Finding Shortest Non-separating and Non-contractible Cycles for Topologically Embedded Graphs

Sergio Cabello[1,*] and Bojan Mohar[2,**]

[1] Department of Mathematics, Institute for Mathematics, Physics and Mechanics, Slovenia
sergio.cabello@imfm.uni-lj.si

[2] Department of Mathematics, Faculty of Mathematics and Physics, University of Ljubljana, Slovenia
bojan.mohar@fmf.uni-lj.si

Abstract. We present an algorithm for finding shortest surface non-separating cycles in graphs with given edge-lengths that are embedded on surfaces. The time complexity is $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$, where V is the number of vertices in the graph and g is the genus of the surface. If $g = o(V^{1/3-\varepsilon})$, this represents a considerable improvement over previous results by Thomassen, and Erickson and Har-Peled. We also give algorithms to find a shortest non-contractible cycle in $O(g^{O(g)}V^{3/2})$ time, improving previous results for fixed genus.

This result can be applied for computing the (non-separating) face-width of embedded graphs. Using similar ideas we provide the first near-linear running time algorithm for computing the face-width of a graph embedded on the projective plane, and an algorithm to find the face-width of embedded toroidal graphs in $O(V^{5/4}\log V)$ time.

1 Introduction

Cutting a surface for reducing its topological complexity is a common technique used in geometric computing and topological graph theory. Erickson and Har-Peled [9] discuss the relevance of cutting a surface to get a topological disk in computer graphics. Colin de Verdière [5] describes applications that algorithmical problems involving curves on topological surfaces have in other fields.

Many results in topological graph theory rely on the concept of face-width, sometimes called representativity, which is a parameter that quantifies local planarity and density of embeddings. The face-width is closely related to the edge-width, the minimum number of vertices of any shortest non-contractible cycle of an embedded graph [17]. Among some relevant applications, face-width

* Partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship.

** Supported in part by the Ministry of Higher Education, Science and Technology of Slovenia, Research Project L1–5014–0101–04 and Research Program P1–0297.

G.S. Brodal and S. Leonardi (Eds.): ESA 2005, LNCS 3669, pp. 131–142, 2005.

plays a fundamental role in the graph minors theory of Robertson and Seymour, and large face-width implies that there exists a collection of cycles that are far apart from each other, and after cutting along them, a planar graph is obtained. By doing so, many computational problems for locally planar graphs on general surfaces can be reduced to corresponding problems on planar graphs. See [17, Chapter 5] for further details. The efficiency of algorithmical counterparts of several of these results passes through the efficient computation of face-width.

The same can be said for the non-separating counterparts of the width parameters, where the surface non-separating (i.e., nonzero-homologous) cycles are considered instead of non-contractible ones. In this work, we focus on what may be considered the most natural problem for graphs embedded on surfaces: finding a shortest non-contractible and a shortest surface non-separating cycle. Our results give polynomial-time improvements over previous algorithms for low-genus embeddings of graphs (in the non-separating case) or for embeddings of graphs in a fixed surface (in the non-contractible case). In particular, we improve previous algorithms for computing the face-width and the edge-width of embedded graphs. In our approach, we reduce the problem to that of computing the distance between a few pairs of vertices, what some authors have called the *k-pairs shortest path problem.*

1.1 Overview of the Results

Let G be a graph with V vertices and E edges embedded on a (possibly non-orientable) surface Σ of genus g, and with positive weights on the edges, representing edge-lengths. Our main contributions are the following:

- We find a shortest surface non-separating cycle of G in $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$ time, or $O(g^{3/2}V^{3/2})$ if $g = O(V^{1-\varepsilon})$ for some constant $\varepsilon > 0$. This result relies on a characterization of the surface non-separating cycles given in Section 3. The algorithmical implications of this characterization are described in Section 4.
- For any fixed surface, we find a shortest non-contractible cycle in $O(V^{3/2})$ time. This is achieved by considering a small portion of the universal cover. See Section 5.
- We compute the non-separating face-width and edge-width of G in $O(g^{3/2}V^{3/2} + g^{5/2}V^{1/2})$ time. For fixed surfaces, we can also compute the face-width and edge-width of G in $O(V^{3/2})$ time. For graphs embedded on the projective plane or the torus we can compute the face-width in near-linear or $O(V^{5/4}\log V)$ time, respectively. This is described in Section 6.

Although the general approach is common in all our results, the details are quite different for each case. The overview of the technique is as follows. We find a set of generators either for the first homology group (in the non-separating case) or the fundamental group (in the non-contractible case) that is made of a few geodesic paths. It is then possible to show that shortest cycles we are interested in (non-separating or non-contractible ones) intersect these generators according to certain patterns, and this allows us to reduce the problem to computing distances between pairs of vertices in associated graphs.

We next describe the most relevant related work, and in Section 2 we introduce the basic background. The rest of the sections are as described above.

1.2 Related Previous Work

Thomassen [20] was the first to give a polynomial time algorithm for finding a shortest non-separating and a shortest non-contractible cycle in a graph on a surface; see also [17, Chapter 4]. Although Thomassen does not claim any specific running time, his algorithm tries a quadratic number of cycles, and for each one it has to decide if it is non-separating or non-contractible. This yields a rough estimate $O(V(V+g)^2)$ for its running time. More generally, his algorithm can be used for computing in polynomial time a shortest cycle in any class $\mathcal{C}$ of cycles that satisfy the so-called 3-*path-condition*: if u, v are vertices of G and P_1, P_2, P_3 are internally disjoint paths joining u and v, and if two of the three cycles $C_{i,j} = P_i \cup P_j$ $(i \neq j)$ are not in $\mathcal{C}$, then also the third one is not in $\mathcal{C}$. The class of one-sided cycles for embedded graphs is another relevant family of cycles that satisfy the 3-path-condition.

Erickson and Har-Peled [9] considered the problem of computing a planarizing subgraph of minimum length, that is, a subgraph $C \subseteq G$ of minimum length such that $\Sigma \setminus C$ is a topological disk. They show that the problem is NP-hard when genus is not fixed, provide a polynomial time algorithm for fixed surfaces, and provide efficient approximation algorithms. More relevant for our work, they show that a shortest non-contractible (resp. non-separating) loop through a fixed vertex can be computed in $O(V \log V + g)$ (resp. $O((V+g)\log V)$) time, and therefore a shortest non-contractible (resp. non-separating) cycle can be computed in $O(V^2 \log V + Vg)$ (resp. $O(V(V+g)\log V)$) time. They also provide an algorithm that in $O(g(V+g)\log V)$ time finds a non-separating (or non-contractible) cycle whose length is at most twice the length of a shortest one.

Several other algorithmical problems for graphs embedded on surfaces have been considered. Colin de Verdière and Lazarus [6,7] considered the problem of finding a shortest cycle in a given homotopy class, a system of loops homotopic to a given one, and finding optimal pants decompositions. Eppstein [8] discusses how to use the tree-cotree partition for dynamically maintaining properties from an embedded graph under several operations. Very recently, Erickson and Whittlesey [10] present algorithms to determine a shortest set of loops generating the fundamental group. Other known results for curves embedded on topological surfaces include [2,3,16,21]; see also [18,19] and references therein.

2 Background

Topology. We consider *surfaces* Σ that are connected, compact, Hausdorff topological spaces in which each point has a neighborhood that is homeomorphic to $\mathbb{R}^2$; in particular, they do not have boundary. A *loop* is a continuous function of the circle S^1 in Σ. Two loops are *homotopic* if there is a continuous deformation of one onto the other, that is, if there is a continuous function from the cylinder $S^1 \times [0,1]$ to Σ such that each boundary of the cylinder is mapped to one of the loops. A loop is *contractible* if it is homotopic to a constant (a loop whose image is a single point); otherwise it is *non-contractible.* A loop is *surface separating* (or *zero-homologous*) if it can be expressed as the symmetric difference of boundaries of topological disks embedded in Σ; otherwise it is *non-separating.* In particular, any non-separating loop is a non-contractible loop. We refer to [13] and to [17, Chapter 4] for additional details.

Representation of Embedded Graphs. We will assume the Heffter-Edmonds-Ringel representation of embedded graphs: it is enough to specify for each vertex v the circular ordering of the edges emanating from v and for each edge $e \in E(G)$ its *signature* $\lambda(e) \in \{+1,-1\}$. The negative signature of e tells that the selected circular ordering around vertices changes from clockwise to anti-clockwise when passing from one end of the edge to the other. For orientable surfaces, all the signatures can be made positive, and there is no need to specify it. This representation uniquely determines the embedding of G, up to homeomorphism, and one can compute the set of facial walks in linear time.

Let V denote the number of vertices in G and let g be the (Eurler) genus of the surface Σ in which G is embedded. It follows from Euler's formula that G has $\Theta(V+g)$ edges. Asymptotically, we may consider $V+g$ as the measure of the size of the input.

We use the notation $G \between C$ for the surface obtained by cutting G along a cycle C. Each vertex $v \in C$ gives rise to two vertices v', v'' in $G \between C$. If C is a two-sided cycle, then it gives rise to two cycles C' and C'' in $G \between C$ whose vertices are $\{v' \mid v \in V(C)\}$ and $\{v'' \mid v \in V(C)\}$, respectively. If C is one-sided, then it gives rise to a cycle C' in $G \between C$ whose length is twice the length of C, in which each vertex v of C corresponds to two diagonally opposite vertices v', v'' on C'. The notation $G \between C$ naturally generalizes to $G \between \mathcal{C}$, where $\mathcal{C}$ is a set of cycles.

Distances in Graphs. In general, we consider simple graphs with *positive* edge-weights, that is, we have a function $w: E \to \mathbb{R}^+$ describing the length of the edges. In a graph G, a *walk* is a sequence of vertices such that any two consecutive vertices are connected by an edge in G; a *path* is a walk where all vertices are distinct; a *loop* is a walk where the first and last vertex are the same; a *cycle* is a loop without repeated vertices; a *segment* is a subwalk. The *length* of a walk is the sum of the weights of its edges, counted with multiplicity.

For two vertices $u, v \in V(G)$, the *distance* in G, denoted $d_G(u,v)$, is the minimum length of a path in G from u to v. A *shortest-path tree* from a vertex v is a tree T such that for any vertex u we have $d_G(v,u) = d_T(v,u)$; it can be computed in $O(V \log V + E) = O(V \log V + g)$ time [12], or in $O(V)$ time if $g = O(V^{1-\varepsilon})$ for any positive, fixed ε [14]. When all the edge-weights are equal to one, a breadth-first-search tree is a shortest-path tree.

We assume non-negative real edge-weights, and our algorithms run in the comparison based model of computation, that is, we only add and compare (sums of) edge weights. For integer weights and word-RAM model of computation, some logarithmic improvements may be possible.

Width of Embeddings. The *edge-width* $\mathtt{ew}(G)$ (*non-separating edge-width* $\mathtt{ew}_0(G)$) of a graph G embedded in a surface is defined as the minimum number of vertices in a non-contractible (resp. surface non-separating) cycle. The *face-width* $\mathtt{fw}(G)$ (*non-separating face-width* $\mathtt{fw}_0(G)$) is the smallest number k such that there exist facial walks $W_1, \ldots, W_k$ whose union contains a non-contractible (resp. surface non-separating) cycle.

2.1 k-Pairs Distance Problem

Consider the *k-pairs distance problem*: given a graph G with positive edge-weights and k pairs $(s_1, t_1), \ldots, (s_k, t_k)$ of vertices of G, compute the distances $d_G(s_i, t_i)$ for $i = 1, \ldots, k$. Djidjev [4] and Fakcharoenphol and Rao [11] (slightly improved by Klein [15] for non-negative edge-lengths) describe data structures for shortest path queries in planar graphs. We will need the following special case.

Lemma 1. *For a planar graph of order V, the k-pairs distance problem can be solved in $O(\min\{V^{3/2} + k\sqrt{V}, V \log^2 V + k\sqrt{V} \log^2 V\})$ time.*

For a graph G embedded on a surface of genus g, there exist a set $S \subset V(G)$ of size $O(\sqrt{gV})$ such that $G - S$ is planar. It can be computed in time linear in the size of the graph [8]. Since $G - S$ is planar, we can then use the previous lemma to get the following result.

Lemma 2. *The k-pairs distance problem can be solved in $O(\sqrt{gV}(V \log V + g + k))$ time, and in $O(\sqrt{gV}(V + k))$ time if $g = O(V^{1-\varepsilon})$ for some $\varepsilon > 0$.*

Proof. (Sketch) We compute in $O(V + g)$ time a vertex set $S \subset V(G)$ of size $O(\sqrt{gV})$ such that $G - S$ is a planar graph. Making a shortest path tree from each vertex $s \in S$, we compute all the values $d_G(s, v)$ for $s \in S, v \in V(G)$. We define the restricted distances $d_G^S(s_i, t_i) = \min_{s \in S}\{d_G(s_i, s) + d_G(s, t_i)\}$, and compute for each pair (s_i, t_i) the value $d_G^S(s_i, t_i)$

If s_i and t_i are in different connected components of $G - S$, it is clear that $d_G(s_i, t_i) = d_G^S(s_i, t_i)$. If s_i, t_i are in the same component G_j of $G - S$ we have $d_G(s_i, t_i) = \min\{d_{G_j}(s_i, t_i),\ d_G^S(s_i, t_i)\}$. We can compute $d_{G_j}(s_i, t_i)$ for all the pairs (s_i, t_i) in a component G_j using Lemma 1, and the lemma follows because each pair (s_i, t_i) is in one component. □

3 Separating vs. Non-separating Cycles

In this section we characterize the surface non-separating cycles using the concept of crossing. Let $Q = u_0 u_1 \ldots u_k u_0$ and $Q' = v_0 v_1 \ldots v_l v_0$ be cycles in the embedded graph G. If Q, Q' do not have any common edge, for each pair of common vertices $u_i = v_j$ we count a crossing if the edges $u_{i-1}u_i, u_i u_{i+1}$ of Q and the edges $v_{j-1}v_j, v_j v_{j+1}$ of Q' alternate in the local rotation around $u_i = v_j$; the resulting number is $cr(Q, Q')$. If Q, Q' are distinct and have a set of edges E' in common, then $cr(Q, Q')$ is the number of crossings after contracting G along

E'. If $Q = Q'$, then we define $cr(Q, Q') = 0$ if Q is two-sided, and $cr(Q, Q') = 1$ if Q is one-sided; we do this for consistency in later developments.

We introduce the concept of ($\mathbb{Z}_2$-)homology; see any textbook of algebraic topology for a comprehensive treatment. A set of edges E' is a *1-chain*; it is a *1-cycle* if each vertex has even degree in E'; in particular, every cycle in the graph is a 1-cycle, and also the symmetric difference of 1-cycles is a 1-cycle. The set of 1-cycles with the symmetric difference operation $+$ is an Abelian group, denoted by $\mathcal{C}_1(G)$. This group can also be viewed as a vector space over $\mathbb{Z}_2$ and is henceforth called the *cycle space* of the graph G. If f is a closed walk in G, the edges that appear an odd number of times in f form a 1-cycle. For convenience, we will denote the 1-cycle corresponding to f by the same symbol f.

Two 1-chains E_1, E_2 are *homologically equivalent* if there is a family of facial walks $f_1, \ldots, f_t$ of the embedded graph G such that $E_1 + f_1 + \cdots + f_t = E_2$. Being homologically equivalent is an equivalence relation compatible with the symmetric difference of sets. The 1-cycles that are homologically equivalent to the empty set, form a subgroup $\mathcal{B}_1(G)$ of $\mathcal{C}_1(G)$. The quotient group $H_1(G) = \mathcal{C}_1(G)/\mathcal{B}_1(G)$ is called the homology group of the embedded graph G.

A set L of 1-chains generates the homology group if for any loop l in G, there is a subset $L' \subset L$ such that l is homologically equivalent with $\sum_{l' \in L'} l'$. There are sets of generators consisting of g 1-chains. It is known that any generating set of the fundamental group is also a generating set of the homology group $H_1(G)$.

If $\mathcal{L} = \{L_1, \ldots, L_g\}$ is a set of 1-cycles that generate $H_1(G)$, then every L_i $(1 \leq i \leq g)$ contains a cycle Q_i such that the set $\mathcal{Q} = \{Q_1, \ldots, Q_g\}$ generates $H_1(G)$. This follows from the exchange property of bases of a vector space since $H_1(G)$ can also be viewed as a vector space over $\mathbb{Z}_2$.

A cycle in G is surface non-separating if and only if it is homologically equivalent to the empty set. We have the following characterization of non-separating cycles involving parity of crossing numbers.

Lemma 3. *Let $\mathcal{Q} = \{Q_1, \ldots, Q_g\}$ be a set of cycles that generate the homology group $H_1(G)$. A cycle Q in G is non-separating if and only if there is some cycle $Q_i \in \mathcal{Q}$ such that Q and Q_i cross an odd number of times, that is, $cr(Q, Q_i) \equiv 1 \pmod 2$.*

Proof. Let $f_0, \ldots, f_r$ be the 1-cycles that correspond to the facial walks. Then $f_0 = f_1 + \cdots + f_r$ and $\mathcal{Q} \cup \{f_1, \ldots, f_r\}$ is a generating set of $\mathcal{C}_1(G)$. If C is a 1-cycle, then $C = \sum_{j \in J} Q_j + \sum_{i \in I} f_i$. We define $cr_C(Q)$ as the modulo 2 value of

$$\sum_{j \in J} cr(Q, Q_j) + \sum_{i \in I} cr(Q, f_i) = \sum_{j \in J} cr(Q, Q_j) \mod 2.$$

It is easy to see that $cr_C : \mathcal{C}_1(G) \to \mathbb{Z}_2$ is a homomorphism. Since $cr(Q, f_i) = 0$ for every facial walk f_i, cr_C determines also a homomorphism $H_1(G) \to \mathbb{Z}_2$.

If Q is a surface separating cycle, then it corresponds to the trivial element of $H_1(G)$, so every homomorphism maps it to 0. In particular, for every j, $cr(Q, Q_j) = cr_{Q_j}(Q) = 0 \mod 2$.

Let Q be a non-separating cycle and consider $\tilde{G} = G \not\!\!\! \not\, Q$. Take a vertex $v \in Q$, which gives rise to two vertices $v', v'' \in \tilde{G}$. Since Q is non-separating, there is a simple path P in $\tilde{G}$ connecting v', v''. The path P is a loop in G (not necessarily a cycle) that crosses Q exactly once.

Since $\mathcal{Q}$ generates the homology group, there is a subset $\mathcal{Q}' \subset \mathcal{Q}$ such that the loop P and $\sum_{Q_i \in \mathcal{Q}'} Q_i$ are homological. But then $1 = cr_P(Q) = \sum_{Q_i \in \mathcal{Q}'} cr(P, Q_i) \mod 2$, which means that for some $Q_i \in \mathcal{Q}'$, it holds $cr(P, Q_i) \equiv 1 \pmod 2$. □

4 Shortest Non-separating Cycle

We use the tree-cotree decomposition for embedded graphs introduced by Eppstein [8]. Let T be a spanning tree of G rooted at $x \in V(G)$. For any edge $e = uv \in E(G) \setminus T$, we denote by $loop(T, e)$ the closed walk in G obtained by following the path in T from x to u, the edge uv, and the path in T from v to x; we use $cycle(T, e)$ for the cycle obtained by removing the repeated edges in $loop(T, e)$. A subset of edges $C \subseteq E(G)$ is a *cotree* of G if $C^* = \{e^* \in E(G^*) \mid e \in C\}$ is a spanning tree of the dual graph G^*. A *tree-cotree* partition of G is a triple (T, C, X) of disjoint subsets of $E(G)$ such that T forms a spanning tree of G, C is cotree of G, and $E(G) = T \cup C \cup X$. Euler's formula implies that if (T, C, X) is a tree-cotree partition, then $\{loop(T, e) \mid e \in X\}$ contains g loops and it generates the fundamental group of the surface; see, e.g., [8]. As a consequence, $\{cycle(T, e) \mid e \in X\}$ generates the homology group H_1.

Let T_x be a shortest-path tree from vertex $x \in V(G)$. Let us fix any tree-cotree partition (T_x, C_x, X_x), and let $\mathcal{Q}_x = \{cycle(T_x, e) \mid e \in X_x\}$. For a cycle $Q \in \mathcal{Q}_x$, let $\mathcal{Q}_Q$ be the set of cycles that cross Q an odd number of times. Since $\mathcal{Q}_x$ generates the homology group, Lemma 3 implies that $\bigcup_{Q \in \mathcal{Q}_x} \mathcal{Q}_Q$ is precisely the set of non-separating cycles. We will compute a shortest cycle in $\mathcal{Q}_Q$, for each $Q \in \mathcal{Q}_x$, and take the shortest cycle among all them; this will be a shortest non-separating cycle.

We next show how to compute a shortest cycle in $\mathcal{Q}_Q$ for $Q \in \mathcal{Q}_x$. Firstly, we use that T_x is a shortest-path tree to argue that we only need to consider cycles that intersect Q exactly once; a similar idea is used by Erickson and Har-Peled [9] for their 2-approximation algorithm. Secondly, we reduce the problem of finding a shortest cycle in $\mathcal{Q}_Q$ to an $O(V)$-pairs distance problem.

Lemma 4. *Among the shortest cycles in $\mathcal{Q}_Q$, where $Q \in \mathcal{Q}_x$, there is one that crosses Q exactly once.*

Proof. (Sketch) Let Q_0 be a *shortest* cycle in $\mathcal{Q}_Q$ for which the number $Int(Q, Q_0)$ of connected components of $Q \cap Q_0$ is minimum. We claim that $Int(Q, Q_0) \leq 2$, and therefore $cr(Q, Q_0) = 1$ because $\mathcal{Q}_Q$ is the set of cycles crossing Q an odd number of times, and each crossing is an intersection. Using the 3-path-condition and that the cycle Q is made of two shortest paths, it is not difficult to show that $Int(Q, Q_0) \geq 3$ cannot happen. □

Lemma 5. *For any $Q \in \mathcal{Q}_x$, we can compute a shortest cycle in $\mathcal{Q}_Q$ in $O((V \log V + g)\sqrt{gV})$ time, or $O(V\sqrt{gV})$ time if $g = O(V^{1-\varepsilon})$.*

Proof. Consider the graph $\tilde{G} = G \not\!\!\! \backslash\!\!\! \backslash Q$, which is embedded in a surface of Euler genus $g-1$ (if Q is a 1-sided curve in Σ) or $g-2$ (if Q is 2-sided). Each vertex v on Q gives rise to two copies v', v'' of v in $\tilde{G}$.

In G, a cycle that crosses Q exactly once (at vertex v, say) gives rise to a path in $\tilde{G}$ from v' to v'' (and vice versa). Therefore, finding a shortest cycle in $\mathcal{Q}_Q$ is equivalent to finding a shortest path in $\tilde{G}$ between pairs of the form (v', v'') with v on Q. In $\tilde{G}$, we have $O(V)$ pairs (v', v'') with v on Q, and using Lemma 2 we can find a closest pair (v_0', v_0'') in $O((V \log V + g)\sqrt{gV})$ time, or $O(V\sqrt{gV})$ if $g = O(V^{1-\varepsilon})$. We use a single source shortest path algorithm to find in $\tilde{G}$ a shortest path from v_0' to v_0'', and hence a shortest cycle in $\mathcal{Q}_Q$. □

Theorem 1. *Let G be a graph with V vertices embedded on a surface of genus g. We can find a shortest surface non-separating cycle in $O((gV \log V + g^2)\sqrt{gV})$ time, or $O((gV)^{3/2})$ time if $g = O(V^{1-\varepsilon})$.*

Proof. Since $\bigcup_{Q \in \mathcal{Q}_x} \mathcal{Q}_Q$ is precisely the set of non-separating cycles, we find a shortest non-separating cycle by using the previous lemma for each $Q \in \mathcal{Q}_x$, and taking the shortest among them. The running time follows because Q_x contains $O(g)$ loops. □

Observe that the algorithm by Erickson and Har-Peled [9] outperforms our result for $g = \Omega(V^{1/3} \log^{2/3} V)$. Therefore, we can recap concluding that a shortest non-separating cycle can be computed in $O(\min\{(gV)^{3/2}, V(V+g)\log V\})$ time.

5 Shortest Non-contractible Cycle

Like in the previous section, we consider a shortest-path tree T_x from vertex $x \in V(G)$, and we fix a tree-cotree partition (T_x, C_x, X_x). Consider the set of loops $L_x = \{loop(T_x, e) \mid e \in X_x\}$, which generates the fundamental group with base point x. By increasing the number of vertices to $O(gV)$, we can assume that L_x consists of cycles (instead of loops) whose pairwise intersection is x. This can be shown by slightly modifying G in such a way that L_x can be transformed without harm.

Lemma 6. *The problem is reduced to finding a shortest non-contractible cycle in an embedded graph $\tilde{G}$ of $O(gV)$ vertices with a given set of cycles $\mathcal{Q}_x$ such that: $\mathcal{Q}_x$ generates the fundamental group with basepoint x, the pairwise intersection of cycles from $\mathcal{Q}_x$ is only x, and each cycle from $\mathcal{Q}_x$ consists of two shortest paths from x plus an edge. This reduction can be done in $O(gV)$ time.*

Proof. (Sketch) The first goal is to change the graph G in such a way that the loops in L_x will all become cycles. Then we handle the pairwise intersections between them. The procedure is as follows. Consider a non-simple loop l_0 in

L_x whose repeated segment P_0 is shortest, and replace the vertices in P_0 in the graph as shown in Figure 1. We skip a detailed description since it involves much notation, but the idea should be clear from the figure. Under this transformation, the rest of loops (or cycles) in L_x remain the same except that their segment common with P_0 is replaced with the corresponding new segments. We repeat this procedure until L_x consists of only cycles; we need $O(g)$ repetitions. This achieves the first goal, and the second one can be achieved doing a similar transfomation if we consider at each step the pair of cycles that have a longest segment in common. □

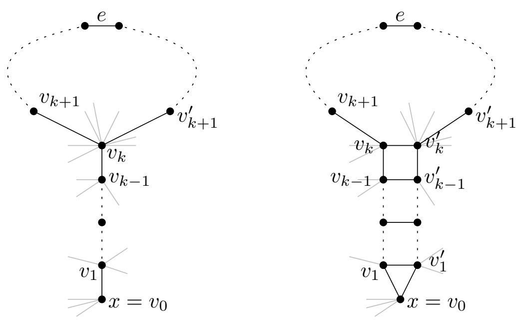

Fig. 1. Changing G such that a loop $l_0 \in L_x$ becomes a cycle. The edges $v_i v'_i$ have length 0.

Therefore, from now on, we only consider scenarios as stated in Lemma 6. Let $\mathcal{Q}^*$ be the set of shortest non-contractible cycles in $\tilde{G}$. Using arguments similar to Lemma 4, we can show the following.

Lemma 7. *There is a cycle $Q \in \mathcal{Q}^*$ that crosses each cycle in $\mathcal{Q}_x$ at most twice.*

Consider the set $D = \Sigma$ ✂ $\mathcal{Q}_x$ and the corresponding graph $G_P = \tilde{G}$ ✂ $\mathcal{Q}_x$. Since $\mathcal{Q}_x$ is a set of cycles that generate the fundamental group and they only intersect at x, it follows that D is a topological disk, and G_P is a planar graph. We can then use D and G_P as building blocks to construct a portion of the universal cover where a shortest non-contractible cycle has to lift.

Theorem 2. *Let G be a graph with V vertices embedded on a surface of genus g. We can find a shortest non-contractible cycle in $O(g^{O(g)} V^{3/2})$ time.*

Proof. According to Lemma 6, we assume that $\tilde{G}$ has $O(gV)$ vertices and we are given a set of cycles $\mathcal{Q}_x$ that generate the fundamental group with base point x, whose pairwise intersection is x, and such that each cycle of $\mathcal{Q}_x$ consists of two shortest paths plus an edge. Moreover, because of Lemma 7, there is a shortest non-contractible cycle crossing each cycle of $\mathcal{Q}_x$ at most twice.

Consider the topological disk $D = \Sigma$ ✂ $\mathcal{Q}_x$ and let U be the universal cover that is obtained by gluing copies of D along the cycles in $\mathcal{Q}_x$. Let G_U be the universal cover of the graph $\tilde{G}$ that is naturally embedded in U. The graph G_U is an infinite planar graph, unless Σ is the projective plane $\mathbb{P}^2$, in which case G_U is finite.

Let us fix a copy D_0 of D, and let U_0 be the portion of the universal cover U which is reachable from D_0 by visiting at most $2g$ different copies of D. Since each copy of D is adjacent to $2|\mathcal{Q}_x| \leq 2g$ copies of D, U_0 consists of $(2g)^{2g} = g^{O(g)}$ copies of D. The portion G_{U_0} of the graph G_U that is contained in U_0 can be constructed in $O(g^{O(g)} gV) = O(g^{O(g)} V)$ time. We assign to the edges in G_{U_0} the same weights they have in G.

A cycle is non-contractible if and only if its lift in U finishes in different copies of the same vertex. Each time that we pass from a copy of D to another copy we must intersect a cycle in $\mathcal{Q}_x$. Using the previous lemma, we conclude that there is a shortest non-contractible cycle whose lift intersects at most $2|\mathcal{Q}_x| = O(g)$ copies of D. That is, there exists a shortest non-contractible cycle in G whose lifting to U starts in D_0 and is contained G_{U_0}.

We can then find a shortest non-contractible cycle by computing, for each vertex $v \in D_0$, the distance in G_{U_0} from the vertex v to all the other copies of v that are in G_{U_0}. Each vertex $v \in D_0$ has $O(g^{O(g)})$ copies in G_{U_0}. Therefore, the problem reduces to computing the shortest distance in G_{U_0} between $O(g^{O(g)} V)$ pairs of vertices. Since G_{U_0} is a planar graph with $O(g^{O(g)} V)$ vertices, we can compute these distances using Lemma 1 in $O(g^{O(g)} V \sqrt{g^{O(g)} V}) = O(g^{O(g)} V^{3/2})$ time. $\square$

Observe that, for a fixed surface, the running time of the algorithm is $O(V^{3/2})$. However, for most values of g as a function of V (when $g \geq c \frac{\log V}{\log \log V}$ for a certain constant c), the near-quadratic time algorithm by Erickson and Har-Peled [9] is better.

6 Edge-Width and Face-Width

When edge-lengths are all equal to 1, shortest non-contractible and surface non-separating cycles determine combinatorial width parameters (cf. [17, Chapter 5]). Since their computation is of considerable interest in topological graph theory, it makes sense to consider this special case in more details.

6.1 Arbitrary Embedded Graphs

The (*non-separating*) *edge-width* $\mathtt{ew}(G)$ (and $\mathtt{ew}_0(G)$, respectively) of an embedded graph G is the minimum number of vertices in a non-contractible (surface non-separating) cycle, which can be computed by setting $w(e) = 1$ for all edges e in G and running the algorithms from previous sections. For computing the (*non-separating*) *face-width* $\mathtt{fw}(G)$ (and $\mathtt{fw}_0(G)$, respectively) of an embedded graph G, it is convenient to consider its vertex-face incidence graph Γ: a bipartite graph whose vertices are faces and vertices of G, and there is an edge between face f and vertex v if and only if v is on the face f. The construction of Γ takes linear time from an embedding of G, and it holds that $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma)$

and $\mathtt{fw}_0(G) = \frac{1}{2}\mathtt{ew}_0(\Gamma)$ [17]. In this setting, since a breadth-first-search tree is a shortest-path tree, a log factor can be shaved off.

Theorem 3. *For a graph G embedded in a surface of genus g, we can compute its non-separating edge-width and face-width in $O(g^{3/2}V^{3/2}+g^{5/2}V^{1/2})$ time and its edge-width and face-width in $O(g^{O(g)}V^{3/2})$ time.*

Although in general it can happen that $\mathtt{ew}(G) = \Omega(V)$, there are non-trivial bounds on the face-width $\mathtt{fw}(G)$. Albertson and Hutchinson [1] showed that the edge-width of a *triangulation* is at most $\sqrt{2V}$. Since the vertex-face graph Γ has a natural embedding in the same surface as G as a quadrangulation, we can add edges to it to obtain a triangulation T, and conclude that $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma) \leq \mathtt{ew}(T) \leq \sqrt{2V}$.

6.2 Face-Width in the Projective Plane and the Torus

For the special cases when G is embedded in the projective plane $\mathbb{P}^2$ or the torus $\mathbb{T}$, we can improve the running time for computing the face-width. The idea is to use an algorithm for computing the edge-width whose running time depends on the value $\mathtt{ew}(G)$. We only describe the technique for the projective plane.

Lemma 8. *Let G be a graph embedded in $\mathbb{P}^2$. If $\mathtt{ew}(G) \leq t$, then we can compute $\mathtt{ew}(G)$ and find a shortest non-contractible cycle in $O(V\log^2 V + t\sqrt{V}\log^2 V)$ time.*

Proof. (Sketch) Since the sphere is the universal cover of the projective plane $\mathbb{P}^2$, we can consider the cover of G on the sphere, the so-called double cover D_G of the embedding of G, which is a planar graph. Each vertex v of G gives rise to two copies v, v' in D_G, and a shortest non-contractible loop passing through a vertex $v \in V(G)$ is equivalent to a shortest path in D_G between the vertices v and v'.

We compute in $O(V\log V)$ time a non-contractible cycle Q of G of length at most $2\mathtt{ew}(G) \leq 2t$ [9]. Any non-contractible cycle in G has to intersect Q at some vertex, and therefore the problem reduces to find two copies $v, v' \in D_G$ of the same vertex $v \in Q$ that minimize their distance in d_G. This requires $|Q| \leq 2t$ pairs of distances in D_G, which can be solved using Lemma 1. □

Like before, consider the vertex-face incidence graph Γ which can be constructed in linear time. From the bounds in Section 6.1, we know that the edge-width of Γ is $O(\sqrt{V})$, and computing the face-width reduces to computing the edge-width of a graph knowing a priori that $\mathtt{ew}(\Gamma) = 2\mathtt{fw}(G) = O(\sqrt{V})$. Using the previous lemma we conclude the following.

Theorem 4. *If G is embedded in $\mathbb{P}^2$ we can find $\mathtt{fw}(G)$ in $O(V\log^2 V)$ time.*

For the torus, we have the following result, whose proof we omit.

Theorem 5. *If G is embedded in $\mathbb{T}$ we can find $\mathtt{fw}(G)$ in $O(V^{5/4}\log V)$ time.*

References

1. M. O. Albertson and J. P. Hutchinson. On the independence ratio of a graph. *J. Graph Theory*, 2:1–8, 1978.
2. T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Comput. Syst. Sci.*, 58:297–325, 1999. Preliminary version in FOCS'95.
3. T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete Comput. Geom.*, 14:93–110, 1995.
4. H. Djidjev. On-line algorithms for shortest path problems on planar digraphs. In *WG'96*, volume 1197 of *LNCS*, pages 151–165, 1997.
5. É. Colin de Verdière. *Shortening of curves and decomposition of surfaces.* PhD thesis, University Paris 7, December 2003.
6. É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. In *FOCS 2002*, pages 627–636, 2002. To appear in Discrete Comput. Geom.
7. É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. In *GD 2003*, volume 2912 of *LNCS*, 2004.
8. D. Eppstein. Dynamic generators of topologically embedded graphs. In *SODA 2003*, pages 599–608, 2003.
9. J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.*, 31:37–59, 2004. Preliminary version in SoCG'02.
10. J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *SODA 2005*, 2005.
11. J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. In *FOCS 2001*, pages 232–242, 2001.
12. M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34:596–615, 1987. Preliminary version in FOCS'84.
13. A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2001.
14. M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, 1997.
15. P. N. Klein. Multiple-source shortest paths in planar graphs. In *SODA 2005*, 2005.
16. F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *SOCG 2001*, pages 80–89, 2001.
17. B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, Baltimore, 2001.
18. A. Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *J. Combin. Theory Ser. B*, 51:127–159, 1991.
19. A. Schrijver. Paths in graphs and curves on surfaces. In *First European Congress of Mathematics, Vol. II*, volume 120 of *Progr. Math.*, pages 381–406. Birkhäuser, 1994.
20. C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Combin. Theory, Ser. B*, 48:155–177, 1990.
21. G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *SOCG 1990*, pages 102–111, 1990.