Proceedings of Symposia in Applied Mathematics

# Combinatorial Optimization of Cycles and Bases

Jeff Erickson

Abstract. We survey algorithms and hardness results for two important classes of topology optimization problems: computing minimum-weight cycles in a given homotopy or homology class, and computing minimum-weight cycle bases for the fundamental group or various homology groups.

## 1. Introduction

Identification of topological features is an important subproblem in several geometric applications. In many of these applications, it is important for these features to be represented as compactly as possible.

For example, a common method for building geometric models is to reconstruct a surface from a set of sample points, obtained from range finders, laser scanners, or some other physical device. Gaps and measurement errors in the point cloud data induce errors in the reconstructed surface, which often take the form of spurious handles or tunnels. For example, the original model of *David*'s head constructed by Stanford's Digital Michelangelo Project [**130**] has 340 small tunnels, none of which are present in the original marble sculpture [**101**]. Because most surface simplification and parametrization methods deliberately preserve the topology of the input surface, topological noise must be identified, localized, and removed before these other algorithms can be applied [**74**, **101**, **191**, **193**].

Another example arises in VLSI routing [**46**, **86**, **129**], map simplification [**53**, **54**, **81**], and graph drawing [**70**, **75**]. Given a rough sketch of one or more paths in a planar environment with fixed obstacles—possibly representing roads or rivers near cities or other geographic features, or wires between components on a chip—we want to produce a topologically equivalent set of paths that are as short or as simple as possible, perhaps subject to some tolerance constraints.

Similar optimization problems also arise in higher-dimensional simplicial complexes. For example, several researchers model higher-order connectivity properties

2000 *Mathematics Subject Classification.* Primary 68W05, 68Q25; Secondary 57M05, 55U05.
*Key words and phrases.* computational topology, topological graph theory, algorithms.
Portions of this work were done while the author was visiting IST Austria. This work was partially supported by NSF grant CCF 09-15519.

©0000 (copyright holder)

of sensor networks using simplicial complexes. In one formulation, complete coverage of an area by a sensor network is indicated by the presence of certain nontrivial second homology classes in the associated simplicial complex [**57, 58, 89, 178**]; smaller generators indicate that fewer sensors are required for coverage. Moreover, if coverage is incomplete, holes in the network are indicated by nontrivial one-dimensional homology classes; localizing those holes makes repairing or routing around them easier.

Similar topological optimization problems arise in shape analysis [**63, 64**], low-distortion surface parametrization [**99, 174, 184**], probabilistic embedding of high-genus graphs into planar graphs [**20, 128, 171**], computing crossing numbers of graphs [**117**], algorithms for graph isomorphism [**116**], approximation of optimal traveling salesman tours [**60, 88**] and Steiner trees [**16–18**], data visualization and analysis [**31, 32, 50**], shape modeling [**33**], localization of invariant sets of differential equations [**113**], minimal surface computation [**71, 176**], image and volume segmentation [**24, 92, 94, 119**], and sub-sampling point cloud data for topological inference [**154**].

This survey gives an overview of algorithms and hardness results for four classes of topological optimization problems:

**Optimal Homotopy Basis:** Given a space $\Sigma$ and a basepoint $x$, find an optimal set of loops whose homotopy classes generate the fundamental group $\pi_1(\Sigma, x)$.

**Homotopy Localization:** Given a cycle $\gamma$ in a space $\Sigma$, find a shortest cycle in $\Sigma$ that is homotopic to $\gamma$.

**Optimal Homology Basis:** Given a space $\Sigma$ and an integer $p$, find an optimal set of $p$-cycles whose homology classes generate the homology group $H_p(\Sigma)$.

**Homology Localization:** Given a $p$-cycle $\boldsymbol{c}$ in a space $\Sigma$, find an optimal $p$-cycle in $\Sigma$ that is homologous to $\boldsymbol{c}$.

In the interest of finiteness, the survey is limited to *exact* and *efficient* algorithms for these problems; I do not even attempt to cover the vast literature on approximation algorithms, numerical methods, and practical heuristics for topological optimization. For similar reasons, the survey attempts only to give a high-level overview of the most important results; many crucial technical details are mentioned only in passing or ignored altogether.

I assume the reader is familiar with basic results in algebraic topology (cell complexes, surface classification, homotopy, covering spaces, homology, relative homology, Poincaré-Lefschetz duality) [**103, 150, 175**], graph algorithms (graph data structures, depth-first search, shortest paths, minimum spanning trees, NP-completeness) [**52, 122, 179**], and combinatorial optimization (flows, cuts, circulations, linear programming, LP duality) [**4, 122, 166**].

## 2. Input Assumptions

**2.1. Combinatorial spaces.** Our focus on efficient, exact algorithms necessarily limits the scope of the problems we can consider. With one exception, we consider only *combinatorial* topological spaces as input: finite cell complexes whose cells do not have geometry per se, but where each cell $c$ has an associated non-negative *weight* $w(c)$. For simplicity of exposition, we restrict our attention to simplicial complexes; however, most of the algorithms we discuss can be applied with little or no modification to finite regular CW-complexes. For questions about homotopy, we consider only paths, loops, and cycles in the 1-skeleton; the weight (or 'length') of a cycle is the sum of the weights of its edges. For questions about homology, the output is either a subcomplex of the $p$-skeleton, whose weight is the sum of the weights of its cells, or a real or integer $p$-chain, whose cost is the weighted sum of its coefficients.

Unless noted otherwise, the time bounds reported here assume that the input complex is (effectively) represented by a sequence of boundary matrices, each stored explicitly in a standard two-dimensional array. (For sparse complexes, representing each boundary matrix using a sparse-matrix data structure leads to faster algorithms, at least in practice, especially in combination with simplification heuristics [**10**, **30**, **56**, **84**, **113**, **147**, **194**, **195**].) For each index $i$, we let $n_i$ denote the number of $i$-dimensional cells in the input complex, and we let $n$ denote the total number of cells of all dimensions.

**2.2. Combinatorial surfaces.** For problems related to homotopy of loops and cycles, we must further restrict the class of input spaces to combinatorial *2-manifolds*. Classical results of Markov and others [**136**–**138**] imply that most computational questions about homotopy are undecidable for general 2-complexes or manifolds of dimension 4 and higher. Thurston and Perelman's geometrization theorem [**123**, **145**, **182**] implies that the homotopy problems we consider are decidable in 3-manifolds [**23**, **139**, **159**]; however, the few explicit algorithms that are known are extremely complex, and no complexity bounds are known. The homology problems we consider are all decidable for arbitrary finite complexes, but (not surprisingly) they can be solved more efficiently in combinatorial 2-manifolds than in general spaces.

To simplify exposition, this survey explicitly considers only *orientable* surfaces; however, most of the results we describe apply to non-orientable surfaces with little or no modification.

The surface algorithms we consider are most easily described in the language of topological graph theory [**90**, **125**, **144**]. An *embedding* of an undirected graph $G$ on an abstract 2-manifold $\Sigma$ maps vertices of $G$ to distinct points in $\Sigma$ and edges of $G$ to interior-disjoint curves in $\Sigma$. The *faces* of an embedding are maximal connected subsets of $\Sigma$ that are disjoint from the image of the graph. An embedding is *cellular* (or *2-cell* [**144**]) if each of its faces is homeomorphic to an open disk. A combinatorial surface is simply a cellular embedding of a graph on an abstract 2-manifold.

Any cellular embedding of a graph on an orientable surface can be represented combinatorially by a *rotation system*, which encodes the cyclic order of edges around each vertex. To define rotation systems more formally, we represent each edge $uv$ in $G$ by a pair of directed edges or *darts* $u{\to}v$ and $v{\to}u$; we say that the dart $u{\to}v$ *leaves* $u$ and *enters* $v$. A rotation system is a pair $(\sigma, \rho)$ of permutations of the darts, such that $\sigma(u{\to}v)$ is the next dart leaving $u$ after $u{\to}v$ in counterclockwise order around $u$ (with respect to some fixed orientation of the surface), and $\rho$ is the reversal permutation $\rho(u{\to}v) = v{\to}u$. The permutation $\sigma \circ \rho$ encodes the clockwise order of darts around each 2-cell.

Any cellularly embedded graph with $n_0$ vertices, $n_1$ edges, and $n_2$ faces lies on a surface $\Sigma$ with Euler characteristic $\chi = n_0 - n_1 + n_2 = 2 - 2g$. Assuming without loss of generality that no vertex has degree less than 3, it follows that $n_1 = O(n_0 + g)$

and $n_2 = O(n_0 + g)$. Let $n = n_0 + n_1 + n_2$ denote the total complexity of the input surface.

Any graph $G$ embedded on any surface $\Sigma$ has an associated *dual graph* $G^*$, defined intuitively by giving each edge of $G$ a clockwise quarter-turn around its midpoint. More formally, the dual graph $G^*$ has a vertex $f^*$ for each face $f$ of $G$, and two vertices in $G^*$ are joined by an edge $e^*$ if and only if the corresponding faces of $G$ are separated by an edge $e$. The dual graph $G^*$ has a natural cellular embedding in the same surface $\Sigma$, determined by the rotation system $(\sigma \circ \rho, \rho)$. Each face $v^*$ of this embedding corresponds to a vertex $v$ of the primal graph $G$. Duality can be extended to the darts of $G$ by defining $(u{\to}v)^* = (\ell^*){\to}(r^*)$, where $\ell$ and $r$ are respectively the faces on the left and right sides of $u{\to}v$. See Figure 1.

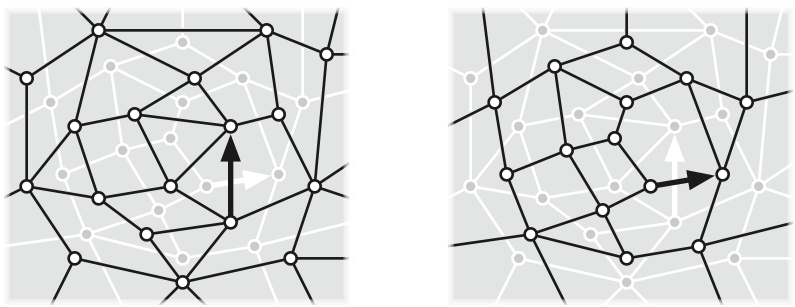

FIGURE 1. A portion of an embedded graph $G$ and its dual $G^*$, with one dart and its dual emphasized.

For any subgraph $H$ of $G$, we abuse notation by letting $H^*$ denote the corresponding subgraph of the dual graph $G^*$, and letting $G \setminus H$ denote the subgraph of $G$ obtained by deleting the *edges* of $H$. In particular, we have $(G \setminus H)^* = G^* \setminus H^*$.

For algorithms that act on combinatorial surfaces, we assume that the input is a data structure of size $O(n)$ that efficiently supports standard graph operations, such as enumerating neighbors of vertices in constant time each, both in the primal graph and in the dual graph. Several such data structures are known [**12**, **100**, **131**, **148**, **189**]. It is straightforward to determine the Euler characteristic, and therefore the genus, of a given surface in $O(n)$ time by counting vertices, edges, and faces.

To simplify our exposition, we implicitly assume that shortest vertex-to-vertex paths are unique in every weighted graph we consider. This assumption can be enforced automatically using standard perturbation techniques [**149**], but in fact, none of the algorithms we describe actually require this assumption.

**2.3. Why not solve the real problem?** The problems we consider are obviously well-defined for continuous geometric spaces, such as piecewise-linear or Riemannian surfaces (with some appropriate discrete or implicit representation). Unfortunately, at least with the current state of the art, these spaces permit only inefficient or approximate solutions.

Almost all the algorithms we describe here rely heavily on the ability to compute exact shortest paths. Shortest paths in combinatorial spaces can be computed in $O(n \log n)$ time using Dijkstra's algorithm in the 1-skeleton. In general Riemannian surfaces, shortest paths have no analytic representation, and therefore cannot be computed exactly even in principle. For piecewise-linear surfaces, existing shortest-path algorithms, both exact [**42**, **133**, **143**, **158**, **164**, **165**, **177**] and approximate [**1**, **5**], are efficient only under the assumption that any shortest path crosses each edge of the input complex at most a constant number of times. (Some common numerical algorithms [**118**, **170**] require even stronger geometric assumptions.) The crossing assumption is reasonable in practice—for example, it holds if the input complex is PL-embedded in some ambient Euclidean space (in which case any shortest path crosses any edge at most *once*), or if all face angles are larger than some fixed constant—but it does not hold in general.

As an elementary bad example, consider the piecewise-linear annulus defined by identifying the non-horizontal edges of the Euclidean trapezoid with vertices $(0, 0)$, $(1, 0)$, $(x, 1)$, $(x + 1, 1)$, for some arbitrarily large integer $x$, as shown in Figure 2. (Essentially the same example appears as Figure 1 in Alexandrov's seminal paper on convex polyhedral metrics [**7**].) The shortest path in this annulus between its two vertices is a vertical segment that crosses the oblique edge $x - 1$ times. All existing shortest-path algorithm require at least constant time for each crossing; thus, their total running time is *unbounded* as a function of the combinatorial complexity of the input.

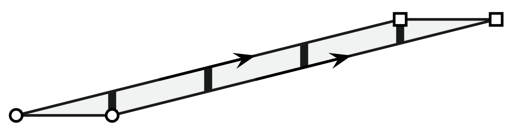

FIGURE 2. A shortest path in a piecewise-linear annulus

Even for piecewise-linear surfaces embedded in $\mathbb{R}^3$, shortest-path algorithms can only be efficient in a model of computation that supports exact constant-time real arithmetic, including square roots. Even for polyhedra with integer vertices, most vertex-to-vertex geodesics have irrational lengths, representable analytically only with deeply nested radicals. Admittedly, this does not appear to be a significant problem in practice; it is relatively easy to implement existing shortest-path algorithms to compute paths that are optimal up to floating-point precision [**177**].

Subject to those caveats, a few of the algorithms we describe for combinatorial surfaces can be extended to embedded piecewise-linear surfaces, with some loss of efficiency, by replacing Dijkstra's algorithm with a piecewise-linear shortest-path algorithm. For example, Erickson and Whittlesey's algorithm [**80**] (described in Section 3.2) can be modified to compute an optimal homotopy basis with a given basepoint, in an embedded piecewise-linear surface, in $O(n^2)$ real arithmetic operations. However, for most of the problems we consider, no efficient algorithms are known for piecewise-linear surfaces.

## —— PART I. HOMOTOPY ——

### 3. Homotopy Bases

Suppose we are given an undirected graph $G$ with non-negatively weighted edges, a cellular embedding of $G$ on an orientable surface $\Sigma$ of genus $g$, and a vertex $x$ of $G$. A *homotopy basis* is a set of $2g$ loops based at $x$ whose homotopy classes generate the fundamental group $\pi_1(\Sigma, x)$. Our goal in this section is to compute a homotopy basis of minimum total length.

The standard structure for a homotopy basis is a *system of loops*, which is a set $\{\ell_1, \ell_2, \ldots, \ell_{2g}\}$ of $2g$ loops in $\Sigma$, each with basepoint $x$, such that the subsurface $\Sigma \setminus (\ell_1 \cup \ell_2 \cup \cdots \cup \ell_{2g})$ is an open topological disk; see Figure 3. The resulting disk is called a *(reduced) polygonal schema*.

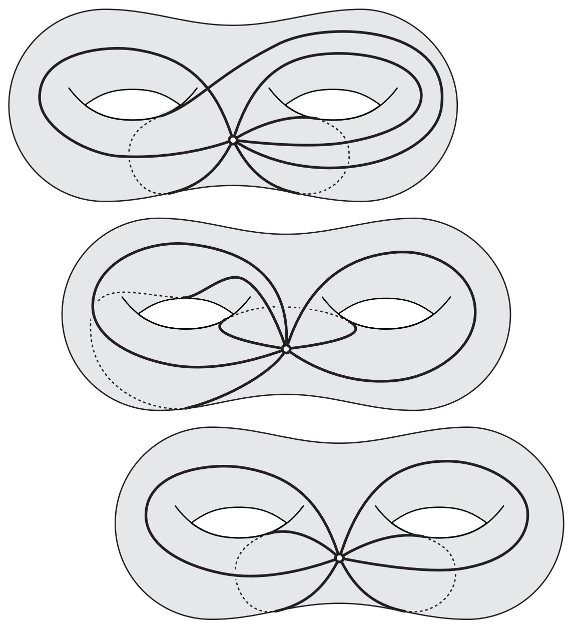

FIGURE 3. A homotopy basis that is not a system of loops, a non-canonical system of loops, and a canonical system of loops for a surface of genus 2.

Suppose we assign each loop $\ell_i$ an arbitrary orientation. Each loop $\ell_i$ appears as two paths on the boundary of the polygonal schema, once in each orientation. The cyclic sequence of these directed boundary paths is the *gluing pattern* of the polygonal schema. We obtain a one-relator presentation of $\pi_1(\Sigma, x)$, in which the loops $\ell_i$ are generators and the gluing pattern is the relator.

Many statements and proofs of the surface-classification theorem [**6, 22, 156**] rely on the existence of a *canonical* system of loops with the gluing pattern

$$\ell_1\,\ell_2\,\bar\ell_1\,\bar\ell_2\,\ell_3\,\ell_4\,\bar\ell_3\,\bar\ell_4 \cdots \ell_{2g-1}\,\ell_{2g}\,\bar\ell_{2g-1}\,\bar\ell_{2g};$$

see the bottom of Figure 3. We emphasize that most of the algorithms we describe do *not* compute canonical systems of loops; indeed, it is open whether the shortest canonical system of loops can be computed in polynomial time. Fortunately, most applications of systems of loops do not require this canonical structure.

**3.1. Without Optimization.** If we don't care about optimization, we can compute a system of loops for any combinatorial surface in $O(n)$ time, using a straightforward extension of the textbook algorithm [**168, 169**] to construct a (non-minimal) presentation of the fundamental group of an arbitrary cell complex. The algorithm was first described explicitly by Eppstein [**76**], but is implicit in earlier work of several authors [**65, 98, 161, 172, 180**].

A *spanning tree* of $G$ is a connected acyclic subgraph of $G$ that includes every vertex of $G$; a *spanning cotree* of $G$ is a subgraph $C$ such that the corresponding dual subgraph $C^*$ is a spanning tree of $G^*$.

A *tree-cotree decomposition* [**13**] is a partition of $G$ into three edge-disjoint subgraphs: a spanning tree $T$, a spanning cotree $C$, and the leftover edges $L = G \backslash (T \cup C)$. In any tree-cotree decomposition $(T, L, C)$ of any graph embedded on an orientable surface of genus $g$, the set $L$ contains exactly $n_1 - (n_0 - 1) - (n_2 - 1) = 2g$ edges. (In particular, if $g = 0$, then $L = \varnothing$, and we recover the classical result that the complement of any spanning tree of a planar graph is a spanning cotree [**187**].) For each edge $e$ in $G$, let $\ell_x(T, e)$ denote the loop obtained by concatenating the unique path in $T$ from $x$ to one endpoint of $e$, the edge $e$ itself, and the unique path in $T$ back to $x$. The set of $2g$ loops $\mathcal{L} = \{\ell_x(T, e) \mid e \in L\}$ is a system of loops.

We can construct a tree-cotree decomposition in $O(n)$ time by computing an arbitrary spanning tree $T$ of $G$, for example by breadth- or depth-first search, and then computing an arbitrary spanning tree $C^*$ of the dual subgraph $(G \setminus T)^*$. (Alternatively, we can compute a spanning tree $C^*$ of $G^*$ first, and then compute a spanning tree $T$ of $G \setminus C$.) The sequence of edges traversed by any loop $\ell_x(T, e)$ can then be extracted from $T$ and $e$ in $O(1)$ time per edge. Thus, we can construct a system of loops in $O(n+k)$ time, where $k$ denotes the total complexity of the output (the sum over all loops of the number of edges in each loop). Each loop $\ell_x(T, e)$ traverses each edge of the combinatorial surface at most twice, so $k = O(gn)$, and this bound is tight in the worst case.

THEOREM 3.1. *Given a combinatorial surface $\Sigma$ with complexity $n$ and genus $g$, we can construct a homotopy basis for $\Sigma$ in $\Theta(n + k) = O(gn)$ time.*

**3.2. Optimization.** To construct the *minimum-length* homotopy basis with a given basepoint $x$, Erickson and Whittlesey [**80**] modify the previous algorithm by choosing a particular *greedy* tree-cotree decomposition $(T, L, C)$. In this greedy decomposition,

- $T$ is the *shortest-path tree* rooted at the basepoint $x$, and
- $C^*$ is a *maximum-weight* spanning tree of the dual subgraph $(G \setminus T)^*$, where the weight of each dual edge $e^*$ is the length of the corresponding primal loop $\ell_x(T, e)$.

We call the system of loops defined by this tree-cotree decomposition the *greedy system of loops*. Erickson and Whittlesey [**80**] proved that the greedy system of loops is the shortest system of loops with basepoint $x$ using a complex exchange argument. Here we describe a simpler proof due to Colin de Verdière [**47**].

A *pointed homology basis* is a set of $2g$ loops with a common basepoint $x$ whose homology classes generate the first homology group $H_1(\Sigma; \mathbb{Z}_2) \cong (\mathbb{Z}_2)^{2g}$. The Hurewicz theorem implies that any homotopy basis (and therefore any system of loops) is also a pointed homology basis. Thus, it suffices to prove that the greedy system of loops is the pointed homology basis of minimum total length.

The core of Colin de Verdière's proof is the following exchange argument, which extends a similar characterization of shortest non-contractible and non-separating cycles by Thomassen [**181**]. For any loop $\ell$, let $[\ell]$ denote the *homology class* of $\ell$, which we identify as a vector in $(\mathbb{Z}_2)^{2g}$. A set of $2g$ loops is a pointed homology basis if and only if the corresponding set of $2g$ homology classes is linearly independent.

LEMMA 3.2. *Every loop in a minimum-length pointed homology basis has the form $\ell_x(T, e)$ for some edge $e$, where $T$ is the shortest-path tree rooted at $x$.*

PROOF. We regard the graph $G$ as a continuous metric space, in which any edge of length $w$ is isometric to the real interval $[0, w]$. Let $\overline{\alpha}$ denote the reversal of any directed path $\alpha$. Let $\alpha \cdot \beta$ denote the concatenation of two directed paths $\alpha$ and $\beta$ with matching endpoints.

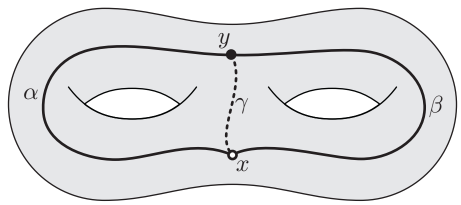


FIGURE 4. Three paths from $x$ to $y$.

Fix a loop $\ell$ with basepoint $x$ and a pointed homology basis $\mathcal{L}$ that contains $\ell$. Let $y$ be the midpoint of $\ell$, so that $\ell$ can be decomposed two paths from $x$ to $y$ of equal length. Call these paths $\alpha$ and $\beta$, so that $\ell = \alpha \cdot \overline{\beta}$; see Figure 4. If we assume shortest vertex-to-vertex paths are unique, the point $y$ lies in the interior of an edge of $G$.

Suppose there is a third path $\gamma$ from $x$ to $y$ that is shorter than both $\alpha$ and $\beta$, as illustrated in Figure 4. Then the loops $\ell^\flat = \alpha \cdot \overline{\gamma}$ and $\ell^\sharp = \gamma \cdot \overline{\beta}$ are both shorter than $\ell = \alpha \cdot \overline{\beta}$. We immediately have $[\ell] = [\ell^\flat] + [\ell^\sharp]$, which implies that either $\mathcal{L} \cup \{\ell^\sharp\} \setminus \{\ell\}$ or $\mathcal{L} \cup \{\ell^\flat\} \setminus \{\ell\}$ is a pointed homology basis of smaller total length than $\mathcal{L}$. Thus, $\mathcal{L}$ is not a minimum-length pointed homology basis.

We conclude that if $\ell$ is a member of any minimum-length pointed homology basis, then $\alpha$ and $\beta$ must be *shortest* paths from $x$ to $y$. It follows that $\ell = \ell_x(T, e)$, where $T$ is the shortest-path tree rooted at $x$ and $e$ is the edge of $G$ that contains $y$. □

We are now faced with the following problem: From the set of $O(n)$ loops $\ell_x(T, e)$, extract a subset of $2g$ loops of minimum total length whose homology classes are linearly independent. Erickson and Whittlesey [**80**] observe that the loops $\ell_x(T, e_1), \ell_x(T, e_2), \ldots, \ell_x(T, e_k)$ have linearly independent homology classes if and only if deleting those loops from the surface $\Sigma$ leaves a connected subsurface, or equivalently, if the dual subgraph $(G \setminus T)^* \setminus \{e_1^*, e_2^*, \ldots, e_k^*\}$ is connected. Thus, we seek the minimum-weight set of $2g$ edges in the dual subgraph $(G \setminus T)^*$ whose deletion leaves the graph connected. These are precisely the edges that are *not* in the maximum-weight spanning tree of $(G \setminus T)^*$.

The greedy tree-cotree decomposition can be constructed in $O(n \log n)$ time using textbook algorithms for shortest-path trees and minimum spanning trees. If $g = O(n^{1-\varepsilon})$ for some constant $\varepsilon > 0$, the time bound can be reduced to $O(n)$, using a more recent shortest-path algorithm of Henzinger *et al.* based on graph separators [**105**] and a careful implementation of Borůvka's minimum-spanning tree algorithm [**21**, **135**]. The greedy system of loops can be extracted from this decomposition in $O(1)$ time per output edge.

THEOREM 3.3. *Given a combinatorial surface* $\Sigma$ *with complexity* $n$ *and genus* $g$*, and a vertex* $x$ *of* $\Sigma$*, we can construct a system of loops with basepoint* $x$*, of minimum total length, in* $O(n \log n + k) = O(n \log n + gn)$ *time, or in* $O(n+k) = O(gn)$ *time if* $g = O(n^{1-\varepsilon})$ *for some constant* $\varepsilon > 0$.

If no basepoint is specified in advance, we can compute the globally shortest system of loops in $O(n^2 \log n + k)$ time by running the previous algorithm at every vertex. No faster algorithm is known.

**3.3. Related Results.** The tree-cotree algorithm can also be used to construct a system of loops in the dual graph $G^*$. Using this dual system of loops, one can label each edge of $G$ with a string of length $O(g)$, in $O(gn)$ time, so that the concatenation of labels along any path encodes the homotopy type of that path. A version of this encoding can be used to quickly determine whether two paths or cycles of length $k$ are homotopic in $O(n + k)$ time, via Dehn's algorithm [**59**, **61**].

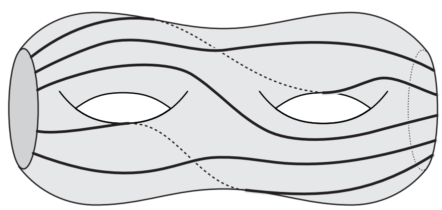

FIGURE 5. A system of five arcs for a surface with genus 2 and two boundary components

A slightly different substructure is needed to cut a surface *with boundary* into a disk. An *arc* in a surface $\Sigma$ with boundary is a path whose endpoints lie on the boundary of $\Sigma$; a set of arcs $\{\alpha_1, \alpha_2, \ldots, \alpha_\beta\}$ such that $\Sigma \setminus (\alpha_1 \cup \alpha_2 \cup \cdots \cup \alpha_\beta)$ is a topological disk is called a *system of arcs*. See Figure 5. Euler's formula implies that if $\Sigma$ has genus $g$ and $b$ boundary components, then any system of arcs for $\Sigma$ has exactly $\beta = 2g + b - 1$ elements; the number $\beta$ is the first *Betti number* of $\Sigma$. The minimum-length system of arcs can be constructed in $O(n \log n + k) = O(n \log n + (g + b)n)$ time using a variant of the greedy tree-cotree construction [**34**, **48**, **79**]. In fact, this greedy system of arcs is the minimum-length basis for the relative homology group $H_1(\Sigma, \partial\Sigma; \mathbb{Z}_2)$.

A *cut graph* is a subgraph $X$ of the graph $G$ such that $\Sigma \setminus X$ is homeomorphic to an open disk; for example, the union of all loops in a system of loops is a cut graph. Colin de Verdière [**47**] also described a similar greedy algorithm to compute the minimum-length cut graph with a prescribed set of vertices of degree greater than two. Computing the minimum-length cut graph with no prescribed vertices is NP-hard [**78**].

The algorithms described so far in this section do not necessarily construct *canonical* systems of loops. Brahana's proof of the surface classification theorem [**22**] describes an algorithm to transform any system of loops into a canonical system by cutting and pasting the polygonal schema; a more efficient transformation algorithm was later described by Vegter and Yap [**185**]. Lazarus *et al.* [**126**] described the first direct algorithm to construct a canonical system of loops in $O(gn)$ time. The minimum-length system of loops homotopic to a given system can be computed in polynomial time by an algorithm of Colin de Verdière and Lazarus [**48, 49**]. However, it is open whether the globally shortest canonical homotopy basis can be computed in polynomial time.

## 4. Shortest Homotopic Paths and Cycles

The *shortest homotopic path* problem asks, given a path $\pi$ in some combinatorial surface $\Sigma$, to compute the shortest path in $\Sigma$ that is homotopic to $\pi$. Similarly, the *shortest homotopic cycle* problem asks, given a cycle $\gamma$ in some combinatorial surface $\Sigma$, to compute the shortest cycle in $\Sigma$ that is homotopic to $\gamma$. The input and output curves need not be paths and cycles in the graph-theoretic sense; they may visit vertices and edges multiple times.

The definition of the universal cover $\tilde{\Sigma}$ of $\Sigma$ implies that the shortest path homotopic to any path $\pi$ is the projection of the shortest path in $\tilde{\Sigma}$ between the endpoints of some lift $\tilde{\pi}$ of $\pi$. This characterization does not directly yield an algorithm, however, because the universal cover is usually infinite. Algorithms to solve the homotopic shortest path problem instead first construct a finite, simply-connected, and ideally small *relevant region* of $\tilde{\Sigma}$, and then compute a shortest path between the endpoints of $\tilde{\pi}$ within this relevant region.

A slightly different strategy is needed to compute shortest homotopic cycles, because cycles lift to *infinite* paths in the universal cover; we discuss the necessary modifications in Section 4.4.

**4.1. Warm up: Polygons with Holes.** The earliest algorithms for computing shortest homotopic paths and cycles considered the Euclidean setting, where the the input path $\pi$ is a chain of $k$ line segments, and the environment is a polygon $P$ with holes of total complexity $n$. Hershberger and Snoeyink [**106**] described an efficient algorithm for this special case, simplifying an earlier algorithm of Leiserson and Maley [**129**]. Hershberger and Snoeyink's algorithm proceeds in five stages, illustrated in Figures 6 and 7 on the next page.

First, in a preprocessing phase, *triangulate* the polygon $P$ and assign each diagonal in the triangulation a unique label. The triangulation can be computed in $O(n \log n)$ time using textbook computational geometry algorithms [**55**]; faster algorithms are known when the number of holes is small [**11, 39, 167**].

Second, compute the *crossing sequence* of the input path $\pi$: the sequence of labels of diagonals crossed by the path, in order along the path. This stage is straightforward to implement in $O(k + x)$ time, where $x = O(kn)$ is the length of the crossing sequence.

Third, *reduce* the crossing sequence of $\pi$ by repeatedly removing adjacent pairs of the same diagonal label; the reduced crossing sequence can be computed in $O(x)$ time. The reduction is justified by the observation that the shortest path in any homotopy class cannot cross the same diagonal $e$ twice consecutively; otherwise, replacing the subpath between the two crossings with a sub-segment of $e$ would yield a shorter homotopic path. The reduction of the crossing sequence mirrors a homotopy from $\pi$ to the (unknown) shortest homotopic path $\pi'$. See Figure 6.

Fourth, construct the *sleeve* of triangles defined by the reduced crossing sequence of $\pi$. The sleeve is a topological disk (but not necessarily a simple Euclidean polygon) constructed by gluing together copies of the triangles in the triangulation

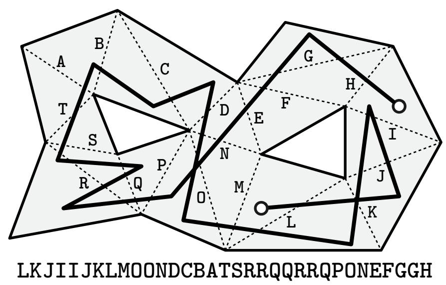


FIGURE 6. Reducing a crossing sequence

of $P$. Specifically, we start with a copy the triangle containing the starting point of $\pi$, and then for each label in the reduced crossing sequence, we attach a new copy of the triangle just beyond the corresponding diagonal. See Figure 7. The sleeve can be constructed in $O(1)$ time per triangle, or $O(x)$ time altogether.

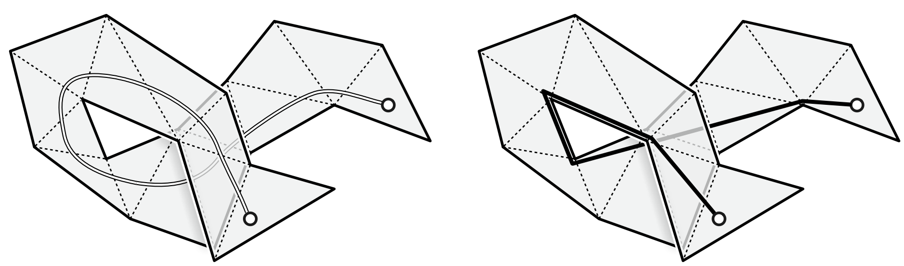

FIGURE 7. The sleeve defined by the reduced crossing sequence in Figure 6, and the shortest path within this sleeve

Finally, compute the shortest path in the sleeve between the endpoints of $\pi$; this is the shortest path in $P$ homotopic to $\pi$. This shortest path can be computed in $O(1)$ time per sleeve triangle using the *funnel* algorithm independently proposed by Tompa [**183**], Chazelle [**38**], Lee and Preparata [**127**], and Leiserson and Maley [**129**]. Although the funnel algorithm was designed to compute shortest paths in simple Euclidean polygons, it works without modification in any simply-connected domain obtained by gluing Euclidean triangles along common edges, provided all triangle vertices lie on the boundary.

THEOREM 4.1 (Hershberger and Snoeyink [**106**]). *Let $P$ be any polygon with holes with total complexity $n$, and let $\pi$ be a polygonal chain with $k$ edges. The shortest path in $P$ homotopic to $\pi$ can be computed in $O(n \log n + kn)$ time.*

**4.2. Surfaces with Boundary.** Colin de Verdière and Erickson [**48**] describe algorithms to compute shortest homotopic paths and cycles in arbitrary combinatorial surfaces. For surfaces with boundary, their algorithm follows almost exactly the same outline as Hershberger and Snoeyink's.

Here we sketch a variant of their algorithm for surfaces with boundary, essentially due to Colin de Verdière and Lazarus [**49**]. Suppose we are given a combinatorial surface $\Sigma$ with complexity $n$, genus $g$, and $b$ boundary components. In the preprocessing phase, we first compute a *greedy system of arcs* $\{\alpha_1, \alpha_2, \ldots, \alpha_\beta\}$ for $\Sigma$ in $O(n \log n + (g+b)n)$ time, as described in Section 3.3.

The greedy system of arcs has the following important property, motivated by results of Hass and Scott [**102**]. We say that two paths $\sigma$ and $\tau$ form a *bigon* if some subpath of $\sigma$ and some subpath of $\tau$ are homotopic.

LEMMA 4.2. *For any path $\pi$, there is a shortest path homotopic to $\pi$ that does not define a bigon with any arc in the greedy system of arcs.*

PROOF. Recall that the greedy system of arcs is the minimum-length collection of arcs that generates the relative homology group $H_1(\Sigma, \partial\Sigma; \mathbb{Z}_2)$. It follows that each arc $\alpha_i$ in the greedy system of arcs is a shortest path in its relative homology class, and therefore in its homotopy class.

Define the *crossing number* of a path $\pi$ to be the total number of times $\pi$ crosses the arcs $\alpha_i$ in the greedy system.

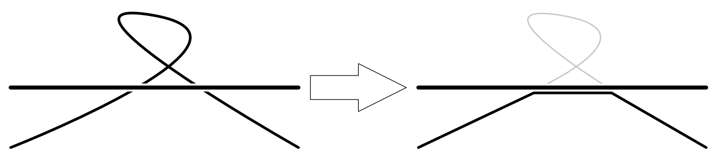

FIGURE 8. Let bigons by bygones.

If $\pi$ forms a bigon with some arc $\alpha_i$ in the greedy system of arcs, we can replace some subpath of $\pi$ with a homotopic subpath of $\alpha_i$, as shown in Figure 8. This new path is homotopic to $\pi$, is no longer than $\pi$, and has smaller crossing number than $\pi$. It follows that among all shortest paths homotopic to $\pi$, the path with smallest crossing number does not define a bigon with any arc $\alpha_i$. $\square$

Now suppose we are given a path $\pi$ with complexity $k$. We compute the *signed* crossing sequence of the directed input path $\pi$ with respect to these arcs; each time $\pi$ crosses an arc $\alpha_i$ from left to right (respectively, from right to left), the signed crossing sequence contains the symbol $i^+$ (respectively, $i^-$). The signed crossing sequence is then reduced by repeatedly removing pairs of the form $i^+i^-$ or $i^-i^+$. As in the Euclidean setting, the reduction mirrors a homotopy of $\pi$ that removes bigons one at a time; Lemma 4.2 implies that the reduced crossing sequence of $\pi$ is the crossing sequence of some shortest path homotopic to $\pi$.

To construct the relevant region $R$ of the universal cover, we glue together a sequence of $x+1$ copies of the polygonal schema $D = \Sigma \setminus (\alpha_1 \cup \alpha_2 \cup \cdots \cup \alpha_\beta)$, where $x = O((g+b)k)$ is the length of the reduced crossing sequence. The relevant region is a combinatorial *disk* with complexity $O(nx) = O((g+b)nk)$. Finally, we compute the shortest path in $R$ between the first vertex of $\pi$ in the first copy of $D$ to the last vertex of $\pi$ in the last copy of $D$, using a linear-time shortest-path algorithm for planar graphs [**105**].

THEOREM 4.3 (Colin de Verdière and Erickson [**48**]). *Let $\Sigma$ be a combinatorial surface with genus $g$ and $b \geq 1$ boundary components, and let $\pi$ be a path of $k$ edges in $\Sigma$. The shortest path in $\Sigma$ homotopic to $\pi$ can be computed in $O(n \log n + (g+b)nk)$ time.*

**4.3. Surfaces without Boundary.** For orientable surfaces with no boundary, Colin de Verdière and Erickson require a different decomposition of the surface into disks. For surfaces with genus at least 2, their preprocessing algorithm computes a *tight octagonal decomposition* of the given surface $\Sigma$ in $O(gn \log n)$ time. This is a set of $O(g)$ cycles $\{\gamma_1, \gamma_2, \ldots\}$, each as short as possible in its homotopy class, that decompose the surface into octagons meeting four at a vertex; see Figure 9. The algorithm to construct this decomposition is fairly technical and relies on several other results [**29**, **49**, **78**, **85**].

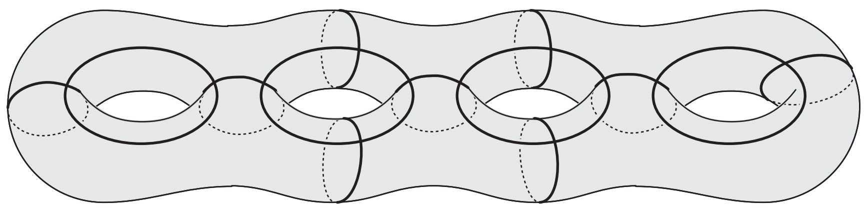

FIGURE 9. A tight octagonal decomposition.

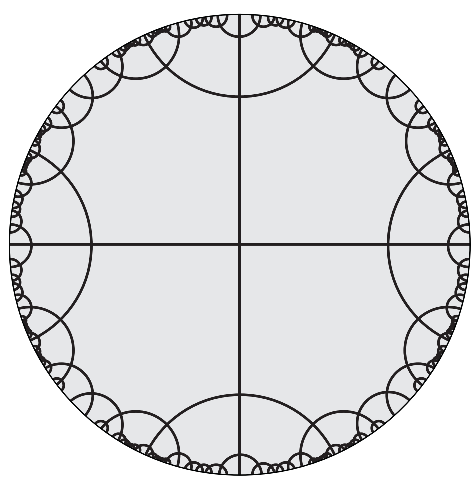

FIGURE 10. The universal cover of a tight octagonal decomposition.

The universal cover of a tight octagonal decomposition is combinatorially isomorphic to a regular tiling of the hyperbolic plane by right-angled octagons. In particular, each cycle in the decomposition lifts to a family of infinite geodesics in the universal cover $\tilde{\Sigma}$, each of which crosses any other shortest path in $\tilde{\Sigma}$ at most once. Evoking the regular hyperbolic structure, Colin de Verdière and Erickson call these infinite geodesics *lines*. These lines are drawn as circular arcs in Figure 10, following the Poincaré disk model of the hyperbolic plane.

Now let $\pi$ be a given path in $\Sigma$, and let $\tilde{\pi}$ be an arbitrary lift of $\pi$ to $\tilde{\Sigma}$. Let $X$ denote the set of lines (lifts of cycles $\gamma_i$) that $\tilde{\pi}$ crosses, and let $X'$ be the set of lines that $\tilde{\pi}$ crosses an odd number of times. Any path in $\tilde{\Sigma}$ between the endpoints in $\tilde{\pi}$ must cross every line in $X'$ an odd number of times. Moreover, an easy variant of Lemma 4.2 implies that the shortest such path crosses each line in $X'$ *exactly* once and does not cross any line not in $X'$. For this reason, we call the lines in $X'$ *relevant*.

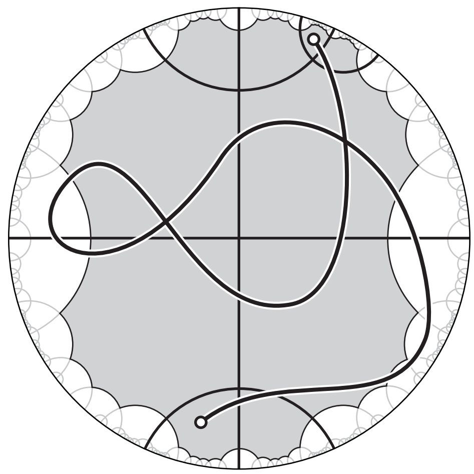

FIGURE 11. The relevant lines and relevant region of a lifted path.

Let $R(X)$ denote the complex of octagons reachable from the initial point of $\tilde{\pi}$ by crossing only (a subset of) lines in $X$; see Figure 11. A classical result of Dehn [**59**] implies that $R(X)$ is the union of at most $O(|X|)$ octagons. Colin de Verdière and Erickson describe an incremental algorithm to construct $R(X)$ in $O(x)$ time, where $x$ is the length of the signed crossing sequence of $\pi$ with respect to the cycles $\gamma_i$ [**48**]. Equivalently, $x$ is the total number of times the lifted path $\tilde{\pi}$ crosses the lines in $X$. Because the cycles in the tight octagonal decomposition may share edges, a single edge in the input path may cross $\Theta(g)$ cycles simultaneously; thus, we have an upper bound $x = O(kg)$.

The subset $X'$ can be identified in $O(x)$ time by performing a breadth-first search in the dual 1-skeleton of $R(X)$. Alternatively, $X'$ can be identified directly by reducing the signed crossing sequence of $\pi$ with respect to the cycles $\gamma_i$. The crossing sequence can no longer be reduced by merely canceling bigons—see the example in Figure 11—but the regular hyperbolic tiling allows fast reduction in $O(x)$ time using techniques from small cancellation theory [**134**,**140**]. Let $x'$ denote the length of the reduced crossing sequence; again, because $x' \leq x$, we have $x = O(kg)$.

Finally, let $R(X')$ denote the complex of octagons reachable from the initial point of $\tilde{\pi}$ by crossing only (a subset of) lines in $X'$; again, this complex can be constructed in $O(x')$ time. After filling each octagon in $R(X')$ with the corresponding planar portion of the input graph $G$, Colin de Verdière and Erickson's algorithm computes the shortest path $\tilde{\pi}'$ between the endpoints of $\tilde{\pi}$ in the resulting planar graph [**105**]. The projection of $\tilde{\pi}'$ to the original graph $G$ is the shortest path homotopic to $\pi$. Altogether, the query phase of their algorithm runs in $O(x + x'n) = O(gnk)$ time, where as usual $k$ is the number of edges in the input path.

THEOREM 4.4 (Colin de Verdière and Erickson [**48**]). *Let $\Sigma$ be an orientable combinatorial surface with genus $g \geq 2$ and no boundary, and let $\pi$ be a path of $k$*

*edges in* $\Sigma$. *The shortest path in* $\Sigma$ *homotopic to* $\pi$ *can be computed in* $O(gn \log n + gnk)$ *time.*

For shortest homotopic paths on the torus, where no hyperbolic structure is possible, a similar algorithm runs $O(n \log n + nk^2)$ time. Instead of a tight octagonal decomposition, the algorithm cuts the surface into a disk with a system of loops, where each loop is as short as possible in its homotopy class. Shortest homotopic paths in any non-orientable surface $\Sigma$ can be computed by searching the oriented double cover of $\Sigma$.

**4.4. Shortest Homotopic Cycles.** As promised, we now describe the necessary modifications to find shortest homotopic cycles. We cannot directly apply the previous algorithm, because any lift of a non-contractible cycle to the universal cover is an *infinite* path; see Figure 9.

Let $\gamma$ be a non-contractible cycle in some combinatorial surface $\Sigma$. In the simplest nontrivial special case of this problem, $\Sigma$ is a combinatorial *annulus* and $\gamma$ is one of its boundary cycles; we call any cycle homotopic to $\gamma$ a *generating* cycle. Using an argument similar to Lemma 4.2, Itai and Shiloach [**110**] observed that the shortest generating cycle crosses any shortest path between the two boundary cycles exactly once. Thus, one can compute the shortest generating cycle by cutting $\Sigma$ along a shortest path $\sigma$ between the two boundaries; duplicating every vertex and edge of $\sigma$; and then, for each vertex $v$ of $\sigma$, computing the shortest path between the two copies of $v$ in the resulting planar graph. See Figure 12.

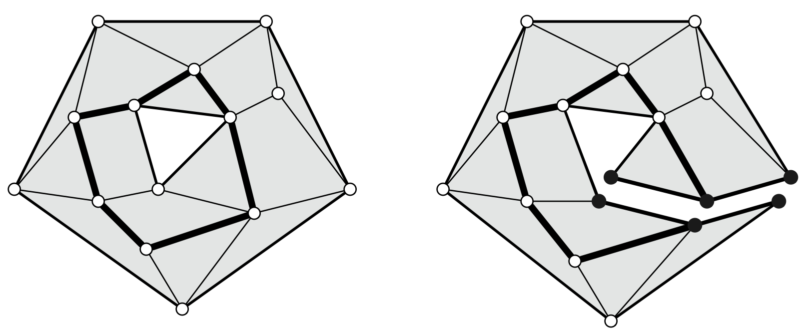

FIGURE 12. Finding the shortest nontrivial cycle in an annulus by cutting it along a shortest boundary-to-boundary path.

Itai and Shiloach applied Dijkstra's shortest-path algorithm at each vertex of $\sigma$, immediately obtaining a running time of $O(n^2 \log n)$ [**110**]. Reif [**160**] improved the running time of this algorithm to $O(n \log^2 n)$ using a divide-and-conquer strategy. Frederickson [**85**] further improved the running time to $O(n \log n)$ using a recursive separator decomposition [**132**] to speed up the shortest-path computations. The same improvement can be obtained using more recent algorithms for shortest paths [**105**,**121**] and maximum flows [**19**,**77**] in planar graphs. Most recently, Italiano *et al.* [**111**] improved the running time to $O(n \log \log n)$ using a more careful separator decomposition and other algorithmic tools for planar graphs [**82**,**85**,**142**].

Colin de Verdière and Erickson [**48**] describe a reduction from the more general shortest homotopic cycle problem to this special case. At a very high level, the algorithm identifies an infinite *periodic* relevant region $R$ in the universal cover that contains a lift of the shortest homotopic cycle. The algorithm actually constructs one period $R_0$ of this infinite relevant region, identifies corresponding boundary paths of $R_0$ to obtain a combinatorial annulus $A$, and finally computes the shortest generating cycle of $A$. Yin *et al.* [**192**] sketch a similar algorithm to compute shortest homotopic cycles; however, their description omits several key details and offers no time analysis.

More concretely, suppose the surface $\Sigma$ has genus 2 and no boundary. Let $\gamma\colon \mathbb{R}/\mathbb{Z} \to \Sigma$ denote the input cycle, and let $\tilde{\gamma}\colon \mathbb{R} \to \tilde{\Sigma}$ be one of its lifts to the universal cover. There is a translation $\tau\colon \tilde{\Sigma} \to \tilde{\Sigma}$ such that $\tau(\tilde{\gamma}(t)) = \tilde{\gamma}(t+1)$ for all $t$. Fix an arbitrary lift $\tilde{v}_0$ of some vertex $v$ of $\gamma$, and for each positive integer $i$, let $\tilde{v}_i = \tau(\tilde{v}_{i+1})$. The set $X'$ of all lines in the tight octagonal tiling of $\tilde{\Sigma}$ that separate $\tilde{v}_0$ from $\tilde{v}_4$ can be computed using the same algorithm as for shortest homotopic paths. Let $\ell_2$ be any line that separates $\tilde{v}_0$ and $\tilde{v}_1$ from $\tilde{v}_2$ and $\tilde{v}_3$, and let $\ell_3 = \tau(\ell_2)$; both of these lines lie in the set $X'$. Let $R(X')$ be the complex of octagons reachable from $\tilde{v}_0$ by crossing only lines in $X'$. Finally, let $R_0$ be the portion of $R(X')$ that lies between $\ell_2$ and $\ell_3$; identifying the segments of $\ell_2$ and $\ell_3$ on the boundary of $R_0$ gives us the annulus $A$.

Combining Colin de Verdière and Erickson's reduction with the $O(n \log\log n)$-time algorithm of Italiano *et al.* for the annulus [**111**] gives us the following result.

THEOREM 4.5. *Let $\Sigma$ be a combinatorial surface with genus $g$ and $b$ boundaries, and let $\gamma$ be a cycle of $k$ edges in $\Sigma$. The shortest cycle homotopic to $\gamma$ can be computed in $O(n\log n + (g+b)nk \log\log nk)$ time if $b > 0$, in $O(gn \log n + gnk \log\log gnk)$ time if $b = 0$ and $g > 1$, and in $O(n \log n + nk^2 \log\log nk)$ time if $g = 1$ and $b = 0$.*

No efficient algorithm is known for computing shortest homotopic paths in *non-orientable* surfaces.

# —— PART II. HOMOLOGY ——

## 5. General Remarks

We now turn from homotopy to homology. Unlike the homotopy problems we have considered so far, questions about homology are decidable for any finite regular cell complex. Not surprisingly, however, these problems can be solved more efficiently for combinatorial 2-manifolds than for general complexes, so we consider algorithms for surfaces separately.

Before we consider any algorithms, we must carefully define the functions we wish to optimize. Fix a simplicial complex $\Sigma$ and a coefficient ring $R$. A *$p$-chain* is a formal linear combination of oriented $p$-simplices in $\Sigma$, which we identify with a vector $\boldsymbol{c} = (c_1, c_2, \dots, c_{n_p}) \in R^{n_p}$. Fix a vector $\boldsymbol{w} = (w_1, w_2, \dots, w_{n_p}) \in \mathbb{R}^{n_p}$ that assigns a non-negative weight to each $p$-cell in $\Sigma$. In the interest of developing *exact* optimization algorithms, we restrict our attention to two definitions of the "weight" of a $p$-chain:

- The *weighted $L_0$-norm* is the sum of the weights of all cells with non-zero coefficients:
$$\|\boldsymbol{c}\|_{0,\boldsymbol{w}} := \sum_{i : c_i \neq 0} w_i,$$

- The *weighted $L_1$-norm* is the weighted sum of the absolute values of the coefficients:
$$\|\boldsymbol{c}\|_{1,\boldsymbol{w}} := \sum_i w_i\,|c_i|.$$

The weighted $L_0$-norm is well-defined for any coefficient ring $R$; the weighted $L_1$-norm is well-defined only when $R$ is a sub-ring of the reals. We call a $p$-chain *unitary* if every coefficient lies in the set $\{-1, 0, 1\}$; the weighted $L_0$- and $L_1$-norms of a chain are equal if and only if the chain is unitary.

Formally, a $p$-cycle is any $p$-chain that lies in the kernel of the boundary map $\partial_p \colon R^{n_p} \to R^{n_{p-1}}$. However, when we seek $p$-cycles or homology bases with minimum weighted $L_0$-norm, it is convenient to conflate any $p$-cycle $\boldsymbol{c}$ with the subset of $p$-cells with non-zero coefficients $c_i$. To avoid confusion over the multiple meanings of the word “cycle”, we consistently refer to the elements of a $p$th homology class as *$p$-cycles*, and closed walks in the 1-skeleton of $\Sigma$ as *loops*.

Finally, a *$p$th homology basis* is a minimum-cardinality set of $p$-cycles whose homology classes generate the $p$th homology group $H_p(\Sigma; R)$. We define the *weight* of a homology basis to be the sum of the weighted $L_0$-norms of its constituent $p$-cycles. Although the total $L_1$-norm of a homology basis is well-defined when $R \subseteq \mathbb{R}$, the corresponding optimization problem is uninteresting. If $R = \mathbb{Z}$, then every $p$-cycle in the $L_1$-minimal homology basis is unitary; thus, the $L_1$-minimal homology basis is also the $L_0$-minimal homology basis. On the other hand, if $R$ contains numbers arbitrarily close to zero (for example, if $R = \mathbb{Q}$), there are homology bases whose weighted $L_1$-norm is arbitrarily close to zero.

## 6. Homology Bases

Let $\Sigma$ be an arbitrary simplicial complex with weighted cells, and let $p$ be a positive integer. Our goal in this section is to find a $p$th homology basis whose total weighted $L_0$-norm is as small as possible. Because we are optimizing the weighted $L_0$-norm, we need not distinguish between a $p$-cycle and the subset of $p$-cells with non-zero coefficients. In particular, any *first* homology basis for $\Sigma$ is a minimal set of simple *loops* in the 1-skeleton of $\Sigma$; we seek to minimize the total length of these loops. See Figure 13.

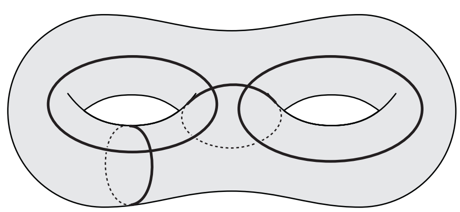

Figure 13. A first homology basis for a surface of genus 2.

**6.1. Surfaces.** For combinatorial surfaces, any system of loops is also a first homology basis, for any coefficient ring; thus, Eppstein’s tree-cotree algorithm constructs a homology basis for any combinatorial surface in $O(n + k) = O(gn)$ time. We can reduce the output size by replacing each loop $\ell_x(T, e)$ with the unique *simple* loop in the graph $T \cup \{e\}$, but the output size is still $\Theta(gn)$ in the worst case.

However, the minimum-weight first homology basis is not necessarily consistent with any tree-cotree decomposition.

Erickson and Whittlesey [**80**] describe an efficient algorithm to compute the optimal first homology basis with respect to any coefficient *field*, generalizing an earlier algorithm of Horton [**108**] to compute the shortest cycle basis of an edge-weighted graph. Their algorithm is simplified by the following observation of Dey *et al.* [**66**]: Every generator in the optimal *homology* basis is also a generator in the optimal *homotopy* basis for some basepoint. Thus, in $O(n^2 \log n + gn^2)$ time, we can compute a set of $O(gn)$ candidate loops that must include the optimal homology basis. Again, the $n^2 \log n$ term can be removed from the running time if $g = n^{1-\varepsilon}$ for any $\varepsilon > 0$. The homotopy basis algorithm automatically computes the length of each candidate loop.

We can also compute vectors of length $2g$ that encode the homology class of each candidate loop using an elementary form of Poincaré duality. Recall that we associate two *darts* $u{\to}v$ and $v{\to}u$ with each edge $uv$ of the embedded graph $G$. We first construct a *dual* homology basis $\{\lambda_1^*, \lambda_2^*, \dots, \lambda_{2g}^*\}$ using an arbitrary tree-cotree decomposition $(T, L, C)$; each element $\lambda_i^*$ of the dual homology basis is a loop in the dual graph $G^*$, obtained by adding an edge in $L^*$ to the dual spanning tree $C^*$. We orient the loops $\lambda_i^*$ arbitrarily. We then label each dart $u{\to}v$ in the primal graph with a vector $h(u{\to}v) \in R^{2g}$, called the *homology signature* of the edge, whose $i$th coordinate is $+1$ if the dart crosses $\lambda_i^*$ from left to right, $-1$ if the dart crosses $\lambda_i^*$ from right to left, and 0 otherwise. The homology class of any directed loop $\gamma$ in $G$ is then the sum of the homology signatures of its edges. By accumulating homology signatures along every root-to-leaf path in every shortest-path tree, we can compute the homology classes of all $O(gn)$ candidate loops in $O(gn^2)$ total time.

We are now faced with a standard *matroid optimization* problem: Given a collection of $O(gn)$ vectors, each with non-negative weight, find a subset of minimum total weight that generates the *vector space* $R^{2g}$. This problem can be solved by a greedy algorithm, similar to Kruskal's classical minimum-spanning-tree algorithm [**124**]. Starting with an empty basis, we consider the vectors one at a time in order of increasing weight; whenever we encounter a vector that is linearly independent of the vectors already in the basis, add it to the basis. Sorting the vectors takes $O(gn \log n)$ comparisons, and for each vector, we need $O(g^2)$ arithmetic operations to test linear independence.

Theorem 6.1 (Erickson and Whittlesey [**80**]). *Let $\Sigma$ be a combinatorial surface with complexity $n$ and genus $g$. An optimal first homology basis for $\Sigma$, with respect to any coefficient field, can be computed in $O(n^2 \log n + gn^2 + g^3 n)$ time, or in $O(gn^2 + g^3 n)$ time if $g = O(n^{1-\varepsilon})$ for any $\varepsilon > 0$.*

The restriction to coefficient fields is unfortunately necessary. For coefficient rings without division, homology groups are not vector spaces, and thus a maximal linearly independent set of homology classes is not necessarily a basis. Gortler and Thurston [**91**] have shown that Erickson and Whittlesey's greedy algorithm can return a set of $2g$ loops that do *not* generate the first homology group, although they lie in independent $\mathbb{Z}$-homology classes, even when $g = 2$. It is natural to conjecture that computing an optimal $\mathbb{Z}$-homology basis is NP-hard, but no such result is known, even for more general complexes.

**6.2. Complexes.** If we do not care about optimization, we can compute a homology basis of any dimension, for any simplicial complex, over any coefficient ring $R$, using the classical Poincaré-Smith reduction algorithm [**155, 173**], which requires $O(n^3)$ arithmetic operations over the coefficient ring $R$. This is not the fastest algorithm known. Homology over any field can be computed by an algorithm of Bunch and Hopcroft [**25**], whose complexity is dominated by the time to multiply two $n \times n$ matrices; the fastest algorithm known for that problem requires only $O(n^{2.376})$ arithmetic operations [**51**]. On the other hand, these running times are misleading when $R = \mathbb{Z}$, as careless implementations can produce intermediate integers with exponentially many bits in the worst case [**83**]. The first polynomial-time algorithm for computing integer homology was described by Kannan and Bachem [**114**]; for a sample of more recent results, see Dumas *et al.* [**68**,**69**] and Eberly *et al.* [**72**].

Chen and Freedman [**41**] and Dey *et al.* [**66**] extend Erickson and Whittlesey's greedy strategy to compute optimal first homology bases with $\mathbb{Z}_2$ coefficients, for an arbitrary simplicial complex, in polynomial time. Instead, both algorithms modify the input complex by gluing disks to all loops in the evolving basis. A new loop $\gamma$ is accepted into the evolving basis if and only if gluing a disk to $\gamma$ decreases the rank $\beta_1$ of the first homology group. Chen and Freedman's algorithm [**41**] recomputes $\beta_1$ from scratch at each iteration, using an algorithm for sparse Gaussian elimination over finite fields [**190**]; their algorithm runs in $O(\beta_1 n^3 \log^2 n)$ time with high probability. Dey *et al.* [**66**] exploit persistent homology [**73**,**74**,**195**] to achieve a running time of $O(n^4)$. Both algorithms can be extended to compute optimal homology bases over any coefficient *field*; in this more general setting, the algorithm of Dey *et al.* [**66**] requires $O(n^4)$ arithmetic operations. Again, the restriction to coefficient fields is necessary; no polynomial-time algorithm or NP-hardness result is known for computing optimal $\mathbb{Z}$-homology bases.

Chen and Friedman further generalize their algorithm to compute minimum-cost bases for higher-dimensional homology groups, where the cost of a single generator is the *radius* of the smallest ball that contains it [**41**]. However, for the arguably more natural weighted $L_0$-norm, they also prove that computing optimal $p$th homology bases (over $\mathbb{Z}_2$) is NP-hard, for any $p \geq 2$ [**40**].

## 7. Homology Localization over $\mathbb{Z}_2$

We now turn to finding optimal representatives in a single given homology class. The complexity of this problem depends critically on the choice of coefficient ring. Somewhat surprisingly, in light of results for homology bases, optimization over finite fields turns out to be significantly harder than optimization either over the reals or (at least for manifolds) the integers.

We first consider one-dimensional homology over $\mathbb{Z}_2$. With this coefficient field, a 1-chain in any simplicial complex $\Sigma$ is a *subgraph* of the 1-skeleton; a 1-cycle is a subgraph in which every vertex has even degree (henceforth, an *even subgraph*); a *1-boundary* is the boundary of the union of a subset of 2-cells; and two even subgraphs are $\mathbb{Z}_2$-homologous if and only if their symmetric difference is a boundary subgraph. The weight of an even subgraph is just the sum of the weights of its edges; this is equivalent to the weighted $L_0$-norm. Our goal in this section is to find a minimum-weight even subgraph $\mathbb{Z}_2$-homologous to a given even subgraph.

Unfortunately, any reasonable variant of this problem is NP-hard, even in combinatorial surfaces. An argument of Chambers *et al.* [**34**] can be modified to show that finding the minimum-weight *connected* even subgraph $\mathbb{Z}_2$-homologous to a given simple loop is NP-hard, by reduction from the Hamiltonian cycle problem in planar grid graphs [**109**]. A refinement of this argument by Cabello *et al.* [**28**] implies that finding the shortest *simple cycle* in a given $\mathbb{Z}_2$-homology class is NP-hard. Later Chambers *et al.* [**36**] proved that finding the optimal even subgraph $\mathbb{Z}_2$-homologous to a given simple loop is NP-hard, by reduction from the minimum cut problem in graphs with negative edges [**141**]. For more general complexes, Chen and Friedman [**40**] prove that even *approximating* the minimum-weight even subgraph in a given $\mathbb{Z}_2$-homology class by a constant factor is NP-hard, even when the rank of the first $\mathbb{Z}_2$-homology group is 1, by reduction from the nearest codeword problem [**9**].

However, for combinatorial surfaces with constant genus, it is possible to find minimal representatives in *every* $\mathbb{Z}_2$-homology class in $O(n \log n)$ time. Specifically, Erickson and Nayyeri [**79**] describe an algorithm to compute either the shortest loop or the shortest even subgraph in a given $\mathbb{Z}_2$-homology class in $2^{O(g)} n \log n$ time, simplifying and improving an earlier algorithm by Chambers *et al.* [**34**] that runs in $g^{O(g)} n \log n$ time.

Erickson and Nayyeri's algorithm first constructs the *$\mathbb{Z}_2$-homology cover* $\overline{\Sigma}$, which is the unique connected covering space of $\Sigma$ whose group of deck transformations is $H_1(\Sigma; \mathbb{Z}_2) \cong (\mathbb{Z}_2)^{2g}$. They give two different but equivalent descriptions of the construction, one in terms of voltage graphs [**95**, Chapter 4], the other directly topological. We sketch the second formulation here. The construction is easier to visualize for a simple surface with boundary; see Figure 14.

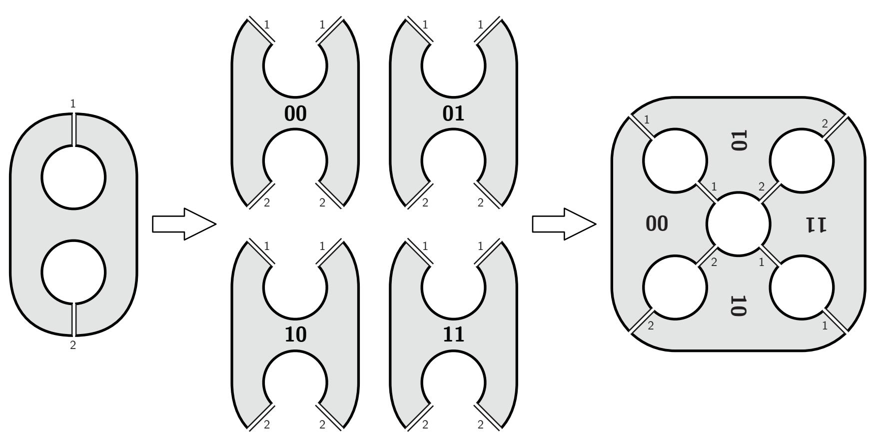


FIGURE 14. Constructing the $\mathbb{Z}_2$-homology cover of a pair of pants.

Let $\{\ell_1, \ell_2, \dots, \ell_{2g}\}$ be any system of loops for $\Sigma$, such as the one constructed from a tree-cotree decomposition by Eppstein's algorithm; for surfaces with boundary, we use a system of arcs instead, as shown in Figure 14. The surface $D := \Sigma \setminus (\ell_1 \cup \cdots \cup \ell_{2g})$ is a topological disk; each loop $\ell_i$ appears on the boundary of $D$ as two boundary segments $\ell_i^+$ and $\ell_i^-$. For each homology class $h \in (\mathbb{Z}_2)^{2g}$, we create a disjoint copy $(D, h)$ of $D$; for each index $i$, let $(\ell_i^+, h)$ and $(\ell_i^-, h)$ denote the copies of $\ell_i^+$ and $\ell_i^-$ in the disk $(D, h)$. (In Figure 14, each copy $(D, h)$ is labeled by a 2-bit string representing the homology class $h$.) For each index $i$, let $b_i$ denote the $2g$-bit vector whose $i$th bit is equal 1 and whose other $2g-1$ bits are all equal to 0. The $\mathbb{Z}_2$-homology cover $\overline{\Sigma}$ is constructed by gluing the copies of $D$ together by identifying boundary paths $(\ell_i^+, h)$ and $(\ell_i^-, h \oplus b_i)$, for every index $i$ and every homology class $h$, where $\oplus$ represents bitwise exclusive-or. (For example, in Figure 14, the copies $D_{00}$ and $D_{01}$ are glued together along a copy of $\ell_1$.) The resulting combinatorial surface has $\overline{n} = 2^{2g}n$ vertices, each labeled by a pair $(v, h)$ for some vertex $v$ in $\Sigma$ and some homology class $h$, and genus $\overline{g} = 2^{2g}(g-1)+1$. The entire construction takes $2^{O(g)}n$ time.

Because each connected component of an even subgraph has a closed Euler tour, we can reasonably regard any even subgraph as a collection of vertex-disjoint loops. We define the $\mathbb{Z}_2$-homology class of a loop as the $\mathbb{Z}_2$-homology class of its *carrier*: the subgraph of edges that the loop traverses an odd number of times. The weight of a loop is defined as the sum of the weights of its edges, *counted with appropriate multiplicity*; thus, if a loop traverses any edge more than once, its weight is larger than the weight of its carrier.

First consider the related problem of finding the shortest *loop* in a given $\mathbb{Z}_2$-homology class; we emphasize that we must consider loops that repeat edges, because the shortest loop in a given $\mathbb{Z}_2$-homology class need not be simple. Let $\ell$ be any loop in $\Sigma$, and let $[\ell]$ denote its $\mathbb{Z}_2$-homology class. The loop $\ell$ is the projection of a path in $\overline{\Sigma}$ from $(v, 0)$ to $(v, [\ell])$, where $v$ is any vertex in $\ell$. Thus, the shortest loop in homology class $h$ is the projection of the shortest path in $\overline{\Sigma}$ from some vertex $(v, 0)$ to the corresponding vertex $(v, h)$. This shortest path can be found in $O(n \cdot \overline{n} \log \overline{n}) = 2^{O(g)}n^2 \log n$ time by computing a shortest-path tree at every vertex $(v, 0)$. Erickson and Nayyeri [**79**] reduce the running time to $O(\overline{gn} \log \overline{n}) = 2^{O(g)}n \log n$ using more complex shortest-path data structures [**26**, **27**, **121**].

The minimum-weight even subgraph in any $\mathbb{Z}_2$-homology class has at most $g$ connected components, each of which is (the carrier of) the shortest loop in its own $\mathbb{Z}_2$-homology class. To find the optimal even subgraph, Erickson and Nayyeri first compute the shortest loop in *every* $\mathbb{Z}_2$-homology class and then assemble the components using a simple dynamic programming algorithm. Specifically, let $C(h, k)$ denote the minimum total weight of any set of at most $k$ loops whose homology classes sum to $h$. This function obeys the recurrence

$$C(h, k) \;=\; \min_{h'} \big(C(h', k-1) + C(h \oplus h', 1)\big),$$

where $h'$ ranges over all $\mathbb{Z}_2$-homology classes. The base cases are $C(0, k) = 0$ and $C(h, 1)$, which has already been computed for each $h$. The dynamic programming algorithm computes $C(h, g)$ in $2^{O(g)}$ additional time.

THEOREM 7.1 (Erickson and Nayyeri [**79**]). *Given a combinatorial surface $\Sigma$ with complexity $n$ and genus $g$, the minimum-weight even subgraph of $\Sigma$ in any (in fact,* ***every****) $\mathbb{Z}_2$-homology class can be computed in $2^{O(g)}n \log n$ time.*

This algorithm can be used directly to compute minimum cuts in surface-embedded graphs. Fix a graph $G$, where every edge has a non-negative *capacity*, and two vertices $s$ and $t$. An $(s, t)$*-cut* is a subset of edges of $G$ that contains at least one edge in every path from $s$ and $t$. Itai and Shiloach [**110**] proved that the minimum-capacity $(s,t)$-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, minimum cuts in undirected planar graphs can be computed by finding the shortest generating cycle in an annulus, as described in Section 4.4. Chambers *et al.* [**36**] generalized Itai and Shiloach's result to higher-genus surfaces, by proving that the minimum-capacity $(s,t)$-cut is dual to the minimum-weight even subgraph in $G^*$ that is $\mathbb{Z}_2$-homologous with the boundary of $s^*$ in the punctured surface $\Sigma \setminus (s^* \cup t^*)$. This result together with Theorem 7.1 immediately implies the following.

THEOREM 7.2 (Erickson and Nayyeri [**79**]). *Given an undirected graph $G$ with non-negative edge capacities, embedded on a surface $\Sigma$ with genus $g$, and two vertices $s$ and $t$, a minimum $(s,t)$-cut in $G$ can be computed in $2^{O(g)} n \log n$ time.*

Alternatively, recent results of Italiano *et al.* [**111**] also improve the running time of the algorithm of Chambers *et al.* [**34**] from $g^{O(g)} n \log n$ to $g^{O(g)} n \log\log n$. The resulting algorithm is faster than Erickson and Nayyeri's algorithm for graphs of constant genus, but it is also considerably more complex.

No similar algorithm is known for directed graphs. We consider the dual maximum-flow problem in Section 8.5.

Finally, Erickson and Nayyeri's algorithm can be generalized to $p$-manifold complexes of dimension any $p > 2$, using an arbitrary basis for first cohomology group $H^1(\Sigma; \mathbb{Z}_2) \cong H_{p-1}(\Sigma; \mathbb{Z}_2)$ (or the first relative cohomology group $H^1(\Sigma, \partial\Sigma; \mathbb{Z}_2) \cong H_{p-1}(\Sigma; \mathbb{Z}_2)$ if the manifold has boundary) in place of a system of loops or arcs. Each element of such a cohomology basis is a subgraph of the 1-skeleton of the complex. Thus, the cohomology basis can be used to construct the 1-skeleton of the $\mathbb{Z}_2$-homology cover using the same voltage-graph construction. The data structures used to accelerate shortest-path computations in surface graphs have no higher-dimensional analogue; otherwise, the remainder of the algorithm is unchanged. If we use the standard Poincaré-Smith reduction algorithm to compute the (relative) cohomology basis, the resulting algorithm requires $O(n^3) + 2^{O(\beta)} n^2 \log n$ real arithmetic operations, where $\beta$ is the rank of the first $\mathbb{Z}_2$-homology group.

## 8. Homology Localization over $\mathbb{R}$ and $\mathbb{Z}$

We now switch to finding representatives in real and integer homology classes whose weighted $L_1$-norm is minimized. For homology over the *reals*, optimal homologous chains can be computed in polynomial time via linear programming. If the input complex satisfies certain conditions—in particular, if the input is an orientable $(p+1)$-manifold—the resulting linear programs actually have integral solutions and thus can be used without modification to find optimal representatives in *integer* homology classes. In general, however, finding optimal $\mathbb{Z}$-homologous chains is NP-hard.

**8.1. Real homology via linear programming.** Fix a simplicial complex $\Sigma$ and a non-negative integer $p \geq 0$. To simplify notation, let $m$ and $n$ respectively denote the number of $p$-simplices and $(p+1)$-simplices in $\Sigma$. In real simplicial homology, a *p-chain* is a formal linear combination of oriented $p$-simplices in $\Sigma$, which we identify with a real vector $\boldsymbol{c} = (c_1, c_2, \ldots, c_m) \in \mathbb{R}^m$. Two $p$-chains are $\mathbb{R}$-homologous if their difference (as vectors) lies in the kernel of the boundary map $\partial_{p+1} \colon \mathbb{R}^n \to \mathbb{R}^m$. Given a $p$-chain $\boldsymbol{c}$ and a weight vector $\boldsymbol{w} \in \mathbb{R}^m$, our goal is to find a $p$-chain $\boldsymbol{x}$ with minimum weighted $L_1$-norm that is $\mathbb{R}$-homologous to $\boldsymbol{c}$.

This optimization problem can be solved in polynomial time by formulating it as a linear program as follows [**62**, **176**]:

$$\text{(LP)} \qquad \begin{array}{rl} \text{minimize} & \sum_i (x_i^+ + x_i^-) \cdot w_i \\ \text{subject to} & \boldsymbol{x}^+ - \boldsymbol{x}^- = \boldsymbol{c} + [\partial_{p+1}]\boldsymbol{y} \\ & \boldsymbol{x}^+ \geq 0 \\ & \boldsymbol{x}^- \geq 0 \end{array}$$

Here, the variable $\boldsymbol{x} = \boldsymbol{x}^+ - \boldsymbol{x}^- \in \mathbb{R}^m$ represents the output $p$-chain, the variable $\boldsymbol{y} \in \mathbb{R}^n$ represents an unknown $(p+1)$-chain, and $[\partial_{p+1}]$ is the $m \times n$ boundary matrix. The output vector $\boldsymbol{x}$ is split into two *non-negative* vectors $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ so that the weighted $L_1$-norm can be represented as a linear objective function.

We emphasize that this linear programming formulation does *not* require the input $p$-chain $\boldsymbol{c}$ to have zero boundary. Thus, for example, we can use this formulation to find a minimal surface with prescribed boundary in (the 2-skeleton of) a triangulation of $\mathbb{R}^3$, by letting $\boldsymbol{c}$ be any 2-chain with the desired boundary [**62**, **71**, **176**].

**8.2. Integer homology is hard.** Althaus and Fink [**8**] proved that finding minimum $\mathbb{Z}$-homologous *unitary* 2-chains is NP-hard, by reduction from 3-dimensional matching [**87**, **115**]. Dunfield and Hirani [**71**] removed the coefficient restriction, proving NP-hardness by reduction from 1-in-3-SAT [**87**,**163**]; their proof simplifies an earlier proof by Agol *et al.* [**3**] that finding a minimum-area surface with a given boundary in a *piecewise-linear* 3-manifold is NP-hard.

The minimum-cost representative in a given *integer* homology class is the solution to an *integer* program, obtained by adding the constraints $\boldsymbol{x}^+, \boldsymbol{x}^- \in \mathbb{Z}^m$ and $\boldsymbol{y} \in \mathbb{Z}^n$ to the linear program (LP). Integer programming is well-known to be NP-hard in general [**87**, **115**]. However, some interesting families of of integer programs can be solved in polynomial time, and these can be exploited to compute optimal $\mathbb{Z}$-homologous $p$-chains in polynomial time in certain families of spaces.

**8.3. Total unimodularity.** A matrix is *totally unimodular* if every square minor has determinant $-1$, 0, or 1. Cramer's rule implies that for any totally unimodular matrix $A$ and any integer vector $\boldsymbol{b}$, every vertex of the polyhedron $\{\boldsymbol{x} \mid A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq 0\}$ is integral [**107**]. Thus, any linear program with a totally unimodular constraint matrix $A$ and an integral constraint vector $\boldsymbol{b}$ has an integral solution. In other words, a totally unimodular integer program can be solved in polynomial time by dropping the integrality constraint and solving the resulting linear program.

Call a simplicial complex *totally $p$-unimodular* if its $(p+1)$th boundary matrix $[\partial_{p+1}]$ is totally unimodular. For such complexes, the linear program (LP) automatically has integral solutions, and thus can be used to find optimal $\mathbb{Z}$-homologous chains in polynomial time [**62**].

Total unimodularity is of central importance in combinatorial optimization, but its first application was actually topological. In the same paper that first describes the standard reduction algorithm to compute homology [**155**], Poincaré observed (in modern terminology) that the $p$th homology group $H_p(\Sigma; \mathbb{Z})$ of any totally $p$-unimodular simplicial complex $\Sigma$ is torsion-free. This observation is a straightforward consequence of the reduction algorithm; each diagonal element in the Smith normal form of any matrix is the greatest common divisor of all subdeterminants of a certain size [**173**].

Poincaré also described a simple condition involving cycles of elements in the boundary matrix that implies total unimodularity. Poincaré's condition is more easily explained in topological terms. Following Dey *et al.* [**62**], a *cycle complex* is a pure simplicial complex whose dual 1-skeleton is a cycle, and a *Möbius complex* is a non-orientable cycle complex. For example, a 2-dimensional Möbius complex is a triangulation of the Möbius band with all vertices on the boundary, and 1-dimensional Möbius complexes do not exist. Poincaré proved by induction that any simplicial complex $\Sigma$ with no $(p+1)$-dimensional Möbius subcomplex is totally $p$-unimodular [**155**, Section 6]. (It follows immediately that the incidence matrix of any directed graph—that is, the 1st boundary matrix of any 1-dimensional simplicial complex—is totally unimodular; this theorem is often attributed to Heller and Tomkins [**104**].)

As observed by Dey *et al.* [**62**], Poincaré's theorem implies that all orientable $(p+1)$-manifolds (possibly with boundary) are totally $p$-unimodular, as are all simplicial complexes embedded in $\mathbb{R}^{p+1}$. It follows that optimal $\mathbb{Z}$-homologous $p$-chains in such complexes can be computed in polynomial time. We discuss a more efficient algorithm for these special cases in the next section. Grady [**93, 94**] sketches a slightly weaker condition than total $p$-unimodularity that still supports polynomial-time solutions.

Dey *et al.* [**62**] recently extended these results in several directions. First, they proved that a complex $\Sigma$ is totally $p$-unimodular if and only if the relative homology group $H_p(L, L_0)$ is torsion-free for all pure subcomplexes $L_0 \subset L \subseteq \Sigma$ such that $L_0$ has dimension $p$ and $L$ has dimension $p+1$. Note that a $(p+1)$-dimensional cycle complex $L$ is a Möbius complex if and only if $H_p(L, \partial L)$ has nontrivial torsion. They also proved that any 2-dimensional complex is totally 1-unimodular *if and only if* it has no Möbius subcomplex; this equivalence does not extend to higher dimensions.

Finally, Dey *et al.* [**62**] proved that optimal *unitary* $\mathbb{Z}$-homologous $p$-chains in totally $p$-unimodular complexes can be computed in polynomial time, by adding the constraints $\boldsymbol{x}^+ \leq 1$ and $\boldsymbol{x}^- \leq 1$ to (LP) and solving the resulting linear program. The solution $\boldsymbol{x}$ to this augmented linear program is not necessarily the homologous chain with minimum weighted $L_0$-norm [**62**, Remark 3.11].

**8.4. Manifolds and circulations.** Motivated by a problem in minimal surface construction, Sullivan [**176**] developed a polynomial-time algorithm for the special case where $\Sigma$ is an orientable $(p+1)$-manifold, exploiting both linear programming duality and Poincaré duality. For example, when $p = 2$, Sullivan's algorithm finds minimum-weight homologous 2-chains (intuitively, discrete surfaces) in triangulated 3-manifolds. Essentially the same algorithm was rediscovered by Bueller *et al.* [**24, 92, 119**]; see also recent related results of Grady [**93, 94**].

Let $G$ denote the dual 1-skeleton of $\Sigma$; this graph has a vertex for every $(p+1)$-cell of $\Sigma$ and an edge for every $p$-cell of $\Sigma$. Conveniently, $G$ has $n$ vertices and $m$ edges. Recall that each edge $e$ in $G$ is represented by a symmetric pair of directed edges or *darts*; each dart is dual to one of the orientations of the $p$-cell whose dual edge is $e$. The dual of (LP) is another linear program, which has a dual variable for each dart:

$$\text{(LP}^*\text{)}\qquad \begin{array}{lll} \text{maximize} & \sum_{u\to v} \varphi_{u\to v}\, c_{u\to v} & \\ \text{subject to} & \sum_u \varphi_{u\to v} = \sum_u \varphi_{v\to u} & \text{for every vertex } v \\ & \varphi_{u\to v} \le w_{uv} & \text{for every dart } u\to v \\ & \varphi_{u\to v} \ge 0 & \text{for every dart } u\to v \end{array}$$

Here, $w_{uv}$ the weight of the $p$-cell whose dual edge is $uv$, and $c_{u\to v}$ is the coefficient of the input chain $\boldsymbol{c}$ for the *oriented* $p$-cell whose dual dart is $u\to v$. In particular, we have $c_{u\to v} = -c_{v\to u}$ for every dart $u\to v$.

Up to a sign change, this dual linear program describes the standard *minimum-cost circulation* problem. Intuitively, the dual variable $\varphi_{u\to v}$ represents an amount of *flow* traversing edge $uv$ from $u$ to $v$; without loss of generality, we can assume that either $\varphi_{u\to v} = 0$ or $\varphi_{v\to u} = 0$ for every edge $uv$. The equality constraint states that the total flow into any vertex equals the total flow out of that vertex; a vector $\boldsymbol{\varphi}$ that satisfies this constraint is called a *circulation*. Restated in topological language, a circulation $\varphi$ is a real 1-cycle in $G$, or equivalently, a real *1-cocycle* in the primal complex $\Sigma$. The weight of the corresponding $p$-cell in $\Sigma$ is interpreted as the *capacity* of the dual edge $uv$; a circulation $\boldsymbol{\varphi}$ that satisfies the capacity and non-negativity constraints is said to be *feasible*. In the objective function, each coefficient of the input chain $\boldsymbol{c}$ is interpreted as the *cost* of sending one unit of flow in either direction across the corresponding dual edge.

Several specialized algorithms are known for the minimum-cost circulation problem that are faster than general-purpose linear programming algorithms [**4**, **166**]. Sullivan [**176**] described an algorithm (independently proposed by Röck [**162**] and by Bland and Jensen [**14**]) that runs in $O(mn^2 \log n)$ time if the input chain $\boldsymbol{c}$ is unitary, and a second algorithm that runs in $O(mn)$ time if in addition every $p$-cell has weight 1. The fastest algorithm known (in terms of $m$ and $n$) for the general minimum-cost circulation problem, due to Orlin [**153**], runs in $O(m^2 \log n + mn \log^2 n)$ time. Each of these algorithms either returns or can be modified to return a solution $\boldsymbol{x}$ to the primal linear program (LP) along with the minimum-cost circulation $\boldsymbol{\varphi}$. Moreover, if the input $p$-chain $\boldsymbol{c}$ is integral, the $p$-chain $\boldsymbol{x}$ output by these algorithms is also integral, which implies that $\boldsymbol{x}$ is also the optimal $\mathbb{Z}$-homologous $p$-chain.

THEOREM 8.1 (Sullivan [**176**] and Orlin [**153**]). *Let $\Sigma$ be an orientable combinatorial $(p+1)$-manifold $\Sigma$ with $n$ $(p+1)$-cells and $m$ $p$-cells, and let $\boldsymbol{c}$ be an integer $p$-chain in $\Sigma$. A $p$-chain $\mathbb{Z}$-homologous to $\boldsymbol{c}$ with minimum weighted $L_1$-norm can be computed in $O(m^2 \log n + mn \log^2 n)$ time.*

**8.5. Back to surfaces.** Finally, we we describe a recent algorithm of Chambers *et al.* [**37**] to compute optimal $\mathbb{R}$-homologous circulations (real 1-cycles) in combinatorial surfaces. Their algorithm is a generalization of algorithms for computing maximum flows in planar graphs, which have been an object of study for more than 50 years; see Weihe [**188**] or Borradaile and Klein [**15**, **19**] for a detailed history. Unlike the algorithms described in the previous sections, this algorithm requires that the input chain is a 1-cycle; that is, its boundary *must* be empty.

Let $G$ be a cellularly embedded graph on some orientable surface $\Sigma$. Given a circulation $\boldsymbol{c}$ in $G$, our goal is to compute another circulation $\boldsymbol{\phi}$ with minimum weighted $L_1$-norm that is homologous with $\boldsymbol{c}$. Like Sullivan's algorithm [**176**], the algorithm of Chambers *et al.* considers the LP-dual formulation as a *minimum-cost* circulation problem in the dual graph $G^*$, as described by (LP$^*$). We emphasize that the primal and dual circulation problems are distinct. In the primal circulation problem, the edges of $G$ have non-negative weights but infinite capacities; whereas, in the dual problem, the edges of $G^*$ have both weights (costs) and non-negative capacities. Specifically, the cost of each dual dart in $G^*$ is the coefficient of the input chain $\boldsymbol{c}$ for the corresponding primal dart in $G$, and the capacity of each dual dart is the weight of the corresponding primal dart. Moreover, in the primal problem, the solution $\boldsymbol{\phi}$ is restricted to a particular homology class; whereas, the dual problem imposes no such restriction.

Homology between circulations in $G^*$ can be characterized in terms of cycles in the primal graph $G$ as follows. A *cocycle* $\lambda^*$ in $G^*$ is any subgraph dual to a directed cycle $\lambda$ in $G$. For any flow $\boldsymbol{\varphi}$, let $\boldsymbol{\varphi}(\lambda^*) = \sum_{u\to v \in \lambda^*} \varphi_{u\to v}$ denote the total flow through the edges of $\lambda^*$. Chambers *et al.* [**37**] observe that two circulations $\varphi$ and $\psi$ are homologous if and only if $\phi(\lambda) = \psi(\lambda)$ for every cocycle $\lambda$.

Recall that a circulation $\boldsymbol{\varphi}$ is *feasible* if $\varphi_{u\to v} \le w_{u\to v}$ for every dart $u\to v$ in $G^*$; call a homology class of circulations *feasible* if it contains a feasible circulation. Let $w(\lambda^*)$ denote sum of the capacities of the edges in any cocycle $\lambda^*$. Generalizing an observation of Venkatesan [**186**] for planar networks, Chambers *et al.* [**37**] prove that the homology class of a circulation $\boldsymbol{\varphi}$ is feasible if and only if $\boldsymbol{\varphi}(\lambda^*) \le w(\lambda^*)$ for every cocycle $\lambda^*$. Moreover, this condition can be checked by solving a single-source shortest path problem in the primal graph $G$, but with different, possibly negative, edge weights. Because the necessary edge weights may be negative, Dijkstra's algorithm cannot be used to solve this shortest-path problem; Chambers *et al.* describe a suitable shortest-path algorithm that runs in $O(g^2 n \log^2 n)$ time, generalizing an earlier algorithm for planar graphs by Klein *et al.* [**120**, **146**]. Their algorithm returns either a feasible circulation homologous to $\boldsymbol{\varphi}$ or a cocycle that is over-saturated by $\boldsymbol{\varphi}$.

To simplify notation, let $\boldsymbol{c}(\boldsymbol{\varphi}) := \sum_{u\to v} \varphi_{u\to v} c_{u\to v}$ denote the total cost of a circulation $\boldsymbol{\varphi}$ in $G^*$; this is the objective function in (LP$^*$). The input circulation $\boldsymbol{c}$ can be expressed as a weighted sum of directed cycles. It follows that the dual cost function is *homology-invariant*; that is, $\boldsymbol{c}(\boldsymbol{\varphi}) = \boldsymbol{c}(\boldsymbol{\psi})$ for any homologous circulations $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ in $G^*$. Moreover, for each cocycle $\lambda$ in $G$, the inequality $\boldsymbol{\varphi}(\lambda^*) \le w(\lambda^*)$ is a *linear* constraint on the homology class of $\boldsymbol{\varphi}$. Thus, the set of all feasible homology classes is a convex polyhedron in $H_1(\Sigma, \mathbb{R}) \cong \mathbb{R}^{2g}$, and finding the feasible homology class of minimum cost is a $(2g)$-dimensional linear programming problem. More careful analysis reveals that this linear program is just a linear projection of the $O(n)$-dimensional min-cost circulation linear program (LP$^*$) into the homology subspace $\mathbb{R}^{2g}$.

Unfortunately, this linear program appears to have $n^{O(g)}$ non-redundant constraints, so it cannot be solved directly, but it can be solved using implicit methods that apply the new shortest-path algorithm as a membership and separation oracle. If the edge capacities are integers less than $C$, the central-cut ellipsoid method [**96**, **97**] solves the linear program in $O(g^8 n \log^2 n \log^2 C)$ time. Alternatively, multidimensional parametric search [**2**, **44**, **45**, **152**], together with a parallel shortest-path algorithm of Cohen [**43**], gives us a combinatorial algorithm that runs in $g^{O(g)} n^{3/2}$ arithmetic operations, for arbitrary capacities. When $g$ is constant, both time bounds are faster by roughly a factor of $\sqrt{n}$ than the fastest minimum-cost circulation algorithms for general sparse graphs [**67**, **153**].

## References


[1] Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, and Kasturi R. Varadarajan, *Approximating shortest paths on a convex polytope in three dimensions*, J. ACM **44** (1997), no. 4, 567–584.

[2] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo, *An efficient multi-dimensional searching technique and its applications*, Tech. Rep. CS-1993-20, Dept. Comp. Sci., Duke Univ., August 1993.

[3] Ian Agol, Joel Hass, and William P. Thurston, *The computational complexity of knot genus and spanning area*, Trans. Amer. Math. Soc. **358** (2006), no. 9, 3821–3850.

[4] Ravindra K. Ahuja, Thomas L. Magnanti, and James Orlin, *Network flows: Theory, algorithms, and applications*, Prentice Hall, 1993.

[5] Lyudmil Aleksandrov, Anil Maheshwari, and Jörg-Rüdiger Sack, *Determining approximate shortest paths on weighted polyhedral surfaces*, J. ACM **52** (2005), no. 1, 25–53.

[6] James W. Alexander, II, *Normal forms for one- and two-sided surfaces*, Ann. Math. **16** (1914–1915), no. 1/4, 158–161.

[7] Alexandr D. Alexandrov, *Existence of a convex polyhedron and of a convex surface with a given metric*, Rec. Math. [Mat. Sbornik] N. S. **11(53)** (1942), no. 1–2, 15–65, In Russian, with English summary.

[8] Ernst Althaus and Christian Fink, *A polyhedral approach to surface reconstruction from planar contours*, Proc. 9th Int. Conf. Integer Prog. Combin. Optim., Lecture Notes Comput. Sci., vol. 2337, Springer-Verlag, 2006, pp. 258–272.

[9] Sanjeev Arora, Laszlo Babai, Jacques Stern, and Z Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, J. Comput. Syst. Sci. **54** (1997), no. 2, 317–331.

[10] Dominique Attali, André Lieutier, and David Salinas, *Efficient data structure for representing and simplifying simplicial complexes in high dimensions*, Proc. 27th Ann. Symp. Comput. Geom., 2011, pp. 501–509.

[11] Reuven Bar-Yehuda and Bernard Chazelle, *Triangulating disjoint Jordan chains*, Int. J. Comput. Geom. Appl. **4** (1994), no. 4, 475–481.

[12] Bruce G. Baumgart, *Winged edge polyhedron representation*, Tech. Report CS-TR-72-320, Dept. Comput. Sci., Stanford Univ., 1972.

[13] Norman Biggs, *Spanning trees of dual graphs*, J. Comb. Theory **11** (1971), 127–131.

[14] Robert G. Bland and David L. Jensen, *On the computational behavior of a polynomial-time network flow algorithm*, Math. Program. **54** (1992), no. 1, 1–39.

[15] Glencora Borradaile, *Exploiting planarity for network flow and connectivity problems*, Ph.D. thesis, Brown University, May 2008.

[16] Glencora Borradaile, Erik D. Demaine, and Siamak Tazari, *Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs*, Proc. 26th Int. Symp. Theoretical Aspects Comput. Sci., Leibniz Int. Proc. Informatics, vol. 3, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2009, pp. 171–182.

[17] Glencora Borradaile, Claire Kenyon-Mathieu, and Philip N. Klein, *A polynomial-time approximation scheme for Steiner tree in planar graphs*, Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms, 2007, pp. 1285–1294.

[18] ———, *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, 2007, pp. 275–286.

[19] Glencora Borradaile and Philip Klein, *An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph*, J. ACM **56** (2009), no. 2, 9:1–30.

[20] Glencora Borradaile, James R. Lee, and Anastasios Sidiropoulos, *Randomly removing g handles at once*, Proc. 25th Ann. Symp. Comput. Geom., 2009, pp. 371–376.

[21] Otakar Borůvka, *O jistém problému minimálním [About a certain minimal problem]*, Práce Moravské Přírodovědecké Společnosti v Brně III **3** (1926), 37–58, English translation in [**151**].

[22] Henry R. Brahana, *Systems of circuits on two-dimensional manifolds*, Ann. Math. **23** (1922), no. 2, 144–168.

[23] Martin R. Bridson, *Combings of semidirect products and 3-manifold groups*, Geom. Funct. Anal. **3** (1993), no. 3, 263–278.

[24] Chris Buehler, Steven J. Gortler, Michael F. Cohen, and Leonard McMillan, *Minimal surfaces for stereo*, Proc. 7th European Conf. Comput. Vision, vol. 3, 2002, pp. 885–899.

[25] James R. Bunch and John E. Hopcroft, *Triangular factorization and inversion by fast matrix multiplication*, Math. Comput. **28** (1974), no. 125, 231–236.

[26] Sergio Cabello and Erin W. Chambers, *Multiple source shortest paths in a genus g graph*, Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms, 2007, pp. 89–97.

[27] Sergio Cabello, Erin W. Chambers, and Jeff Erickson, *Multiple-source shortest paths in embedded graphs*, Preprint, February 2012, ArXiv:1202.0314. Full version of [**26**].

[28] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus, *Finding cycles with topological properties in embedded graphs*, SIAM J. Discrete Math. **25** (2011), 1600–1614.

[29] Sergio Cabello, Matt DeVos, Jeff Erickson, and Bojan Mohar, *Finding one tight cycle*, ACM Trans. Algorithms **6** (2010), no. 4, article 61.

[30] David Canino, Leila De Floriani, and Kenneth Weiss, *IA*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions*, Computers and Graphics **35** (2011), no. 3, 747–753, Proc. Shape Modeling International 2011.

[31] Erik Carlsson, Gunnar Carlsson, and Vin de Silva, *An algebraic topological method for feature identification*, Int. J. Comput. Geom. Appl. **16** (2006), no. 4, 291–314.

[32] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian, *On the local behavior of spaces of natural images*, Int. J. Comput. Vision **76** (2008), no. 1, 1–12.

[33] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J. Guibas, *Persistence barcodes for shapes*, Int. J. Shape Modeling **11** (2005), no. 2, 149–187.

[34] Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Francis Lazarus, and Kim Whittlesey, *Splitting (complicated) surfaces is hard*, Comput. Geom. Theory Appl. **41** (2008), no. 1–2, 94–110.

[35] Erin W. Chambers, Jeff Erickson, and Amir Nayyeri, *Homology flows, cohomology cuts*, Proc. 42nd Ann. ACM Symp. Theory Comput., 2009, pp. 273–282.

[36] ———, *Minimum cuts and shortest homologous cycles*, Proc. 25th Ann. Symp. Comput. Geom., 2009, pp. 377–385.

[37] ———, *Homology flows, cohomology cuts*, SIAM J. Comput. (to appear), Full version of [**35**].

[38] Bernard Chazelle, *A theorem on polygon cutting with applications*, Proc. 23rd Ann. IEEE Symp. Found. Comput. Sci., 1982, pp. 339–349.

[39] ———, *Triangulating a simple polygon in linear time*, Discrete Comput. Geom. **6** (1991), 485–524.

[40] Chao Chen and Daniel Freedman, *Hardness results for homology localization*, Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms, 2010, pp. 1594–1604.

[41] ———, *Measuring and computing natural generators for homology groups*, Comput. Geom. Theory Appl. **43** (2010), no. 2, 169–181.

[42] Jindong Chen and Yijie Han, *Shortest paths on a polyhedron, part I: Computing shortest paths*, Int. J. Comput. Geom. Appl. **6** (1996), no. 2, 127–144.

[43] Edith Cohen, *Efficient parallel shortest-paths in digraphs with a separator decomposition*, J. Algorithms **21** (1996), 331–357.

[44] Edith Cohen and Nimrod Megiddo, *Maximizing concave functions in fixed dimension*, Complexity in Numerical Optimization (Panos M. Pardalos, ed.), World Scientific, 1993, pp. 74–87.

[45] ———, *Strongly polynomial-time and NC algorithms for detecting cycles in periodic graphs*, J. Assoc. Comput. Mach. **40** (1993), no. 4, 791–830.

[46] Richard Cole and Alan Siegel, *River routing every which way, but loose*, Proc. 25th Ann. IEEE Symp. Found. Comput. Sci., 1984, pp. 65–73.

[47] Éric Colin de Verdière, *Shortest cut graph of a surface with prescribed vertex set*, Proc. 18th Ann. Europ. Symp. Algorithms, Lecture Notes Comput. Sci., vol. 6347, 2010, pp. 100–111.

[48] Éric Colin de Verdière and Jeff Erickson, *Tightening non-simple paths and cycles on surfaces*, SIAM J. Comput. **39** (2010), no. 8, 3784–3813.

[49] Éric Colin de Verdière and Francis Lazarus, *Optimal system of loops on an orientable surface*, Discrete Comput. Geom. **33** (2005), no. 3, 507–534.

[50] Anne Collins, Afra Zomorodian, Gunnar Carlsson, and Leonidas J. Guibas, *A barcode shape descriptor for curve point cloud data*, Comput. & Graphics **28** (2004), no. 6, 881–894.

[51] Don Coppersmith and Shmuel Winograd, *Matrix multiplication via arithmetic progressions*, J. Symb. Comput. **9** (1990), no. 3, 251–280.

[52] Thomas H. Cormen, Charles E. Leiseron, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms*, 3rd ed., MIT Press, 2009.

[53] Shervin Daneshpajouh, Mohammad Ali Abam, Lasse Deleuran, and Mohammad Ghodsi, *Computing strongly homotopic line simplification in the plane*, Proc. 27th Europ Workshop Comput. Geom., 2011, pp. 185–188.

[54] Marc de Berg, Marc van Kreveld, and Stefan Schirra, *Topologically correct subdivision simplification using the bandwidth criterion*, Cartography and Geographic Inform. Syst. **25** (1998), no. 4, 243–257.

[55] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, *Computational geometry: Algorithms and applications*, 3rd ed., Springer-Verlag, 2008.

[56] Leila de Floriani and Annie Hui, *Data structures for simplicial complexes: an analysis and a comparison*, Proc. 3rd Eurographics Symp. Geom. Processing, 2005, pp. 119–128.

[57] Vin de Silva and Robert Ghrist, *Homological sensor networks*, Notices Amer. Math. Soc. **54** (2007), no. 1, 10–17.

[58] Vin de Silva, Robert Ghrist, and Abubakr Muhammad, *Blind swarms for coverage in 2-D*, Proc. Robotics: Science and Systems, 2005, pp. 335–342.

[59] Max Dehn, *Transformation der Kurven auf zweiseitigen Flächen*, Math. Ann. **72** (1912), no. 3, 413–421.

[60] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar, *Approximation algorithms via contraction decomposition*, Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms, 2007, pp. 278–287.

[61] Tamal K. Dey and Sumanta Guha, *Transforming curves on surfaces*, J. Comput. System Sci. **58** (1999), 297–325.

[62] Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy, *Optimal homologous cycles, total unimodularity, and linear programming*, Proc. 42nd Ann. ACM Symp. Theory Comput., 2010, pp. 221–230.

[63] Tamal K. Dey, Kuiyu Li, and Jian Sun, *On computing handle and tunnel loops*, IEEE Proc. Int. Conf. Cyberworlds, 2007, pp. 357–366.

[64] Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner, *Computing geometry-aware handle and tunnel loops in 3D models*, ACM Trans. Graphics **27** (2008), no. 3, 1–9, Proc. SIGGRAPH 2008.

[65] Tamal K. Dey and Haijo Schipper, *A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection*, Discrete Comput. Geom. **14** (1995), no. 1, 93–110.

[66] Tamal K. Dey, Jian Sun, and Yusu Wang, *Approximating loops in a shortest homology basis from point data*, Proc. 26th Ann. Symp. Comput. Geom., 2010, pp. 166–175.

[67] Samuel I. Diatch and Daniel A. Spielman, *Faster lossy generalized flow via interior point algorithms*, Proc. 40th Ann. ACM Symp. Theory Comput., 2008, pp. 451–460.

[68] Jean-Guillaume Dumas, Frank Heckenbach, B. David Saunders, and Volkmar Welker, *Computing simplicial homology based on efficient Smith normal form algorithms*, Algebra, Geometry, and Software Systems (Michael Joswig and Nobuki Takayama, eds.), Springer-Verlag, 2003, pp. 177–206.

[69] Jean-Guillaume Dumas, B. David Saunders, and Gilles Villard, *On efficient sparse integer matrix Smith normal form computations*, J. Symb. Comput. **32** (2001), 71–99.

[70] Christian A. Duncan, Alon Efrat, Stephen G. Kobourov, and Carola Wenk, *Drawing with fat edges*, Int. J. Found. Comput. Sci. **17** (2006), no. 5, 1143–1164.

[71] Nathan M. Dunfield and Anil N. Hirani, *The least spanning area of a knot and the optimal bounding chain problem*, Proc. 27th Ann. Symp. Comput. Geom., 2011, pp. 135–144.

[72] Wayne Eberly, Mark Giesbrecht, and Gilled Villard, *On computing the determinant and Smith form of an integer matrix*, Proc. 41st IEEE Symp. Found. Comput. Sci., 2000, pp. 675–685.

[73] Herbert Edelsbrunner and John Harer, *Persistent homology—a survey*, Essays on Discrete and Computational Geometry: Twenty Years Later (Jacob E. Goodman, János Pach, and Richard Pollack, eds.), Contemporary Mathematics, no. 453, American Mathematical Society, 2008, pp. 257–282.

[74] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian, *Topological persistence and simplification*, Discrete Comput. Geom. **28** (2002), 511–533.

[75] Alon Efrat, Stephen G. Kobourov, Michael Stepp, and Carola Wenk, *Growing fat graphs*, Proc. 18th Ann. Symp. Comput. Geom., 2002, pp. 277–278.

[76] David Eppstein, *Dynamic generators of topologically embedded graphs*, Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms, 2003, pp. 599–608.

[77] Jeff Erickson, *Parametric shortest paths and maximum flows in planar graphs*, Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms, 2010, pp. 794–804.

[78] Jeff Erickson and Sariel Har-Peled, *Optimally cutting a surface into a disk*, Discrete Comput. Geom. **31** (2004), 37–59.

[79] Jeff Erickson and Amir Nayyeri, *Minimum cuts and shortest non-separating cycles via homology covers*, Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms, 2011, pp. 1166–1176.

[80] Jeff Erickson and Kim Whittlesey, *Greedy optimal homotopy and homology generators*, Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms, 2005, pp. 1038–1046.

[81] Regina Estowski and Joseph S. B. Mitchell, *Simplifying a polygonal subdivision while keeping it simple*, Proc. 17th Ann. Symp. Comput. Geom., 2001, pp. 40–49.

[82] Jittat Fakcharoenphol and Satish Rao, *Planar graphs, negative weight edges, shortest paths, and near linear time*, J. Comput. Syst. Sci. **72** (2006), no. 5, 868–889.

[83] Xin Gui Fang and Goerge Havas, *On the worst-case complexity of integer Gaussian elimination*, Proc. 1997 Int. Symp. Symb. Alg. Comput., 1997, pp. 28–31.

[84] Leila De Floriani, Annie Hui, Daniele Panozzo, and David Canino, *A dimension-independent data structure for simplicial complexes*, Proc. 19th International Meshing Roundtable (Suzanne Shontz, ed.), Springer-Verlag, 2010, pp. 403–420.

[85] Greg N. Frederickson, *Fast algorithms for shortest paths in planar graphs with applications*, SIAM J. Comput. **16** (1987), no. 6, 1004–1004.

[86] Shaodi Gao, Mark Jerrum, Michael Kaufmann, Kurt Mehlhorn, and Wolfgang Rülling, *On continuous homotopic one layer routing*, Proc. 4th Ann. Symp. Comput. Geom., 1988, pp. 392–402.

[87] Michael R. Garey and David S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman, New York, NY, 1979.

[88] Shayan Oveis Gharan and Amin Saberi, *The asymmetric traveling salesman problem on graphs with bounded genus*, Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms, 2011, pp. 967–975.

[89] Robert W. Ghrist and Abubakr Muhammad, *Coverage and hole-detection in sensor networks via homology*, Proc 4th. Int. Symp. Inform. Proc. Sensor Networks, 2005, pp. 254–260.

[90] Peter Giblin, *Graphs, surfaces and homology*, 3rd ed., Cambridge Univ. Press, 2010.

[91] Steven Gortler and Dylan Thurston, *Personal communication*, 2005.

[92] Steven J. Gortler and Danil Kirsanov, *A discrete global minimization algorithm for continuous variational problems*, Comput. Sci. Tech. Rep. TR-14-04, Harvard Univ., 2004.

[93] Leo Grady, *Computing exact discrete minimal surfaces: Extending and solving the shortest path problem in 3D with applicaton to segmentation*, Proc. IEEE CS Conf. Comput. Vis. Pattern Recog., vol. 1, 2006, pp. 67–78.

[94] ———, *Minimal surfaces extend shortest path segmentation methods to 3D*, IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010), no. 2, 321–334.

[95] Jonathan L. Gross and Thomas W. Tucker, *Topological graph theory*, Dover Publications, 2001.

[96] Martin Grötschel, László Lovász, and Alexander Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica **1** (1981), no. 2, 169–197.

[97] ———, *Geometric algorithms and combinatorial optimization*, 2nd ed., Algorithms and Combinatorics, no. 2, Springer-Verlag, 1993.

[98] Xianfeng Gu, Steven J. Gortler, and Hughes Hoppe, *Geometry images*, ACM Trans. Graphics **21** (2002), no. 3, 355–361.

[99] Xianfeng Gu and Shing-Tung Yau, *Global conformal surface parameterization*, Proc. Eurographics/ACM SIGGRAPH Symp. Geom. Process., 2003, pp. 127–137.
[100] Leonidas J. Guibas and Jorge Stolfi, *Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams*, ACM Trans. Graphics **4** (1985), no. 2, 75–123.
[101] Igor Guskov and Zoë Wood, *Topological noise removal*, Proc. Graphics Interface, 2001, pp. 19–26.
[102] Joel Hass and Peter Scott, *Intersections of curves on surfaces*, Israel J. Math. **51** (1985), 90–120.
[103] Allen Hatcher, *Algebraic topology*, Cambridge Univ. Press, 2002.
[104] Isidore Heller and Charles Brown Tomkins, *An extension of a theorem of Dantzig's*, Linear Inequalities and Related Systems (Harold W. Kuhn and Albert William Tucker, eds.), Annals of Mathematical Studies, no. 38, Princeton University Press, 1956, pp. 215–221.
[105] Monika R. Henzinger, Philip Klein, Satish Rao, and Sairam Subramanian, *Faster shortest-path algorithms for planar graphs*, J. Comput. Syst. Sci. **55** (1997), no. 1, 3–23.
[106] John Hershberger and Jack Snoeyink, *Computing minimum length paths of a given homotopy class*, Comput. Geom. Theory Appl. **4** (1994), 63–98.
[107] Alan J. Hoffman and Joseph B. Kruskal, *Integral boundary points of convex polyhedra*, Linear Inequalities and Related Systems (Harold W. Kuhn and Albert William Tucker, eds.), Annals of Mathematical Studies, no. 38, Princeton University Press, 1956, pp. 223–246.
[108] Joseph D. Horton, *A polynomial-time algorithm to find the shortest cycle basis of a graph*, SIAM J. Comput. **16** (1987), 358–366.
[109] Alon Itai, Christos H. Papadimitriou, and Jayme Luiz Szwarcfiter, *Hamilton paths in grid graphs*, SIAM J. Comput. **11** (1982), 676–686.
[110] Alon Itai and Yossi Shiloach, *Maximum flow in planar networks*, SIAM J. Comput. **8** (1979), 135–150.
[111] Giuseppe F. Italiano, Yahav Nussbaum, Piotr Sankowski, and Christian Wulff-Nilsen, *Improved algorithms for min cut and max flow in undirected planar graphs*, Proc. 43rd Ann. ACM Symp. Theory Comput., 2011, pp. 313–322.
[112] Donald B. Johnson and Shankar M. Venkatesan, *Partition of planar flow networks (preliminary version)*, Proc. 24th IEEE Symp. Found. Comput. Sci., 1983, pp. 259–264.
[113] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek, *Computational homology*, Applied Mathematical Sciences, vol. 157, Springer-Verlag, 2004.
[114] Ravindran Kannan and Achim Bachem, *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, SIAM J. Comput. **8** (1979), no. 4, 499–507.
[115] Richard M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations (E. Miller and J. W. Thatcher, eds.), Plenum Press, New York, 1972, pp. 85–103.
[116] Ken-ichi Kawarabayashi and Bojan Mohar, *Graph and map isomorphism and all polyhedral embeddings in linear time*, Proc. 40th Ann. ACM Symp. Theory Comput., 2008, pp. 471–480.
[117] Ken-ichi Kawarabayashi and Bruce Reed, *Computing crossing number in linear time*, Proc. 39th Ann ACM Symp. Theory Comput., 2007, pp. 382–390.
[118] Ron Kimmel and James A. Sethian, *Computing geodesic paths on manifolds*, Proc. Nat. Acad. Sci. USA **95** (1998), 8431–8435.
[119] Danil Kirsanov, *Minimal discrete curves and surfaces*, Ph.D. thesis, Div. Engin. Appl. Sci., Harvard Univ., September 2004.
[120] Philip Klein, Shay Mozes, and Oren Weimann, *Shortest paths in directed planar graphs with negative lengths: A linear-space $O(n \log^2 n)$-time algorithm*, ACM Trans. Algorithms **6** (2010), no. 2, article 30.
[121] Philip N. Klein, *Multiple-source shortest paths in planar graphs*, Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms, 2005, pp. 146–155.
[122] Jon Kleinberg and Éva Tardos, *Algorithm design*, Addison-Wesley, 2005.
[123] Bruce Kleiner and John Lott, *Notes on Perelman's papers*, Geom. Topol. **12** (2008), no. 5, 2587–2855.
[124] Joseph. B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc. **7** (1956), no. 1, 48–50.

[125] Sergei K. Lando and Alexander K. Zvonkin, *Graphs on surfaces and their applications*, Low-Dimensional Topology, no. II, Springer-Verlag, 2004.
[126] Francis Lazarus, Michel Pocchiola, Gert Vegter, and Anne Verroust, *Computing a canonical polygonal schema of an orientable triangulated surface*, Proc. 17th Ann. Symp. Comput. Geom., 2001, pp. 80–89.
[127] Der-Tsai Lee and Franco P. Preparata, *Euclidean shortest paths in the presence of rectilinear barriers*, Networks **14** (1984), 393–410.
[128] James R. Lee and Anastasios Sidiropoulos, *Genus and the geometry of the cut graph*, Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms, 2010, pp. 193–201.
[129] Charles E. Leiserson and F. Miller Maley, *Algorithms for routing and testing routability of planar VLSI layouts*, Proc. 17th Ann. ACM Symp. Theory Comput., 1985, pp. 69–78.
[130] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean E. Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk, *The digital Michelangelo project: 3D scanning of large statues*, Proc. 27th Ann. Conf. Comput. Graph. (SIGGRAPH), 2000, pp. 131–144.
[131] Sóstenes Lins, *Graph-encoded maps*, J. Comb. Theory Ser. B **32** (1982), 171–181.
[132] Richard J. Lipton and Robert E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Applied Math. **36** (1979), no. 2, 177–189.
[133] Yong-Jin Liu, Qian-Yi Zhou, and Shi-Min Hu, *Handling degenerate cases in exact geodesic computation on triangle meshes*, The Visual Computer **23** (2007), no. 9, 661–668.
[134] Roger C. Lyndon and Paul E. Schupp, *Combinatorial group theory*, Springer-Verlag, 1977.
[135] Martin Mareš, *Two linear time algorithms for MST on minor closed graph classes*, Archivum Mathematicum **40** (2004), no. 3, 315–320.
[136] Andrei Andreyevich Markov, *Impossibility of algorithms for recognizing some properties of associative systems*, Dokl. Akad. Nauk SSSR **77** (1951), 953–956, In Russian.
[137] ———, *The insolubility of the problem of homeomorphy*, Dokl. Akad. Nauk SSSR **121** (1958), 218–220.
[138] ———, *Unsolvability of certain problems in topology*, Dokl. Akad. Nauk SSSR **123** (1958), 978–980.
[139] Sergeĭ Vladomirivich Matveev, *Algorithmic topology and classification of 3-manifolds*, Algorithms and Computation in Mathematics, no. 9, Springer-Verlag, 2003.
[140] Jon McCammond and Daniel Wise, *Fans and ladders in combinatorial group theory*, Proc. London Math. Soc. **84** (2002), 499–644.
[141] S. Thomas McCormick, M. R. Rao, and Giovanni Rinaldi, *Easy and difficult objective functions for max cut*, Math. Program., Ser. B **94** (2003), 459–466.
[142] Gary L. Miller, *Finding small simple cycle separators for 2-connected planar graphs*, J. Comput. System Sci. **32** (1986), no. 3, 265–279.
[143] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou, *The discrete geodesic problem*, SIAM J. Comput. **16** (1987), 647–668.
[144] Bojan Mohar and Carsten Thomassen, *Graphs on surfaces*, Johns Hopkins Univ. Press, 2001.
[145] John Morgan and Gang Tian, *Ricci flow and the Poincaré conjecture*, Clay Mathematics Monographs, no. 3, American Mathematical Society, 2007.
[146] Shay Mozes and Christian Wulff-Nilsen, *Shortest paths in planar graphs with real lengths in $O(n \log^2 n/ \log\log n)$ time*, Proc. 18th Ann. Europ. Symp. Algorithms, Lecture Notes Comput. Sci., no. 6347, Springer-Verlag, 2010, pp. 206–217.
[147] Marian Mrozek, *Čech type approach to computing homology of maps*, Discrete Comput. Geom. **44** (2010), 546–576.
[148] David E. Muller and Franco P. Preparata, *Finding the intersection of two convex polyhedra*, Theoret. Comput. Sci. **7** (1978), 217–236.
[149] Ketan Mulmuley, Umesh Vazirani, and Vijay Vazirani, *Matching is as easy as matrix inversion*, Combinatorica **7** (1987), 105–113.
[150] James R. Munkres, *Topology*, 2nd ed., Prentice-Hall, 2000.
[151] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová, *Otakar Borůvka on minimum spanning tree problem: Translation of both the 1926 papers, comments, history*, Discrete Math. **233** (2001), no. 1–3, 3–36.
[152] Carolyn Haibt Norton, Serge A. Plotkin, and Éva Tardos, *Using separation algorithms in fixed dimension*, J. Algorithms **13** (1992), no. 1, 79–98.

[153] James B. Orlin, *A faster strongly polynomial minimum cost flow algorithm*, Oper. Res. **41** (1993), no. 2, 338–350.

[154] Steve Y. Oudot, Leonidas J. Guibas, Jie Gao, and Yue Wang, *Geodesic Delaunay triangulations in bounded planar domains*, ACM Trans. Algorithms **6** (2010), article 67.

[155] Henri Poincaré, *Second complément à l'Analysis Situs*, Proc. London Math. Soc. **32** (1900), 277–308, English translation in [**157**].

[156] ———, *Cinquième complement à l'analysis situs*, Rendiconti del Circulo Matematico di Palermo **18** (1904), 45—110, English translation in [**157**].

[157] ———, *Papers on topology:* Analysis Situs *and its five supplements*, History of Mathematics, vol. 37, American Mathematical Society, 2010, Translated from the French and with an introduction by John Stillwell.

[158] Konrad Polthier and Marchis Schmies, *Geodesic flow on polyhedral surfaces*, Data Visualization: Proc. Eurographics Worksh. Scientific Visualization, Springer Verlag, 1999, pp. 179–188.

[159] Jean-Philippe Préaux, *Congugacy problems in groups of orientable geometrizable 3-manifolds*, Topology **45** (2006), 171–208.

[160] John Reif, *Minimum s-t cut of a planar undirected network in $O(n \log^2 n)$ time*, SIAM J. Comput. **12** (1983), 71–81.

[161] R. Bruce Richter and Herbert Shank, *The cycle space of an embedded graph*, J. Graph Theory **8** (1984), 365–369.

[162] H. Röck, *Scaling techniques for minimal cost network flows*, Discrete Structures and Algorithms [Proc. 5th Workshop Graph-Theoretic Concepts Comput. Sci.] (U. Pape, ed.), Hanser, Münich, 1980, pp. 181–191.

[163] Thomas J. Schaefer, *The complexity of satisfiability problems*, Proc. 10th ACM Symp. Theory Comput., 1978, pp. 216–226.

[164] Yevgeny Schreiber, *An optimal-time algorithm for shortest paths on realistic polyhedra*, Discrete Comput. Geom. **43** (2010), no. 1, 21–53.

[165] Yevgeny Schreiber and Micha Sharir, *An optimal-time algorithm for shortest paths on a convex polytope in three dimensions*, Proc. 22nd Ann. Symp. Comput. Geom., 2006, pp. 30–39.

[166] Alexander Schrijver, *Combinatorial optimization: Polyhedra and efficiency*, Algorithms and Combinatorics, no. 24, Springer-Verlag, 2003.

[167] Raimund Seidel, *A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons*, Comput. Geom. Theory Appl. **1** (1991), 51–64.

[168] Herbert Seifert and William Threlfall, *Lehrbook der Topologie*, Teubner, Leipzig, 1934, Reprinted by AMS Chelsea, 2003. English translation in [**169**].

[169] ———, *A textbook of topology*, Pure and Applied Mathematics, vol. 89, Academic Press, New York, 1980, Edited by Joan S. Birman and Julian Eisner. Translated from [**168**] by Michael A. Goldman.

[170] James A. Sethian and Alexander Vladimirsky, *Fast methods for the Eikonal and related Hamilton–Jacobi equations on unstructured meshes*, Proc. Nat. Acad. Sci. USA **97** (2000), no. 11, 5699–5703.

[171] Anastasios Sidiropoulos, *Optimal stochastic planarization*, Proc. 51st IEEE Symp. Found. Comput. Sci., 2010.

[172] Martin Škoviera, *Spanning subgraphs of embedded graphs*, Czech. Math. J. **42** (1992), no. 2, 235–239.

[173] Henry John Stephen Smith, *On systems of linear indeterminate equations and congruences*, Phil. Trans. Royal Soc. London **151** (1861), 293–326.

[174] Boris Springborn, Peter Schröder, and Ulrich Pinkall, *Conformal equivalence of triangle meshes*, ACM Trans. Graphics **27** (2008), no. 3, article 77, Proc. SIGGRAPH 2008.

[175] John Stillwell, *Classical topology and combinatorial group theory*, 2nd ed., Graduate Texts in Mathematics, no. 72, Springer-Verlag, 1993.

[176] John Matthew Sullivan, *A crystalline approximation theorem for hypersurfaces*, Ph.D. thesis, Princeton Univ., October 1990.

[177] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe, *Fast exact and approximate geodesics on meshes*, ACM Trans. Graph. **24** (2005), no. 3, 553–560.

[178] Alireza Tahbaz-Salehi and Ali Jadbabaie, *Distributed coverage verification in sensor networks without location information*, IEEE Trans. Automatic Control **55** (2010), no. 8, 1837–1849.

[179] Robert Endre Tarjan, *Data structures and network algorithms*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 44, SIAM, 1983.

[180] Gabriel Taubin and Jarek Rossignak, *Geometric compression through topological surgery*, ACM Trans. Graphics **17** (1998), no. 2, 84–115.

[181] Carsten Thomassen, *Embeddings of graphs with no short noncontractible cycles*, J. Comb. Theory Ser. B **48** (1990), no. 2, 155–177.

[182] William P. Thurston, *The geometry and topology of 3-manifolds*, Princeton University lecture notes, 1980.

[183] Martin Tompa, *An optimal solution to a wire-routing problem*, J. Comput. Syst. Sci. **23** (1981), 127–150.

[184] Yiying Tong, Pierre Alliez, David Cohen-Steiner, and Mathieu Desbrun, *Designing quadrangulations with discrete harmonic forms*, Proc. Eurographics Symp. Geom. Proc., 2006, pp. 201–210.

[185] Gert Vegter and Chee-Keng Yap, *Computational complexity of combinatorial surfaces*, Proc. 6th Ann. Symp. Comput. Geom., 1990, pp. 102–111.

[186] Shankar M. Venkatesan, *Algorithms for network flows*, Ph.D. thesis, The Pennsylvania State University, 1983, Cited in [**112**].

[187] Karl Georg Christian von Staudt, *Geometrie der Lage*, Verlag von Bauer and Rapse (Julius Merz), Nürnberg, 1847.

[188] Karsten Weihe, *Maximum $(s,t)$-flows in planar networks in $O(|V|\log|V|)$-time*, J. Comput. Syst. Sci. **55** (1997), no. 3, 454–476.

[189] Kevin Weiler, *Edge-based data structures for solid modeling in curved-surface environments*, IEEE Comput. Graph. Appl. **5** (1985), no. 1, 21–40.

[190] Douglas H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory **IT-32** (1986), no. 1, 54–62.

[191] Zoë Wood, Hughes Hoppe, Mathieu Desbrun, and Peter Schröder, *Removing excess topology from isosurfaces*, ACM Trans. Graphics **23** (2004), no. 2, 190–208.

[192] Xiaotian Yin, Miao Jin, and Xianfeng Gu, *Computing shortest cycles using universal covering space*, Vis. Comput. **23** (2007), no. 12, 999–1004.

[193] Qian-Yi Zhou, Tao Ju, and Shi-Min Hu, *Topology repair of solid models using skeletons*, IEEE Trans. Vis. Comput. Graph. **13** (2007), no. 4, 675–685.

[194] Afra Zomorodian, *The tidy set: A minimal simplicial set for computing homology of clique complexes*, Proc. 26th Ann. Symp. Comput. Geom., 2010, pp. 257–266.

[195] Afra Zomorodian and Gunnar Carlsson, *Computing persistent homology*, Discrete Comput. Geom. **33** (2005), no. 2, 249–274.

Department of Computer Science, University of Illinois, Urbana-Champaign
*E-mail address*: `jeffe@cs.illinois.edu`
*URL*: `http://www.cs.illinois.edu/~jeffe`