph F_g, the number of possible homeomorphic distinct subsequent forbidden$_{g+1}$ subgraphs F_{g+1} is at most $[F_g]^{O(g)}$

7. Bounds on Total Time and the Size and Number of Forbidden$_g$ Subgraphs

Lemma 7.1 $[F_g] \le \exp(O(g)!)$

Proof by induction. F_1 is known to have branchsize $O(1)$. For any $g \ge 1$, our inductive assumption is that $[F_g] \le \exp((cg)!)$, for some constant $c > 1$. The bound $[F_g]^{O(g)}$ on the branchsize of F_{g+1} proved in Lemma 6.1 implies that F_{g+1} is of branchsize $[F_g]^{c'g}$, for some constant $c' > 1$. Thus for $c > c'$, we have $[F_{g+1}] \le \exp((cg)!\ c'g) \le \exp((c(g+1))!)$. **Q.E.D.**

Our main Imbedding Algorithm 2.1 requires for each value of $g' = 0,1,...,g = \text{genus}(G)$, the construction of all possible skeletal subimbeddings $I(H_{g'})$ of $SI_{g'}(F_g,T)$, which can be computed by the Algorithm of Section 2.2 in deterministic time $(g[F_{g'}])^{O(g')} = \exp(O(g'!))$. For each possible skeletal subimbedding Algorithm 3.2 spends time (obtained by summing the times given in Theorems 3.1, 3.3 and 5.1) at most $|G|[H_{g'}]g^{O(g')}+2^{O(g')}|G|^{O(1)}$. Then the time needed for each g' is

$$\exp(O(g'!))\ (\ |G|[H_{g'}]2^{O(g')}+2^{O(g')}|G|^{O(1)}\ ) + [F_{g'}]^{O(g')},$$

where $[F_{g'}]^{O(g')}$ is the time given in Lemma 6.1. The sum of these enumeration times for $g' = 0,1,...,g=\text{genus}(G)$ gives the total deterministic time of our Imbedding Algorithm 2.1 as $\exp(O(g!))\ (\ |G|[H_g]2^{O(g)}+2^{O(g)}n^{O(1)}+|G|^{O(1)}\ ) = \exp(O(g!))\ |G|^{O(1)}$. This completes our analysis of the time complexity of the algorithm and implies our main result:

Theorem 7.1 Given an input graph G of genus g, our Imbedding Algorithm 2.1 takes time $\exp(O(g)!)\ |G|^{O(1)}$. Furthermore if $g > 0$, our algorithm yields a forbidden$_{g-1}$ subgraph F_g of branchsize $[F_g] \le \exp(O(g!))$.

A simple parallel implementation of our algorithm gives:

Theorem 7.2 Given an input graph G of genus g, the genus g imbedding problem takes parallel time $O(g!) + (\log n)^{O(1)}$ using $\exp(O(g!))\ n^{O(1)}$ processors.

Note that there are at most b^b homeomorphic distinct graphs of branchsize $b \le \exp(O(g!))$.

This immediately implies that the number of homeomorphic distinct forbidden$_g$ subgraphs for any graph imbedding onto a surface of genus g, is at most a triple exponential function of g. However, we can get an even smaller bound.

Theorem 7.3 The number $F(g)$ of homeomorphic distinct forbidden$_g$ subgraphs for graph imbeddings onto a surface of genus g, is at most $\exp(O(g!))$.

Proof by induction. The number of homeomorphic distinct forbidden$_0$ subgraphs is $F(0) = 2$. For any $g \ge 0$, our inductive assumption is that

$$F(g) \le \exp((dg)!),\ \text{for some constant } d > 1.$$

By Lemma 6.3, from any fixed forbidden$_g$ subgraph F_g, the number of possible homeomorphic distinct subsequent forbidden$_{g+1}$ subgraphs F_{g+1} is at most $[F_g]^{d'g}$ for some constant $d' > 1$. The bound of Lemma 7.1 on the branchsize $[F_g] \le \exp(c(g!)$,for some constant $c > 1$ immediately implies that

$$F(g+1) \le F(g) \max([F_g])^{d'g} \le F(g) \exp((c\ (g)!)^{d'g}.$$

Thus, for sufficiently large d, we have

$F(g+1) \le \exp((dg)!) \exp(c\ (g)!\ \ d'g) \le \exp((d(g+1))!))$. **Q.E.D.**

8. Conclusion

It would be of great interest to provide lower bounds on the number of forbidden subgraphs as a function of g.

Acknowledgements

We gratefully acknowledge the careful reading, insightful comments and editorial suggestions made Deganit Armon.

References

Alpert, S.R. The genera of amalgamations of graphs. *Trans. Amer. Math. Soc.*, 178 (1973) 1-39.

Alpert, S.R., and Gross J. L. Graph Imbedding Problems, in *Research Problems* (Richard Guy, ed.).

Battle, J., Harary, F., Kodama, Y., and Youngs, J.W.T. "Additivity of the Genus of a Graph". *Bull. Amer. Math. Soc. 68* (1962), 565-568.

Bender, E.A., and Richmond, L.B. 3-Edge-Connected Embeddings Have Few Singular Edges, *J. Graph Theory*, 14(1990), 475-477.

Djidjev, H.N., A linear algorithm for partitioning graphs of fixed genus, Serdica Bulgaricae mathematicae publications, Vol 11, (1985), 369-387.

Edmonds, J. A combinatorial representation for polyhedral surfaces. *Not. Am. Math. Soc.* 7 (1960), 646.

Fellows, M. R., and Langston, M.A, Nonconstructive Advances in Polynomial-Time Complexity, *Info. Proc. Letters* 26, (1987), 157-162.

Fellows, M. R., and Langston, M.A, Nonconstructive Tools for Proving Polynomial-Time Decidability", J of the ACM 35, (1988) 727-739.

Filotti, I.S. An algorithm for imbedding cubic graphs in the torus, *J. Comput. Syst. Sci.* 20 (1980), 255-276.

Filotti, I.S., Miller, G.L., and Reif, J.H. On determining the genus of a graph in $0(v^{0(g)})$ steps. In *Proceedings of the 11th ACM Symposium on Theory of Computing*, Atlanta, Ga., April, 1979, pp. 27-37.

Furst, Merrick L., Gross, Jonathan L., and McGeoch, Lyle A. Finding a Maximum-Genus Graph Imbedding, *JACM*, Vol 35, No. 3, July 1988, pp. 523-534

Gross, J.L., and Rosen, R.H. A linear time planarity algorithm for 2-complexes. *J. ACM* 26, 4 (Oct. 1979), 611-617.

Gross, J.L., and Tucker, T.W. *Topological Graph Theory*. Wiley Interscience, New York, 1987.

Hopcroft, J., and Tarjan, R. Efficient planarity testing. *J. ACM 21*, (Oct. 1974), 549-568.

Khuller, S., S.G. Mitchell, and V.V. Vazirani, "Processor efficient parallel algorithms for the two disjoint paths problem, and for finding a Kuratowski homeomorph", 30th Annual Symposium on Foundations of Computer Sceince", Durham, NC, Oct, 1989, p 300-305

Klein, P. and J.H. Reif, An Efficient parallel algorithm for Planarity", J. Comp. Sys. Sci., vol 37, 1988.

Kruskal, J. "Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture", *Trans. Amer. Math. Soc. 95* (1960), 210-225.

Kuratowski, "Sur le problem des courbes gauches en topologie", *Fund. Math. 15* (1930), 271-283.

Massey, W.S. *Algebraic Topology: An Introduction,* Harcourt, Brace, and World, New York, 1967.

Miller, G. L. "Isomorphism Testing of Graphs of Bounded Genus. *Proceedings of 12th Annual ACM Symposium on Theory of Computing*, ACM, Apr. 1980, 225-235.

ibid, *Lecture Notes in Computer Science.* Vol. 158: Testing and Canonical Forms for k-Contractible Graphs (A Generalization of Bounded Valence and Bounded Genus). In *Foundations of Computation Theory,* Springer-Verlag, N.Y., 1983, 310-327.

ibid, "Isomorphism of k-contractible Graphs (a Generalization of Bounded Valence and Bounded Genus)". *Information & Control 65,* 1-2 (Jan/Feb 1983), 1-20.

ibid, An Additivity Theorem for the Genus of a Graph, MIT, TR-85-342.

Ramachandran, V. and J.H. Reif, An optimal parallel algorithm for planarity, 30th Annual Symposium on Foundations of Computer Sceince", Durham, NC, Oct 1989.

Reif, J. On the Complexity of Extending Partial Imbeddings, Computer Science Dept, Univ Rochester, Technical Report TR33, Oct 1978.

ibid, Polynomial Time Recognition of Graphs of Fixed Genus, Computer Science Dept, Univ Rochester, Technical Report TR33, Oct 1978.

Robertson, N., and Seymour, P.D. "Graph minors. I. Excluding a forest", *J. Combinatorial Theory* (Ser. B), 39-61.

ibid, "Graph minors. II. Algorithmic aspects of tree-width", *J. Algorithms,* 7 (1986), 309-322.

ibid, "Graph minors. III. Planar tree-width", *J. Combinatorial Theory* (Ser. B), 49-64.

ibid, "Graph minors. IV. Tree-width and well-quasi-ordering". submitted.

ibid, "Graph minors. V.. Excluding a Planar Graph, *J. Combinatorial Theory* (Ser. B), (1986) 92-114.

ibid, "Graph Minors. VI. Disjoint paths across a disc", preprint, Sept 1986

ibid, "Graph Minors. XII. Disjoint paths on a surface", preprint, Sept 1986

ibid, "Graph minors. VIII. A Kuratowski theorem for general surfaces", preprint, Sept 1986

ibid, "Graph Minors. XIII. The Disjoint paths problem", preprint, Sept 1986

ibid, "Graph Minors. XVI. Wagner's Conjecture", preprint, to appear

ibid, "Graph Minors - a Survey", in *Surveys in Combinatorics* (I. Anderson, ed.), *Cambridge Univ. Press* (1985), 153-171.

Thomassen, "The graph genus problem is NP-complete", *J. of Algorithms*, 10(1989), 568-576.

Tutte, W.T. "Combinatorial Oriented Maps". *Can. J. Math. 31,* 5 (1979), 986-1004.

Wagner, K. "Uber Einer Eigenshaft der Ebener Complexe", *Math. Ann.*14 (1937), 570-590.

White, A.T. *Graphs, Groups, and Surfaces, North-Holland,* Amsterdam, 1973.

Youngs, J.W.T. Minimal imbeddings and the genus of a graph, *J. Math. Mech.*, 12 (1963) 303-315. MR 26 #3043.

Appendix A The 2-Satisfiability Problem

Let F be a conjunction of 2-literal clauses each of form (L_{J_i} **or** L_{K_i}). Let G be a graph with vertex set {*true*, *false*, all literals and their complements}.

For each clause with just one literal L let there be edge from *true* to L and from (**not** L) to *false*. For each clause with two distinct literals (L_{J_i} *or* L_{K_i}) add an edge from (**not** L_{J_i}) to L_{K_i} and an edge from (**not** L_{K_i}) to L_{J_i} .Thus G has a directed edge for each logical implication resulting from a single clause of F.

Theorem A.1. F is unsatisfiable iff
(1) there is a path from *true* to *false* or
(2) there is a literal L such that there is a cycle containing L and (**not** L).

Proof: Clearly, there is a logical implication from a literal L to another L' iff there is a path from L to L'. Thus if either of the latter cases (1) and (2) hold, the formula is not satisfiable.

If neither of the latter cases (1), (2) hold then we claim that we can assign to the literals **true** and **false** so as to satisfy the formula.This is done as follows:

(Initialize) First compute the strongly connected components of G and collapse each strong component into a single node. All the literals in such a node have implications to and from each other, so they must have the same truth value on any satisfying assignment. We will now label nodes in such a way so as to satisfy the formula (i.e., we will not violate any clause). Let S be the node *true* if it exists and otherwise let S be the empty set.

(Loop)

(a) If S is not empty, choose any node r from S. We label all nodes w reached from r with **true**, and label **false** any remaining nodes which contain the complement of any literal in each reachable node w.

(b) Choose if possible any node r' such that r' is any so far unlabeled node with entering edges (if any) only from nodes labelled **false**. Then we assign r' **false** and for each literal L in v, we add the node r containing (**not** L) to S and set r to **true**. Then we go back to the Loop.

(c) If there is no such r' to choose and S is empty, then we terminate.

Claim 1. All nodes are eventually labelled.

Claim 2. If the set S initially contains *true*, then we never reach node *false* in the first iteration, nor do we reach both a literal and its complement.
The second case follows from the assumption of condition 2.

Claim 3. We never reach from a root r chosen from S a node w already labeled **false** (even within a given iteration).
The above three claims imply that we satisfy the formula. **Q.E.D.**

Corollary A.1. Given a 2-satisfiability problem with formula F we can determine in $O(|F|)$ time if F is satisfiable. If F is satisfiable, we can find a satisfying assignment. If F is not satisfiable, then we can find in $O(|F|)$ time a sequence of implications satisfying cases (1) or (2) of Theorem A.1

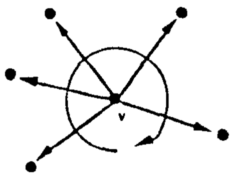

Figure 1.1
The cyclic order of directed edges around a vertex.

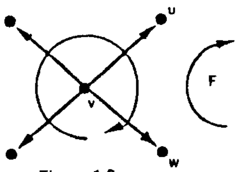

Figure 1.2

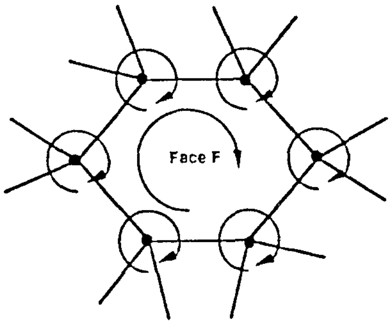

Figure 1.3
A face F defined by an orbit of the permutation of directed edges.

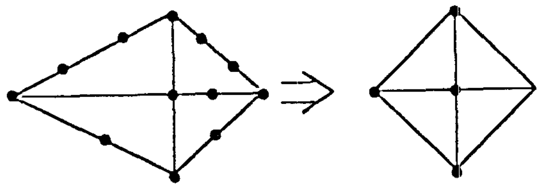

Figure 1.4
Homeomorphic contraction of a graph.

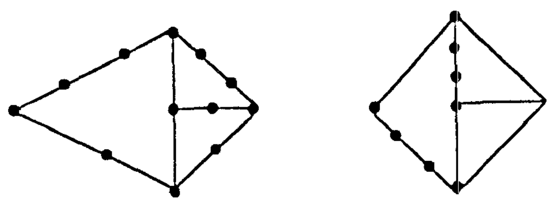

Figure 1.5
Homeomorphic graphs.

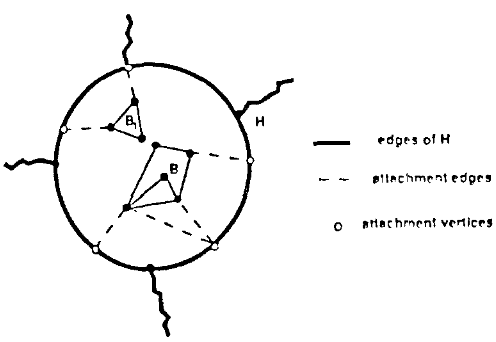

Figure 2.1. Bridges B and B1

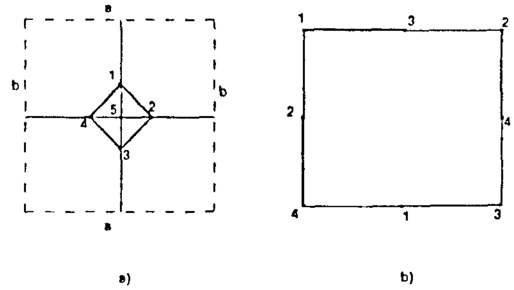

Figure 3.1. Internal vertices and edges
a) An imbedding of $K5$ on the torus
b) The outer face of the imbedding in a). Bold edges and all vertices are internal.

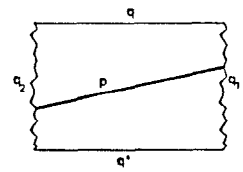

Figure 3.2.

A splitting imbedded path p

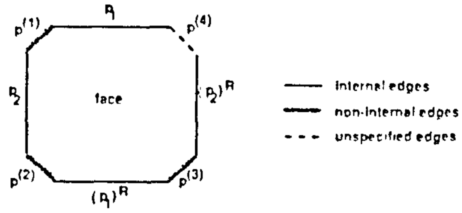

Figure 3.3. A toroidal pair of paths

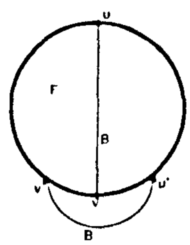

Figure 4.1. Interlacing bridges B and B'

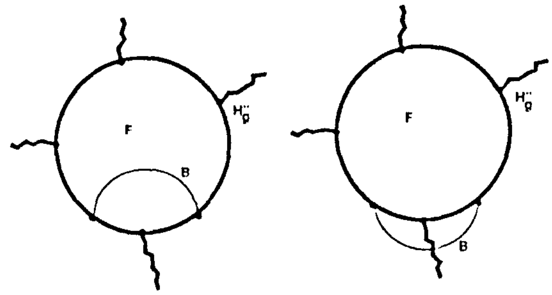

Figure 4.2. $v_{B,F}$ is fixed a) to true; b) to false