



HAL
open science

Computing shortest closed curves on non-orientable surfaces

Denys Bulavka, Éric Colin de Verdière, Niloufar Fuladi

► **To cite this version:**

Denys Bulavka, Éric Colin de Verdière, Niloufar Fuladi. Computing shortest closed curves on non-orientable surfaces. International Symposium on Computational Geometry, Jun 2024, Athens, Greece. 10.4230/LIPICs.SoCG.2024.28 . hal-04732932

HAL Id: hal-04732932

<https://hal.science/hal-04732932v1>

Submitted on 11 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Computing Shortest Closed Curves on Non-Orientable Surfaces

Denys Bulavka ✉

Einstein Institute of Mathematics, Hebrew University, Jerusalem, Israel

Éric Colin de Verdière ✉

LIGM, CNRS, Univ. Gustave Eiffel, F-77454 Marne-la-Vallée, France

Niloufar Fuladi ✉

LORIA, CNRS, INRIA, Université de Lorraine, F-54000 Nancy, France

Abstract

We initiate the study of computing shortest non-separating simple closed curves with some given topological properties on non-orientable surfaces. While, for orientable surfaces, any two non-separating simple closed curves are related by a self-homeomorphism of the surface, and computing shortest such curves has been vastly studied, for non-orientable ones the classification of non-separating simple closed curves up to ambient homeomorphism is subtler, depending on whether the curve is one-sided or two-sided, and whether it is orienting or not (whether it cuts the surface into an orientable one).

We prove that computing a shortest orienting (weakly) simple closed curve on a non-orientable combinatorial surface is NP-hard but fixed-parameter tractable in the genus of the surface. In contrast, we can compute a shortest non-separating non-orienting (weakly) simple closed curve with given sidedness in $g^{O(1)} \cdot n \log n$ time, where g is the genus and n the size of the surface.

For these algorithms, we develop tools that can be of independent interest, to compute a variation on canonical systems of loops for non-orientable surfaces based on the computation of an orienting curve, and some covering spaces that are essentially quotients of homology covers.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Mathematics of computing → Graphs and surfaces

Keywords and phrases Surface, Graph, Algorithm, Non-orientable surface

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.28

Related Version *Full Version*: <http://arxiv.org/abs/2403.11749>

Funding *Denys Bulavka*: Work done while this author was at Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic. This work was initiated while this author was visiting LIGM, funded by the French National Research Agency grant ANR-17-CE40-0033 (SoS), by the grant no. 21-32817S of the Czech Science Foundation (GAČR) and by Charles University project PRIMUS/21/SCI/014.

Éric Colin de Verdière: The work of this author is partly funded by the French National Research Agency grants ANR-17-CE40-0033 (SoS) and ANR-19-CE40-0014 (MIN-MAX).

Niloufar Fuladi: Work done while this author was at LIGM, CNRS, Univ. Gustave Eiffel, F-77454 Marne-la-Vallée, France. The work of this author is partly funded by the French National Research Agency grants ANR-17-CE40-0033 (SoS) and ANR-19-CE40-0014 (MIN-MAX).

Acknowledgements We would like to thank Arnaud de Mesmay for stimulating discussions, and the anonymous reviewers for their useful comments.



© Denys Bulavka, Éric Colin de Verdière, and Niloufar Fuladi;
licensed under Creative Commons License CC-BY 4.0

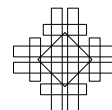
40th International Symposium on Computational Geometry (SoCG 2024).

Editors: Wolfgang Mulzer and Jeff M. Phillips; Article No. 28; pp. 28:1–28:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

In computational topology of graphs on surfaces, much effort has been devoted to computing shortest closed curves with prescribed topological properties on a given surface. Most notably, the computation of shortest non-contractible, or shortest non-separating, closed curves on a combinatorial surface has been studied, under various scenarios, in at least a dozen papers in the last twenty years [11, Table 23.2]. Also, algorithms have been given to compute shortest splitting closed curves [7], shortest essential closed curves [19], shortest closed curves within some non-trivial homotopy class [6], and shortest closed curves within a given homotopy class [12]. In all these cases, the purpose is to compute a shortest closed curve in a given equivalence class, for various notions of equivalence.

Identifying two closed curves on a given surface whenever there is a self-homeomorphism of the surface mapping one to the other is certainly one of the most natural equivalence relations. In particular, this is the most refined relation if we are only given the input surface, and it is relevant in particular in the context of mapping class groups [20, Section 1.3.1]. Under this notion, on an orientable surface, any two simple non-separating closed curves are equivalent: Any non-separating simple closed curve cuts the surface into an orientable surface that has (oriented) genus one less than the original surface and with two boundary components. However, for non-orientable surfaces, it turns out that the classification is subtler: Excluding some low-genus surfaces, a non-separating simple closed curve can be two-sided (have a neighborhood homeomorphic to an annulus) or one-sided (in which case it has a neighborhood homeomorphic to a Möbius band); furthermore it can be orienting (when cutting along it yields an orientable surface with boundary) or not.

In this paper, we study the complexity of computing shortest non-separating simple closed curves in non-orientable surfaces, under the constraint of being either one-sided or two-sided, either orienting or not, developing, in passing, tools to handle non-orientable surfaces algorithmically. Before describing our results in detail, we survey previous works.

1.1 Previous works

One of the most basic and studied questions in topological algorithms for graphs on surfaces is that of computing a shortest non-contractible or non-separating closed curve (the length of such a curve is called *edge-width* in topological graph theory [1] or *systole* in Riemannian geometry [24]). Algorithmically, the simplest setup for graphs on surfaces is that of *combinatorial surfaces*: On a surface \mathcal{S} , one is given an embedding of a graph G that is cellular (all faces are homeomorphic to open disks); each edge of the graph has a positive weight. The goal is to compute a shortest closed walk in G that is non-trivial on \mathcal{S} either in homotopy or in homology. It turns out that such closed walks are simple, so they are, respectively, a shortest cycle that does not bound a disk on \mathcal{S} , or that does not separate \mathcal{S} .

Algorithms for computing shortest non-contractible or non-separating closed curves on surfaces have been developed since the early 1990s [31]. The current fastest algorithm in terms of the size of the input, the number n of vertices, edges, and faces of G , due to Erickson and Har-Peled [16], runs in $O(n^2 \log n)$ time. However, it is typical to view the genus g of the surface as a small parameter, and under this perspective it is worth mentioning the algorithms by Cabello et al. [5], which runs in $O(g^2 n \log n)$ for generic weights, and Fox [21], which runs in $2^{O(g)} \cdot n \log \log n$, but for orientable surfaces only. We refer to a survey [11, Table 23.2] for many other results.

Other topological properties have also been considered. Chambers et al. [7] study the complexity of computing a shortest “simple” closed curve that splits the (orientable) surface into two pieces, neither of which is a disk. Erickson and Worah [19] give a near-quadratic

time algorithm to compute a shortest essential “simple” closed curve on an orientable surface with boundary. Cabello et al. [6] provide a near-linear time algorithm (for fixed genus) to compute a shortest “simple” closed curve within some (unspecified) non-trivial homotopy class. In all these problems, one cannot expect in general the output closed curve to be a simple cycle in the input graph G : It sometimes has to repeat vertices and edges of G , but it is *weakly simple* [9] in the sense that it can be made simple by an arbitrary perturbation of the curve on the surface. In order to store weakly simple curves, both for the output and at intermediate steps of the algorithm, it is convenient to use the dual framework of *cross-metric surface* setting [12], in which these curves are really simple.

Very few works devote tools specifically to non-orientable surfaces. On the combinatorial side, Matoušek et al. [28] carefully describe the result of cutting a non-orientable surface along a simple arc or closed curve, and in particular emphasize that there are various flavors of non-separating simple closed curves. Very recently, Fuladi et al. [22] prove the existence of a canonical system of loops on a non-orientable surface in which each loop has multiplicity $O(1)$, and show that such a system of loops can be computed in polynomial time. There are some good reasons not to neglect non-orientable surfaces [22, Introduction]: Random surfaces are almost surely non-orientable; graphs embeddable on an orientable surface of Euler genus $2g$ are embeddable on a non-orientable surface of genus $2g + 1$, while graphs embeddable on the projective plane can have arbitrarily large orientable genus; non-orientable surfaces appear naturally, e.g., in the graph structure theorem of Robertson and Seymour [29].

1.2 Our results

We obtain the following results on orienting (simple) closed curves on non-orientable surfaces:

► **Theorem 1.1.** *It is NP-hard to decide, given a cross-metric surface (\mathcal{S}, G^*) and an integer k , whether a shortest orienting closed curve on (\mathcal{S}, G^*) has length at most k .*

► **Theorem 1.2.** *Given a non-orientable cross-metric surface (\mathcal{S}, G^*) of genus g and size n , we can compute a shortest orienting closed curve in (\mathcal{S}, G^*) in $2^{O(g)} \cdot n \log n$ time. Such a shortest closed curve has multiplicity at most two.*

It turns out that orienting curves are always non-separating, and that their sidedness is prescribed by the genus of the surface (see Lemma 2.3 below). In contrast, computing shortest non-orienting (simple) closed curves can be done in polynomial time:

► **Theorem 1.3.** *Given a non-orientable cross-metric surface (\mathcal{S}, G^*) of genus g and size n , we can compute a shortest non-separating non-orienting one-sided (respectively, two-sided) closed curve in (\mathcal{S}, G^*) in $O(\text{poly}(g)n \log n)$ time. More precisely:*

- *We can compute a shortest non-separating non-orienting one-sided closed curve in $O(g^3 n \log n)$ time if g is even, and in $O(g^4 n \log n)$ time if g is odd;*
- *we can compute a shortest non-separating non-orienting two-sided closed curve in $O(g^4 n \log n)$ time if g is odd, and in $O(g^5 n \log n)$ time if g is even.*

Such shortest closed curves have multiplicity at most two.

These results implicitly assume that such curves exist (equivalently, $g \geq 2$ in the first case and $g \geq 3$ in the second one; see Lemma 2.3). They are stated in the cross-metric surface model, but their outputs immediately translate to shortest weakly simple closed curves in the dual combinatorial surface (\mathcal{S}, G) .

In passing, we develop tools of independent interest for non-orientable surfaces. First, we give an algorithm to compute an orienting curve in linear time (Proposition 4.1), refining a construction given by Matoušek et al. [28]. Second, we introduce an analog of canonical

systems of loops for non-orientable surfaces with a new combinatorial pattern, the *standard systems of loops*, and show that we can compute one in asymptotically the same amount of time as canonical systems of loops on orientable surfaces (Proposition 5.1); we remark that Fuladi et al. [22] compute a canonical system of loops in polynomial time, but they do not provide a precise estimate on the running time, which is likely higher than ours. Such standard systems of loops can be used to compute homeomorphisms between non-orientable surfaces; moreover, some algorithms for surface-embedded graphs rely on canonical systems of loops in the orientable case [23, 28], and for non-orientable surfaces they could use standard systems of loops instead. Third, we introduce *subhomology covers*, more compact than the *homology covers* of Chambers et al. [8]¹, but capturing exactly the topological information that we need to classify orienting/non-orienting, one-sided/two-sided closed curves (Section 7).

1.3 Techniques and organization of the paper

After the preliminaries (Section 2), we first show that our two algorithmic results (Theorems 1.2 and 1.3) follow from a single common statement (Theorem 3.1). In short, the topological properties that we are considering are homological: Knowing the homology class of a closed curve is enough to decide whether it is separating or not, one-sided or two-sided, orienting or not. Moreover, the non-separating non-orienting curves with given sidedness are characterized by the fact that they belong to the union of $\text{poly}(g)$ affine subspaces of small codimension in the homology group. The rest of the paper is devoted to the proof of Theorem 3.1. In Section 4, we give a linear-time algorithm to compute an orienting curve. This serves as a first step to compute a standard system of loops in Section 5. Section 6 provides a way to convert homology from a canonical basis to a standard one. At this point, given a closed curve, we are able to decide whether it has the desired topological properties efficiently. In Section 7, we introduce *subhomology covers*, and then show how to compute a shortest path in the subhomology cover that, when projected, will become the desired closed curve on the surface (Section 8); this uses a third kind of system of loops, the shortest system of loops [18, 10]. Section 9 concludes the algorithm.

The NP-hardness proof (Theorem 1.1) is omitted; in the same spirit as the NP-hardness proof of computing a shortest splitting closed curve by Chambers et al. [7], it is a reduction from the Hamiltonian cycle problem in grid graphs.

2 Preliminaries

2.1 Curves and graphs on surfaces

We use standard terminology of topology of surfaces and graphs drawn on them; see, e.g., Armstrong [2] or, for a more algorithmic perspective, Colin de Verdière [11]. In this paper, the **genus** of a surface denotes its Euler genus (which equals twice the standard genus for orientable surfaces). Unless specified otherwise, surfaces are without boundary.

A simple closed curve on a surface \mathcal{S} is **two-sided** if it has a neighborhood homeomorphic to the annulus. Otherwise, it has a closed neighborhood homeomorphic to the Möbius band and it is called **one-sided**. A simple closed curve is called **non-separating** if the surface (with boundary) we obtain by cutting along it is connected; otherwise the curve is **separating**. Note that a separating curve is two-sided. A simple closed curve on a non-orientable surface is called **orienting** if by cutting along it, we obtain an orientable surface (with boundary).

¹ Chambers et al. [8] combines several conference abstracts, including one by Erickson and Nayyeri [17]; the material that we use from [8] appeared first in [17].

A **homotopy** between two closed curves is a continuous family of closed curves between them. A closed curve is **contractible** if it is homotopic to a constant closed curve.

Finally, a **covering space** of \mathcal{S} is a topological space $\tilde{\mathcal{S}}$ with a continuous map $\pi: \tilde{\mathcal{S}} \rightarrow \mathcal{S}$ satisfying the local homeomorphism property, i.e., for every point x in \mathcal{S} there exists an open neighborhood U of x in \mathcal{S} and pairwise disjoint open sets U_1, \dots, U_d in $\tilde{\mathcal{S}}$ such that $\bigcup_{i=1}^d U_i = \pi^{-1}(U)$ and π restricted to each U_i is a homeomorphism with U . A **lift** of a path p on \mathcal{S} is a path \tilde{p} on $\tilde{\mathcal{S}}$ such that $\pi \circ \tilde{p} = p$.

2.2 Combinatorial and cross-metric surfaces

A **combinatorial surface** (\mathcal{S}, G) is the data of a surface \mathcal{S} together with a positively edge-weighted graph G cellularly embedded on \mathcal{S} ; in this model, the curves (or the edges of the graphs) considered later are restricted to be walks in G . The length of a curve c is the sum of the weights of the edges of G used by c , counted with multiplicity. A closed curve (or a graph) on a combinatorial surface is **simple** if it is actually *weakly simple*; namely, if it admits an arbitrarily small perturbation on \mathcal{S} that turns it into a simple closed curve (or graph) on \mathcal{S} .

Weakly simple closed curves and graphs can traverse the same edge of G or visit the same vertex of G more than once. To keep track of these multiplicities, it is often useful to use the concept of *cross-metric surface* [12, Section 1.2], which refines the notion of combinatorial surface in dual form. A **cross-metric surface** (\mathcal{S}, G^*) is the data of a surface \mathcal{S} together with a positively edge-weighted graph G^* cellularly embedded on \mathcal{S} ; in this model, the graphs and curves considered later are in general position with respect to G^* . The length of a path p is the sum of the weights of the edges of G^* crossed by p , with multiplicity. The **multiplicity** of a path or closed curve on (\mathcal{S}, G^*) is the maximum number of times it crosses a given edge of G^* . Every (weakly) simple closed curve on the combinatorial surface (\mathcal{S}, G) corresponds to a simple closed curve of the same length on the cross-metric surface (\mathcal{S}, G^*) , and conversely.

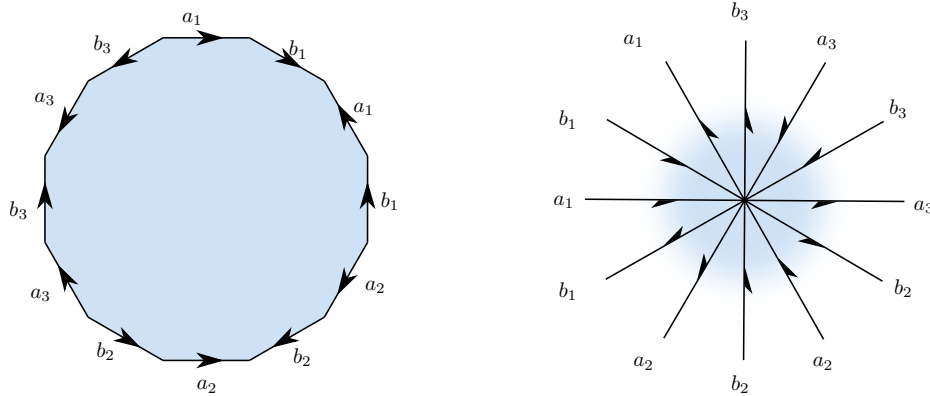
One can represent cellular graph embeddings on (possibly non-orientable) surfaces using, e.g., graph-encoded maps [27, 13]. We represent a graph embedded in the cross-metric surface (\mathcal{S}, G^*) by storing its *overlay* (or *arrangement*) with G^* , and our algorithms work on this representation. The **size** of a combinatorial or cross-metric surface is the number of vertices, edges, and faces of the underlying graph.

2.3 Shortest and canonical systems of loops

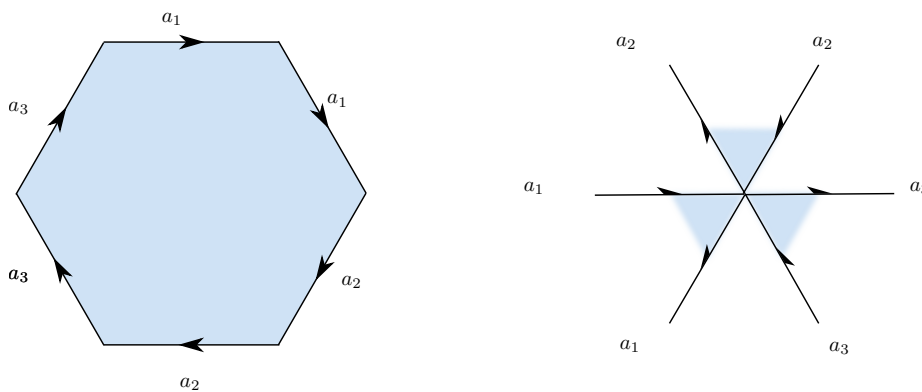
A **system of loops** of a surface \mathcal{S} is a one-vertex graph L embedded on \mathcal{S} that has a single face, which is homeomorphic to a disk. By Euler's formula, L has exactly g loops, where g is the (Euler) genus of \mathcal{S} . Cutting \mathcal{S} along L results in a $2g$ -gon, its **polygonal schema**, with edges of its boundary identified in pairs. The following result is a corollary of an algorithm to compute shortest systems of loops, given by Erickson and Whittlesey [18], and later extended to non-orientable surfaces by Colin de Verdière [10]:

► **Lemma 2.1** ([18, 10]). *Let (\mathcal{S}, G) be a combinatorial surface, orientable or not, of Euler genus g and size n . We can compute a set of $2g$ shortest paths on G such that each non-contractible closed walk in G intersects one of these shortest paths, in $O(gn + n \log n)$ time.*

One can describe a polygonal schema (see Figures 1 and 2) by assigning a symbol and an orientation to each loop in L , and listing the symbols corresponding to the loops encountered along the boundary of the polygonal schema, in clockwise order say, indicating a loop by a bar when encountered with the opposite orientation. A **canonical system of loops for**



■ **Figure 1** Two equivalent views of a canonical system of loops of an orientable surface of Euler genus 6. Left: A polygonal schema of the form $a_1 b_1 \bar{a}_1 \bar{b}_1 a_2 b_2 \bar{a}_2 \bar{b}_2 a_3 b_3 \bar{a}_3 \bar{b}_3$. To reconstruct the surface, one identifies the edges of the polygon in pairs, respecting the orientations of the arrows. Right: By gluing together the corners of the polygonal schema, we see that all the vertices of the polygon get actually identified into a single vertex on the surface. The “half-arrows” indicate one side of each loop: In the present case, all loops are two-sided, which is reflected by the fact that the right side of a loop when leaving the vertex is still the right side of the loop when it comes back to the vertex.



■ **Figure 2** Two equivalent views of a canonical system of loops of a non-orientable surface of Euler genus 3. Left: A polygonal schema of the form $a_1 a_1 a_2 a_2 a_3 a_3$ with its top (visible) face blue and its bottom (hidden) face white. Right: The cyclic ordering of the edges around the vertex after identifying the edges of the polygonal schema. All loops are one-sided, which is reflected by the fact that the right side of each loop when leaving the vertex becomes the left side of that loop when it comes back to the vertex. The colors of the corners indicate the side of the polygon.

an orientable surface of Euler genus $2g$ is a system of loops L such that the polygonal schema associated to L has the form $a_1 b_1 \bar{a}_1 \bar{b}_1 \dots a_g b_g \bar{a}_g \bar{b}_g$ (Figure 1). A **canonical system of loops for a non-orientable surface** of genus g is a system of one-sided loops L such that the polygonal schema associated to L has the form $a_1 a_1 a_2 a_2 \dots a_g a_g$ (Figure 2). We will use the following result by Lazarus et al. [26] to compute a canonical system of loops of an orientable surface:

► **Lemma 2.2** ([26]). *Let (S, G^*) be an orientable cross-metric surface with genus g and size n , and let b be an arbitrary point in (S, G^*) . We can compute a canonical system of loops of (S, G^*) based at b in $O(gn)$ time, such that each loop has multiplicity at most four.*

2.4 Topological characterizations via signatures

Let $L = (\ell_1, \dots, \ell_g)$ be any system of loops and let p be a path on \mathcal{S} in general position with respect to L . The **signature** $\sigma(p, L)$ of p with respect to L is the vector in \mathbb{Z}_2^g whose i th component is the mod two number of crossings of p with ℓ_i . (Although we will not use this, we remark that $\sigma(p, L)$ expresses the homology of p in the homology basis dual to L [8].)

The following lemma relates the topological type of a curve on a non-orientable surface to its signature with respect to a canonical system of loops. The proof is omitted.

► **Lemma 2.3** (see Schaefer and Štefankovič [30]). *Let L be a canonical system of loops on a non-orientable surface and γ a simple closed curve in general position with respect to L .*

1. γ is one-sided if and only if $\sigma(\gamma, L)$ has an odd number of 1 elements;
2. γ is orienting if and only if all the elements in $\sigma(\gamma, L)$ are 1;
3. γ is separating if and only if all the elements in $\sigma(\gamma, L)$ are 0.

Hence, orienting curves are non-separating, and their sidedness is prescribed by the genus.

3 Main technical result

All our algorithms will use the following theorem. As a motivation, we first show how Theorems 1.2 and 1.3 follow from it, while postponing its proof until the end of the paper. The reader may as well skip this section at first reading, and come back to it later.

► **Theorem 3.1.** *Let (\mathcal{S}, G^*) be a cross-metric surface of Euler genus g and size n . Let k be an integer, $\rho : \mathbb{Z}_2^g \rightarrow \mathbb{Z}_2^k$ a linear map, and $A \subseteq \mathbb{Z}_2^k$.*

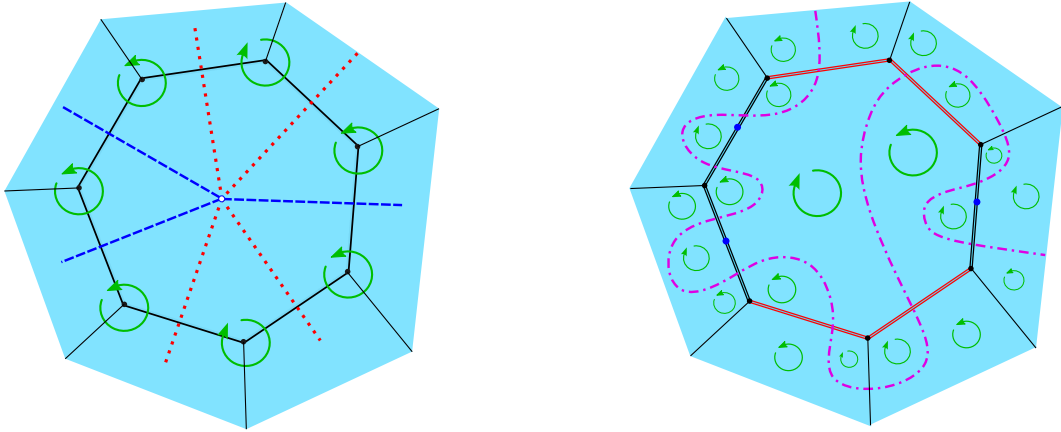
Then, for some (unspecified) canonical system of loops L that depends only on (\mathcal{S}, G^) , we can compute in $O(g^3 8^k k n \log n)$ time a shortest closed curve c on (\mathcal{S}, G^*) such that $\rho(\sigma(c, L)) \in A$ (if such a curve exists). This curve is simple and has multiplicity at most two.*

We emphasize that the canonical system of loops L is not provided in the input of the algorithm, and is actually never computed explicitly.

Proof of Theorem 1.2, assuming Theorem 3.1. Recall that a curve is orienting if and only if it crosses every loop of a canonical system of loops an odd number of times (Lemma 2.3). Thus, to compute a shortest orienting closed curve, we apply Theorem 3.1 with $k = g$, ρ the identity, and A a single vector, the all-ones vector. ◀

Proof of Theorem 1.3, assuming Theorem 3.1. Let $L = \{\ell_1, \dots, \ell_g\}$ be the canonical system of loops from Theorem 3.1 (which we know exists, even if we do not compute it). Let c be a closed curve. A loop ℓ_i is *even* (with respect to c) if ℓ_i and c cross an even number of times. Similarly, ℓ_i is *odd* if ℓ_i and c cross an odd number of times. The *oddity number* of c is the number of odd loops with respect to c . We first consider the problem of computing shortest (non-separating) non-orienting, one-sided closed curves:

- If g is even, recall from Lemma 2.3 that a closed curve is non-separating, non-orienting, and one-sided if and only if its oddity number is odd. To compute a shortest such curve, it suffices to apply Theorem 3.1 with $k = 1$, $\rho(c_1, \dots, c_g) = \sum_i c_i$, and $A = \{1\}$.
- If g is odd, recall from Lemma 2.3 that a closed curve is non-separating, non-orienting, and one-sided if and only if its oddity number is odd, but different from g . To compute a shortest such curve, we do the following. For every $i = 1, \dots, g$, we apply Theorem 3.1 with $k = 2$, $\rho(c_1, \dots, c_g) = (c_1 + \dots + c_g, c_i)$, and $A = (1, 0)$. This computes a shortest closed curve with odd oddity number such that the i th loop of L is even. We return the shortest curve over all $i = 1, \dots, g$.



■ **Figure 3** Left: G and G^* with arbitrary orientations of the faces of G and the inconsistent edges of G (in red, dotted lines); the dual graph G^* is denoted in black lines. Right: an example of a cycle passing through all the edges and the derived orientation.

The case of shortest non-separating, non-orienting, two-sided closed curves is analogous and therefore omitted. ◀

4 Computing an orienting curve

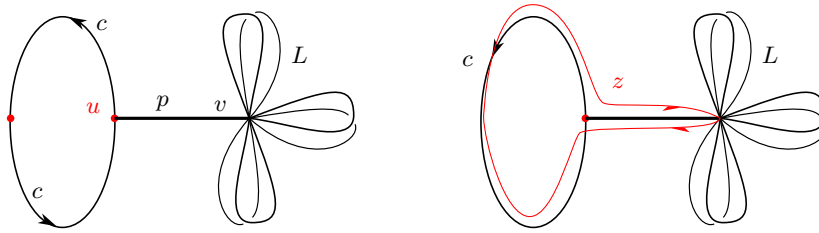
► **Proposition 4.1.** *Let (\mathcal{S}, G^*) be a non-orientable cross-metric surface of size n . We can compute an orienting curve on (\mathcal{S}, G^*) with multiplicity at most two in time $O(n)$.*

Matoušek et al. [28, Proposition 5.5] proved the existence of such curve. We improve on their argument by providing a linear-time algorithm for its computation. We will use the following lemma (below and in Section 9), which is a variation of a very classical result: Every connected graph with even degrees has an Eulerian cycle.

► **Lemma 4.2.** *Let (\mathcal{S}, G) be a combinatorial surface of size n . Let E be the set of edges of G . Let $\mu : E \rightarrow \mathbb{Z}_+$ be a map such that (i) for each vertex v of G , the sum of the values of $\mu(e)$, for each e incident to v , is even, and (ii) the subgraph of G induced by the edges e such that $\mu(e) \geq 1$ is connected. Then one can compute a simple cycle in the dual cross-metric surface (\mathcal{S}, G^*) such that, for each edge e of G , the cycle crosses the dual edge e^* exactly $\mu(e)$ times, in time linear in n plus the sum of the values of μ .*

We emphasize that, in Condition (ii), the considered subgraph cannot have isolated vertices, because it is induced by a set of edges, even though G itself may have vertices incident only to edges e such that $\mu(e) = 0$.

Proof of Proposition 4.1. Let G be the graph dual to G^* . We first choose an arbitrary orientation of every face of G . Let I be the set of *inconsistent* edges of G , which bound two faces of G with inconsistent orientations. If we start traversing the faces incident to a vertex of G by going around the vertex, in a complete turn, we must change orientations an even number of times to get back to the orientation of the initial face; this implies that each vertex of G is incident to an even number of edges in I . Thus the subgraph of G made of the edges in I has all its vertices of even degree, and cutting the surface along it yields an orientable surface. Moreover, we can compute I in linear time. See Figure 3. Now, for each edge e of G , let $\mu(e) = 1$ if $e \in I$ and $\mu(e) = 2$ otherwise. We apply Lemma 4.2 in the combinatorial



■ **Figure 4** The proof of Proposition 5.1, in the case where g is odd. Left: A pictorial view of the orientable surface \mathcal{S}' . The one-sided curve c is repeated twice on the boundary B of \mathcal{S}' . Right: On \mathcal{S} , the construction of the loop z , which is one-sided because c is one-sided.

surface (\mathcal{S}, G) (the hypotheses are obviously satisfied) to compute, in linear time, a simple cycle c in the cross-metric surface (\mathcal{S}, G^*) that crosses each edge e of G^* exactly $\mu(e)$ times. It is not hard to see that c is orienting (see Figure 3, right) which concludes the proof. ◀

5 Computing a standard system of loops

In this section, we introduce standard systems of loops and show that we can compute one efficiently. We believe that this result can be of independent interest: It would perhaps be more natural to compute a canonical system of loops; but on non-orientable surfaces it is only known to compute one in polynomial time [22] (the precise worst-case running time is certainly larger than $O(gn)$). We thus propose this alternate notion of *standard* systems of loops, which can be computed as quickly as the canonical systems of loops for orientable surfaces. In the next section, we show that for our purposes we can convert from one representation (in terms of parity of crossings) to the other.

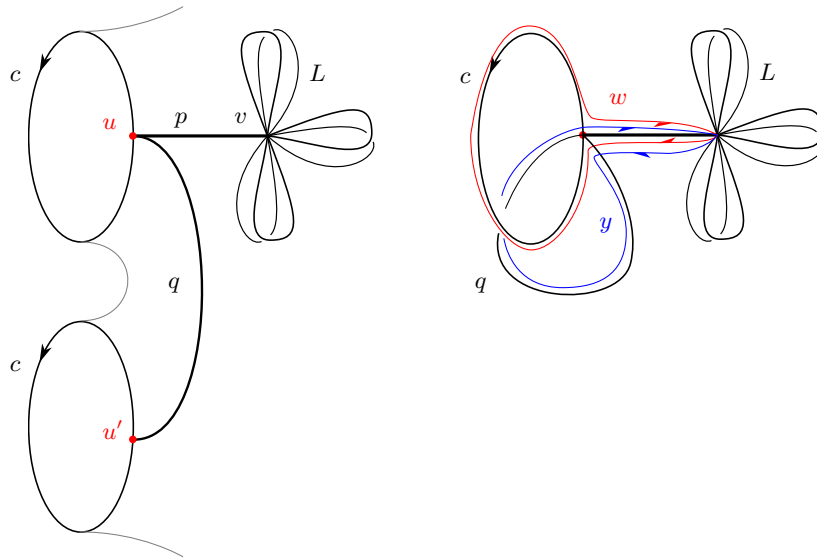
A **standard system of loops** of a non-orientable surface \mathcal{S} of genus g is a system of loops such that the loops appear in the following order around the boundary of the corresponding polygonal schema (where bar denotes reversal and $p = \lfloor \frac{g-1}{2} \rfloor$): $zza_1b_1\bar{a}_1\bar{b}_1 \dots a_p b_p \bar{a}_p \bar{b}_p$ if g is odd, and $yw\bar{y}wa_1b_1\bar{a}_1\bar{b}_1 \dots a_p b_p \bar{a}_p \bar{b}_p$ if g is even.

► **Proposition 5.1.** *Let (\mathcal{S}, G^*) be a non-orientable cross-metric surface of genus g and size n . In $O(gn)$ time, we can compute a standard system of loops on (\mathcal{S}, G^*) such that each loop has multiplicity at most 100.*

Proof. We first compute an orienting simple closed curve c with multiplicity at most two (Proposition 4.1). Let \mathcal{S}' be the orientable surface with boundary obtained by cutting \mathcal{S} along c . We remark that each edge of G^* corresponds to at most three edges in \mathcal{S}' , and these edges induce naturally a cross-metric structure on \mathcal{S}' .

Let us first assume that g is odd. Then c is one-sided, and \mathcal{S}' has a single boundary component B . Let u be an arbitrary point of B . Let $\bar{\mathcal{S}}'$ be the surface obtained by attaching a disk to the boundary component of \mathcal{S}' . In a first step, in $O(gn)$ time, we compute a canonical system of loops L of $\bar{\mathcal{S}}'$, with basepoint v in the face incident with u , that has multiplicity at most four and does not cross any edge of B . We do this as follows. First, starting from \mathcal{S}' , we shrink the boundary component B to a point. Then, we apply the algorithm by Lazarus et al. [26] (Lemma 2.2), obtaining a canonical system of loops of the resulting surface based at v . Finally, we expand back the point to the boundary component B .

Let us now connect u to the basepoint v with a path p that arrives at v at a suitable corner around v (Figure 4). By this we mean the following: The cyclic ordering of the edges at v is $a_1\bar{b}_1\bar{a}_1b_1 \dots a_g\bar{b}_g\bar{a}_gb_g$ (where bar indicates the origin of an edge, and no bar indicates



■ **Figure 5** The proof of Proposition 5.1, in the case where g is even. Left: A pictorial view of the orientable surface \mathcal{S}' . The two-sided curve c corresponds to two boundary components of \mathcal{S}' . Right: On \mathcal{S} , the construction of the loops w and y .

the target of an edge), see Figure 1; we make p arrive between two consecutive groups of the form $a_i \bar{b}_i \bar{a}_i b_i$. For this purpose, let f be the face containing the basepoint v of L ; it is incident with u . The loops cut f into *subfaces*; let f' be the subface incident with u . Starting from u in f' , we draw a path p that goes to a portion of L that lies on the boundary of f' , and then runs along L (possibly exiting f) until we get to the basepoint v at a suitable corner. Because, when running along the boundary of the polygonal schema, the suitable corners appear regularly, every four corners (see Figure 1), this can be done by running along at most two loops; each of these loops has multiplicity at most four on \mathcal{S}' , and thus p has multiplicity at most eight on \mathcal{S}' . This path p is computed in $O(n)$ time.

Finally, the set of loops L , together with the loop z that is the concatenation of the reversal of p , c , and (a slightly translated copy of) p , is a standard system of loops of \mathcal{S} , which is computed in $O(gn)$ time. Moreover, every edge of G^* is crossed at most twice by c and at most 24 times by p , so the multiplicity of each loop is at most 50 with respect to G^* .

The case where g is even is slightly more complicated, since cutting along c results in two boundary components, which we must reconnect before applying the same technique as above. See Figure 5. The details are omitted. ◀

6 Converting between canonical and standard signatures

Our next lemma describes a matrix allowing to change the coordinates in homology, from the standard basis to some canonical one. The motivation is that the topological properties we need are best expressed in terms of crossings with a canonical system of loops, while we can more easily compute a standard system of loops. The proof is omitted.

► **Lemma 6.1.** *Assume L' is a standard system of loops on a non-orientable surface. There exists a canonical system of loops L such that, for each path p , we have $\sigma(p, L) = \varphi(\sigma(p, L'))$, where $\varphi : \mathbb{Z}_2^g \rightarrow \mathbb{Z}_2^g$ is the invertible linear map described by the following matrices depending on the parity of g .*

$$A_{\text{odd}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{bmatrix} \quad A_{\text{even}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 1 & 0 & 1 \end{bmatrix}$$

We remark that we actually never compute the canonical system of loops L .

7 Subhomology covers

In this section, we introduce subhomology covers, which are covering spaces related to quotients of the homology group. They generalize *cyclic double covers* introduced by Erickson [14] and used by Borradaile et al. [3] and are inspired from *homology covers* by Chambers et al. [8], but there are important differences with the latter: They can capture not necessarily the homology group, but arbitrary quotients of the homology group (which allows for faster algorithms), and they are defined on surfaces without boundary. This tool is not restricted to non-orientable surfaces and we present it for arbitrary surfaces. Our construction is inspired from the *voltage construction* by Gross and Tucker [25, Chapter 4].

Throughout this section, let (\mathcal{S}, G) be a combinatorial surface (orientable or not) of Euler genus g and size n , L a system of loops in general position with respect to G (we will only need the case where L is the standard system of loops, computed in Proposition 5.1, but we do not assume this for the construction). Furthermore, let k be an integer and $\rho : \mathbb{Z}_2^g \rightarrow \mathbb{Z}_2^k$ a linear map. We define $\alpha : E \rightarrow \mathbb{Z}_2^k$ by $\alpha(e) = \rho(\sigma(e, L))$.

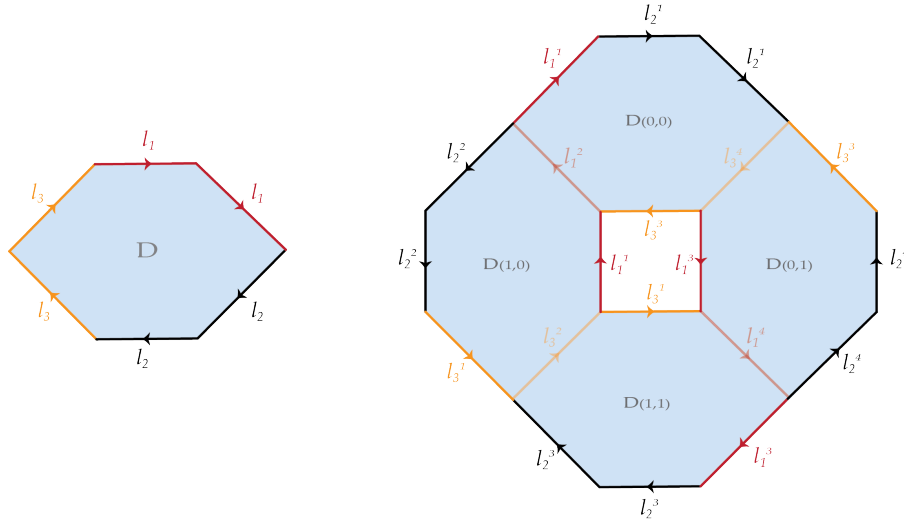
► **Lemma 7.1.** *The map α satisfies the Kirchhoff voltage law on every face: For every face of G with boundary edges e_1, \dots, e_m , we have that $\sum_{i=1}^m \alpha(e_i) = 0$.*

Proof. The boundary of a face bounds a disk and consequently is separating. Thus every loop in L intersects it an even number of times. ◀

We define the graph \tilde{G} as a graph with vertices (v, ν) , one for every vertex v in G and ν in \mathbb{Z}_2^k . For each edge e in G , connecting the vertices u and v , and every ν in \mathbb{Z}_2^k , there is an edge in \tilde{G} connecting (u, ν) and $(v, \nu + \alpha(e))$. We now observe that each facial cycle in G (which bounds a face of G in \mathcal{S}) lifts to a cycle in \tilde{G} , by the Kirchhoff voltage law. By attaching a disk to the lift of each such cycle, we obtain a combinatorial surface $(\tilde{\mathcal{S}}, \tilde{G})$, which is naturally a (possibly non-connected) covering space of \mathcal{S} and is called the **subhomology cover associated to α** . An alternate way to build $(\tilde{\mathcal{S}}, \tilde{G})$ is to cut \mathcal{S} along L , obtaining a disk D , and to glue together 2^k copies of these disks, in such a way that the copy ν gets attached via a lift of loop $\ell_i \in L$ to the copy $\nu + \rho(s_i)$, where $s_i \in \mathbb{Z}_2^g$ has a single non-zero entry, the i th one (Figure 6); however, we will not need this equivalence.

► **Lemma 7.2.** *Assume that one is given the combinatorial surface (\mathcal{S}, G) together with the map α . Then, in $O(2^k kn)$ time, we can compute the combinatorial surface $(\tilde{\mathcal{S}}, \tilde{G})$ that is the subhomology cover of (\mathcal{S}, G) associated to α . Moreover, $\chi(\tilde{\mathcal{S}}) = 2^k \chi(\mathcal{S})$.*

Again, the system of loops L is not part of the input; only the map α is specified.



■ **Figure 6** Left: A non-orientable surface of genus 3 cut along a canonical system of loops. Right: A non-orientable 4-sheeted covering space of the surface at left with genus 6 given by $\rho : \mathbb{Z}_3^k \rightarrow \mathbb{Z}_2^k$; $\rho(s_1) = (1, 0)$, $\rho(s_2) = (0, 0)$ and $\rho(s_3) = (0, 1)$. The notation l_i^j indicates the j th lift of the loop l_i .

Proof. The vertices of \tilde{G} are given by the lifts of the vertices of G , that is, by pairs (v, ν) for every vertex v of G and every $\nu \in \mathbb{Z}_2^k$. Every edge e from v to w in G lifts, given $\nu \in \mathbb{Z}_2^k$, to an edge from (v, ν) to $(w, \nu + \alpha(e))$. Finally, each facial cycle in G , incident to vertex v of G , lifts to 2^k faces of \tilde{G} in $\tilde{\mathcal{S}}$, each incident to (v, ν) for $\nu \in \mathbb{Z}_2^k$, as explained above.

The combinatorial map of \tilde{G} can thus be computed in $O(2^k kn)$ time. A more explicit construction would depend on the data structure used. For example, in the graph-encoded map data structure [27, 13], each flag f of G incident to vertex v corresponds to 2^k flags of \tilde{G} , denoted $\{(f, \nu) \mid \nu \in \mathbb{Z}_2^k\}$ where (f, ν) has associated vertex (v, ν) , and we can easily connect the flags via their three involutions, in overall $O(2^k kn)$ time.

The last claim follows from the fact that every vertex, edge, and face of G on \mathcal{S} corresponds to 2^k vertices, edges, and faces of \tilde{G} on $\tilde{\mathcal{S}}$, respectively. ◀

▶ **Lemma 7.3.** *Let c be a closed walk in G , and \tilde{c} be a lift of c in \tilde{G} , with endpoints (v, ν) and (v, ν') . Then $\rho(\sigma(c, L)) = \nu' - \nu$.*

Proof. By construction, if e connects w with w' , then each lift of e connects vertices (w, η) to $(w', \eta + \alpha(e))$ for some $\eta \in \mathbb{Z}_2^k$. Thus $\nu' - \nu$ equals the sum, over all edges e of c , of $\alpha(e) = \rho(\sigma(e, L))$. This, in turn, equals $\rho(\sigma(c, L))$. ◀

8 Computing shortest closed walks with restriction on the signature

Our algorithm will use the subroutine given in the following proposition, which is a variation on the strategy used earlier by Chambers et al. [8, Section 5.2]. It is a first step towards the proof of Theorem 3.1; however, instead of computing a simple closed curve in the cross-metric surface, we only compute a closed walk in the dual combinatorial surface.

▶ **Proposition 8.1.** *Let (\mathcal{S}, G) be a combinatorial surface of Euler genus g and size n . Moreover, let k be an integer, $\rho : \mathbb{Z}_2^g \rightarrow \mathbb{Z}_2^k$ a linear map, and $A \subseteq \mathbb{Z}_2^k$. Assume that for each edge e of G , the value of $\rho(\sigma(e, L))$ is given, for a fixed system of loops L of \mathcal{S} in general position with respect to G . Given this, we can compute in $O(g^3 8^k kn \log n)$ time a shortest closed curve c in G (a shortest closed walk) such that $\rho(\sigma(c, L)) \in A$.*

For the proof, we start by computing the subhomology cover $(\tilde{\mathcal{S}}, \tilde{G})$ associated to $\alpha(e) = \rho(\sigma(e, L))$ using Lemma 7.2. We also apply Lemma 2.1 to compute a set of $2g$ shortest paths p_1, \dots, p_{2g} in G that intersects every non-contractible closed walk in G .

The following lemma is inspired by Chambers et al. [8, Lemma 5.4].

► **Lemma 8.2.** *Some shortest closed curve c in G such that $\rho(\sigma(c, L)) \in A$ has the following form: It is the projection of a shortest path in $(\tilde{\mathcal{S}}, \tilde{G})$ that starts with a subpath of a lift of some p_i and is otherwise disjoint from that lift.*

We will also need the following immediate consequence of results by Erickson et al. [15] and Cabello et al. [5].

► **Lemma 8.3.** *Given a combinatorial surface (\mathcal{S}, G) , orientable or not, of Euler genus g and size n , with a distinguished face b , one can, after an $O(g^2 n \log n)$ -time preprocessing, compute the distance from any given vertex incident with b to any other vertex in $O(\log n)$ time.*

Proof of Proposition 8.1. As indicated above, we first compute the subhomology cover $(\tilde{\mathcal{S}}, \tilde{G})$ in $O(2^k kn)$ time (Lemma 7.2), and the paths p_1, \dots, p_{2g} in $O(gn + n \log n)$ time (Lemma 2.1).

Fix $i = 1, \dots, 2g$. We show below how to compute a shortest path in $(\tilde{\mathcal{S}}, \tilde{G})$ that starts with a subpath of a lift of p_i , is otherwise disjoint from that lift, and projects to a closed curve c such that $\rho(\sigma(c, L)) \in A$.

Let \tilde{p}_i be a lift of p_i and let $(v_0, \nu_0), (v_1, \nu_1), \dots, (v_m, \nu_m)$ be the sequence of vertices on \tilde{p}_i . We consider the combinatorial surface $\tilde{\mathcal{S}}'$ obtained by cutting $\tilde{\mathcal{S}}$ along the path \tilde{p}_i , thus forming a boundary, and then attaching a disk b to the resulting boundary component; each interior vertex (v_t, ν_t) of \tilde{p}_i , $1 \leq t \leq m - 1$, now corresponds to two vertices, $(v_t, \nu_t)^+$ and $(v_t, \nu_t)^-$; each duplicated edge has the same weight as its original. The vertices $(v_t, \nu_t)^\pm$ ($0 \leq t \leq m$, where for convenience $(v_0, \nu_0)^\pm$ and $(v_m, \nu_m)^\pm$ denote (v_0, ν_0) and (v_m, ν_m) , respectively) all lie on the boundary of the face b of $\tilde{\mathcal{S}}'$. By Lemma 7.3, it suffices to compute a shortest path, in this combinatorial surface, among all paths from some vertex $(v_t, \nu_t)^\pm$ to some corresponding vertex in the set $\{(v_t, \nu_t + a) \mid a \in A\}$. Lemma 8.3 allows us to do this: $(\tilde{\mathcal{S}}, \tilde{G})$ is a combinatorial surface of size $O(2^k n)$ and genus $O(2^k g)$ (by Lemma 7.2), and the same holds for the combinatorial surface in which we perform the computation; moreover, there are $O(2^k n)$ pairs of vertices between which we need to compute the distance. Thus, the preprocessing step takes $O(g^2 2^{3k} kn \log n)$ time, and the distance computations take $O(2^k kn \log n)$. Once we have computed a shortest distance between these pairs of points, we can compute an actual shortest path using Dijkstra's algorithm, without overhead.

Applying this for each $i = 1, \dots, 2g$, and returning the projection of the overall shortest path, we obtain the result. ◀

9 Proof of Theorem 3.1

The proof of the following lemma is omitted.

► **Lemma 9.1.** *On a cross-metric surface (\mathcal{S}, G^*) , let L be a system of loops and c_1 and c_2 be two closed curves that cross each edge of G^* with the same parity. Then $\sigma(c_1, L) = \sigma(c_2, L)$.*

Proof of Theorem 3.1. Let (\mathcal{S}, G^*) be our input cross-metric surface, and let G be the dual graph of G^* . We start by computing, in $O(gn)$ time, a standard system of loops L' in general position with respect to G such that each loop crosses each edge of G at most 100 times; for this purpose, somewhat counterintuitively, we apply Proposition 5.1 in the cross-metric surface (\mathcal{S}, G) . For each edge e of G , we can now compute $\sigma(e, L')$, also in $O(gn)$ time.

Let $\rho' := \rho \circ \varphi$, where φ is the map from Lemma 6.1. For each edge e of G , we compute $\rho'(\sigma(e, L'))$, which by Lemma 6.1 equals $\rho(\sigma(e, L))$, for some fixed canonical system of loops L . This takes $O(kgn)$ time. We then apply Proposition 8.1 in the combinatorial surface (\mathcal{S}, G) with the map ρ' and the standard system of loops L' : In $O(g^3 8^k kn \log n)$ time, we compute a shortest closed walk c in G such that $\rho'(\sigma(c, L')) \in A$, or equivalently $\rho(\sigma(c, L)) \in A$.

The remaining part of the proof is to turn c into a simple cycle. First, we build a map μ from the edges of G^* to $\{0, 1, 2\}$ based on c , as follows. Let e be an edge of G . If c does not traverse e , then we set $\mu(e) = 0$. If c traverses e a positive, even number of times, then we set $\mu(e) = 2$. Otherwise, we set $\mu(e) = 1$. Then, we apply Lemma 4.2 to (\mathcal{S}, G) and μ . This computes in linear time a simple closed curve c' in the cross-metric surface (\mathcal{S}, G^*) among those that cross each edge e^* of G^* exactly $\mu(e)$ times. We return c' . Indeed, by definition of μ , it has multiplicity at most two and it is no longer than c , which by Lemma 9.1 implies that it is a shortest closed curve such that $\rho(\sigma(c', L)) \in A$. ◀

10 Conclusion

We conclude with several remarks. First, our main tool, Theorem 3.1, also holds for orientable surfaces. In that case, the proof is simpler, bypassing the detour with standard systems of loops: We compute a canonical system of loops using Lemma 2.2, compute the signature of every edge with respect to that system of loops, apply Proposition 8.1, and conclude as in the non-orientable case. Second, alternatively, we could bypass the computation of a standard system of loops by computing, in time polynomial in g , an arbitrary system of loops L' and then computing the “change of coordinates” matrices between L' and some canonical system of loops L , as in the statement of Lemma 6.1; the latter computation can be done by mimicking the proof of the classification theorem by Brahana [4, 26], but, at each step, remembering only $\sigma(\ell, L')$ for each loop ℓ in the current system. We omit the details here. Third, our techniques allow to compute a shortest overall one-sided closed curve in $O(g^3 n \log n)$ time (without controlling whether it is orienting or not); indeed, just apply Theorem 3.1 with $\rho(c_1, \dots, c_g) = \sum_i c_i$ and $A = \{1\}$. Fourth, our algorithms run in $O(n \log n)$ for fixed genus. It might be possible, in the case of two-sided curves, to obtain an algorithm with running time $O(n \log \log n)$, using the techniques by Chambers et al. [8, Section 4], though with a hidden dependence on the genus that is at least exponential.

Finally, we have only considered non-separating closed curves, and leave open the complexity of computing a shortest separating closed curve with a given topological type (specifying the topology of the two surfaces with boundary resulting from cutting along it). Even on orientable surfaces, the following problem is fixed-parameter tractable in the genus [7, Theorem 6.1] but apparently neither known to be NP-hard nor polynomial-time solvable: Compute the shortest simple closed curve that splits off a surface of (orientable) genus one.

References

- 1 Michael O. Albertson and Joan P. Hutchinson. The independence ratio and genus of a graph. *Transactions of the American Mathematical Society*, 226:161–173, 1977.
- 2 Mark Anthony Armstrong. *Basic topology*. Undergraduate Texts in Mathematics. Springer-Verlag, 1983.
- 3 Glencora Borradaile, Erin Wolf Chambers, Kyle Fox, and Amir Nayyeri. Minimum cycle and homology bases of surface-embedded graphs. *Journal of Computational Geometry*, 8(2):58–79, 2017.
- 4 Henry R. Brahana. Systems of circuits on 2-dimensional manifolds. *Annals of Mathematics*, 23:144–168, 1921.

- 5 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-source shortest paths in embedded graphs. *SIAM Journal on Computing*, 42(4):1542–1571, 2013.
- 6 Sergio Cabello, Matt DeVos, Jeff Erickson, and Bojan Mohar. Finding one tight cycle. *ACM Transactions on Algorithms*, 6(4):Article 61, 2010.
- 7 Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Francis Lazarus, and Kim Whittlesey. Splitting (complicated) surfaces is hard. *Computational Geometry: Theory and Applications*, 41(1–2):94–110, 2008.
- 8 Erin W. Chambers, Jeff Erickson, Kyle Fox, and Amir Nayyeri. Minimum cuts in surface graphs. *SIAM Journal on Computing*, 52(1):156–195, 2023.
- 9 Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1655–1670, 2015.
- 10 Éric Colin de Verdière. Shortest cut graph of a surface with prescribed vertex set. In *Proceedings of the 18th European Symposium on Algorithms (ESA), part 2*, number 6347 in Lecture Notes in Computer Science, pages 100–111, 2010.
- 11 Éric Colin de Verdière. Computational topology of graphs on surfaces. In Jacob E. Goodman, Joseph O’Rourke, and Csaba Toth, editors, *Handbook of Discrete and Computational Geometry*, chapter 23, pages 605–636. CRC Press LLC, third edition, 2018.
- 12 Éric Colin de Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8):3784–3813, 2010.
- 13 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 599–608, 2003.
- 14 Jeff Erickson. Shortest non-trivial cycles in directed surface graphs. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SOCG)*, pages 236–243. ACM, 2011.
- 15 Jeff Erickson, Kyle Fox, and Luvsandondov Lkhamsuren. Holiest minimum-cost paths and flows in surface graphs. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1319–1332, 2018.
- 16 Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.
- 17 Jeff Erickson and Amir Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1166–1176, 2011.
- 18 Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1038–1046, 2005.
- 19 Jeff Erickson and Pratik Worah. Computing the shortest essential cycle. *Discrete & Computational Geometry*, 44(4):912–930, 2010.
- 20 Benson Farb and Dan Margalit. *A primer on mapping class groups*. Princeton University Press, 2011.
- 21 Kyle Fox. Shortest non-trivial cycles in directed and undirected surface graphs. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 352–364, 2013.
- 22 Niloufar Fuladi, Alfredo Hubard, and Arnaud de Mesmay. Short topological decompositions of non-orientable surfaces. *Discrete & Computational Geometry*, pages 1–48, 2023.
- 23 Cyril Gavoille and Claire Hilaire. Minor-universal graph for graphs on surfaces. *arXiv preprint arXiv:2305.06673*, 2023.
- 24 Mikhael Gromov. Filling Riemannian manifolds. *Journal of Differential Geometry*, 18:1–147, 1983.
- 25 Jonathan L. Gross and Thomas W. Tucker. *Topological graph theory*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1987. A Wiley-Interscience Publication.

28:16 Computing Shortest Closed Curves on Non-Orientable Surfaces

- 26 Francis Lazarus, Michel Pocchiola, Gert Vegter, and Anne Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SOCG)*, pages 80–89. ACM, 2001.
- 27 Sóstenes Lins. Graph-encoded maps. *Journal of Combinatorial Theory, Series B*, 32:171–181, 1982.
- 28 Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. Untangling two systems of noncrossing curves. *Israel Journal of Mathematics*, 212:37–79, 2016.
- 29 Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003.
- 30 Marcus Schaefer and Daniel Štefankovič. The degenerate crossing number and higher-genus embeddings. *Journal of Graph Algorithms and Applications*, 26(1):35–58, 2022.
- 31 Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *Journal of Combinatorial Theory, Series B*, 48(2):155–177, 1990.