Finding Shortest Non-Separating and Non-Contractible Cycles for Topologically Embedded Graphs

Sergio Cabello[1,*] 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 embedded on surfaces in $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$ time, 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, which improves 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 [20]. Among some relevant applications, face-width 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,

*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 and Research Program P1–0297.

many computational problems for locally planar graphs on general surfaces can be reduced to corresponding problems on planar graphs. See [20, 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 4. The algorithmical implications of this characterization are described in Section 5.
- 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 6.
- 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. These are particular cases of the results mentioned in the previous paragraphs where a log factor can be shaved off. See Section 7.
- 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 Sections 7.2 and 7.3.

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.

The paper is organized as follows. The remaining of this section describes the most relevant related work, and in Section 2 we introduce the basic background. In Section 3 we describe results on the k-pairs distance problem that we use later on. The rest of the sections are as described above; we conclude in Section 8.

1.2 Related previous work

Thomassen [23] 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 [20, 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 that 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 [7] considered the problem of finding a shortest cycle in a given homotopy class, as well as a system of loops homotopic to a given one. Under some realistic assumption on the edge-lengths, they provide polynomial time algorithms for both problems. The same authors have also given polynomial time algorithms for finding optimal pants decompositions [6].

Eppstein [8] discusses how to use the tree-cotree partition for dynamically maintaining properties from a graph under several operations. For example, he can maintain the minimum and maximum spanning tree under edge insertions, edge deletions, and edge reweightings.

Very recently, Erickson and Whittlesey [10] have shown that the greedy homotopy generators through a fixed basepoint determine a shortest set of loops generating the fundamental group. They can compute this optimal system of loops and represent it implicitly in $O((V+g)\log V)$ time; an explicit representation may need $\Theta(gV)$ space. Other known results for curves embedded on topological surfaces include [2, 3, 18, 24]; see also [21, 22] and references therein.

2 Background

We describe the topological and graph-theoretical background assumed through the paper.

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, we only consider surfaces without 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, Chapter 1] and to [20, Chapter 4] for additional details.

Every graph, viewed as a 1-dimensional cell complex, determines a topological space and we can speak of embeddings (continuous 1-1 maps) in surfaces. It is customary to consider only *2-cell embeddings*, in which every *face* (i.e., a connected component of the surface after we remove the image of the graph) is homeomorphic to an open disk. Also in this paper we make such a restriction and, henceforth, every embedding will be 2-cell. As shown in [20, Chapter 3], such embeddings admit a simple combinatorial description whose development is usually attributed to Heffter, Edmonds, and Ringel. See also the next paragraph.

If G is a graph with V vertices, E edges, and is embedded in a surface Σ with F faces, then *Euler's formula* holds:

$$V - E + F = 2 - g$$

where g is a nonnegative integer, called the (*Euler*) *genus* of Σ.

If Σ is orientable, then $\bar{g} = \frac{1}{2}g$ is also an integer, known as the *genus* of Σ. Since we will meet the genus only in the O-notation, there will be no need to distinguish between g and $\bar{g}$.

Representation of embedded graphs. For computational purposes, an embedded graph can be represented as described by Eppstein [8]. However, for our purposes, the Heffter-Edmonds-Ringel representation will be used. It is enough to specify for each vertex v the circular ordering of the edges emanating from v, where the ordering coincides with that on the surface in a small disk neighbourhood of v. Additionally we need the *signature* $\lambda(e) \in \{+1, -1\}$ for each edge $e \in E(G)$. 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. If the embedding is in an orientable surface, all the signatures can be made positive, and there is no need to specify it. It is known that this representation uniquely determines the embedding of G, up to homeomorphism. Knowing the circular ordering at each vertex and the signatures, one can compute the set of facial walks in linear time. See [20, Chapter 3] if more detail is needed.

We use V to denote the number of vertices in G, and g for the genus of the surface Σ. That the graph G has $\Theta(V + g)$ edges follows from Euler's formula and the fact that the number of faces is at most $\frac{2}{3}E$. Asymptotically, we may consider $V + g$ as the measure of the size of the input. Observe that this is different from the approach followed by some authors, where $n = V + E$ is used for the size of the graph.

We use the notation $G /\!\!/ C$ for the graph obtained by cutting the embedded graph G along a cycle C. Each vertex $v \in C$ gives rise to two vertices v', v'' in $G /\!\!/ C$. If C is a two-sided cycle, then it gives rise to two cycles C' and C'' in $G /\!\!/ C$ 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 $G \not\!\!\!\! \times C$ 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'. Observe that if G is embedded in a surface Σ of Euler genus g, then $G \not\!\!\!\! \times C$ is naturally embedded in the surface obtained after cutting Σ along C and then pasting disks to each of the boundaries that were created. The notation $G \not\!\!\!\! \times C$ naturally generalizes to $G \not\!\!\!\! \times \mathcal{C}$, where $\mathcal{C}$ is a set of cycles.

Distances in graphs. In general, we consider simple graphs with *non-negative* edge-lengths, 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 if they occur on the walk more than once. Note that a shortest non-separating or non-contractible loop has to be a cycle.

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 $d_G(v, u) = d_T(v, u)$. Since $E = O(V + g)$, a shortest-path tree from any given vertex can be computed in $O(V \log V + E) = O(V \log V + g)$ time using Fibonacci heaps [12]. When $g = O(V^{1-\varepsilon})$ for any positive, fixed ε, then a shortest path tree can be constructed in $O(V)$ time[1].

In the special case that all the edge-lengths are equal to one, any breadth-first-search tree is a shortest-path tree from the starting vertex, and can be computed in $O(V + g)$ time.

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. Computing the (non-separating) face-width is equivalent to computing the (non-separating) edge-width in the so-called vertex-face incidence graph (see Section 7).

Model of computation We assume non-negative real edge-lengths, 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. See the survey by Zwick [25] for a discussion.

3 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 $i = 1, \ldots, k$.

[1]Eppstein [8] shows how to compute in linear time a separator S of size $O(\sqrt{gV}) = O(V^{1-\varepsilon/2})$ for G such that $G - S$ is planar. The recursive subdivision that Henzinger et al. [14] require can then be obtained using the division by Eppstein in the first level and then continue in each planar subpiece using their approach

Djidjev [4] and Fakcharoenphol and Rao [11] (slightly improved by Klein [17] 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*

(i) $O(V^{3/2} + k\sqrt{V})$ *time, and in*

(ii) $O(V \log^2 V + k\sqrt{V} \log^2 V)$ *time.*

Proof. For (i), use the data structure by Djidjev [4]: after $O(V^{3/2})$ time for preprocessing, a distance query can be answered in $O(\sqrt{V})$ time.

For (ii), use the data structure by Fakcharoenphol and Rao [11] with the quicker construction by Klein [17]: after $O(V \log^2 V)$ preprocessing time, a distance query can be answered in $O(\sqrt{V} \log^2 V)$ 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 obtain the following result.

Lemma 2 *Let G be a graph embedded on a surface of genus g. 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. 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 $s \in S, v \in V(G)$. Each shortest path tree takes $O(V \log V + V + g) = O(V \log V + g)$ time in general, and $O(V)$ time if $g = O(V^{1-\varepsilon})$. Therefore, we need $O(\sqrt{gV}(V \log V + g))$ time in total, or $O(\sqrt{gV}V)$ if $g = O(V^{1-\varepsilon})$.

We define the restricted distances

$$d_G^S(s_i, t_i) = \min_{s \in S}\{d_G(s_i, s) + d_G(s, t_i)\}.$$

We can compute for each pair (s_i, t_i) the value $d_G^S(s_i, t_i)$ in $O(\sqrt{gV})$ time, which is $O(k\sqrt{gV})$ time for all pairs. So far, we have spent $O(\sqrt{gV}(V \log V + g + k))$ time, or $O(\sqrt{gV}(V + k))$ if $g = O(V^{1-\varepsilon})$.

If s_i and t_i are in different components of $G - S$, it is clear that $d_G(s_i, t_i) = d_G^S(s_i, t_i)$. On the other hand, for a pair (s_i, t_i) in the same connected component G_j of $G - S$ we have

$$d_G(s_i, t_i) = \min\{d_{G_j}(s_i, t_i), d_G^S(s_i, t_i)\}.$$

Let $G_1, \ldots, G_t$ be the connected components of $G-S$. Let k_j be the number of pairs (s_i, t_i) that are in component G_j and let V_j be the number of vertices of the graph G_j. Because each G_j is planar, the values $d_{G_j}(s_i, t_i)$ for the k_j pairs in the component G_j can be computed in $O(V_j^{3/2} + k_j\sqrt{V_j})$ time as shown in Lemma 1. Since each pair (s_i, t_i) goes to at most one component of $G - S$, we have $\sum k_j \leq k$. Therefore, we can compute the distances using $O\big(\sum_j (V_j^{3/2} + k_j\sqrt{V_j})\big) \leq O\big(\sum_j V_j^{3/2} + \sqrt{V}\sum_j k_j\big) \leq O(V^{3/2} + \sqrt{V}k)$ time. This completes the proof. □

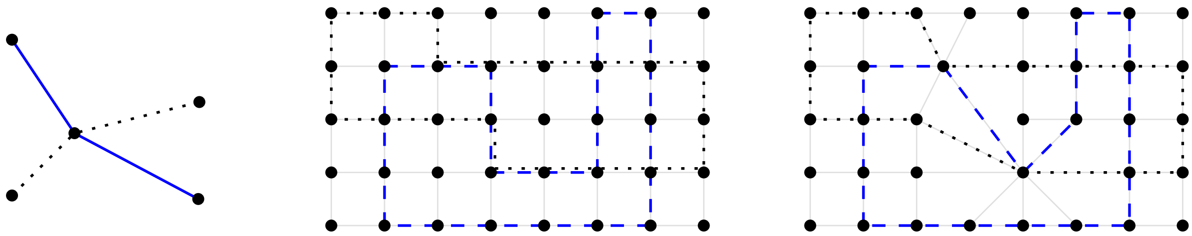

Figure 1: Left: a crossing without shared edges. Center and right: cycles with shared edges and 4 crossings; we count crossings after contracting the common edges.

4 Separating vs. non-separating cycles

In this section we characterize the surface non-separating cycles using the concept of crossing; see Figure 1. Let $Q = u_0u_1 \dots u_ku_0$ and $Q' = v_0v_1 \dots v_lv_0$ 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 $u_{i-1}u_i, u_iu_{i+1}$ of Q and the edges $v_{j-1}v_j, v_jv_{j+1}$ of Q' alternate in the local rotation around $u_i = v_j$ (where the indices are taken modulo $k+1$ and $l+1$, respectively); the number of all crossings is denoted by $cr(Q, Q')$. 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 on 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 $f_1, \dots, f_t$ of the embedded graph G such that $E_1 + f_1 + \dots + f_t = E_2$. 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 $\mathcal{L} = \{L_1, \dots, L_g\}$ is a set of 1-cycles that generate $H_1(G)$, then every L_i $(1 \le i \le g)$ contains a cycle Q_i such that the set $\mathcal{Q} = \{Q_1, \dots, Q_g\}$ 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 $\mathcal{Q} = \{Q_1, \dots, Q_g\}$ be a set of cycles that generate the homology group $H_1(G)$. A cycle Q in G is surface 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 $f_0, \dots, f_r$ be the 1-cycles that correspond to the facial walks. Then $f_0 = f_1 + \cdots + f_r$ and $\mathcal{Q} \cup \{f_1, \dots, f_r\}$ 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

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

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, $cr(Q, Q_j) \equiv cr_{Q_j}(Q) \equiv 0 \pmod 2$.

Let Q be a non-separating cycle and consider $\tilde{G} = G \backslash\!\!\backslash Q$. Take a vertex $v \in V(Q)$, which gives rise to two vertices $v', v'' \in V(\tilde{G})$. 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), but it contains a cycle Q' that crosses Q exactly once.

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

5 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. We describe the whole algorithm in pseudocode at the end of this section.

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. 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.

Let $e = u_1u_2$ be the edge for which $cycle(T_x, e) = Q$, and assume for contradiction that $Int(Q, Q_0) \geq 3$. Then, $Int(Q_0, P[x, u_1]) \geq 2$ or $Int(Q_0, P[x, u_2]) \geq 2$, where $P[u, u']$ denotes the path in T_x from u to u'. We may assume that $P[x, u_1] \cap Q_0$ has at least two connected components. Let v, v' be vertices from different components of $Q_0 \cap Q$ that are *consecutive* along $P[x, u_1]$, that is, it holds that $Q_0 \cap P[v, v'] = \{v, v'\}$. Consider the path $P[v, v']$ in T_x between the vertices v, v', and let P and P' be the segments of Q_0 between v, v'. Observe that by the way v, v' were chosen, $P[v, v']$ does not intersect $Q_0 \setminus \{v, v'\}$, and both the walk P concatenated with $P[v, v']$ and the walk P' concatenated with $P[v, v']$ are indeed cycles.

Let Q_0' be the cycle P concatenated with $P[v, v']$ and let Q_0'' be the cycle P' concatenated with $P[v, v']$. Observe that $length(P[v, v'])$ is smaller or equal to $length(P)$ and to $length(P')$ because $P[v, v']$ is a shortest path. Therefore, $length(Q_0')$ and $length(Q_0'')$ are both at most $length(Q_0)$. Moreover, $cr(Q, Q_0') + cr(Q, Q_0'') = cr(Q, Q_0) = 1 \mod 2$ and therefore it holds that $cr(Q, Q_0') = 1 \mod 2$ or $cr(Q, Q_0') = 1 \mod 2$; assume that $cr(Q, Q_0') = 1 \mod 2$. Then $Q_0' \in \mathcal{Q}_Q$. As shown above, $length(Q_0') \leq length(Q_0)$, and $Int(Q_0') < Int(Q_0)$, which is a contradiction. □

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 $\tilde{G} = G \not\!\!\!\!\times\, Q$, 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 6 *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. We summarize the algorithm that we have described.

Algorithm *Shortest Cycle Crossing Once*(G, Q)
Input: An embedded graph G, a cycle Q
Output: A shortest cycle in G that crosses Q once
$\tilde{G} \leftarrow G \not\!\!/\!\!/ Q$; $\Pi \leftarrow \{(v', v'') \in V(\tilde{G}) \times V(\tilde{G}) \mid v \in Q\}$;
find a pair $(v'_0, v''_0) \in \Pi$ such that $d_{\tilde{G}}(v'_0, v''_0) = \min_{(v', v'') \in \Pi} d_{\tilde{G}}(v', v'')$ using Lemma 2;
return a shortest path in $\tilde{G}$ from v'_0 to v''_0.

Algorithm *Shortest Surface Non-Separating Cycle*(G)
Input: An embedded graph G
Output: A shortest surface non-separating cycle
Fix $x \in V(G)$ and compute a shortest-path tree T_x;
Compute a tree-cotree decomposition (T_x, C_x, X_x);
$\mathcal{Q}_x \leftarrow \{cycle(T_x, e) \mid e \in X_x\}$;
for every cycle $Q \in \mathcal{Q}_x$
  do Shortest Cycle Crossing Once(G, Q);
return the shortest cycle above.

6 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; we give the precise modification in the proof of the following result.

Lemma 7 *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. Throughout this proof we assume that the given embedding is represented in such a way that signatures of edges in T_x are all positive. For vertices $u, u' \in V(G)$, we use $P[u, u']$ for the (unique) path in T_x from u to u'. For a loop $l \in L_x$, we define $split(l)$ as the vertex v on l such that the part that appears twice in l is equal to $P[x, v]$. In particular, $split(l) = x$ if and only if l is a cycle.

Our first goal is to change the graph G and the spanning tree T_x in such a way that the loops in L_x will all become cycles whose pairwise intersection is only the vertex x. To achieve this goal, we proceed as follows.

There is nothing to do if all loops in T_x are cycles. Otherwise, consider a non-simple loop l_0 in L_x whose repeated part $P_0 = P[x, split(l_0)]$ is shortest. Let $x = v_0, v_1, \ldots, v_k = split(l_0)$ be be the consecutive vertices on P_0. Let v_{k+1} and v'_{k+1} be the neighbors of v_k that are on the loop l_0 and are distinct from v_{k-1}. Assume, moreover, that the edges of l_0 around v_k have local rotation $v_{k-1}, v_{k+1}, v'_{k+1}$ (see Figure 2). The edges incident with each vertex v_i $(1 \le i \le k)$ that are not on the path $v_0, v_1, \ldots, v_{k+1}$ can be classified as those on the *left* or on the *right*

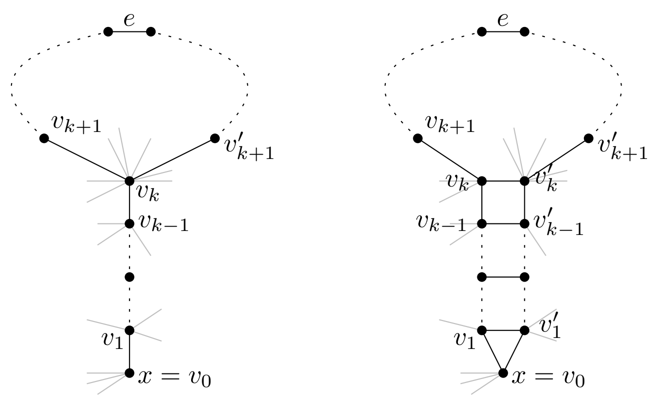

Figure 2: How to modify the graph such that a loop $l_0 = loop(T_x, e) \in L_x$ becomes a cycle. The edges $v_i v'_i$ have length 0.

of that path. Now we replace the path $x, v_1, \ldots, v_k$ in G with two paths $x, v_1, \ldots, v_k$ and $x, v'_1, \ldots, v'_k$ and add edges $v_i v'_i$ $(1 \le i \le k)$ between them. All edges incident with v_i that are on the left of v_i in G are now incident with v_i, while those on the right are incident with v'_i, $i = 1, \ldots, k$. In particular, the edge $v_k v'_{k+1}$ which is on the right of v_k, is replaced by the edge $v'_k v'_{k+1}$. See Figure 2. The edges that correspond to previous edges have the same length in the new graph, while all new edges $v_i v'_i$ have length 0.

The new graph is naturally embedded in Σ as well. We replace the loop l_0 in L_x by the cycle $v_0, v_1, \ldots, v_k, l_0 \setminus P[v_0, v_k], v'_k, v'_{k-1}, \ldots, v'_1, v_0$. 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. They keep being loops (or cycles) because we have chosen l_0 such that $length(P[x, split(l_0)])$ is shortest among non-simple loops in T_x.

We repeat the procedure until L_x consists of only cycles; we need $O(g)$ repetitions. Each cycle in L_x may have two paths from x in common with other cycles. Consider a *longest* path P that two cycles $Q, Q' \in L_x$ have in common. Using the same technique as above, we can modify the graph and the cycles Q, Q' in such a way that they share one path less; details are similar and omitted. We keep repeating this step until any pair of cycles intersects only at x. We have to repeat this step at most $2|L_x| = 2g$ times because, at each step, a new edge adjacent to x that is used by some cycle is created. Therefore, the cycles in the resulting set pairwise intersect only in vertex x.

Let $\tilde{G}$ be the final graph that is obtained and $\tilde{L}_x$ the final set of cycles. Observe that each cycle in $\tilde{L}_x$ is composed of two shortest paths from x plus an edge. It is clear that a shortest non-contractible cycle in $\tilde{G}$ corresponds to a shortest non-contractible cycle in the original graph G, as well as shortest paths in $\tilde{G}$ correspond to shortest paths in G. The problem reduces then to find a shortest non-contractible cycle in $\tilde{G}$. However, observe that the number of vertices has increased; each step may add $O(V)$ vertices, and therefore $\tilde{G}$ consists of $O(gV)$ vertices. This reduction can easily be done in $O(gV)$ time. $\square$

The problem that remains is to find a shortest non-contractible cycle in a graph $\tilde{G}$ with $O(gV)$ vertices where we are given a set of cycles $\mathcal{Q}_x$ that generate the fundamental group with base point x and whose pairwise intersection is x. Moreover, each cycle of $\mathcal{Q}_x$ consists of two shortest paths from x plus an edge. 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 8 *There is a cycle $Q \in \mathcal{Q}^*$ that crosses each cycle in $\mathcal{Q}_x$ at most twice.*

Proof. For each cycle $Q \in \mathcal{Q}^*$, let $Int(Q) = \max\{Int(Q, Q') \mid Q' \in \mathcal{Q}_x\}$, where $Int(Q, Q')$ denotes the number of connected components of $Q \cap Q'$. Let Q_0 be a cycle in $\mathcal{Q}^*$ minimizing $Int(Q)$, that is, $Int(Q_0) = \min\{Int(Q) \mid Q \in \mathcal{Q}^*\}$. We claim that $Int(Q_0) \leq 2$, and therefore $cr(Q', Q_0) \leq 2$ for all $Q' \in \mathcal{Q}_x$ because each crossing is an intersection. Indeed, we can assume for contradiction that $Int(Q_0) \geq 3$; let $Q_1 \in \mathcal{Q}_x$ be a cycle such that $Int(Q_0, Q_1) \geq 3$. We can now use an argumentation as that in Lemma 4, but using the fact that non-contractible cycles satisfy the 3-path property, instead of the argument with cr_C. Details are omitted. □

Consider the set $D = \Sigma \text{✂} \mathcal{Q}_x$ and the corresponding graph $G_P = \tilde{G} \text{✂} \mathcal{Q}_x$. 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. In G_P, each cycle from $\mathcal{Q}_x$ corresponds to two paths on the boundary of D. We can glue an infinite number of copies of D to construct the universal cover of Σ; see [13] for a reference on universal covers. However, because of Lemma 8, we can find a shortest non-contractible cycle by constructing only a portion of this universal cover. These are the main ideas to prove the following result; an algorithm is described below.

Theorem 9 *Let G be a graph with V vertices embedded on a surface of genus g. We can find a shortest non-contractible cycle in $O(g^{O(g)}V)$ time plus the time needed to solve the $O(g^{O(g)}V)$-pairs distance problem in a planar graph of size $O(g^{O(g)}V)$.*

Proof. According to Lemma 7, 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 8, there is a shortest non-contractible cycle crossing each cycle of $\mathcal{Q}_x$ at most twice.

Consider the topological disk $D = \Sigma \text{✂} \mathcal{Q}_x$ 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, the result follows. □

Corollary 10 *Let G be a graph with V vertices embedded on a surface of genus g. A shortest non-contractible cycle in G can be found in $O(g^{O(g)}V^{3/2})$ time.*

Proof. Combine the previous theorem with Lemma 1(i). □

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. We summarize our algorithm below.

Algorithm *Shortest Non-Contractible Cycle*(G)
Input: An embedded graph G
Output: A shortest non-contractible cycle in G

1. Fix $x \in V(G)$ and compute shortest-path tree T_x;
2. Compute a tree-cotree decomposition (T_x, C_x, X_x);
3. $L_x \leftarrow \{loop(T_x, e) \mid e \in X_x\}$;
4. Modify G such that the loops in L_x become cycles $\mathcal{Q}_x$ whose pairwise intersection is equal to x (use Lemma 7);
5. $G_P \leftarrow G /\!\!/ \mathcal{Q}_x$ (G_P is planar);
6. $\mathcal{G} \leftarrow O(g^{O(g)})$ copies of G_P;
7. Glue the graphs in $\mathcal{G}$ to construct G_{U_0}, the portion of the universal cover of G reachable from G_P by crossing at most $2g$ boundaries of G_P;
8. $\Pi \leftarrow \{(v, v') \mid v \in G_P,\ v' \text{ a copy of } v \text{ in distinct copy of } G_P\}$;
9. find a pair $(v_0, v_0') \in \Pi$ such that $d_{G_{U_0}}(v_0, v_0') = \min\{d_{G_{U_0}}(v, v') \mid (v, v') \in \Pi\}$ (use Lemma 1);
10. **return** a shortest path in G_{U_0} from v_0 to v_0' (this is a cycle in G).

7 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. [20, Chapter 5]). Since their computation is of considerable interest in topological graph theory, it makes sense to consider this special case in more details.

7.1 Arbitrary embedded graphs

We next recall the parameters measuring the width of embedded graphs that were introduced in Section 2. 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. Therefore, the non-separating edge-width can be computed by setting $w(e) = 1$ for all edges e in G and running the algorithms from previous sections. The (*non-separating*) *face-width* $\mathtt{fw}(G)$ (and $\mathtt{fw}_0(G)$, respectively) of an embedded graph is the minimum number of faces that any non-contractible (surface non-separating) closed curve in the surface is going to intersect.

For an embedded graph G, 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. (If v and f have multiple incidence, then Γ also has multiple edges joining v and f.) It is easy to see that $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma)$ and $\mathtt{fw}_0(G) = \frac{1}{2}\mathtt{ew}_0(\Gamma)$ [20]. The construction of Γ takes linear time $O(V + g)$ from an embedding of G.

Observe that, for this special case, a breadth-first-search tree is indeed a shortest-path tree, and it can be computed in $O(V + g)$ time, instead of $O(V \log V + g)$ that is required for arbitrary lengths if the genus is high. This improves slightly the running time for this special case.

Theorem 11 *For a graph G embedded in a surface of genus g, the non-separating edge-width and face-width can be computed in $O(g^{3/2}V^{3/2} + g^{5/2}V^{1/2})$ time. The edge-width and face-width of G can be computed in $O(g^{O(g)}V^{3/2})$ time.*

It can happen that $\mathtt{ew}(G) = \Omega(V)$. The situation is different for the face-width $\mathtt{fw}(G)$ for which there exist non-trivial bounds. Hutchinson [15] showed that the edge-width of a *triangulation* in an orientable surface of genus $g \le V$ is $O(\sqrt{V/g}\log g)$, and $O(\log g)$ if $g > V$, improving the previous bound of $\sqrt{2V}$ by Albertson and Hutchinson [1]. A bound for general surfaces that is comparable to the orientable case is proved below.

Theorem 12 *Let G be a graph of order V embedded in a surface of genus g. Then $\mathtt{fw}(G) = O(\sqrt{V/g}\log g)$ if $V \ge g$, and $\mathtt{fw}(G) = O(\log g)$ if $V < g$.*

Proof. The vertex-face incidence graph Γ has a natural embedding as a quadrangulation in the same surface as G, that is, all facial walks consist of exactly four edges. Let $T = \Gamma + E(G)$ be the triangulation obtained from Γ by adding all edges of G in the quadrangular faces of Γ. Then $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma) \le \mathtt{ew}(T)$.

If the surface is orientable then the aforementioned result of Hutchinson shows that $\mathtt{ew}(T) = O(\sqrt{|V(T)|/g}\log g)$ if $|V(T)| \ge g$, and $O(\log g)$ if $g > |V(T)|$. Since $|V(T)| = V + |F(G)| = O(V + g)$, the first bound reduces to $\mathtt{ew}(T) = O(\sqrt{V/g}\log g)$. It is clear that this completes the proof in the orientable case.

Let us now show that the bound of Hutchinson can be extended to nonorientable surfaces. To show this, we form the orientable double cover D_T of T which is also a triangulation and its genus $\tilde{g}$ is less than $2g$. Combinatorially, the orientable double cover D_G is constructed as follows: for each vertex $v \in V(G)$ we place two vertices v, v' in $V(D_G)$, for each edge $uv \in E(G)$ we place edges $uv, u'v'$ in $E(D_G)$ if the signature of uv is $\lambda(uv) = +1$, and edges $uv', u'v$ in $E(D_G)$ if $\lambda(uv) = -1$; and the circular order of edges around vertices $u, u' \in V(D_G)$ is the same as around $u \in V(G)$.

By [15], D_T contains a noncontractible cycle $\tilde{C}$ of length $O(\sqrt{2V/(2g)}\log 2g)$ if $V \ge g$, and $O(\log 2g)$ if $g > V$. The projection C of $\tilde{C}$ to T is a closed walk in T which is noncontractible because of the homotopy lifting property (cf. [19]). This walk contains a noncontractible cycle of length at most $|V(\tilde{C})| = \mathtt{ew}(D_T)$, so $\mathtt{ew}(T) \le \mathtt{ew}(D_T)$, which is what we were to prove. $\square$

7.2 Face-width in the projective plane

For the special case when G is embedded in the projective plane $\mathbb{P}^2$, 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 $\mathtt{ew}(G)$. This is achieved by combining three ideas: an algorithm of Erickson and Har-Peled [9] to compute a non-contractible cycle which is at most twice as long as a shortest non-contractible cycle; a double cover of G which is planar; and Lemma 1(ii) for distance queries on planar graphs. The result is as follows.

Lemma 13 *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. Since the sphere is the universal cover of the projective plane $\mathbb{P}^2$, we can consider the cover of G on the sphere, which is nothing else than the orientable double cover D_G of the embedding of G that was introduced in the proof of Theorem 12.

It is well-known that D_G is a planar graph embedded on the sphere, and that it is a cover of G. Moreover, 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 can compute in $O(V \log V)$ time a non-contractible cycle Q of G of length at most $2\mathtt{ew}(G) \leq 2t$ using the results of [9]. Actually, this can be done in $O(V)$ time by using a breadth-first-search tree. Any non-contractible cycle in G has to intersect Q at some vertex. In particular, every shortest non-contractible cycle intersects Q. By finding for each vertex $v \in V(Q)$ the distance between v and v' in D_G, we find a vertex $v_0 \in Q$ passing through a shortest non-contractible loop. This requires to compute $|Q| \leq 2t$ pairs of distances in D_G, which using Lemma 1 takes $O(V \log^2 V + t\sqrt{V} \log^2 V)$ time in total.

Once we know a vertex v_0 which is contained in a shortest non-contractible cycle, we can use breadth-first-search in D_G from the vertex v_0 to find a shortest path from v_0 to its couple v_0' in D_G. This gives a shortest non-contractible cycle in G. □

Like before, consider the vertex-face incidence graph Γ which can be constructed in linear time. From the bounds in Section 7.1, we know that the edge-width of Γ is $O(\sqrt{V})$. Therefore, the problem reduces to that of 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 with $t = O(\sqrt{V})$, we conclude the following.

Theorem 14 *Let G be a graph embedded in $\mathbb{P}^2$. We can compute the face-width of G in $O(V \log^2 V)$ time.*

Juvan and Mohar [16] obtained a linear time algorithm for deciding if $\mathtt{fw}(G) \leq k$, where k is a fixed constant. They needed the special case when $k = 4$ in an algorithm for testing embeddability in the torus.

7.3 Face-width in the torus

We next describe an algorithm for computing the face-width for a graph G embedded on the torus $\mathbb{T}$. Consider the vertex-face incidence graph Γ; we will compute $\mathtt{ew}(\Gamma) = 2\mathtt{fw}(G)$. Let us observe that on a fixed surface, $|V(\Gamma)| = \Theta(V)$.

We compute in $O(V \log V)$ time a non-contractible cycle Q of Γ of length at most $2\mathtt{ew}(\Gamma)$ using the results of [9]. We know that $|Q| = O(\sqrt{V})$ by the aforementioned result of Albertson and Hutchinson [1]. See also Lemma 15 below.

Fix a vertex $v_0 \in V(Q)$ and find in $O(V \log V)$ time a shortest non-contractible loop l from v_0. Let A be the non-contractible cycle obtained by removing the repeated edges from l. We know that A consists of two shortest paths from a vertex $u_0 \in V(\Gamma)$ plus an edge; $u_0 = v_0$ if l was a cycle, and $u_0 \neq v_0$ otherwise. Moreover $|A| \leq |l| \leq |Q| = O(\sqrt{V})$.

The graph $\Gamma_A = \Gamma /\!\!/ A$ is planar. Let v', v'' be the two copies of $v \in A$ in Γ_A. Consider the $O(\sqrt{V})$ pairs $\Pi = \{(v', v'') \mid v \in V(A)\}$ and find a pair $(v_1', v_1'') \in \Pi$ such that $d_{\Gamma_A}(v_1', v_1'') = \min\{d_{\Gamma_A}(v', v'') \mid (v', v'') \in \Pi\}$ in $O(V \log^2 V)$ time using Lemma 1. Let P_{v_1} be a shortest path in Γ_A from v_1' to v_1'', which is a non-contractible cycle B in Γ. The graph $\Gamma_B = \Gamma /\!\!/ B$ is planar as well.

We may assume that

$$|B| \geq \frac{\sqrt{2}}{2}|A|. \tag{1}$$

If not, then we repeat the above procedure by starting with B playing the role of the cycle Q. Then we find new cycles A and B. If (1) is violated again, then the length of the new cycle B would be strictly smaller than one half of the length of the former cycle A. This would imply that $|B| < \mathtt{ew}(\Gamma)$, a contradiction.

Using Menger's theorem for vertex-disjoint paths, we can prove the following bound.

Lemma 15 *It holds that* $|A| \cdot |B| = O(V)$.

Proof. Let M be the maximum number of vertex-disjoint paths from A' to A'', the copies of A in Γ_A. By Menger's theorem, M is equal to the cardinality of a minimum (A', A'')-separator S in γ_A. Since Γ_A is embedded in a cylinder with A' and A'' being the cycles on the boundary, the separator S gives rise to a closed curve γ in the torus homotopic to A that intersects Γ precisely in the vertices in S. Since Γ is a quadrangulation, the curve γ determines a non-contractible cycle C in Γ of length at most $2|S| = 2M$. In particular, $M \geq \frac{1}{2}\mathtt{ew}(\Gamma) \geq \frac{1}{4}|A|$.

Let $R_1, \ldots, R_M$ be disjoint (A', A'')-paths. Each R_i together with a segment on A'' determines a path from a vertex $v' \in V(A')$ to its mate v'' in A'' of length at most $|R_i| + \frac{1}{2}|A|$. Consequently, $|R_i| + \frac{1}{2}|A| \geq |B|$. Since $R_1, \ldots, R_M$ are vertex disjoint and (1) holds, we get:

$$V \geq \sum_{i=1}^{M} |R_i| \geq M(|B| - \tfrac{1}{2}|A|) \geq \tfrac{1}{4}|A|(1 - \tfrac{\sqrt{2}}{2})|B|.$$

This completes the proof. $\square$

Cycles A and B constructed above can be used for a fast computation of the face-width of G.

Theorem 16 *Let G be a graph embedded in the torus. We can compute the face-width of G in $O(V^{5/4} \log V)$ time.*

Proof. Every non-contractible cycle in the torus is surface non-separating. Since the cycles A, B generate the fundamental group, Lemma 3 tells us that we can assume that a shortest non-contractible cycle crosses either A or B an odd number of times.

Let us first assume that a shortest non-contractible cycle crosses A an odd number of times. Since A is composed of two shortest paths (plus an edge), the argument used in the proof of Lemma 4 shows that there is a shortest non-contractible cycle that crosses A exactly once. In this case, B is a shortest non-contractible cycle, and we are done.

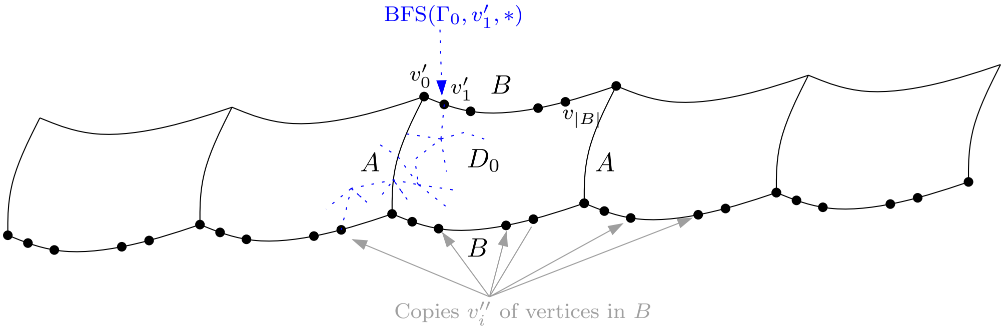

Figure 3: Portion of the universal cover reachable by crossing at most twice A and none B. We show some of the concepts uses in the proof.

Let us now consider the case where a shortest non-contractible cycle crosses B an odd number of times. The argument used in Lemma 4 shows that we can assume that a shortest non-contractible cycle crosses A at most twice. Let C be one such shortest non-contractible cycle crossing B a *minimum* number of times and A at most twice. Then C crosses B exactly once. To see this, consider two copies of $\Gamma_A = \Gamma$ ✂ A and glue them along one copy of A to obtain the graph $\tilde{\Gamma}$. Since C crosses A at most twice, C has a lift to $\tilde{\Gamma}$. However, in each copy of Γ_A, the cycle C intersects B at most once because B is a shortest path from A' to A'', and therefore C crosses B at most twice. Since we assume that C crosses B an odd number of times, we conclude that they cross exactly once.

Let $\Gamma_B = \Gamma$ ✂ B. We distinguish two cases depending on the length of A:

- If $|A| \geq V^{1/4} \log V$, then $|B| = O(V^{3/4}/\log V)$ because of Lemma 15. We compute the distance from v' to v'' for any $v \in B$. Since Γ_B is a planar graph and we have to compute distances between $|B| = O(V^{3/4}/\log V)$ pairs, we use Lemma 1 to find in $O(V^{5/4} \log V)$ time a point $v_0 \in B$ that contains a shortest non-contractible cycle. From this, we easily compute a shortest non-contractible cycle in Γ.

- If $|A| < V^{1/4} \log V$, we proceed as follows. Consider the topological disk $D = \mathbb{T}$ ✂ $(A \cup B)$, make its copy D_0, and construct the portion U_0 of the universal cover reachable from D_0 by crossing A at most twice and without crossing B. This needs five copies of D; see Figure 3. Let Γ_0 be the cover of Γ naturally embedded in U_0 by gluing copies of D.

  In D_0, let $v'_1, v'_2, \ldots, v'_{|B|}$ be the consecutive vertices of one copy of B. The shortest cycle we are seeking corresponds to a shortest path connecting v'_i with some copy v''_i of v'_i.

  We make for each v'_i a BFS$(\Gamma_0, v'_i, |A| - 1)$, where BFS$(\Gamma_0, u, t)$ is a breadth first search tree from vertex u in the graph Γ_0 clipped at depth t, that is, only including vertices up to depth t. If a shortest non-contractible cycle has length strictly smaller than $|A|$, then its lift in Γ_0 has to be contained in BFS$(\Gamma_0, v'_i, |A| - 1)$ for some $v'_i \in B$. Therefore, once we have BFS$(\Gamma_0, v'_i, |A| - 1)$ for all $v'_i \in B$ we can easily find a shortest non-contractible cycle.

  We claim that it takes $O(V^{5/4} \log V)$ time to construct the trees BFS$(\Gamma_0, v'_i, |A| - 1)$ for all $i = 1, \ldots, |B|$. Observe that the proof of the claim will finish the proof of the theorem. The proof is as follows. Define the sets $B_k = \{v'_{k+4i|A|} \mid i = 0, \ldots, \lfloor |B|/4|A| \rfloor\}$

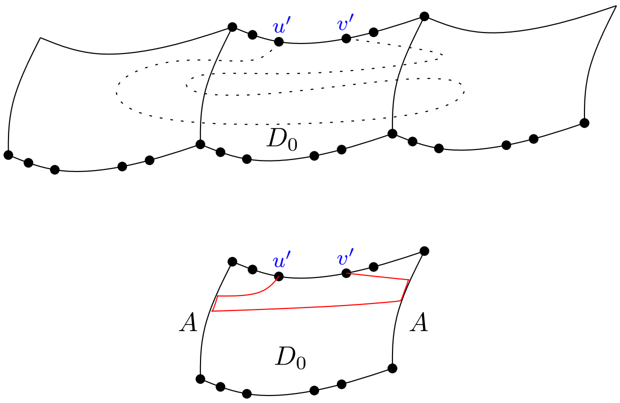

Figure 4: A path of length l in Γ_0 yields a path of length at most $l + 2|A|$ in D_0.

for $k = 1, \ldots, 4|A|$; that is, B_k consists of v'_k and each $4|A|$-th vertex along B. Observe that $B \setminus \bigcup_k B_k = \{v'_{|B|-(|B| \bmod 4|A|)}, \ldots, v'_{|B|-1}\}$ consists of $O(|A|) = O(V^{1/4} \log V)$ vertices. Therefore, the trees $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for $u \in B \setminus \bigcup_k B_k$ can be computed in $O(V^{1/4} \log V) \cdot O(V) = O(V^{5/4} \log V)$ time.

If $u', v' \in B_k$, then $d_{\Gamma \scissors A}(u', v') \geq 4|A|$ because B is a shortest path in $\Gamma \scissors A$. This implies that $d_{\Gamma_0}(u', v') \geq 2|A|$ because any shortest path of length l in Γ_0 can be clipped by D_0 to obtain a path of length at most $l + 2|A|$; see Figure 4. Therefore, if $u', v' \in B_k$, we have $\mathrm{BFS}(\Gamma_0, u', |A| - 1) \cap \mathrm{BFS}(\Gamma_0, v', |A| - 1) = \emptyset$. Since for any fixed k, each edge of Γ_0 appears at most once in the trees $\{\mathrm{BFS}(\Gamma_0, u, |A| - 1) \mid u \in B_k\}$, we can compute $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for all $u \in B_k$ in $O(n)$ time. The parameter k takes the values $1, \ldots, 4|A| = O(V^{1/4} \log V)$, and therefore we need $O(V^{1/4} \log V) \cdot O(V) = O(V^{5/4} \log V)$ time to compute $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for all $u \in \bigcup_k B_k$. This finishes the proof of the claim and of the theorem.

□

8 Conclusions

We have presented algorithms for finding shortest non-contractible and surface non-separating cycles for graphs embedded on a surface. For a fixed surface, our algorithms run in $O(V^{3/2})$ time, which is a considerable improvement over previous results. Our algorithms can be used to compute the (non-separating) edge-width and the (non-separating) face-width of embedded graphs.

Our algorithms work for undirected graphs with non-negative edge-lengths. Similar results for directed graphs seem much harder because non-contractible or non-separating cycles do not satisfy the 3-path-condition anymore. Finding shortest cycles with properties that do not satisfy the 3-path-condition remains an elusive problem.

We have also given a near-linear running time algorithm for computing the face-width in the projective plane; for the torus, we show how to compute the face-width in $O(V^{5/4} \log V)$ time. We feel that one of the most appealing open questions is finding near-linear running time algorithms for computing the face-width of graphs embedded on a (possibly fixed) surface. Our approach when dealing with the projective plane and the torus does not seem to extend to surfaces of higher genera.

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. Efficient algorithms for shortest path problems on planar digraphs. In F. d'Amore, P. G. Franciosa, and A. Marchetti-Spaccamela, editors, *WG'96*, volume 1197 of *LNCS*, pages 151–165. Springer, 1997.

[5] É. Colin de Verdière. *Shortening of curves and decomposition of surfaces.* PhD thesis, Université Paris 7, December 2003. Avilable at `http://www.di.ens.fr/users/colin/textes/these.html.en`.

[6] É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. In *11th International Symposium on Graph Drawing, Perugia, Italy*, volume 2912 of *LNCS*, pages 478–490. Springer, 2004.

[7] É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete Comput. Geom.*, 33:507–534, 2005. Preliminary version in FOCS'02.

[8] D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proc. 14th Symp. Discrete Algorithms*, pages 599–608. ACM and SIAM, January 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 *Proc. 16th Symp. Discrete Algorithms*, pages 1038–1046, 2005. `http://compgeom.cs.uiuc.edu/~jeffe/pubs/gohog.html`.

[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. Available at `http://www.math.cornell.edu/~hatcher/`.

[14] M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1):3–23, 1997.

[15] J. P. Hutchinson. On short noncontractible cycles in embedded graphs. *SIAM J. Disc. Math.*, 1:185–192, 1988.

[16] M. Juvan and B. Mohar. An algorithm for embedding graphs in the torus. *Submitted.*

[17] P. N. Klein. Multiple-source shortest paths in planar graphs. In *Proc. 16th ACM-SIAM Sympos. Disc. Alg. (SODA)*, pages 146–155, 2005.

[18] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proc. 17th Symp. Comp. Geom.*, pages 80–89, 2001.

[19] W. S. Massey. *Algebraic Topology: An Introduction.* Springer Verlag, 1967.

[20] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, Baltimore, 2001.

[21] A. Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *J. Combin. Theory Ser. B*, 51:127–159, 1991.

[22] A. Schrijver. Paths in graphs and curves on surfaces. In *First European Congress of Mathematics, Vol. II (Paris, 1992)*, volume 120 of *Progr. Math.*, pages 381–406. Birkhäuser, Basel, 1994.

[23] C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Combin. Theory, Ser. B*, 48:155–177, 1990.

[24] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proc. 6th Ann. ACM Symp. Comput. Geom.*, pages 102–111, 1990.

[25] U. Zwick. Exact and approximate distances in graphs – A survey. In *ESA 2001*, volume 2161 of *LNCS*, pages 33–48. Springer, 2001.