

Minimum Cuts and Shortest Non-Separating Cycles via Homology Covers*

Jeff Erickson

Department of Computer Science
University of Illinois at Urbana-Champaign
jeffe@cs.uiuc.edu

Amir Nayyeri

Department of Computer Science
University of Illinois at Urbana-Champaign
nayyeri2@cs.uiuc.edu

Abstract

Let G be a directed graph with weighted edges, embedded on a surface of genus g . We describe an algorithm to compute a shortest directed cycle in G in any given \mathbb{Z}_2 -homology class in $2^{O(g)}n \log n$ time; this problem is NP-hard even for undirected graphs. We also present two applications of our algorithm. The first is an algorithm to compute a shortest non-separating directed cycle in G in $2^{O(g)}n \log n$ time, improving the recent algorithm of Cabello *et al.* [SOCG 2010] for all $g = o(\log n)$. The second is a combinatorial algorithm to compute minimum (s, t) -cuts in *undirected* surface graphs in $2^{O(g)}n \log n$ time, improving on previous combinatorial algorithms, and in particular the recent of Chambers *et al.* [SOCG 2009], for all $g = o(\log n)$. Unlike earlier algorithms for surface graphs that construct and search finite portions of the universal cover, our algorithms use another canonical covering space, called the \mathbb{Z}_2 -homology cover.

1 Introduction

Algorithms to find optimal subgraphs with certain properties in surface-embedded graphs are a basic ingredient in numerous algorithms in topological graph theory. Structures of interest include minimum spanning trees, single- and multiple-source shortest paths [7, 8, 39, 45], minimum cuts [15, 14], shortest topologically interesting cycles [6, 8, 9, 30, 46, 56], shortest paths or cycles homotopic to a given path or cycle [23, 22, 40], shortest cut graphs [21, 30, 31], and shortest generators for homotopy or homology groups [26, 31]. Applications of these basic algorithms include probabilistically embedding high-genus graphs into planar graphs [5, 47, 52], drawing abstract graphs in the plane with the fewest possible crossings [44], testing isomorphism between graphs of fixed genus [43], approximating optimal traveling salesman tours [24] and Steiner trees [1, 2, 3], and computing low-distortion surface parametrizations [54, 57].

In this paper, we describe an algorithm to compute a shortest *directed* cycle in a specified \mathbb{Z}_2 -homology class, in a *directed* n -vertex graph embedded on a surface of genus g , in $2^{O(g)}n \log n$ time. (See Section 2 for a brief overview of \mathbb{Z}_2 -homology.) We are unaware of any previously published algorithm for this problem; however, an algorithm of Chambers *et al.* [13] can be modified to find shortest homologous cycles in *undirected* graphs in $g^{O(g)}n \log n$ time. This problem can be shown to be NP-hard, even for undirected graphs, by a reduction from the traveling salesman problem in grid graphs [13].

We also describe two applications of our main result. As an immediate corollary, we obtain an algorithm to compute a shortest non-separating cycle in a surface-embedded directed graph in $2^{O(g)}n \log n$ time (Corollary 5.5). Although the corresponding problem in undirected graphs is well studied, the first nontrivial results for directed graphs were only recently published by Cabello *et al.* [9], who describe an algorithm that runs in $O(g^{1/2}n^{3/2} \log n)$ time. Our algorithm is faster when $g = o(\log n)$.

Our second application (Section 6) is a combinatorial algorithm to compute minimum (s, t) -cuts in *undirected* surface graphs in $2^{O(g)}n \log n$ time. Like Chambers *et al.* [15], we reduce the minimum cut problem to the more general (and NP-hard) problem of computing a minimum-cost *subgraph* in any given \mathbb{Z}_2 -homology class. We solve this more general problem by combining our algorithm to find minimum *cycles* in each homology class with a straightforward dynamic programming algorithm. The resulting algorithm is both simpler and faster than the minimum-cut algorithm of Chambers *et al.* [15], which runs in $g^{O(g)}n \log n$ time. When $g = o(\log n)$, our algorithm is also faster than the best combinatorial minimum-cut algorithms for arbitrary sparse graphs, which run in $O(n^2 \log n)$ time [53, 35].

Many previous algorithms for finding optimal substructures are based on a common approach, first suggested by Kutz [46]. An exchange argument implies an upper bound on the number of times that the target

*This research was partially supported by NSF grant CCF 09-15519. See <http://www.cs.uiuc.edu/~jeffe/pubs/homcover.html> for the most recent version of this paper.

structure can cross a shortest path; for example, the shortest non-separating cycle intersects any shortest path at most once [12]. Ultimately, this crossing bound follows from the observation that two shortest paths in a generic undirected surface graph cross at most once. The crossing bound implies that the target structure lies in one of a finite number of *homotopy* classes. (See Section 2 for definitions.) For each candidate homotopy class, these algorithms construct and search a relevant finite portion of the (infinite) universal cover of the input surface, using planar shortest-path algorithms [39, 45].

For directed surface graphs, however, this approach appears doomed from the start. Two *directed* shortest paths can cross arbitrarily many times; similarly, a shortest non-separating directed cycle can cross a directed shortest path arbitrarily many times. Even for undirected graphs, finding optimal cycles in a given *homology* class by enumerating *homotopy* classes seems unnecessarily baroque. In this paper, we abandon crossing bounds and homotopy methods entirely, and work directly with homology instead, at every stage of our algorithm and its analysis. Instead of the universal cover, our algorithms construct and search another canonical covering space, called the \mathbb{Z}_2 -*homology cover*, which we describe in detail in Section 4. We also use a recent generalization of Klein’s seminal multiple-source shortest path algorithm [45] to higher-genus embedded graphs [7, 8].

1.1 Related Results

Minimum cuts. Minimum cuts and maximum flows have been studied in planar graphs for more than half a century, starting with the seminal work of Ford and Fulkerson [33] and others [37]. A long series of results eventually led to an algorithm by Reif [50] to compute minimum (s, t) -cuts in *undirected* planar networks in near-linear time; Frederickson [34] later improved the running time of Reif’s algorithm to $O(n \log n)$. The only efficient method known to compute minimum cuts in *directed* planar graphs is to compute a maximum flow [58, 4, 29]. Janiga and Koubek described an adaptation of Reif’s algorithm, to *directed* planar networks in near-linear time [42]; however, their algorithm has a subtle error [49].

Surprisingly little is known about computing minimum cuts in natural generalizations of planar graphs. Chambers *et al.* [15] described an algorithm to compute a minimum (s, t) -cut in an *undirected* graph embedded on a surface of genus g , for any specified vertices s and t , in $g^{O(g)} n \log n$ time. The fastest algorithm to compute minimum (s, t) -cuts in directed surface graphs, also due to Chambers *et al.* [14], computes a maximum (s, t) -flow in near-linear time for graphs of constant genus and polynomially-bounded capacities.

Shortest non-trivial cycles. The problem of finding shortest topologically nontrivial cycles in embedded *undirected* graphs has a long history. Itai and Shiloach [41] observed that the minimum (s, t) -cut in an undirected planar graph G is dual to the minimum-cost cycle that separates faces s^* and t^* in the dual graph G^* . Thus, Frederickson’s minimum cut algorithm [34] computes the shortest nontrivial cycle in a combinatorial annulus in $O(n \log n)$ time. Thomassen [56] developed the first efficient algorithm for graphs on arbitrary surfaces, which runs in $O(n^3)$ time and exploits the so-called *3-path condition*; see also Mohar and Thomassen [48, Sect. 4.3]. Erickson and Har-Peled described a faster algorithm that runs in $O(n^2 \log n)$ time [30]. This is the fastest algorithm known for arbitrary surface-embedded graphs; however, several faster algorithms are known when the genus g of the underlying surface is small [12, 6, 46, 7, 8]. For related results, see [10, 11, 13, 32].

The history for *directed* embedded graphs is much shorter, in part because neither Thomassen’s 3-path condition nor Cabello and Mohar’s crossing condition hold. The shortest nontrivial *directed* cycle in an annular graph is dual to either the minimum (s, t) -cut or the minimum (t, s) -cut in the directed planar dual graph, whichever has smaller capacity. Both of these cuts can be computed in $O(n \log n)$ time using planar flow algorithms.¹ Very recently, Cabello *et al.* [9] describe an algorithm to find a shortest non-contractible or non-separating cycle in a directed surface graph in $O(\sqrt{g} n^{3/2} \log n)$ time, using a subtle generalization of Thomassen’s 3-path condition.

Shortest equivalent cycles. Several authors have considered the related question of finding the shortest cycle in a surface graph that is either homotopic or homologous to a given cycle. Colin de Verdière and Erickson [22] describe an algorithm to compute the shortest cycle homotopic to a given cycle in a combinatorial surface in $O(gnk \log nk)$ time, where k is the number of edges in the input cycle, improving and generalizing previous results of Colin de Verdière and Lazarus for *simple* cycles [23]. An argument of Chambers *et al.* [13] implies that finding the shortest cycle (either simple or not) in a given *homology* class in a surface graph is NP-hard; Chen and Friedman [17, 18] proved that the corresponding problem in simplicial complexes is NP-hard to approximate within any constant factor.

The minimum-cut algorithm of Chambers *et al.* [15] can be used to compute the minimum-cost *even subgraph* in a given \mathbb{Z}_2 -homology class in $g^{O(g)} n \log n$ time; this problem is also NP-hard in general. Chambers *et al.* [14]

¹It appears that Jeniga and Koubeck’s algorithm [42] always correctly computes the smaller of these two cuts.

described algorithms to find the minimum-cost *circulation* in a given real or integer homology class in a directed surface-embedded graph in polynomial time; Dey *et al.* [25] generalized this result to arbitrary chains of arbitrary dimension in arbitrary simplicial complexes. For other related results, see [19, 26, 31, 59].

2 Notation and Terminology

We begin by recalling several useful definitions related to surface-embedded graphs. For further background, we refer the reader to Gross and Tucker [36] or Mohar and Thomassen [48] for topological graph theory, and to Hatcher [38] or Stillwell [55] for surface topology and homology.

Surfaces and curves. A *surface* (more formally, a *2-manifold with boundary*) Σ is a compact Hausdorff space in which every point has an open neighborhood homeomorphic to either the plane \mathbb{R}^2 or a closed halfplane $\{(x, y) \in \mathbb{R}^2 \mid x \geq 0\}$. The points with halfplane neighborhoods make up the *boundary* of Σ ; the complement of the boundary is the *interior* of Σ . Every component of the boundary is homeomorphic to a circle. A surface is *non-orientable* if it contains a subset homeomorphic to the Möbius band, and *orientable* otherwise.

A *path* in a surface Σ is a continuous function $p: [0, 1] \rightarrow \Sigma$. A *loop* is a path whose endpoints $p(0)$ and $p(1)$ coincide; we refer to this common endpoint as the *basepoint* of the loop. An *arc* is a path whose endpoints lie on the boundary of Σ . A *cycle* is a continuous function $\gamma: S^1 \rightarrow \Sigma$; the only difference between a cycle and a loop is that a loop has a distinguished basepoint. We collectively refer to paths, loops, arcs, and cycles as *curves*. A curve is *simple* if it is injective, except for the basepoint in the case of loops; we usually do not distinguish between simple curves and their images in Σ . The *reversal* $\text{rev}(p)$ of a path p is defined by setting $\text{rev}(p)(t) = p(1 - t)$. The *concatenation* $p \cdot q$ of two paths p and q with $p(1) = q(0)$ is the path created by setting $(p \cdot q)(t) = p(2t)$ for all $t \leq 1/2$ and $(p \cdot q)(t) = q(2t - 1)$ for all $t \geq 1/2$.

The *genus* of a surface Σ is the maximum number of disjoint simple cycles $\gamma_1, \gamma_2, \dots, \gamma_g$ in the interior of Σ whose complement $\Sigma \setminus (\gamma_1 \cup \dots \cup \gamma_g)$ is connected. We will consider only compact, connected, orientable surfaces. Up to homeomorphism, there is exactly one such surface with any genus $g \geq 0$ and any number of boundary components $b \geq 0$; the *Euler characteristic* χ of this surface is $\chi := 2 - 2g + b$.

Graphs, embeddings, and duality. An *embedding* of an undirected graph G on a surface Σ maps vertices to distinct points and edges to interior-disjoint curves.

The *faces* of the embedding are maximal connected subsets of Σ that are disjoint from the image of the graph. An embedding is *cellular* if each of its faces is homeomorphic to the plane; in particular, in any cellular embedding, each component of the boundary of Σ must be covered by a cycle of edges in G . Euler’s formula implies that any cellularly embedded graph with n vertices, m edges, and f faces lies on a surface with Euler characteristic $\chi = n - m + f$, which implies that $m = O(n + g)$ and $f = O(n + g)$. We consider only cellular embeddings of genus $g = O(n)$, so that the overall complexity of the embedding is $O(n)$.

Any undirected graph G embedded on a surface Σ with boundary has a *dual graph* G^* , defined as follows.² The dual graph G^* has a vertex f^* for each face f of G , including the boundary cycles, and an edge e^* for each edge e in G (including boundary edges) joining the vertices dual to the faces that e separates. For each boundary cycle δ of G , we refer to the corresponding vertex δ^* of G^* as a *dual boundary vertex*. The dual graph G^* has a natural cellular embedding in the surface Σ^* obtained from Σ by gluing a disk to each boundary cycle; each face of this embedding corresponds to a vertex of G . See Figure 1. (Duality can be extended to directed graphs [14], but our results do not require this extension.)

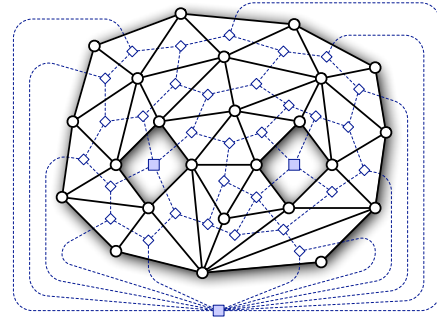


Figure 1. A cellularly embedded graph G (solid lines) on a pair of pants (genus 0 with 3 boundaries), and its dual graph G^* (dashed lines). Dual boundary vertices are indicated by squares.

For any subgraph $F = (U, D)$ of $G = (V, E)$, we write $G \setminus F$ to denote the edge-complement $(V, E \setminus D)$. Also, when the graph G is fixed, we abuse notation by writing F^* to denote the subgraph of G^* corresponding to a subgraph F of G ; each edge in F^* is the dual of a unique edge in F . In particular, we have the identity $(G \setminus F)^* = G^* \setminus F^*$.

For most of the problems we consider, the input consists of a *directed* edge-weighted graph G with a cellular embedding on some surface. We use the notation

²Our definition differs slightly from the one proposed by Erickson and Colin de Verdière [22].

$u \rightarrow v$ to denote the directed edge from vertex u to vertex v . Without loss of generality, we consider only *symmetric* directed graphs, in which the reversal $v \rightarrow u$ of any edge $u \rightarrow v$ is another edge. We also assume that in the cellular embedding, the images of any edge and its reversal coincide (but with different orientations). Thus, like Cabello *et al.* [9, Section 2.3], we implicitly model directed graphs as *undirected graphs with asymmetric edge weights*.

Cellular graph embeddings are equivalent to the **combinatorial surfaces** introduced by Colin de Verdière [20] and used by several authors to formulate optimization problems for surface-embedded graphs. A combinatorial surface consists of an abstract surface Σ together with a cellularly embedded graph G with (possibly asymmetrically) weighted edges. Paths and cycles in a combinatorial surface are directed walks in its graph; the length of any such walk is the sum of its (directed) edge weights, counted with appropriate multiplicity.

Homotopy and homology. Two paths α and β in Σ are *homotopic* if one can be continuously deformed into the other. More formally, a **homotopy** between α and β is a continuous map $h: [0, 1] \times [0, 1] \rightarrow \Sigma$ such that $h(0, \cdot) = \alpha$, $h(1, \cdot) = \beta$, $h(\cdot, 0) = \alpha(0) = \beta(0)$, and $h(\cdot, 1) = \alpha(1) = \beta(1)$. Homotopy defines an equivalence relation over the set of paths with any fixed pair of endpoints. The set of homotopy classes of loops in Σ with basepoint x defines a group $\pi_1(\Sigma, x)$ under concatenation, called the *fundamental group* of Σ . (For all points x and y , the groups $\pi_1(\Sigma, x)$ and $\pi_1(\Sigma, y)$ are isomorphic.)

A cycle γ is *contractible* if it is homotopic to a constant map; a simple cycle γ is **separating** if $\Sigma \setminus \gamma$ is disconnected.

Homology is a coarser equivalence relation than homotopy, with nicer algebraic properties; intuitively, two cycles are homologous if together they define the boundary of some subset of the surface. Like several earlier papers [16, 17, 26, 27, 15], we will consider only one-dimensional cellular homology with coefficients in the finite field \mathbb{Z}_2 ; this restriction allows us to radically simplify our definitions.

Fix a cellular embedding of an undirected graph G on a surface Σ with genus g and b boundaries. An **even subgraph** is a subgraph of G in which every node has even degree, or equivalently, the union of edge-disjoint cycles. An even subgraph is **null-homologous** if it is the boundary of the union of a subset of faces of G . Two even subgraphs η and η' are **homologous**, or in the same **homology class**, if their symmetric difference $\eta \oplus \eta'$ is null-homologous; See Figure 2. The set of all homology classes of even subgraphs defines the **first homology group** of Σ , denoted $H_1(\Sigma; \mathbb{Z}_2)$, which is isomorphic to the finite vector space $(\mathbb{Z}_2)^\beta$, where

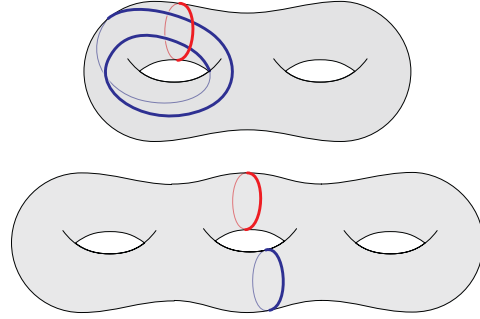


Figure 2. Pairs of cycles that are homologous but not homotopic. (Lighter portions of the curves are on the back side of the surface.)

$\beta = 2g + \max\{b - 1, 0\}$. The rank β is called the **first Betti number** of Σ .

We define the **carrier** of a (not necessarily simple) cycle γ in G to be the even subgraph of edges that γ traverses an odd number of times; two cycles are homologous if their carriers are homologous. A cycle γ is **\mathbb{Z}_2 -minimal** if it has minimum total length among all cycles homologous with γ . Similarly, a loop ℓ with basepoint v is **\mathbb{Z}_2 -minimal** if it has minimum length among all loops based at v that are homologous with ℓ . Every \mathbb{Z}_2 -minimal cycle can be regarded as a \mathbb{Z}_2 -minimal loop through any of its vertices.

Covering maps and covering spaces. A continuous map $\pi: \Sigma' \rightarrow \Sigma$ between two surfaces is called a **covering map** if each point $x \in \Sigma$ lies in an open neighborhood $U \subset \Sigma$ such that (1) $\pi^{-1}(U)$ is a countable union of disjoint open sets $U_1 \cup U_2 \cup \dots$ and (2) for each i , the restriction $\pi|_{U_i}: U_i \rightarrow U$ is a homeomorphism. If there is a covering map π from Σ' to Σ , we call Σ' a **covering space** of Σ . The **universal cover** $\tilde{\Sigma}$ is the unique simply-connected covering space of Σ (up to homeomorphism). For any path p in Σ such that $\pi(x') = p(0)$ for some point $x' \in \Sigma'$, there is a unique path p' in Σ' , called a **lift** of p , such that $p'(0) = x'$ and $\pi \circ p' = p$. Conversely, for any path p' in Σ' , the path $\pi \circ p'$ is called a **projection** of p' .

A **deck transformation** for a covering map $\pi: \Sigma' \rightarrow \Sigma$ is an automorphism $d: \Sigma' \rightarrow \Sigma'$ such that $\pi \circ d = \pi$; the set of all deck transformations defines a group under composition. For any normal subgroup N of the fundamental group $\pi_1(\Sigma, x)$, there is a unique path-connected covering space of Σ (up to homeomorphism) whose group of deck transformations is isomorphic to the quotient group $\pi_1(\Sigma, x)/N$. For example, the universal cover $\tilde{\Sigma}$ is the unique path-connected covering space whose deck transformation group is the entire fundamental group $\pi_1(\Sigma, x)$. The **\mathbb{Z}_2 -homology cover** $\bar{\Sigma}$ of Σ is the unique path-connected covering space whose group of deck transformations is the first homology

group $H_1(\Sigma, \mathbb{Z}_2)$. We give an equivalent combinatorial definition in the next section.

3 Homology Signatures

Throughout the paper, we fix a directed graph $G = (V, E)$, a non-negative weight function $w : E \rightarrow \mathbb{R}$, and a cellular embedding of G on a surface Σ of genus g with b boundaries. Without loss of generality, we assume that the underlying surface Σ has at least one boundary; otherwise, we can remove an arbitrary face of G from Σ without affecting its homology at all. Let $\delta_1, \dots, \delta_b$ denote the boundary cycles of Σ , and let $\beta = 2g + b - 1$ denote the first Betti number of Σ .

In this section, we describe a standard method for preprocessing a combinatorial surface in $O(\beta n)$ time, so that the \mathbb{Z}_2 -homology class of any cycle γ can be computed in $O(\beta)$ time per edge. Our algorithm associates a vector of β bits with each edge e , called the *signature* of e ; the homology class of any cycle is characterized by the bit-wise exclusive-or of the signatures of its edges. These homology signatures are essentially equivalent to the *crossing parity vectors* described by Chambers *et al.* [15], but our construction is slightly more flexible.

Our construction is based on one of two natural generalizations of tree-cotree decompositions [28] to surfaces with boundary; the second generalization is introduced in Section 5. We define a *tree-coforest decomposition* of G to be any partition (T, F, X) of the edges of G into three edge-disjoint subgraphs with the following properties:

- T is a spanning tree of G .
- F^* is a spanning forest of G^* , that is, an acyclic subgraph that contains every vertex.
- Each component of F^* contains a single dual boundary vertex δ_i^* .

Euler's formula implies that there are exactly β edges in X ; arbitrarily index these edges e_1, \dots, e_β . For each edge $e_i \in X$, adding the corresponding dual edge e_i^* to F^* creates a new dual path p_i , which is either a simple path between distinct boundary vertices, or a nontrivial loop from a boundary vertex back to itself; in the second case, p_i may traverse some edges of F^* twice. We can treat each path p_i as a simple arc in the abstract surface Σ ; cutting along these β arcs transforms Σ into a topological disk. See Figure 3.

Finally, for each edge e in G , we define its signature $[e]$ to be the β -bit vector whose i th bit is equal to 1 if and only if e crosses p_i (that is, if p_i traverses the dual edge e^*) exactly once. The signature $[\eta]$ of an even subgraph η is the bitwise exclusive-or of the signatures of its edges. Similarly, the signature $[\gamma]$ of a cycle γ is the

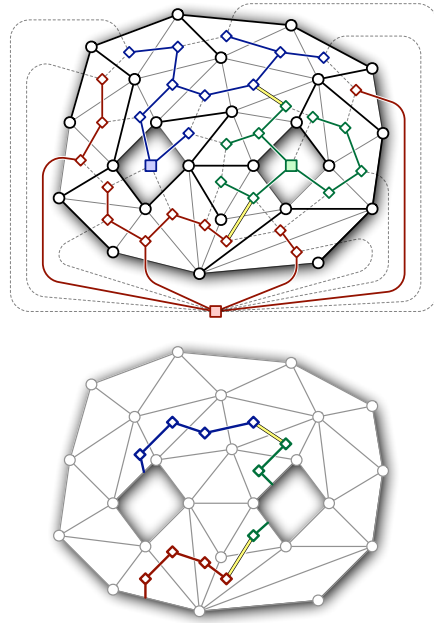


Figure 3. Top: A tree-coforest decomposition of the graph in Figure 1; doubled lines indicate edges in X . Bottom: The resulting system of dual arcs. Compare with Figure 5.

bitwise exclusive-or of the signatures of the edges that γ traverses an odd number of times.

Let $h \oplus h'$ denote the bitwise exclusive-or of two homology signatures h and h' , or equivalently, their sum as elements of the homology group $(\mathbb{Z}_2)^\beta$. The identities $[\eta \oplus \eta'] = [\eta] \oplus [\eta']$ and $[\gamma \cdot \gamma'] = [\gamma] \oplus [\gamma']$ follow directly from the definitions.

Lemma 3.1. *We can preprocess G in $O(\beta n)$ time, so that the signature $[\gamma]$ of any cycle can be computed in $O(\beta)$ time per edge.*

Proof: A tree-coforest decomposition can be computed in $O(n)$ time as follows. First construct a graph H by identifying all the dual boundary vertices in G^* to a single vertex. Compute a spanning tree of H by whatever-first search; the edges of this spanning tree define an appropriate dual spanning forest F^* . Construct the subgraph $G \setminus F$ and compute a spanning tree T via whatever-first search. Finally, let $X = G \setminus (T \cup F)$. With the decomposition in hand, it is straightforward to compute each path p_i in $O(n)$ time, and then compute each edge signature in $O(\beta)$ time. \square

Lemma 3.2. *An even subgraph η of G is null-homologous in Σ if and only if $[\eta] = 0$.*

Proof: Let η be a null-homologous even subgraph of G . Then by definition, η is the boundary of the union of a

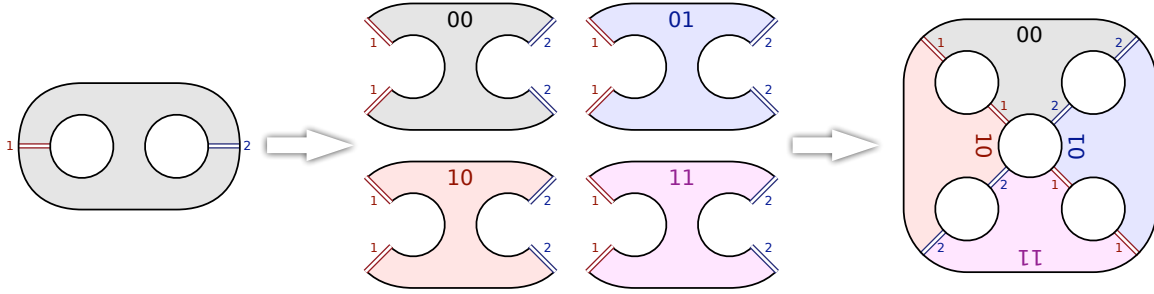


Figure 4. Constructing the \mathbb{Z}_2 -homology cover of a pair of pants (a genus zero surface with three boundaries).

subset Y of faces of G . The boundary of any face f is contractible in Σ and therefore has signature 0. It follows immediately that $[\eta] = [\bigoplus_{f \in Y} \partial f] = \bigoplus_{f \in Y} [\partial f] = 0$.

Conversely, suppose η crosses each arc p_i an even number of times, so $[\eta] = 0$. Let x and y be two intersection points between η and some arc p_i , and let $p_i[x, y]$ be the subpath of p_i between those two points. Replacing tiny segments of η through x and y with two copies of $p_i[x, y]$ does not change the homology class of η , but does reduce the number of intersection points between η and p_i . It follows by induction that η is homologous to another even graph η' that does not intersect any path p_i at all. This even graph lies entirely within the disk $\Sigma \setminus \bigcup_i p_i$, and is therefore null-homologous. \square

The following corollaries are now immediate.

Corollary 3.3. *Two even subgraphs η and η' of G are \mathbb{Z}_2 -homologous in Σ if and only if $[\eta] = [\eta']$.*

Corollary 3.4. *Two cycles γ and γ' in G are \mathbb{Z}_2 -homologous in Σ if and only if $[\gamma] = [\gamma']$.*

4 The \mathbb{Z}_2 -Homology Cover

With the homology signatures in hand, the \mathbb{Z}_2 -homology cover of a combinatorial surface can be defined using a standard *voltage construction* [36, Chapter 4], as follows. Let \overline{G} denote the graph whose vertices are all ordered pairs (v, h) where v is a vertex of G and h is an element of $(\mathbb{Z}_2)^\beta$, and whose edges are the ordered pairs $(u \rightarrow v, h) := (u, h) \rightarrow (v, h \oplus [u \rightarrow v])$ for all edges $u \rightarrow v$ of G and all homology classes $h \in (\mathbb{Z}_2)^\beta$. Let $\pi: \overline{G} \rightarrow G$ denote the covering map $\pi(v, h) = v$; this map projects any cycle in \overline{G} to a cycle in G . To define a cellular embedding of \overline{G} , we declare a cycle in \overline{G} to be a face if and only if its projection is a face of G . The combinatorial surface defined by this embedding is the \mathbb{Z}_2 -homology cover $\overline{\Sigma}$.

Our construction can be interpreted more topologically as follows. Let p_1, \dots, p_β denote the system of dual arcs used to define the homology signatures [e]. The

surface $D := \Sigma \setminus (p_1 \cup \dots \cup p_\beta)$ is a topological disk. Each arc p_i appears on the boundary of D as two segments p_i^+ and p_i^- . For each signature $h \in (\mathbb{Z}_2)^\beta$, we create a disjoint copy (D, h) of D ; for each index i , let (p_i^+, h) and (p_i^-, h) denote the copies of p_i^+ and p_i^- in the disk (D, h) . For each index i , let b_i denote the β -bit vector whose i th bit is equal 1 and whose other $\beta - 1$ bits are all equal to 0. The \mathbb{Z}_2 -homology cover $\overline{\Sigma}$ is constructed by gluing the 2^β copies of D together by identifying boundary paths (p_i^+, h) and $(p_i^-, h \oplus b_i)$, for every index i and homology class h . See Figure 4 for an example.

Lemma 4.1. *The combinatorial surface $\overline{\Sigma}$ has $\overline{n} = 2^\beta n$ vertices, genus $\overline{g} = O(2^\beta \beta)$, and $\overline{b} = O(2^\beta b)$ boundaries, and it can be constructed in $O(2^\beta n)$ time.*

Proof: Let m and f denote the number of edges and faces of Σ , respectively. Recall that the Euler characteristic of Σ is $\chi = n - m + f = 2 - 2g - b = 1 - \beta$. The combinatorial surface $\overline{\Sigma}$ has exactly $\overline{n} = 2^\beta n$ vertices, $2^\beta m$ edges, and $2^\beta f$ faces, so its Euler characteristic is $\overline{\chi} = 2^\beta(1 - \beta)$.

If $b > 1$, then each boundary cycle δ_i has a non-zero homology signature; at least one arc p_j has exactly one endpoint on δ_i . Thus, $\overline{\Sigma}$ has exactly $\overline{b} = 2^{\beta-1} b$ boundary cycles, each of which is a double-cover (in fact, the \mathbb{Z}_2 -homology cover) of some boundary cycle δ_i . It follows that $\overline{\Sigma}$ has genus $\overline{g} = 1 - (\overline{\chi} + \overline{b})/2 = 2^{\beta-2}(4g + b - 4) + 1$. (Somewhat surprisingly, $\overline{\Sigma}$ may have positive genus even when Σ does not!) On the other hand, when $b = 1$, the boundary cycle δ_1 is null-homologous, so $\overline{\Sigma}$ has $\overline{b} = 2^\beta b$ boundary cycles, and thus $\overline{\Sigma}$ has genus $\overline{g} = 1 - (\overline{\chi} + \overline{b})/2 = 2^\beta(g - 1) + 1$.

After computing the homology signatures for Σ in $O(\beta n)$ time, following Lemma 3.1, it is straightforward to construct $\overline{\Sigma}$ in $O(\overline{n}) = O(2^\beta n)$ time. \square

We assign weights to the directed edges of \overline{G} by setting $\overline{w}(u \rightarrow v, h) := w(u \rightarrow v)$ for each edge $u \rightarrow v$ of G and each homology class h . In other words, each directed edge in $\overline{\Sigma}$ inherits the weight of its projection in Σ .

Now consider an arbitrary path p in G , with (possibly equal) endpoints u and v . A straightforward induction argument implies that for any homology class $h \in (\mathbb{Z}_2)^\beta$, the path p is the projection of a unique path from (u, h) to $(v, h \oplus [p])$, which we denote (\mathbf{p}, \mathbf{h}) . Moreover, this lifted path has the same length as its projection: $w(p) = \bar{w}(\mathbf{p}, \mathbf{h})$. The following lemmas are now immediate.

Lemma 4.2. *Every lift of a shortest directed path in G is a shortest directed path in \bar{G} .*

Lemma 4.3. *A loop ℓ in G with basepoint v is \mathbb{Z}_2 -minimal if and only if, for every homology class $h \in (\mathbb{Z}_2)^\beta$, the lifted path (ℓ, h) is a shortest directed path in \bar{G} from (v, h) to $(v, h \oplus [\ell])$.*

5 Computing \mathbb{Z}_2 -Minimal Cycles

The results in the previous section immediately suggest imply an algorithm to compute the shortest directed cycle in a given \mathbb{Z}_2 -homology class h in time $2^{O(\beta)}n^2$: construct the \mathbb{Z}_2 -homology cover, and then compute the shortest path from $(v, 0)$ to (v, h) , for every vertex v in the original graph. In this section, we describe a more complex algorithm that runs in time $2^{O(\beta)}n \log n$.

Lemma 5.1. *In $O(n \log n + \beta n)$ time, we can construct a set S of $O(\beta)$ directed shortest paths in G , such that every non-null-homologous cycle in G intersects at least one path in S .*

Proof: Following Chambers *et al.* [13], we construct a *greedy system of arcs*, using a variant of Erickson and Whittlesey’s algorithm to construct optimal systems of loops [31]. Our algorithm uses a natural generalization of tree-cotree decompositions [28] to surfaces with boundary, essentially dual to the tree-coforest decompositions described in Section 3. A **forest-cotree decomposition** of G is any partition $(\partial G, F, C, X)$ of the edges of G into *four* edge-disjoint subgraphs with the following properties:

- ∂G is the set of all boundary edges of G .
- F is a spanning forest of G , that is, an acyclic subgraph of G that contains every vertex.
- Each component of F contains a single boundary vertex.
- C^* is a spanning tree of $G^* \setminus (\partial G)^*$, that is, a subtree of G^* that contains every vertex *except* the dual boundary vertices δ_i^* .

Euler’s formula implies that there are exactly β edges in X ; arbitrarily label these edges e_1, e_2, \dots, e_β . For each edge $e_i \in X$, the subgraph $F \cup \{e_i\}$ contains a single

nontrivial arc α_i , which is either a simple path between distinct boundary cycles, or a nontrivial loop from a boundary cycle back to itself; in the second case, α_i may traverse some edges of F twice. Cutting along the arcs $\alpha_1, \dots, \alpha_\beta$ transforms Σ into a topological disk. Thus, every non-null-homologous cycle in G must cross at least one arc α_i . See Figure 5.

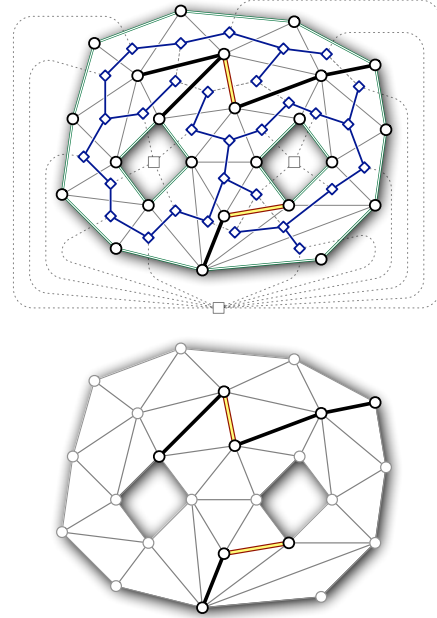


Figure 5. Top: A forest-cotree decomposition of the graph in Figure 1; thick doubled lines indicate edges in X . Bottom: The resulting system of arcs. Compare with Figure 3.

We can easily construct an *arbitrary* forest-cotree decomposition in $O(n)$ time using whatever-first search, as in Lemma 3.1, but we require a decomposition with a particular forest F . Let $G/\partial G$ denote the graph obtained from G by *contracting* the entire subgraph ∂G —both vertices and edges—to a single vertex x . Using Dijkstra’s algorithm, we compute the single-source shortest-path tree T in $G/\partial G$ rooted at x in $O(n \log n)$ time. Let F be the subgraph of G corresponding to T . Each component of F is a tree of shortest paths from a boundary vertex to a subset of the non-boundary vertices of G . With the shortest-path forest F in hand, we can easily construct the rest of the forest-cotree decomposition in $O(n)$ time.

Finally, for each edge $e_i \in X$, let σ_i and τ_i denote the unique directed paths in F from the boundary of G to the endpoints of e_i , and let $S := \{\sigma_1, \dots, \sigma_\beta, \tau_1, \dots, \tau_\beta\}$. By construction of F , every element of S is a (possibly empty) shortest directed path. Moreover, because $\alpha_i = \sigma_i \cdot e_i \cdot \text{rev}(\tau_i)$ for each index i , every non-null-homologous cycle in G must intersect at least one path in S . We can easily compute each path in S in $O(n)$ time. \square

Recall that any path σ from u to v in G is the projection of a unique path $(\sigma, 0)$ from $(u, 0)$ to $(v, [\sigma])$ in \bar{G} .

Lemma 5.2. *Let γ be a \mathbb{Z}_2 -minimal cycle in G , and let σ be any shortest path in G that intersects γ . There is a \mathbb{Z}_2 -minimal cycle γ' homologous to γ , which is the projection of a shortest path (γ', h) in \bar{G} that starts with a subpath of $(\sigma, 0)$ but does not otherwise intersect $(\sigma, 0)$.*

Proof: Let v be the vertex of $\sigma \cap \gamma$ closest to the starting vertex of σ , and let (v, h) be the corresponding vertex of the lifted path $(\sigma, 0)$. Think of γ as a loop based at v . Lemma 4.3 implies that the lifted path (γ, h) is a shortest path from (v, h) to $(v, h \oplus [\gamma])$.

Now let (w, h') be the last vertex along (γ, h) that is also a vertex of $(\sigma, 0)$. Let (γ', h) be the path obtained from (γ, h) by replacing the subpath from (v, h) to (w, h') with the corresponding subpath of $(\sigma, 0)$. By construction, (γ', h) starts with a directed subpath of $(\sigma, 0)$ but does not otherwise intersect $(\sigma, 0)$. Because both (γ, h) and $(\sigma, 0)$ are shortest paths in $\bar{\Sigma}$, the new path (γ', h) has the same length as (γ, h) . Thus, the projected cycle γ' has the same length and homology class as γ , which implies that γ' is \mathbb{Z}_2 -minimal. \square

We emphasize that the modified cycle γ' may intersect σ arbitrarily many times; however, all such intersections lift to intersections between (γ', h) and lifts of σ other than $(\sigma, 0)$.

Our algorithm uses a recent generalization of Klein's seminal multiple-source shortest path algorithm [45] to higher-genus embedded graphs:

Lemma 5.3 (Chambers et al. [7, 8]). *Let G be a directed graph with non-negative edge weights, cellularly embedded on a surface Σ of genus g with $b > 0$ boundaries, and let f be an arbitrary face of G . We can preprocess G in $O(gn \log n)$ time³ and $O(n)$ space, so that the length of the shortest path from any vertex incident to f to any other vertex can be retrieved in $O(\log n)$ time.*

Theorem 5.4. *Let G be a directed graph with non-negative edge weights, cellularly embedded on a surface Σ with first Betti number β , and let γ be a cycle in G with k edges. A shortest directed cycle in Σ that is \mathbb{Z}_2 -homologous to γ can be computed in $O(\beta k + 4^\beta \beta^2 n \log n)$ time.*

³The published version of this algorithm [7] proves a weaker time bound of $O(g^2 n \log n)$; using this version increases the running time of our algorithms by a factor of 2^β .

Proof: We begin by computing homology signatures for the edges of G in $O(\beta n)$ time, as described in Section 3. In $O(\beta k)$ time, we then compute the homology signature $[\gamma]$. If $[\gamma] = 0$, we return the empty walk and halt.

Next, we construct the \mathbb{Z}_2 -homology cover \bar{G} in $O(2^\beta n \log n)$ time, as described in Section 4, as well as the set S of directed shortest paths described in Lemma 5.1. We then look for the shortest path in \bar{G} of the canonical form described in Lemma 5.2, by considering each shortest path $\sigma \in S$ in turn as follows.

Let us write $(\sigma, 0) = (v_0, 0) \rightarrow (v_1, h_1) \rightarrow \dots \rightarrow (v_t, h_t)$. We construct the combinatorial surface $\bar{\Sigma} \mathcal{A}(\sigma, 0)$ by splitting the path $(\sigma, 0)$ into two parallel paths from $(v_0, 0)$ to (v_t, h_t) , which we denote $(\sigma, 0)^+$ and $(\sigma, 0)^-$. For each index $1 \leq i \leq t - 1$, let $(v_i, h_i)^+$ and $(v_i, h_i)^-$ denote the copies of vertex (v_i, h_i) on the paths $(\sigma, 0)^+$ and $(\sigma, 0)^-$, respectively. The paths $(\sigma, 0)^+$ and $(\sigma, 0)^-$ bound a new common face $f_{(\sigma, 0)}$ in $\bar{\Sigma} \mathcal{A}(\sigma, 0)$.

Lemma 5.2 implies that if any \mathbb{Z}_2 -minimal cycle homologous to γ intersects σ , then some \mathbb{Z}_2 -minimal cycle homologous to γ is the projection of a shortest path in $\bar{\Sigma} \mathcal{A}(\sigma, 0)$ from some vertex $(v_i, h_i)^\pm$ to the corresponding vertex $(v_i, h_i \oplus [\gamma])$. To compute these shortest paths, we implicitly compute the shortest path in $\bar{\Sigma} \mathcal{A}(\sigma, 0)$ from every vertex on the boundary of $f_{(\sigma, 0)}$ to every vertex of $\bar{\Sigma} \mathcal{A}(\sigma, 0)$, using Lemma 5.3. The resulting algorithm runs in $O(\bar{g} \bar{n} \log \bar{n}) = O(4^\beta \beta^2 n \log n)$ time, by Lemma 4.1. \square

By running this algorithm 2^β times, we can compute the shortest directed cycle in Σ in every \mathbb{Z}_2 -homology class, in $O(8^\beta \beta^2 n \log n)$ time. In particular, we can compute the shortest directed cycle in Σ that has nontrivial \mathbb{Z}_2 -homology. When the original surface has no boundary, this is just the shortest non-separating cycle in Σ . Very recently, Cabello et al. [9] described an algorithm to compute the shortest non-separating cycle in any surface-embedded directed graph in $O(g^{1/2} n^{3/2} \log n)$ time; our new algorithm is faster whenever $g \leq (\lg n)/13$.

Corollary 5.5. *Given a directed graph G with n vertices with non-negative edge weights, cellularly embedded on a surface with genus g , we can compute the shortest directed cycle in G that is non-separating in Σ in $O(64^g g^2 n \log n)$ time.*

Cabello et al. [9] also described an algorithm to find the shortest non-contractible cycle in an embedded directed graph in $O(g^{1/2} n^{3/2} \log n)$ time. Our approach does not lead to a faster algorithm for this problem; contractibility is inherently a property of homotopy, not homology.

6 Faster Minimum Cuts

We now apply our algorithm for computing \mathbb{Z}_2 -minimal cycles to the classical minimum cut problem for *undirected* surface-embedded graphs. Our approach is motivated by the following lemma of Chambers *et al.* [15, Lemma 3.1]:

Lemma 6.1. *Let H be an undirected graph with non-negative edge capacities, embedded on a surface Π without boundary, and let s and t be vertices of H . Finally, let X be the minimum-capacity (s, t) -cut in H . Then X^* is the minimum-weight even subgraph of H^* that is \mathbb{Z}_2 -homologous with the boundary of s^* in the surface $\Pi \setminus (s^* \cup t^*)$.*

We must emphasize that this characterization applies only to *undirected* graphs, or equivalently, symmetric directed graphs where the capacity of any edge is equal to the capacity of its reversal. Even for directed *planar* graphs, where the dual of the minimum cut is a simple cycle, the characterisation is incorrect: The minimum-weight cycle \mathbb{Z}_2 -homologous to ∂s^* in the annulus $\Pi \setminus (s^* \cup t^*)$ is either the dual of the minimum (s, t) -cut or the dual of the minimum (t, s) -cut. (Of course, if the graph H is undirected, the minimum (s, t) -cut and the minimum (t, s) -cut coincide.)

Like Chambers *et al.* [15], we actually describe an algorithm to compute the minimum-weight even subgraph in every \mathbb{Z}_2 -homology class. Theorem 5.4 immediately implies that we can compute a minimum-weight cycle in every \mathbb{Z}_2 -homology class in $O(8^\beta \beta^2 n \log n)$ time. However, the minimum weight *even subgraph* in a given homology class may not be (the carrier of) a \mathbb{Z}_2 -minimal cycle. In particular, if a \mathbb{Z}_2 -minimal cycle γ traverses any edge more than once, then every minimum-weight even subgraph with signature $[\gamma]$ must be disconnected.

However, any *connected* \mathbb{Z}_2 -minimal even subgraph is the carrier of a \mathbb{Z}_2 -minimal cycle, and the components of any \mathbb{Z}_2 -minimal even subgraph are themselves \mathbb{Z}_2 -minimal even subgraphs. Thus, we can assemble a \mathbb{Z}_2 -minimal even subgraph in any homology class from a subset of the \mathbb{Z}_2 -minimal cycles we have already computed. The following lemma puts an upper bound on the number of cycles we need.

Lemma 6.2. *Every \mathbb{Z}_2 -minimal even subgraph of G has at most $g + b - 1$ components.*

Proof: Let $\gamma_1, \dots, \gamma_{g+b}$ be disjoint simple cycles on an abstract surface Σ of genus g with b boundaries, and consider the surface $\Sigma' = \Sigma \setminus (\gamma_1 \cup \dots \cup \gamma_{g+b})$. The definition of genus implies that Σ' cannot be connected; indeed, Σ' must have at least $b + 1$ components. So the

pigeonhole principle implies that some component Σ'' of Σ does not contain any of the boundary cycles of Σ . The boundary of Σ'' is therefore null-homologous.

Following Chambers *et al.* [15], call a cycle in G *weakly simple* if it traverses each edge of G at most once and never crosses itself. Any weakly simple cycle in G can be perturbed into a simple cycle in an arbitrarily small neighborhood of G . A *cycle decomposition* of an even subgraph η is a set of edge-disjoint, non-crossing, weakly simple cycles whose union is η . It is easy to prove that every even subgraph has a cycle decomposition [15, Lemma 3.2].

Now let η be an even subgraph of G with more than $g + b - 1$ components. Each component has a cycle decomposition, so η must have a cycle decomposition with more than $g + b - 1$ elements. Thus, the argument in the first paragraph implies that some subgraph of η must be null-homologous. We conclude that η is not \mathbb{Z}_2 -minimal. \square

Theorem 6.3. *Let G be an undirected graph with non-negative edge weights, cellularly embedded on a surface Σ with first Betti number β . A minimum-weight even subgraph of G in each \mathbb{Z}_2 -homology class can be computed in $O(8^\beta \beta^2 n \log n)$ time.*

Proof: Our algorithm computes a minimum-weight cycle γ_h in every \mathbb{Z}_2 -homology class h in $O(8^\beta \beta^2 n \log n)$ time, via Theorem 5.4, and then assemble these \mathbb{Z}_2 -minimal cycles into \mathbb{Z}_2 -minimal even subgraphs using dynamic programming.

For each homology class $h \in (\mathbb{Z}_2)^\beta$ and each integer $1 \leq k \leq g + b - 1$, let $C(h, k)$ denote the minimum total weight of any set of at most k cycles in G whose homology classes sum to h . Lemma 6.2 implies that the minimum weight of any even subgraph in homology class h is exactly $C(h, g + b - 1)$. This function obeys the following straightforward recurrence:

$$C(h, k) = \min \left\{ C(h_1, k-1) + C(h_2, 1) \mid h_1 \oplus h_2 = h \right\}.$$

This recurrence has two base cases: $C(0, k) = 0$ for any integer k , and for any homology class h , the value $C(h, 1)$ is just the length of γ_h . A standard dynamic programming algorithm computes $C(h, g + b - 1)$ for all 2^β homology classes h in $O(4^\beta \beta)$ time. We can then assemble the actual minimum-weight even subgraphs in each homology class in $O(\beta n)$ time. The total time for this phase of the algorithm is $O(4^\beta \beta + 2^\beta \beta n)$, which is dominated by the time to compute all the \mathbb{Z}_2 -minimal cycles. \square

Lemma 6.1 and Theorem 6.3 immediately give us the following result:

Corollary 6.4. *Given an undirected graph H with non-negative edge capacities, embedded on a surface Σ with genus g , and two vertices let s and t , we can compute a minimum (s, t) -cut in H in $O(16^g g^2 n \log n)$ time.*

References

- [1] G. Borradaile, E. D. Demaine, and S. Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Proc. 26th Int. Symp. Theoretical Aspects Comput. Sci.*, 171–182, 2009. Leibniz Int. Proc. Informatics 3, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. (<http://drops.dagstuhl.de/opus/volltexte/2009/1835>).
- [2] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1285–1294, 2007.
- [3] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. Steiner tree in planar graphs: An $O(n \log n)$ approximation scheme with singly-exponential dependence on epsilon. *Proc. 10th Workshop on Algorithms and Data Structures*, 275–286, 2007.
- [4] G. Borradaile and P. Klein. An $O(n \log n)$ algorithm for maximum st -flow in a directed planar graph. *J. ACM* 56(2): 9:1–30, 2009.
- [5] G. Borradaile, J. R. Lee, and A. Sidiropoulos. Randomly removing g handles at once. *Proc. 25th Ann. ACM Symp. Comput. Geom.*, 371–376, 2009.
- [6] S. Cabello. Many distances in planar graphs. *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1213–1220, 2006.
- [7] S. Cabello and E. W. Chambers. Multiple source shortest paths in a genus g graph. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 89–97, 2007.
- [8] S. Cabello, E. W. Chambers, and J. Erickson. Multiple-source shortest paths in embedded graphs. Full version of [7], in preparation.
- [9] S. Cabello, É. Colin de Verdière, and F. Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *Proc. 26th Ann. ACM Symp. Comput. Geom.*, 156–165, 2010.
- [10] S. Cabello, É. Colin de Verdière, and F. Lazarus. Output-sensitive algorithm for the edge-width of an emedded graph. *Proc. 26th Ann. ACM Symp. Comput. Geom.*, 147–155, 2010.
- [11] S. Cabello, M. DeVos, J. Erickson, and B. Mohar. Finding one tight cycle. *ACM Trans. Algorithms* 6(4):article 61, 2010.
- [12] S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete Comput. Geom.* 37:213–235, 2007.
- [13] E. W. Chambers, É. Colin de Verdière, J. Erickson, F. Lazarus, and K. Whittlesey. Splitting (complicated) surfaces is hard. *Comput. Geom. Theory Appl.* 41(1–2):94–110, 2008.
- [14] E. W. Chambers, J. Erickson, and A. Nayyeri. Homology flows, cohomology cuts. *Proc. 42nd Ann. ACM Symp. Theory Comput.*, 273–282, 2009. Full version available at <http://www.cs.uiuc.edu/~jeffe/pubs/surflow.html>.
- [15] E. W. Chambers, J. Erickson, and A. Nayyeri. Minimum cuts and shortest homologous cycles. *Proc. 25th Ann. ACM Symp. Comput. Geom.*, 377–385, 2009.
- [16] C. Chen and D. Freedman. Quantifying homology classes II: Localization and stability. Preprint, 2007. (<http://arxiv.org/abs/0709.2512v2>).
- [17] C. Chen and D. Freedman. Quantifying homology classes. *Proc. 25th Ann. Symp. Theoretical Aspects Comput. Sci.*, 169–180, 2008. Dagstuhl Seminar Proceedings 08001. (<http://drops.dagstuhl.de/opus/volltexte/2008/1343/>).
- [18] C. Chen and D. Freedman. Hardness results for homology localization. *Proc. 21st ACM-SIAM Symp. Discrete Algorithms*, 1594–1604, 2010.
- [19] C. Chen and D. Freedman. Measuring and computing natural generators for homology groups. *Comput. Geom. Theory Appl.* 43(2):169–181, 2010.
- [20] É. Colin de Verdière. *Raccourcissement de courbes et décomposition de surfaces [Shortening of Curves and Decomposition of Surfaces]*. Ph.D. thesis, University of Paris 7, Dec. 2003. (<http://www.di.ens.fr/~colin/textes/these.html>).
- [21] É. Colin de Verdière. Shortest cut graph of a surface with prescribed vertex set. *Proc. 18th Ann. Europ. Symp. Algorithms*, to appear, 2010. (<http://www.di.ens.fr/~colin/textes/10cutgraph.pdf>).
- [22] É. Colin de Verdière and J. Erickson. Tightening non-simple paths and cycles on surfaces. *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms*, 192–201, 2006. Full version available at <http://www.cs.uiuc.edu/~jeffe/pubs/octagons.html>.
- [23] É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. *J. ACM* 54(4), 2007.
- [24] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 278–287, 2007.
- [25] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *Proc. 42nd Ann. ACM Symp. Theory Comput.*, 221–230, 2010.
- [26] T. K. Dey, K. Li, and J. Sun. On computing handle and tunnel loops. *IEEE Proc. Int. Conf. Cyberworlds*, 357–366, 2007.
- [27] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3D models. *ACM Trans. Graphics* 27(3):1–9, 2008. Proc. SIGGRAPH 2008.
- [28] D. Eppstein. Dynamic generators of topologically embedded graphs. *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, 599–608, 2003.
- [29] J. Erickson. Parametric shortest paths and maximum flows in planar graphs. *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, 794–804, 2010.

- [30] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31:37–59, 2004.
- [31] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1038–1046, 2005.
- [32] J. Erickson and P. Worah. Computing the shortest essential cycle. *Discrete Comput. Geom.* 44(4):912–930, 2010. (<http://www.cs.uiuc.edu/~jeffe/pubs/essential.html>).
- [33] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian J. Math.* 8(399–404), 1956. First published as Research Memorandum RM-1400, The RAND Corporation, Santa Monica, California, November 19, 1954.
- [34] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM J. Comput.* 16(6):1004–1004, 1987.
- [35] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.* 35(4):921–940, 1988.
- [36] J. L. Gross and T. W. Tucker. *Topological graph theory*. Dover Publications, 2001.
- [37] T. E. Harris and F. S. Ross. Fundamentals of a method for evaluating rail net capacities. Memorandum RM-1573, The RAND Corporation, Santa Monica, California, October 24, 1955. Cited in [51].
- [38] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001. (<http://www.math.cornell.edu/~hatcher/>).
- [39] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.* 55(1):3–23, 1997.
- [40] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.* 4:63–98, 1994.
- [41] A. Itai and Y. Shiloach. Maximum flow in planar networks. *SIAM J. Comput.* 8:135–150, 1979.
- [42] L. Janiga and V. Koubek. Minimum cut in directed planar networks. *Kybernetika* 28(1):37–49, 1992.
- [43] K. Kawarabayashi and B. Mohar. Graph and map isomorphism and all polyhedral embeddings in linear time. *Proc. 40th Ann. ACM Symp. Theory Comput.*, 471–480, 2008.
- [44] K. Kawarabayashi and B. Reed. Computing crossing number in linear time. *Proc. 39th Ann. ACM Symp. Theory Comput.*, 382–390, 2007.
- [45] P. N. Klein. Multiple-source shortest paths in planar graphs. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 146–155, 2005.
- [46] M. Kutz. Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time. *Proc. 22nd Ann. ACM Symp. Comput. Geom.*, 430–438, 2006.
- [47] J. R. Lee and A. Sidiropoulos. Genus and the geometry of the cut graph. *Proc. 21st ACM-SIAM Symp. Discrete Algorithms*, 193–201, 2010.
- [48] B. Mohar and C. Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, 2001.
- [49] Y. Nussbaum. Personal communication, January 2010.
- [50] J. Reif. Minimum s - t cut of a planar undirected network in $O(n \log^2 n)$ time. *SIAM J. Comput.* 12:71–81, 1983.
- [51] A. Schrijver. On the history of combinatorial optimization (till 1960). *Handbook of Discrete Optimization*, 1–68, 2005. Elsevier.
- [52] A. Sidiropoulos. Optimal stochastic planarization. *Proc. 51st IEEE Symp. Found. Comput. Sci.*, to appear, 2010. (<http://arxiv.org/abs/1004.1666>).
- [53] D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.* 26(3):362–391, 1983.
- [54] B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. *ACM Trans. Graphics* 27(3), Article 77 (11 pages), 2008. Proc. SIGGRAPH 2008.
- [55] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, 2nd edition. Graduate Texts in Mathematics 72. Springer-Verlag, 1993.
- [56] C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory Ser. B* 48(2):155–177, 1990.
- [57] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. *Proc. Eurographics Symp. Geom. Proc.*, 201–210, 2006.
- [58] K. Weihe. Maximum (s, t) -flows in planar networks in $O(|V| \log |V|)$ -time. *J. Comput. Syst. Sci.* 55(3):454–476, 1997.
- [59] A. Zomorodian and G. Carlsson. Localized homology. *Proc. IEEE Int. Conf. Shape Modeling Appl. (SMA)*, 189–198, 2007.