

Topologically Trivial Closed Walks in Directed Surface Graphs*

Jeff Erickson and Yipu Wang

University of Illinois at Urbana-Champaign

March 21, 2019

Abstract

Let G be a directed graph with n vertices and m edges, embedded on a surface S , possibly with boundary, with first Betti number β . We consider the complexity of finding closed directed walks in G that are either contractible (trivial in homotopy) or bounding (trivial in integer homology) in S . Specifically, we describe algorithms to determine whether G contains a simple contractible cycle in $O(n + m)$ time, or a contractible closed walk in $O(n + m)$ time, or a bounding closed walk in $O(\beta(n + m))$ time. Our algorithms rely on subtle relationships between strong connectivity in G and in the dual graph G^* ; our contractible-closed-walk algorithm also relies on a seminal topological result of Hass and Scott. We also prove that detecting simple bounding cycles is NP-hard.

We also describe three polynomial-time algorithms to compute shortest contractible closed walks, depending on whether the fundamental group of the surface is free, abelian, or hyperbolic. A key step in our algorithm for hyperbolic surfaces is the construction of a context-free grammar with $O(g^2L^2)$ non-terminals that generates all contractible closed walks of length at most L , and only contractible closed walks, in a system of quads of genus $g \geq 2$. Finally, we show that computing shortest simple contractible cycles, shortest simple bounding cycles, and shortest bounding closed walks are all NP-hard.

*So they did what any savvy business would do.
They hired a consultant. They brought in a contractor.
I'm sorry, not a **contractor**—a **contractor**.
A man who made words smaller by combining them
or apostrophizing them.*

— Gary Gulman, *Conan*, July 13, 2016

*Research on this paper was partially supported by NSF grant CCF-1408763. An extended abstract of this paper will be presented at the 33rd International Symposium on Computational Geometry [33].

1 Introduction

A key step in several algorithms for surface graphs is finding a shortest closed walk and/or simple cycle in the input graph with some interesting topological property. There is a large body of work on finding short interesting walks and cycles in undirected surface graphs, starting with Thomassen’s seminal *3-path condition* [54, 63]. For example, efficient algorithms are known for computing shortest non-contractible and non-separating cycles [7, 12, 29, 30, 50], shortest contractible closed walks [10], simple cycles that are shortest in their own homotopy class [11], and shortest closed walks in a *given* homotopy [19] or homology class [31], and for detecting simple cycles that are either contractible, non-contractible, or non-separating [9]. On the other hand, several related problems are known to be NP-hard, including computing shortest splitting closed walks [13], computing shortest separating cycles [6], computing shortest closed walks in a given homology class [14], and deciding whether a surface graph contains a simple separating or splitting cycle [9].

Directed surface graphs are much less understood, in part because they do not share convenient properties of undirected graphs, such as Thomassen’s 3-path condition [54, 63], or the assumption that if shortest paths are unique, then two shortest paths cross at most once [28]. The first progress in this direction was a pair of algorithms by Cabello, Colin de Verdière, and Lazarus [8], which compute shortest non-contractible and non-separating cycles in directed surface graphs, with running times $O(n^2 \log n)$ and $O(g^{1/2} n^{3/2} \log n)$. Erickson and Nayyeri described an algorithm to compute the shortest non-separating cycles in $2^{O(g)} n \log n$ time [31]. Later Fox [37] described algorithms to compute shortest non-contractible cycles in $O(\beta^3 n \log n)$ time and shortest non-separating cycles in $O(\beta^2 n \log n)$ time on surfaces with first Betti number β . (For all these bounds, the input size n is the total number of vertices, edges, and faces of the input graph.)

This paper describes the first algorithms and hardness results for finding topologically *trivial* closed walks in directed surface graphs. Our results extend similar results of Cabello, Colin de Verdière, and Lazarus [6, 9] for undirected surface graphs; however, our algorithms require several new techniques, both in design and analysis. (On the other hand, our NP-hardness proofs are actually simpler than the corresponding proofs for undirected graphs!)

We present results for eight different problems, determined by three independent design choices. First, we consider two types of “trivial” closed walks: *contractible* walks, which can be continuously deformed to a point, and *bounding* walks, which are weighted sums of face boundaries. Simple bounding cycles are also called *separating* cycles. (See Section 2 for more detailed definitions.) Second, like Cabello *et al.* [6, 8, 9], we carefully distinguish between closed walks and simple cycles throughout the paper. Finally, we consider two different goals: deciding whether a given directed graph contains a trivial cycle or closed walk, and finding the shortest trivial cycle or closed walk in a given directed graph (possibly with weighted edges). Crucially, our algorithms do *not* assume that the faces of the input embedding are open disks. Our results are summarized in Table 1.

Structure	Surface	Any	Shortest
Simple contractible cycle		$O(n)$	NP-hard
Contractible closed walk	annulus	$O(n)$	$O(n^2 \log \log n)$
	torus	$O(n)$	$O(n^3 \log \log n)$
	with boundary	$O(n)$	$O(\beta^5 n^3)$
	other	$O(n)$	$O(\beta^6 n^9)$
Simple bounding cycle		NP-hard	NP-hard
Bounding closed walk		$O(\beta n)$	NP-hard

Table 1. Our results; β is the first Betti number of the underlying surface.

In Section 3, we describe linear-time algorithms to determine whether a directed surface graph contains a simple contractible cycle or a contractible closed walk, matching similar algorithms for undirected graphs by Cabello *et al.* [9]. Our algorithms are elementary: After removing some obviously useless edges, we report success if and only if some face of the embedding has a (simple) contractible boundary. However, the proofs of correctness require careful analysis of the dual graphs, and the correctness proof for contractible closed walks relies on a subtle topological lemma of Hass and Scott [42]. We emphasize that these problems are nontrivial for directed graphs *even if* every face of the input embedding is a disk; see Figure 1.1.

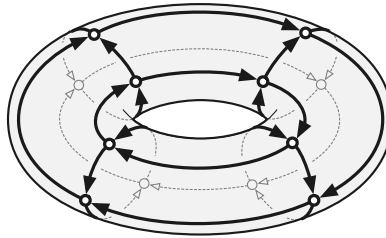


Figure 1.1. A cellularly embedded directed graph with no contractible or bounding closed walks.

In Section 4, we describe an algorithm to determine whether a directed surface graph contains a *bounding* closed walk in $O(\beta n)$ time.¹ Our algorithm exploits a careful analysis of the interplay between strong connectivity in the input graph G and its dual graph G^* . With some additional effort, our algorithm can return an explicit description of a bounding closed walk in $O(n^2)$ time if one exists; we prove that this quadratic upper bound is optimal. This problem can also be reduced to finding zero cycles in periodic (or “dynamic”) graphs [18, 44, 49]. A periodic graph is a graph whose edges are labeled with integer vectors; a zero cycle is a closed walk whose edge labels sum to the zero vector. However, all algorithms known for finding zero cycles rely on linear programming, and thus are much more complex and much less efficient than the specialized algorithm we present.

In Section 5, we prove that detecting simple bounding cycles, finding shortest simple contractible cycles, and finding shortest bounding closed walks are all NP-hard. Cabello [6] described an algorithm to compute the shortest simple contractible cycle in an *undirected* surface graph in $O(n^2 \log n)$ time; thus, our reduction for that problem makes essential use of the fact that the input graph is directed. Cabello and [6] and Cabello *et al.* [9] proved that the other two problems are NP-hard in undirected surface graphs. Our NP-hardness proofs closely follow theirs but are slightly simpler.

Finally, in Sections 6 and 7, we describe three polynomial-time algorithms to compute shortest contractible closed walks. Each of our algorithms is designed for a different class of surfaces, depending whether the surface’s fundamental group is abelian (the annulus and the torus), free (any surface with boundary), or hyperbolic (everything else). Our algorithm for the annulus and torus uses a standard covering-space construction, together with a recent algorithm of Mozes *et al.* [56]. For graphs on surfaces with boundary, we exploit the fact that the set of trivial words for any finitely-generated free group is a context-free language [57, 62]; this observation allows us to reduce to a small instance of the *CFG shortest path* problem [3, 64]. Our algorithm for hyperbolic surfaces also reduces to CFG shortest paths; the main technical hurdle is the construction of an appropriate context-free grammar. Specifically, for any integers $g \geq 2$ and $L \geq 1$, we construct a context-free grammar with $O(g^2 L^2)$ nonterminals, in Chomsky normal form, that generates all contractible closed walks of length L , and only contractible closed walks, in a canonical genus- g surface map called a *system of quads* [34, 51]. Our grammar construction

¹This problem is straightforward for undirected surface graphs, even if we forbid bounding walks with spurs. A connected undirected surface graph supports a spur-free bounding walk if and only if it has either (1) more than one face or (2) more edges than vertices.

exploits classical geometric properties of hyperbolic tilings: linear isoperimetry [21, 22] and exponential growth [53].

2 Background

Directed Graphs

Let G be an arbitrary directed graph, possibly with loops and parallel edges. Each edge of G is directed from one endpoint, called its **tail**, to the other endpoint, called its **head**. An edge is a **loop** if its head and tail coincide. At the risk of confusing the reader, we sometimes write $u \rightarrow v$ to denote an edge with tail u and head v .

A **walk** in G is an alternating sequence of vertices and edges $v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_\ell$, where $v_i \rightarrow v_{i+1}$ is an edge in G for each index i ; this walk is **closed** if $v_0 = v_\ell$. A **simple cycle** is a closed walk that visits each vertex at most once. The **concatenation** of two walks $\omega = v_0 \rightarrow \cdots \rightarrow v_{k-1} \rightarrow v_k$ and $\omega' = v_k \rightarrow v_{k+1} \rightarrow \cdots \rightarrow v_\ell$ is the walk $\omega \cdot \omega' := v_0 \rightarrow \cdots \rightarrow v_{k-1} \rightarrow v_k \rightarrow v_{k+1} \rightarrow \cdots \rightarrow v_\ell$.

An **edge cut** in a directed graph G is a nonempty subset X of edges such that $G \setminus X$ has two components, one containing the tails of edges in X , and other containing the heads of edges in X . A directed graph is **strongly connected** if it contains a directed walk from any vertex to any other vertex, or equivalently, if it contains no edge cuts.

An (integer) **circulation** in a directed graph G is a function $\phi : E(G) \rightarrow \mathbb{N}$ that satisfies a balance constraint $\sum_{u \rightarrow v} \phi(u \rightarrow v) = \sum_{v \rightarrow w} \phi(v \rightarrow w)$ for every vertex v . The **support** of a circulation ϕ is the subset of edges e such that $\phi(e) > 0$. An **Euler tour** of a circulation ϕ is a closed walk that traverses each edge e exactly $\phi(e)$ times; such a walk exists if and only if the support of ϕ is connected.

Surfaces, Embeddings, and Duality

A **surface** is a 2-manifold, possibly with boundary. A surface is **orientable** if it does not contain a Möbius band; we explicitly consider only orientable surfaces in this paper.² A **closed curve** on a surface S is (the image of) a continuous map $\gamma : S^1 \hookrightarrow S$; a closed curve is **simple** if this map is injective. The **boundary** ∂S of S consists of disjoint simple closed curves; the **interior** of S is the complement $S^\circ = S \setminus \partial S$. The **genus** of S is the maximum number of disjoint simple closed curves in S° whose deletion leaves the surface connected. Up to homeomorphism, there is exactly one orientable surface S with genus g and b boundary cycles, for any non-negative integers g and b . The **first Betti number** of S is either $2g$ if $b = 0$, or $2g + b - 1$ if $b > 0$.

An **embedding** of a graph G on a surface S is a continuous map that sends vertices of G to distinct points in S° , and sends edges to interior-disjoint simple paths in S° from their tails to their heads. In particular, if G contains two anti-parallel edges $u \rightarrow v$ and its reversal $v \rightarrow u$, those edges are embedded as interior-disjoint paths. The embedding of G maps every (simple) closed walk in G to a (simple) closed curve in S° ; we usually do not distinguish between a closed walk in G and its image in S .

We explicitly consider graphs with loops and parallel edges; however, without loss of generality, we assume that no loop edge is the boundary of a disk, and no two parallel edges are the boundary of a disk. (That is, no edge is contractible, and no two edges are homotopic.) With this assumption, Euler's formula implies that a graph with n vertices on a surface with first Betti number β has at most $O(n + \beta)$ edges.

A **face** of the embedding is a component of the complement of the image of the graph. An embedding is **cellular** if every face is homeomorphic to an open disk. *Unlike most previous papers, we explicitly*

²As in previous papers [19, 35, 51], all of the problems we consider can be solved for graphs on nonorientable surfaces, with similar running times, by lifting to the oriented double cover.

consider *non-cellular graph embeddings*; a single face may have disconnected boundary and/or positive genus. Each directed edge e in a surface graph lies on the boundary of two (possibly equal) faces, called the *left shore* and *right shore* of e . We sometimes write $f \uparrow f'$ to denote a directed edge whose left shore is f and whose right shore is f' . A **boundary face** of the embedding is any face that intersects the boundary of S .

A **dual walk** is an alternating sequence of faces and edges $f_0 \uparrow f_1 \uparrow \cdots \uparrow f_\ell$, where $f_i \uparrow f_{i+1}$ is an edge in G for each index i . A dual walk is **closed** if its initial face f_0 and final face f_ℓ coincide; a dual walk is **simple** if the faces f_i are distinct (except possibly $f_0 = f_\ell$). A simple closed dual walk is called a **cocycle**. Every minimal edge cut in a surface graph is the disjoint union of at most $g + 1$ cocycles; in particular, every minimal edge cut in a *planar* directed graph is a cocycle and vice versa.

Every surface embedding of a directed graph G defines a directed **dual graph** G^* , with one vertex f^* for each face f of G , and one edge e^* for each edge e of G . Specifically, for any edge e in G , the corresponding dual edge satisfies $\text{tail}(e^*) = \text{left}(e)^*$ and $\text{head}(e^*) = \text{right}(e)^*$. The dual graph G^* can be embedded on the same surface S ; however, unless the embedding of G is cellular, the embedding of G^* is *not* unique. As a consequence, we will treat G^* exclusively as an abstract directed graph. Every dual walk in G corresponds to a walk in G^* ; in particular, every cocycle in G corresponds to a cycle in G^* .

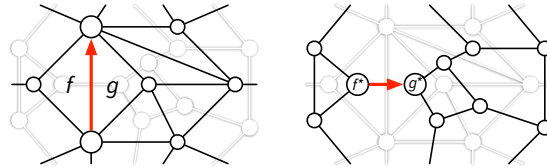


Figure 2.1. Duality in directed surface graphs

Contractible and Bounding

Let $\alpha: [0, 1] \rightarrow S$ and $\beta: [0, 1] \rightarrow S$ be two (not necessarily simple) paths in S with the same endpoints. A **homotopy** between α and β is a continuous function $h: [0, 1]^2 \rightarrow S$ such that $h(s, 0) = \alpha(0) = \beta(0)$ and $h(s, 1) = \alpha(1) = \beta(1)$ for all s , and $h(0, t) = \alpha(t)$ and $h(1, t) = \beta(t)$ for all t . Two paths are **homotopic**, or in the same **homotopy class**, if there is a homotopy between them.

A closed curve γ in S is **contractible** in S if it can be continuously deformed on S to a single point, or more formally, if there is a homotopy from γ to a constant function. A closed walk in a surface graph is contractible if its image under the embedding is a contractible closed curve. The concatenation of two contractible closed walks is contractible. A simple closed curve (or a simple cycle in G) is contractible in S if and only if it is the boundary of a disk in S [27].

Mirroring classical terminology for curves in the plane [1, 58], an **Alexander numbering** for a surface graph G is a function $\alpha: F(G) \rightarrow \mathbb{Z}$ that assigns an integer to each face of G , such that $\alpha(f) = 0$ for every boundary face f . The **boundary** $\partial\alpha$ of an Alexander numbering α is a circulation, defined by setting $\partial\alpha(e) = \alpha(\text{left}(e)) - \alpha(\text{right}(e))$. A closed walk is **bounding** if and only if it is an Euler tour of some boundary circulation. Equivalently, a closed walk (or its underlying circulation) is bounding if and only if its *integer* homology class is trivial. (We refer to reader to Hatcher [43], Giblin [39], or Edelsbrunner and Harer [25] for a more through introduction to homology.) On the sphere or the plane, every closed walk is bounding, and every circulation is a boundary circulation; however, these equivalences do not extend to other surfaces. (See Figure 1.1.) A simple cycle γ in G is bounding in S if and only if $S \setminus \gamma$ is disconnected; simple bounding cycles are usually called **separating**. The concatenation of two bounding closed walks is bounding.

For example, Figure 2.2 shows an Alexander numbering for a directed graph on a surface of genus 2, which has three vertices, six edges, and three faces. The boundary circulation of this Alexander numbering has value 2 on both loops and value 1 on the four edges that are not loops. The closed walk $a \rightarrow a \rightarrow a \rightarrow b \rightarrow c \rightarrow c \rightarrow c \rightarrow b \rightarrow a$ is an Euler tour of this boundary circulation, and thus is a bounding closed walk. (This walk is not contractible, and it is actually the shortest bounding walk in this graph.)

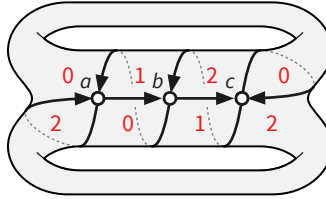


Figure 2.2. An Alexander numbering for a genus-2 surface graph.

Unlike many previous papers, we do not consider homology with coefficients in \mathbb{Z}_2 , because the problems we consider are easy in that setting. A directed surface graph contains a closed walk with trivial \mathbb{Z}_2 -homology if and only if it contains a directed cycle. Our NP-hardness proofs in Section 5 imply that it is NP-hard to find a simple cycle, the shortest cycle, or the shortest closed walk with trivial \mathbb{Z}_2 -homology (or with trivial homology over *any* ring).

3 Contractible Cycles and Walks

3.1 Simple Cycles

Lemma 3.1. *Let G be a directed graph embedded on an orientable surface, and let e be a directed edge that lies in a directed cycle of G . No bounding closed walk (and in particular, no simple contractible cycle) in G traverses e .*

Proof: Let λ be a directed cocycle in G , and let ω be a bounding closed walk. Let $\phi : E(G) \rightarrow \mathbb{Z}$ be the boundary circulation defined by setting $\phi(e)$ to the number of times that ω traverses e , and let $\alpha : F(G) \rightarrow \mathbb{N}$ be an Alexander numbering of ω (so that $\phi = \partial\alpha$). We immediately have

$$\sum_{e \in \lambda} \phi(e) = \sum_{e \in \lambda} (\alpha(\text{right}(e)) - \alpha(\text{left}(e))) = 0.$$

Because $\phi(e) \geq 0$ for every edge e in G , it follows that $\phi(e) = 0$ for every edge $e \in \lambda$. □

In light of this lemma, we can assume without loss of generality that G contains no directed cocycles; in particular, no edge of G has the same face on both sides.

The **boundary** of a face f , denoted ∂f , is the set of edges that have f on one side. The boundary of f is oriented **clockwise** if f is the right shore of any edge in ∂f , and **counterclockwise** if f is the left shore of any edge in ∂f . Each face in G with counterclockwise boundary appears as a source in the dual graph G^* ; each face of G with clockwise boundary appears as a sink in G^* . We call a face **coherent** if its boundary is oriented either clockwise or counterclockwise, and **incoherent** otherwise. See Figure 3.1.

Lemma 3.2. *Let G be a directed graph with no cocycles, embedded on an orientable surface S . If G contains a simple contractible directed cycle, then G has a face whose boundary is a simple contractible directed cycle.*

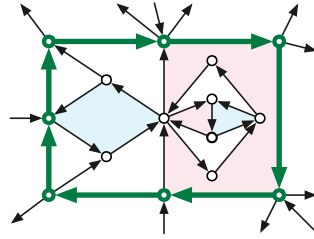


Figure 3.1. A simple clockwise contractible cycle (bold green) in a directed surface graph, enclosing four coherent faces: two counterclockwise (shaded blue) and two clockwise (shaded pink).

Proof: Let γ be a simple contractible cycle in G . This cycle is the boundary of a closed disk D [27]. Without loss of generality, assume γ is oriented clockwise around D ; that is, the right shore of every edge in γ is a face in D .

A dual walk that starts at a face inside D cannot visit the same face more than once, because G has no cocycles (G^* has no cycles), and cannot cross γ , because γ is oriented clockwise (all its dual edges point into D). Thus, D must contain at least one face f with clockwise boundary (a sink in G^*). The closure of f is a subset of the closed disk D , so it must be homeomorphic to a closed disk with zero or more open disks (“holes”) removed from its interior.

If ∂f is a simple directed cycle, we are done. Otherwise, ∂f is the union of two or more edge-disjoint simple directed cycles; consider the non-simple shaded face in Figure 3.1. Each of these cycles is contractible, because it lies in D . Any simple cycle in ∂f that is not the original cycle γ (for example, the boundary of any hole in the closure of f) encloses strictly fewer faces than γ . The lemma now follows immediately by induction on the number of enclosed faces. \square

Theorem 3.3. *Given an arbitrary directed graph G embedded on an arbitrary orientable surface, we can determine in $O(n)$ time whether G contains a simple contractible cycle.*

Proof: The algorithm proceeds as follows. First, we compute the strong components of the dual graph G^* in $O(n)$ time. An edge e of G lies in a cocycle if and only if both endpoints of the dual edge e^* lie in the same strong component of G^* ; we remove all such edges from G in $O(n)$ time. Finally, we examine each face of the remaining subgraph of G by brute force, in $O(n)$ time. If we discover a face whose interior is a disk and whose boundary is a simple directed cycle, we output TRUE; otherwise, we output FALSE. Correctness follows directly from Lemmas 3.1 and 3.2. \square

3.2 Non-simple Closed Walks

It is easy to construct directed surface graphs that contain contractible closed walks but no simple contractible cycles. For example, the one-vertex graph in Figure 3.2 has two Eulerian circuits, both of which are contractible, but the only simple cycles consist of single edges, none of which are contractible.

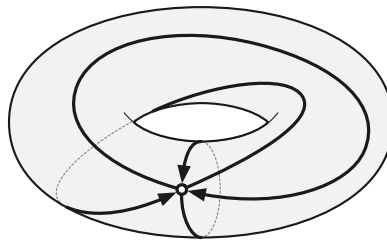


Figure 3.2. A directed surface graph with contractible closed walks but no simple contractible cycles.

We detect contractible closed walks using essentially the same algorithm described in the previous section: After removing all cocycle edges, we look for a face whose boundary is coherent and whose interior is an open disk. However, proving this algorithm correct requires more subtlety.

Fix a graph G embedded on an orientable surface S . Let $\{U_v \mid v \in V(G)\}$ be a collection of disjoint open disks in S , each containing one of the vertices of G ; we refer to each set U_v as a *vertex bubble*. Let $\{U_e \mid e \in E(G)\}$ be another collection of disjoint open disks in S , each containing the portion of an edge of G outside the vertex bubbles, such that $U_e \cap U_v \neq \emptyset$ if and only if v is one of the endpoints of e ; we refer to each disk U_e as an *edge bubble*. Finally, let $U = \bigcup_v U_v \cup \bigcup_e U_e$.

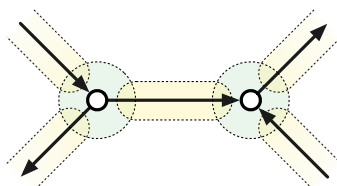


Figure 3.3. Vertex bubbles (green) and edge bubbles (yellow).

We say that a simple closed curve γ in U *follows* a closed walk ω in G if γ visits the vertex bubbles of G in exactly the same sequence as the walk ω visits the corresponding vertices of G , and therefore γ intersects each edge bubble $U_{u \rightarrow v}$ in a disjoint set of simple paths from U_u to U_v , one for each occurrence of the edge $u \rightarrow v$ in ω . A closed walk ω in G is *weakly simple* if U contains a simple closed curve that follows ω , or equivalently (because walks in asymmetric directed graphs cannot contain spurs [16]) if ω is homotopic in U to a simple closed curve.

Lemma 3.4. *Let G be a directed graph embedded on an orientable surface S . If there is a contractible closed walk in G , then there is a weakly simple contractible closed walk in G .*

Proof: Let ω be a contractible closed walk in G . Let $\tilde{\omega}$ be a generic closed curve homotopic to ω inside the graph neighborhood U (and therefore contractible in S) that follows ω and does not self-intersect inside any edge bubble. For each edge $u \rightarrow v$ in G , the intersection $\tilde{\omega} \cap U_{u \rightarrow v}$ consists of pairwise-disjoint simple paths, each directed from U_u to U_v .

A seminal result of Hass and Scott [42, Theorem 2.7] implies that every non-simple contractible closed curve in U contains either a simple contractible closed subpath, which we call a *monogon*, or a pair of simple interior-disjoint subpaths that bound a disk, which we call a *bigon*. We call a bigon *coherent* if one subpath is homotopic to the reverse of the other, and *incoherent* otherwise. See Figure 3.4.



Figure 3.4. Shortening or simplifying a contractible closed curve by smoothing at points of self-intersection. From left to right: A monogon, a coherent bigon, and an incoherent bigon.

First, suppose $\tilde{\omega}$ contains two homotopic subpaths α and β ; these subpaths need not be simple or interior-disjoint. Let $\tilde{\omega}'$ be the curve obtained from $\tilde{\omega}$ by *smoothing* the common endpoints of α and β , as shown on the right of Figure 3.4. The smoothed curve $\tilde{\omega}'$ is homotopic to $\tilde{\omega}$ via a homotopy that swaps the subpaths α and β [15, 23, 59]; it follows that $\tilde{\omega}'$ is contractible. Thus, by repeatedly smoothing pairs of vertices, we can reduce $\tilde{\omega}$ to a contractible curve $\tilde{\omega}''$ with no pair of homotopic subpaths, and therefore no incoherent bigons. (If $\tilde{\omega}$ has no pair of homotopic subpaths, then $\tilde{\omega}'' = \tilde{\omega}$.)

Hass and Scott’s theorem implies that either $\tilde{\omega}''$ is simple, or $\tilde{\omega}''$ contains a monogon, or $\tilde{\omega}''$ contains a coherent bigon. If $\tilde{\omega}''$ contains a monogon, that monogon is itself a simple contractible closed curve.

If $\tilde{\omega}''$ contains a coherent bigon composed of subpaths α and β , their concatenation $\alpha \cdot \beta$ is a simple contractible closed curve. In all three cases, we have discovered a simple contractible closed curve γ in the graph neighborhood U . Moreover, for every edge $u \rightarrow v$ of G , the intersection $\gamma \cap U_{u \rightarrow v}$ is a subset of $\tilde{\omega} \cap U_{u \rightarrow v}$, and therefore consists of disjoint simple paths directed from U_u to U_v . Thus, γ follows a closed walk in G , which is both weakly simple [16] and contractible. \square

Lemma 3.5. *Let G be a directed graph embedded on an orientable surface S . Every weakly simple **bounding** closed walk in G traverses each edge of G at most once.*

Proof: Let ω be a weakly simple bounding closed walk in G . Let γ be a simple closed curve in U that follows ω , as in the previous proof. Because γ is simple and bounding, the set $S \setminus \gamma$ has exactly two components. Color the component on the left of γ blue, and the component on the right of γ red.

Consider a single edge $u \rightarrow v$. Each component of $\gamma \cap U_{u \rightarrow v}$ is a directed path that splits $U_{u \rightarrow v}$ into a blue region on the left and a red region on the right. Thus, the components of $\gamma \cap U_{u \rightarrow v}$ must alternate direction, as shown in Figure 3.5, since otherwise some component of $U_{u \rightarrow v} \setminus \gamma$ would be both red and blue.

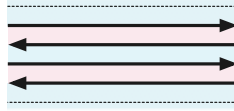


Figure 3.5. A simple contractible cycle in U must alternate directions inside any edge bubble.

By assumption, even if G contains an edge from v to u , that edge is embedded disjointly from $u \rightarrow v$. Thus, every path component of $\tilde{\omega} \cap U_{u \rightarrow v}$ is directed from U_u to U_v . It follows that $\tilde{\omega} \cap U_{u \rightarrow v}$ is either empty or a single directed path, which implies that ω traverses $u \rightarrow v$ at most once. \square

Corollary 3.6. *Let G be a directed graph with n vertices and m non-negatively weighted edges, embedded on an orientable surface, possibly with boundary. The shortest contractible closed walk in G , if such a walk exists, has hop-length at most m .*

Lemma 3.7. *Let G be a directed graph with no cocycles, embedded on an orientable surface. If G contains a contractible closed walk, then G has a face with coherent boundary whose interior is a disk.*

Proof: Let ω be a weakly-simple contractible closed walk in G (as guaranteed by Lemma 3.4). Smoothing ω at its vertices yields a simple contractible closed curve γ in U that follows ω ; this closed curve is the boundary of an open disk D [27]. Following terminology for face boundaries, we say that ω is oriented *clockwise* if D lies to the right of γ and *counterclockwise* otherwise. We say that ω *encloses* a face f if the interior region $f \setminus U$ lies inside the disk D .

Let A denote the area enclosed by ω , defined as the interior of the union of the closure of all faces enclosed by ω . Area A has genus zero and connected boundary ω (by Lemma 3.5), so each component of A is an open disk. If A is disconnected, then the boundary of each component of A is a contractible closed walk that encloses fewer faces than ω . Thus, we can assume without loss of generality that A is a single open disk. However, we *cannot* assume that ω itself is simple; the closure of A might have holes and/or positive genus. (See Figures 3.2 and 3.6.)

Without loss of generality, assume ω is oriented clockwise. A dual walk that starts at a face enclosed by ω cannot visit the same face more than once, because G has no cocycles (G^* is a dag), and cannot cross ω , because ω is oriented clockwise (all its dual edges point into A). Thus, ω must enclose at least one face f with clockwise boundary (a sink in the dual graph G^*).

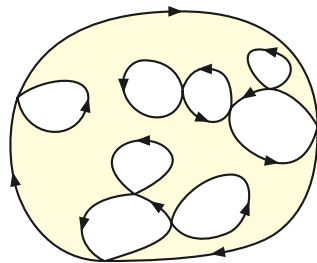


Figure 3.6. A non-simple walk that encloses an open disk.

The interior of f is a subset of the open disk A , so it must be homeomorphic to an open disk with zero or more holes. If f is an open disk (and in particular, if $f = A$), we are done. Otherwise, let W' be closed walk that exactly covers the boundary of any interior hole of f . Because W' lies entirely inside the open disk A , it must be contractible. Thus, W' is a weakly simple contractible closed walk that encloses strictly fewer faces than ω , and the lemma follows immediately by induction. \square

Theorem 3.8. *Given a directed graph G embedded on any surface, we can determine in $O(n)$ time whether there is a contractible closed walk in G .*

Proof: We use essentially the algorithm described by Theorem 3.3: Remove all cocycle edges from G by considering the strong components of the dual graph G^* , and then examine the faces of the remaining subgraph by brute force. If we discover a face with coherent boundary whose interior is a disk, we output TRUE; otherwise, we output FALSE. Correctness follows directly from Lemmas 3.1 and 3.7. \square

In the last stage of the proof, if the face f has an interior hole, then the graph G *must* be disconnected, because the interior of f is a subset of the open disk A . Thus, it may tempting to simplify the proof of Lemma 3.7 by assuming that the graph G is connected. Unfortunately, we cannot *simultaneously* assume that G is connected *and* that G contains no cocycles. Resolving this tension lies at the core of our algorithm for finding bounding closed walks, which we describe next.

4 Bounding Closed Walks

To simplify our presentation, we assume in this section that our input graphs are embedded on surfaces *without* boundary; this assumption is justified by the following observation. Let G be a directed graph embedded on a surface S with $b > 0$ boundary components. Let S^\bullet be the surface without boundary obtained from S by attaching a single disk with $b - 1$ holes to the boundary cycles of S . Thus, every boundary face of G on the original surface S is contained in a single face of G on S^\bullet . If S has first Betti number β , then S^\bullet has genus $O(\beta)$.

Lemma 4.1. *A closed walk ω in G is bounding in S if and only if it is bounding in S^\bullet .*

Proof: Any Alexander numbering for ω on S is also an Alexander numbering for ω on S^\bullet .

Let α be any Alexander numbering for ω on S^\bullet . For any integer k , the function $\alpha + k$ defined as $(\alpha + k)(f) = \alpha(f) + k$ for every face is also an Alexander numbering for ω on S^\bullet . Thus, there is an Alexander numbering α^\bullet of ω such that $\alpha^\bullet(f) = 0$ for the single face of G containing $S^\bullet \setminus S$. The function α^\bullet is an Alexander numbering for ω on S . \square

4.1 Detection

Our algorithm for detecting bounding closed walks relies on two elementary observations. First, we can assume without loss of generality that G is strongly connected, because any closed walk in the input graph G stays within a single strong component of G . On the other hand, Lemma 3.1 implies that we can assume without loss of generality that the input graph G has no cocycles; otherwise, we can remove all cocycles in $O(n)$ time by contracting each strong component of the dual graph G^* to a single dual vertex. Indeed, if the input graph satisfies *both* of these conditions, detecting bounding closed walks is trivial.

Lemma 4.2. *Let G be a strongly connected surface graph with at least one edge, whose dual graph G^* is acyclic. There is a bounding closed walk in G .*

Proof: For each face f of G , let $\alpha(f)$ be the rank of the dual vertex f^* in an arbitrary topological sort of the dual graph G^* . For each edge e in G , we have $\alpha(\text{left}(e)) > \alpha(\text{right}(e))$. Thus, the function $\partial\alpha: e \mapsto \alpha(\text{left}(e)) - \alpha(\text{right}(e))$ is a positive integer boundary circulation with connected support. Any Euler tour of this circulation is a bounding closed walk. \square

Kao and Shannon [45, 47, 48] observed that these two assumptions are actually identical for planar graphs; a planar directed graph G is strongly connected if and only if its dual graph G^* is acyclic. (This important observation is the basis of several efficient algorithms for planar directed graphs [2, 45–48, 61].) And indeed, finding bounding closed walks in planar graphs is trivial, because *every* closed walk in a planar graph is bounding.

However, this equivalence does not extend to directed graphs on more complex surfaces; a strongly connected directed surface graph can contain many directed cycles. Moreover, deleting all the directed cocycles from a strongly connected surface graph can disconnect the graph. More subtly, if H is a disconnected surface graph without cocycles, the induced embeddings of the individual components of H can contain cocycles. See Figure 4.1.

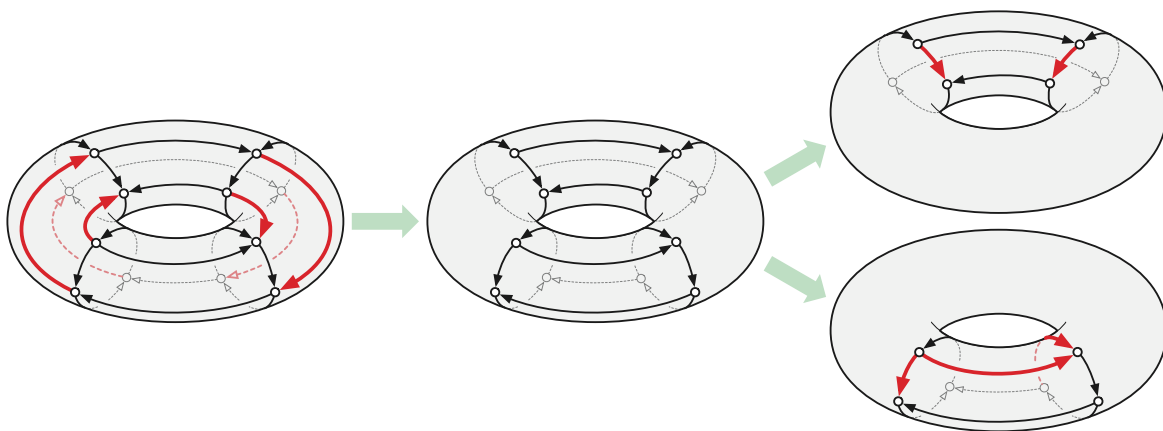


Figure 4.1. A strongly connected directed graph on the torus with cocycles (bold red). Deleting all cocycles disconnects the graph. The components of the disconnected graph have more cocycles. (Only one cocycle is emphasized in each component.)

Our algorithm repeatedly applies both of these simplifications, alternately removing cocycle edges and separating components, eventually reporting success if it ever finds a component that is both strongly connected and cocycle-free. Each iteration can be performed in linear time, and each iteration removes at least one edge from the graph; thus, conservatively, our algorithm runs in $O(n^2)$ time. But in fact, as we show next, our algorithm converges after only $O(g)$ iterations.

Lemma 4.3. *Let H be a directed surface graph without cocycles. Every weak component of H is strongly connected.*

Proof: If a weak component of H is not strongly connected, then H contains a non-empty directed edge cut. Every non-empty directed edge cut is the disjoint union of cocycles. \square

We call any face of G **simple** if it is homeomorphic to an open disk and **non-simple** otherwise. (The boundary of a simple face is *not* necessarily a simple cycle.)

Lemma 4.4. *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, and let G' be a (strong) component of H . Every simple face of G' is also a simple face of G .*

Proof: Let f be an arbitrary simple face of G' . There are two cases to consider.

First, suppose f contains no vertices of G in its interior. Then any cocycle in G that passes through a face inside f must contain at least one edge on the boundary of f . But by definition, every edge on the boundary of f is an edge in G' . It follows that no cocycle in G passes through a face in the interior of f , and thus no cocycle in G includes an edge in the interior of f . We conclude that f is actually a face of G .

On the other hand, suppose some vertex v of G lies inside the open disk f . Then v is disconnected from the boundary of f in H by removing cocycles in G . None of these cocycles contain an edge on the boundary of f , because all such edges survive in H , so G must contain a cocycle whose edges and faces are entirely inside f . This cocycle must be a directed edge cut in G , because every directed cocycle inside a disk is a directed edge cut. But no such edge cut exists, because G is strongly connected. \square

Corollary 4.5. *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, and let G' be a (strong) component of H . Every directed cocycle in G' visits at least one non-simple face of G' .*

We define the **footprint** $\llbracket G \rrbracket$ of a directed surface graph G as the union of the vertices, edges, and simple faces of G . For example, the footprint of the middle graph in Figure 4.1 is the disjoint union of two annuli. The embedding of G is cellular if and only if $\llbracket G \rrbracket$ is the entire surface. Let $\beta_0(G)$ denote the number of (weak) components of G , or equivalently, the number of components of its footprint $\llbracket G \rrbracket$. Let $\beta_1(G)$ denote the first Betti number of $\llbracket G \rrbracket$, which is the rank of the first homology group of $\llbracket G \rrbracket$. If G has at least one non-simple face, then

$$\beta_1(G) := \beta_0(G) - v(G) + e(G) - f_0(G),$$

where $v(G)$, $e(G)$, and $f_0(G)$ denote the number of vertices, edges, and simple faces of G , respectively. If G (and therefore $\llbracket G \rrbracket$) is disconnected, then $\beta_1(G)$ is the sum of the first Betti numbers of its components.

Lemma 4.6. *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, let G' be a (strong) component of H , and let H' be the subgraph of G' obtained by deleting all cocycles. If $H' \neq G'$, then $\beta_1(H') < \beta_1(G')$.*

Proof: Assume $H' \neq G'$, so G' contains at least one cocycle. Corollary 4.5 implies that every cocycle in G' passes through a non-simple face of G' . Thus, every cocycle in G' can be decomposed into edge-disjoint *dual arcs* of the form $f_0 \uparrow f_1 \uparrow \cdots \uparrow f_k$, where the initial face f_0 and final face f_k are non-simple, and the intermediate faces f_i are simple and distinct.

Let α_1 be any dual arc in G' . Because G' is strongly connected, α_1 is *not* a directed edge cut, and therefore $G' \setminus \alpha_1$ is (at least weakly) connected. It follows that $\beta_1(G' \setminus \alpha_1) = \beta_1(G') - 1$.

For any other dual arc α' in G' , the edges in $\alpha' \setminus \alpha_1$ define one or more dual arcs in $G' \setminus \alpha_1$. Thus, we can proceed inductively as follows. Define a sequence of nested subgraphs $G' = G_0 \supset G_1 \supset G_2 \supset \dots \supset G_\ell = H$, where each subgraph G_i is obtained by deleting a dual arc α_i from G_{i-1} , and the final subgraph $G_\ell = H$ contains no dual arcs (and therefore no cocycles). For each index $i > 0$, either α_i is an edge cut in G_{i-1} , which implies $\beta_1(G_i) = \beta_1(G_{i-1})$, or α_i is not an edge cut, which implies $\beta_1(G_i) = \beta_1(G_{i-1}) - 1$. The lemma now follows by induction. \square

Theorem 4.7. *Given a directed graph G embedded on any surface with genus g , we can determine in $O(gn)$ time whether there is a bounding closed walk in G .*

Proof: Assume G is strongly connected, since otherwise, we can consider each strong component of G separately. Let H be the graph obtained by deleting all cocycles of G . We can construct H in $O(n)$ time by computing the strong components of G^* in linear time, and then deleting any edge of G whose incident faces lie in the same strong component of G^* . Lemma 3.1 implies that any bounding closed walk in G is also a bounding closed walk in H .

If H has no edges, we can immediately report that G has no bounding closed walks. If H is weakly connected and has at least one edge, then Lemmas 4.2 and 4.3 imply that H contains a bounding closed walk. Otherwise, we recursively examine each (strong) component of H .

Each vertex of the original input graph G participates in only one subproblem at each level of recursion, so the total time spent at each level of the recursion tree is $O(n)$. Finally, Lemma 4.6 implies that the depth of the recursion tree is at most $O(g)$. \square

Call the subgraph obtained from G by alternately removing cocycles and isolating components the *snarl* of G ; the snarl is the largest subgraph of G in which every component is strongly connected and cocycle-free. We conjecture that the snarl of a surface graph (or at least one nontrivial component thereof) can actually be extracted in $O(n)$ time.

4.2 Explicit Construction

This algorithm described in Theorem 4.7 only reports whether G contains a closed bounding walk; it does not actually compute such a walk. With some additional straightforward bookkeeping, we can compute an *implicit* representation of a bounding closed walk as an Alexander numbering α of the faces of G , without increasing the running time of our algorithm. The complexity of this implicit representation is $O(n)$. If necessary, we can then obtain an *explicit* bounding closed walk, as an alternating sequence of vertices and edges, by computing an Euler tour of the circulation $\partial\alpha$, as described in the proof of Lemma 4.2. Because each Alexander number $\alpha(f)$ is an integer between 0 and $O(n)$, the total length of this Euler tour is $O(n^2)$, and we can compute in $O(n^2)$ time using standard Euler-tour algorithms.

Corollary 4.8. *Let G be a directed surface graph with n vertices. In $O(n^2)$ time, we can either compute an explicit description of a bounding closed walk in G or report correctly that no such walk exists.*

Corollary 4.9. *Let G be a directed surface graph with n vertices. The shortest bounding closed walk in G (if such a walk exists) has length $O(n^2)$.*

For graphs on the sphere or the torus, every bounding closed walk is actually contractible, so Lemmas 3.4 and 3.5 imply that the shortest bounding closed walk actually has length $O(n)$, and Theorem 3.8 implies that we can find an explicit bounding closed walk in $O(n)$ time. For graphs on even slightly more complicated surfaces, however, our quadratic upper bounds are actually tight.

Theorem 4.10. *For any fixed integer $g \geq 2$ and any positive integer n , there is a directed graph with $2n$ vertices and $3n + 1$ edges, embedded on an orientable surface of genus g , in which the shortest bounding closed walk has length $\Omega(n^2)$.*

Proof: It suffices to consider the case $g = 2$. Let G be a directed graph consisting of a directed cycle of length n (“the long cycle”), a chain of n cycles of length 2 (“the short cycles”), and a single cycle of length 1 (“the self-loop”), embedded on a surface of genus 2 as shown in Figure 4.2 (for $n = 5$). This embedding has $n + 1$ faces, of which two are annuli and the rest are disks. The long cycle and the self-loop both lie on the common boundary of both annular faces.

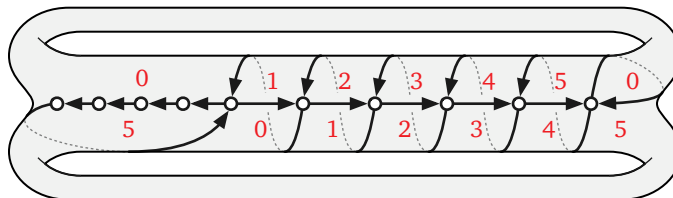


Figure 4.2. A directed surface graph with only long bounding closed walks, with a minimal valid Alexander numbering.

The dual graph G^* consists of a directed path $f_0 \rightarrow f_1 \rightarrow \dots \rightarrow f_{n-1} \rightarrow f_n$, where f_0 and f_n are the annular faces, plus n additional edges directly from f_0 to f_n . Because G^* is a dag, G has no cocycles, and so Lemma 4.2 implies that there is a bounding closed walk in G . Specifically, the function $\alpha(f_i) = i$ for all i is the Alexander numbering of a bounding circulation; again, see Figure 4.2.

The subgraph of edges traversed by any closed walk is strongly connected. Straightforward exhaustive case analysis implies that the only strongly connected subgraph of G that contains both the long cycle and the self-loop is the entire graph G . Thus, every strongly connected proper subgraph H of G omits either the entire long cycle or the self-loop. In either case, faces f_0 and f_n of G belong to a single face of H . Thus, the dual graph H^* is strongly connected, and Lemma 3.1 implies that H cannot support a bounding closed walk. We conclude that every bounding closed walk in G traverses every edge of G at least once.

Now let ω be any bounding closed walk in G that traverses every edge at least once, and let $\alpha: F(G) \rightarrow \mathbb{Z}$ be the Alexander numbering of ω . For each edge e , we must have $\alpha(\text{left}(e)) > \alpha(\text{right}(e))$, so α must be consistent with the only topological order of G^* . It follows that $\alpha(f_n) - \alpha(f_0) \geq n$. Thus, ω must traverse both the long cycle and the self-loop n times, and every other edge of G at least once. We conclude that ω must have length at least $n^2 + 3n$. (This is exactly the length of any bounding closed walk induced by the Alexander numbering $\alpha(f_i) = i$, so our analysis is tight.) \square

One frustrating source of complexity in our construction is the long cycle of degree-2 vertices. A different construction with no degree-2 vertices yields a lower bound of $\Omega(gn)$ on any surface of genus g ; we leave this construction as an amusing exercise for the reader, and a proof or disproof of its optimality as an open problem.

5 NP-hardness

5.1 Simple Bounding Cycles and Shortest Bounding Walks

Our algorithms for detecting contractible closed walks and simple contractible cycles are essentially identical: Remove all cocycles, and then check the boundary of every face. Unfortunately, this similarity does not extend to bounding walks and bounding cycles. In this section, we prove that determining whether a directed surface graph contains a simple bounding cycle is NP-hard. The same reduction also

implies that finding the shortest bounding closed walk in a directed surface graph is NP-hard, even if all edges have weight 1.

Our proof uses a variant of the “heaven and hell” NP-hardness reduction of Cabello *et al.* [9] for the corresponding problem in undirected graphs, which is based in turn on a reduction of Chambers *et al.* [13] for the problem of finding the shortest closed walk in an undirected surface graph that is separating but non-contractible. In fact, our reduction is simpler than either of these earlier arguments, because our underlying graphs are directed.

We reduce from the Hamiltonian cycle problem in directed planar graphs, where every vertex has either in-degree 1 and out-degree 2, or in-degree 2 and out-degree 1. This problem was proved NP-hard by Plesník [60], using a direct reduction from 3SAT.

Let G be a planar directed graph. Fix a planar embedding of G on the sphere (“Earth”). Ultimately, we will construct an embedding of *the same graph* G on a more complex surface S , such that G has a Hamiltonian cycle if and only if G has a closed walk of length at most n that is bounding in S .

As a preliminary step, we partially color the faces of G as follows. Call an edge of G *forced* if it is the only edge leaving a vertex or the only edge entering a vertex (or both). Color the left shore of every forced edge blue, and color the right shore of every forced edge red. A face may be colored both red and blue, or not colored at all; see Figure 5.1.

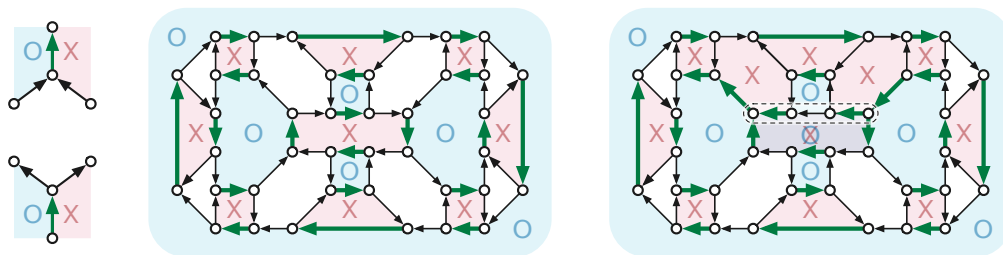


Figure 5.1. Coloring the faces of cubic directed planar graphs; bold green edges are forced. The graphs differ only in the direction of three edges at the top of the central face. The second graph is not Hamiltonian, because its central face is colored both red and blue.

Lemma 5.1. *If G has a Hamiltonian cycle, then no face of G is colored both red and blue.*

Proof: Suppose G contains a Hamiltonian cycle γ . By definition, γ visits each vertex of G exactly once, so γ is a simple closed curve that traverses every forced edge in G . Thus, each blue face is on the left side of γ , and each red face is on the right side of γ . The lemma now follows immediately from the Jordan curve theorem. \square

Assuming no face of G is colored both red and blue, we construct the surface S in two stages. First we delete a small disk from the interior of each blue face and attach the resulting punctures to another punctured sphere, thereby merging all blue faces into a single blue face, which we call *heaven*. Similarly, we delete a small disk from the interior of each red face and attach the resulting holes to a single punctured sphere, merging all the red faces into a single red face, which we call *hell*. Euler’s formula implies that the resulting surface S has genus $O(n)$.

Lemma 5.2. *The following statements are equivalent: (a) G contains a Hamiltonian cycle. (b) G contains a simple cycle that is bounding in S . (c) There is a closed walk in G that is bounding in S and has length at most n .*

Proof: We prove the implication (a) \Rightarrow (b) first. Suppose γ is a Hamiltonian cycle in G . Viewed as a cycle in the original planar embedding, γ separates all blue face from all red faces. Thus, the embedding of γ on S separates heaven from hell; it follows that γ is a bounding cycle in S .

The implication (b) \Rightarrow (c) is trivial.

Finally, we prove the implication (c) \Rightarrow (a). Suppose ω is a closed walk in G that is bounding in S and has length at most n . Fix an Alexander numbering α of the faces of G on S that is consistent with ω . Because ω visits at least one vertex of G , it traverses at least one forced edge. It follows that $\alpha(\text{heaven}) \neq \alpha(\text{hell})$, which implies that ω traverses every forced edge, and thus visits every vertex. We conclude that ω is a Hamiltonian cycle. \square

Theorem 5.3. *Deciding whether a directed surface graph contains a simple bounding cycle is NP-hard.*

Theorem 5.4. *Computing the shortest bounding closed walk in a directed surface graph is NP-hard.*

Both of these NP-hardness results hold even if we insist on cellularly embedded graphs. Without loss of generality, suppose the original input graph G is connected, since otherwise it cannot contain a Hamiltonian cycle. It follows that heaven and hell are the only non-disk faces in G 's embedding on S . Thus, we can extend G to a cellularly embedded graph H by adding a single vertex a in the interior of heaven, with directed edges $v \rightarrow a$ from one vertex v on the boundary of each blue face, and a single vertex z in the interior of hell, with edges $v \rightarrow z$ from one vertex v on the boundary of each red face. Because the two new vertices are sinks, they cannot appear in any (bounding) closed walk in H .

5.2 Shortest Trivial Cycles

Now we prove that finding the shortest simple contractible cycle in a directed surface graph is NP-hard, by reduction from the classical maximum independent set problem. Our reduction is nearly identical to a reduction of Cabello [6], who proved that finding the shortest simple bounding cycle in an undirected surface graph is NP-hard. In the same paper, Cabello described an algorithm to compute the shortest contractible cycle in an undirected surface graph in $O(n^2 \log n)$ time; thus, our NP-hardness proof must make essential use of the fact that the input graph is directed.

Given an undirected graph G with n vertices and m edges and a positive integer k , we first construct a directed planar graph H as follows. Arbitrarily identify the n vertices of G with the integers 0 through $n - 1$. For each integer $1 \leq i \leq n$ the graph H contains a corresponding vertex i , along with two directed paths P_i and Q_i from vertex i to vertex $i + 1 \pmod n$. Path P_i has length $\deg(i) + 1$, and path Q_i has length $\deg(i) + 2$. Label the vertices of P_i as $i \rightarrow [ij_1] \rightarrow [ij_2] \rightarrow \dots \rightarrow [ij_{\deg(i)}] \rightarrow (i + 1) \pmod n$, where $j_1, j_2, \dots, j_{\deg(i)}$ are the neighbors of vertex i in G (in no particular order). Thus, for each edge ij of G , there are two corresponding vertices in H , namely $[ij]$ in P_i and $[ji]$ in P_j . In total, H has $2n + 4m$ vertices and $3n + 4m$ edges.

We embed H on the sphere so that one face is bounded by the paths P_0, P_1, \dots, P_{n-1} , another face is bounded by the paths Q_0, Q_1, \dots, Q_{n-1} , and each of the remaining faces is bounded by P_i and Q_i for some index i . See Figure 5.2.

A simple cycle γ in H is *compliant* if, for every edge ij in G , γ does not use both of the vertices $[ij]$ and $[ji]$ in H .

Lemma 5.5. *G contains an independent set of size k if and only if H contains a simple compliant cycle of length at most $2m + n - k$.*

Proof: First, suppose G has an independent set I of size k . For each integer $0 \leq i \leq n - 1$, let $R_i = P_i$ if $i \in I$, and define $R_i = Q_i$ otherwise. Let γ be the simple cycle formed by concatenating the paths

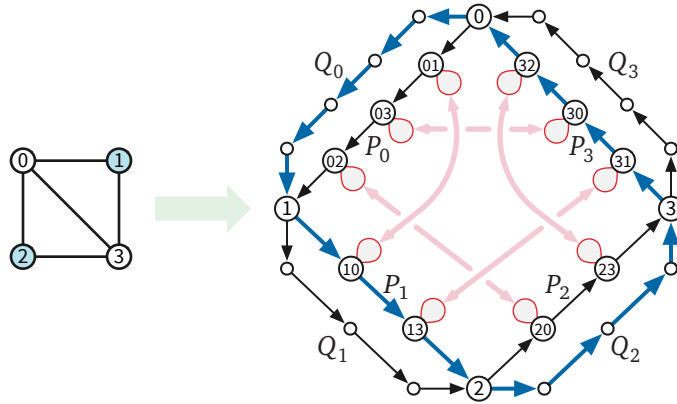


Figure 5.2. Reduction from maximum independent set to shortest contractible cycle in a directed surface graph.

R_0, R_1, \dots, R_{n-1} . This cycle is compliant, because there are no edges in G between the vertices in I . Straightforward calculation implies that the length of γ is $2m + n - k$.

On the other hand, suppose H contains a compliant cycle γ of length $2m + n - k$. Every cycle in H is the concatenation of paths R_0, R_1, \dots, R_{n-1} , where for each index i , either $R_i = P_i$ or $R_i = Q_i$. Because γ is compliant, γ cannot contain both P_i and P_j , for any edge ij of G . Thus, the set of indices $I = \{i \mid R_i = P_i\}$ is an independent set in G . Straightforward calculation implies $|I| = k$. \square

We now convert H into a directed graph H' embedded on a more complex surface S as follows. Let f be the face of H bounded by paths P_0, P_1, \dots, P_{n-1} . For each edge ij of the original graph G , let α_{ij} and α_{ji} denote two small simple closed curves, respectively passing through vertices $[ij]$ and $[ji]$ of H , and otherwise in the interior of f . To define both the new graph H' and the surface S , we remove the interiors of all curves α_{ij} and then identify each curve α_{ij} with its twin α_{ji} , so that vertices $[ij]$ and $[ji]$ are identified and the resulting surface S is orientable. The resulting surface S has genus m .

Lemma 5.6. *For any integer ℓ , there is a simple compliant cycle of length ℓ in H if and only if there is a simple cycle of length ℓ in H' that is contractible in S .*

Proof: First, let γ be any simple cycle in H . The edges in H' are in one-to-one correspondence with edges in H , and sequence of edges corresponding to γ defines a closed walk γ' in H' with the same length at γ . Because the construction of H' and S only involves surgery on the interior of the single face f , the corresponding closed walk γ' in H is contractible in S . Finally, if γ is compliant, then γ' is simple.

Now let γ' be a simple contractible cycle in H' . For each edge ij of the original graph G , the curve $\alpha_{ij} = \alpha_{ji}$ intersects H' at exactly one vertex $[ij] = [ji]$ and therefore crosses γ' at most once. Because γ' is contractible, γ' crosses every closed curve in S an even number of times. It follows that γ' does not cross any curve α_{ij} . Thus, the corresponding sequence of edges in H defines a simple cycle γ . Finally, because γ' visits each vertex $[ij] = [ji]$ at most once, γ is compliant. \square

Theorem 5.7. *Finding the shortest simple contractible cycle in a directed surface graph is NP-hard.*

The statement and proof of Lemma 5.6 still hold without further modification if we replace every instance of the word “contractible” with the word “bounding” or “separating”. Indeed, every bounding cycle in H' is actually contractible.

Theorem 5.8. *Finding the shortest simple bounding cycle in a directed surface graph is NP-hard.*

6 Shortest Contractible Closed Walks: The Easy Cases

Now we describe polynomial-time algorithms to find a shortest contractible closed walk in a directed surface graph with weighted edges. For this problem, it will prove convenient to treat input graphs as undirected, but with asymmetric edge weights, as proposed by Cabello *et al.* [10]. That is, each undirected edge uv in the input graph is composed of two *darts* $u \rightarrow v$ and $v \rightarrow u$ with independent weights; a walk is alternating series of vertices and *darts*; and the length of a walk is the sum of the weights of its darts, counted with appropriate multiplicity. We also assume without loss of generality that the given embedding of G is *cellular*, meaning every face is homeomorphic to an open disk. Both of these assumptions can be enforced by adding at most $O(n + \beta)$ directed edges with (symbolically) infinite weight, where β is the first Betti number of the underlying surface. Finally, to avoid issues with negative cycles, we assume all edge weights are non-negative.

All orientable 2-manifolds can be partitioned into four classes, depending on the structure of their fundamental group.

- The sphere and the disk have *trivial* fundamental groups.
- The annulus and the torus have *abelian* fundamental groups.
- Every surface with boundary has a *free* fundamental group.
- Every surface without boundary and with genus at least 2 has a *hyperbolic* fundamental group.

All closed curves on the sphere or the disk are contractible, so our problem is trivial for those surfaces. We develop a separate polynomial-time algorithm for each of the other three classes.

Our algorithm for the annulus and the torus, described in Section 6.1, constructs a finite relevant portion \bar{G} of the universal cover of the input graph, and then finds the shortest directed cycle in \bar{G} using a recent algorithm of Mozes *et al.* [56]. Our algorithm for surfaces with boundary, described in Section 6.2, exploits an observation by Smikun [62] and Muller and Schupp [57] that the set of trivial words for any finitely-generated free group is a context-free language. After carefully labeling the input graph, we invoke an algorithm of Barrett *et al.* [3] that finds the shortest walk in a given graph whose structure is consistent with a given context-free grammar. We defer discussion of the hyperbolic case to Section 7.

6.1 Annulus and Torus (Abelian)

The universal cover \widehat{G} of a surface graph G is an infinite planar graph, constructed by tiling the plane with an infinite number of copies of a disk called the *fundamental domain*, obtained by cutting the underlying surface along a system of loops [26, 50] or a system of arcs [32]. A closed walk in G is contractible if and only if it is the projection of a closed walk in \widehat{G} . Thus, the shortest contractible walk in G is the projection of the shortest directed cycle in \widehat{G} .

Corollary 3.6 implies that we can find the shortest contractible closed walk in G by searching a relevant region \bar{G} of \widehat{G} , assembled from a finite copies of the fundamental domain, and just large enough to contain at least one lift of every closed walk of length $m = O(n)$ in G . The assembled graph \bar{G} is planar, so we can find the shortest directed cycle in $O(\bar{n} \log \log \bar{n})$ time, where \bar{n} is the number of vertices in \bar{G} , using the recent algorithm of Mozes *et al.* [56].

This approach immediately gives us efficient algorithms when the fundamental group of the underlying surface is *abelian*. There are only two such surfaces:

Theorem 6.1. *Given a directed graph G with n vertices and non-negatively weighted edges embedded on the annulus, we can find the shortest contractible closed walk in G in $O(n^2 \log \log n)$ time.*

Proof: We can compute an appropriate fundamental domain in $O(n)$ time by slicing the annulus along any path in G between the two boundaries of the annulus. \overline{G} consists of at most $O(n)$ copies of the fundamental domain and thus has complexity $O(n^2)$. \square

Theorem 6.2. *Given a directed graph G with n vertices and non-negatively weighted edges embedded on the torus, we can find the shortest contractible closed walk in G in $O(n^3 \log \log n)$ time.*

Proof: We can compute an appropriate fundamental domain in $O(n)$ time by slicing the surface along any system of loops [26, 50]. \overline{G} consists of at most $O(n^2)$ copies of the fundamental domain and thus has complexity $O(n^3)$. \square

In principle, we can use the same covering-space approach to find shortest contractible closed walks in graphs on *arbitrary* surfaces. Unfortunately, for every nontrivial surface other than the annulus and the torus, covering all paths of length m requires exponentially many copies of the fundamental domain.

6.2 Surfaces with Boundary (Free)

The fundamental group of a surface with genus g and $b > 0$ boundary cycles is a *free* group with $\beta = 2g + b - 1$ generators. Smikun [62] and Muller and Schupp [57] observed that the set of trivial words for any finitely-generated free group is a context-free language.³

This observation allows us to reduce finding shortest contractible walks on surfaces with boundary to the *CFG shortest path* problem, introduced by Yannakakis [64]. Given a directed graph G with weighted edges, a labeling $\ell: E(G) \rightarrow \Sigma \cup \{\varepsilon\}$ for some finite set Σ , and a context-free grammar \mathcal{G} over the alphabet Σ , the CFG shortest path problem asks for the shortest walk (if any) in G whose label is in the language generated by \mathcal{G} . An algorithm of Barrett *et al.* [3] solves the CFG shortest path problem in $O(NPn^3)$ time, where N and P are the numbers of nonterminals and productions in \mathcal{G} , respectively, and n is the number of vertices in G , when the grammar \mathcal{G} is in Chomsky normal form and all edge weights in G are non-negative. (Their algorithm was extended to graphs with negative edges but no negative cycles by Bradford and Thomas [4].)

Theorem 6.3. *Given a directed graph G with m non-negatively weighted edges, embedded on a surface S with boundary with first Betti number β , we can find the shortest contractible closed walk in G in $O(\beta^5 m^3)$ time.*

Proof: Our algorithm begins by constructing a *system of dual arcs* for G —a collection of boundary-to-boundary paths $\alpha_1, \alpha_2, \dots, \alpha_\beta$ in G^* that cut the underlying surface S into a disk. Such a system can be constructed in $O(gn)$ time [32], using a natural variant of Eppstein’s tree-cotree construction of systems of loops [26]. If necessary, we subdivide the edges of G (introducing parallel edges into G^*) so that these dual arcs are simple and edge-disjoint, as described by Cabello and Mohar [12] for systems of loops. After subdivision, G has at most βm edges and therefore (because G is symmetric) at most βm vertices. We arbitrarily direct each arc α_i . Finally, we label each directed edge of G with a generator a_i if it crosses the corresponding dual arc α_i from left to right, with an inverse generator \bar{a}_i if it crosses α_i from right to left, and with the empty string otherwise.

Any closed walk in G is naturally labeled with a string obtained by concatenating the labels of its edges in order. A closed walk is contractible if and only if its label can be *reduced* to the empty string by

³More generally, both Smikun and Muller and Schupp proved that the set of trivial words for a finitely-presented group is a context-free language if and only if the group has a free subgroup with finite index.

repeatedly removing substrings of the form $a\bar{a}$ or $\bar{a}a$, where a is a generator and \bar{a} is its inverse. The set of all such reducible strings is generated by the following Chomsky normal form context-free grammar \mathcal{G} :

$$\begin{aligned} I &\rightarrow II \mid A_1\bar{A}_1 \mid \bar{A}_1A_1 \mid A_2\bar{A}_2 \mid \bar{A}_2A_2 \mid \cdots \mid A_\beta\bar{A}_\beta \mid \bar{A}_\beta A_\beta \\ A_i &\rightarrow a_i \mid A_iI \mid IA_i && \text{for all } i \\ \bar{A}_i &\rightarrow \bar{a}_i \mid \bar{A}_iI \mid I\bar{A}_i && \text{for all } i \end{aligned}$$

Specifically, the starting nonterminal I generates reducible strings; each non-terminal A_i generates strings that reduce to the corresponding generator a_i ; each barred non-terminal \bar{A}_i generates strings that reduce to the corresponding inverse generator \bar{a}_i .

We are now left with instance of the CFG shortest path problem with $N = 2\beta + 1$ non-terminals and $P = 8\beta + 1$ productions in a graph with $n = O(\beta m)$ vertices, which can be solved in $O(NPn^3) = O(\beta^5 m^3)$ time by the algorithm of Barrett *et al.* [3]. \square

7 Shortest Contractible Closed Walks: The Hyperbolic Case

Finally, we describe our algorithm to compute shortest contractible closed walks on surfaces with no boundary and with genus at least 2. All such surfaces have *hyperbolic* fundamental groups in the sense of Gromov [41]; as observed by Dehn over a century ago [20], the most natural geometry for the universal cover of these surfaces is the hyperbolic plane. (See Figure 7.2 below.)

Neither of the algorithms described in Section 6 apply to graphs on hyperbolic surfaces. As we already mentioned, adapting the covering-space algorithm from Section 6.1 would require exponentially many copies of the fundamental domain. (See Lemma 7.1 below.) On the other hand, the set of trivial words for a hyperbolic group is *not* a context-free language [57], so we cannot apply our algorithm from Section 6.2 directly. Nevertheless, we solve our problem by reduction to the CFG-shortest path algorithm of Barrett *et al.* [3], by constructing a grammar that generates all *sufficiently short* contractible closed walks in a certain canonical surface map.

Fix an undirected graph G embedded on an orientable surface S with genus $g \geq 2$ and no boundary. As in the previous section, we assume without loss of generality that G is symmetric, the embedding of G is cellular, and all edge weights are non-negative.

7.1 Edge Labeling

The first step in our algorithm is to label the edges of G so that the labels along any closed walk encode its homotopy type. In principle, we can derive such a labeling by reducing G to a system of loops [12, 24, 50]; however, this labeling leads to a less efficient algorithm.⁴ Instead, we reduce G to a different canonical surface decomposition called a *system of quads*, as proposed by Lazarus and Rivaud [51] and Erickson and Whittlesey [34].

Let (T, L, C) be an arbitrary *tree-cotree decomposition* of G , which partitions the edges of G into a spanning tree T , a spanning tree C^* of the dual graph G^* , and exactly $2g$ leftover edges L [26]. Contracting every edge in T and deleting every edge in C reduces G to a *system of loops*, which has one vertex a , one face f , and $2g$ loops L . To construct the system of quads Q , we introduce a new vertex z in the interior of f , add edges between z and every corner of f , and then deleting the edges in L .

Next we label each directed edge e in G with a directed walk $\langle e \rangle$ in Q as follows. Every walk $\langle e \rangle$ starts and ends at a and thus has even length. If $e \in T$, then $\langle e \rangle$ is the empty walk. Otherwise, after contracting T , edge e connects two corners of f ; we define $\langle e \rangle$ as the walk of length 2 in Q from the back

⁴See footnote 5.

corner of e , to z , and then to the front corner of e . (If $e \in L$, there are two possibilities for $\langle e \rangle$; choose one arbitrarily.)

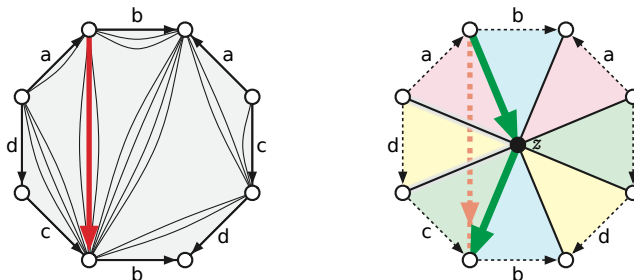


Figure 7.1. Left: A fundamental polygon for a surface of genus 2, obtained by contracting a spanning tree and cutting along a system of loops. Right: Mapping a directed non-tree edge to a path of length 2 in the corresponding system of quads.

Finally, for any closed walk ω in G , let $\langle \omega \rangle$ denote the closed walk in Q obtained by concatenating of the labels of edges in ω . The closed walks ω and $\langle \omega \rangle$ are freely homotopic in S ; in particular, ω is contractible if and only if $\langle \omega \rangle$ is contractible. Moreover, the number of edges in $\langle \omega \rangle$ is at most twice the number of edges in ω .

The universal cover \widehat{Q} of a system of quads Q is isomorphic to a regular tiling of the hyperbolic plane by quadrilaterals meeting at vertices of degree $4g$; see Figure 7.2. The covering map from \widehat{Q} to Q is a graph homomorphism (mapping vertices to vertices and edges to edges), which at the risk of confusing the reader, we also denote $\langle \cdot \rangle: \widehat{Q} \rightarrow Q$. A closed walk in Q is contractible if and only if it is the projection of a closed walk in \widehat{Q} . Thus, a closed walk ω in G is contractible if and only if there is a closed walk $\widehat{\omega}$ in \widehat{Q} such that $\langle \widehat{\omega} \rangle = \langle \omega \rangle$.

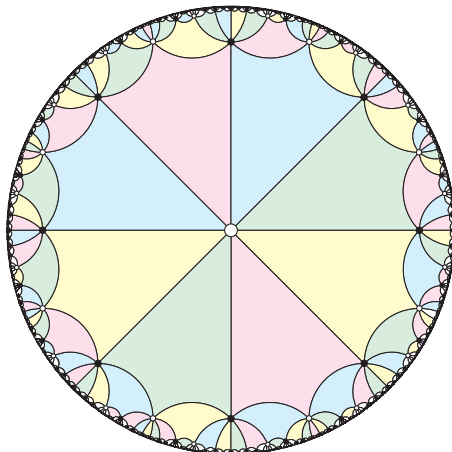


Figure 7.2. The universal cover of a genus-2 system of quads. (Compare with Figure 7.1.)

7.2 Hyperbolic Geometry

Our algorithm exploits two well-known geometric properties of regular hyperbolic tilings: the area of any ball in a hyperbolic tiling grows exponentially with its radius [40, 53], and the area enclosed by any simple cycle grows at most linearly with the cycle's length [21, 22]. In particular, we rely on the following immediate corollary of these two properties: The distance from any point in the interior of a simple cycle in a regular hyperbolic tiling is at most *logarithmic* in the length of that cycle; see Figure 7.3.

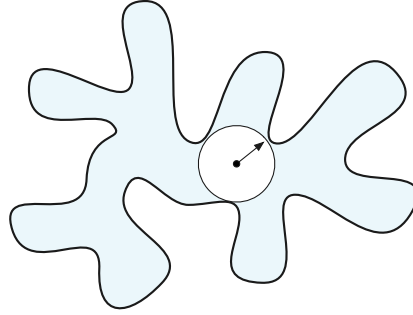


Figure 7.3. Simple closed curves in hyperbolic tilings have logarithmic in-radius.

Classical results already imply this *qualitative* behavior for *all* regular hyperbolic tilings; indeed, Chepoi *et al.* [17] have derived tight *asymptotic* bounds. However, we require *precise* bounds for growth, isoperimetry, and in-radius in the specific tiling \widehat{Q} , because the running time of our eventual algorithm depends *exponentially* on the constants in those bounds.

Fix an arbitrary vertex \widehat{a} of \widehat{Q} called the *basepoint*. For any positive integer r , let $\mathbf{n}(r)$ denote the number of vertices in \widehat{Q} at distance exactly r from \widehat{a} , and let $N(r)$ denote the number of vertices at distance at most r from \widehat{a} .

Lemma 7.1. $N(r) \sim \beta \lambda^r$ for some constants $1 < \beta < 2$ and $4g - 3 < \lambda < 4g - 2$. In particular, $N(r) \geq \lambda^r$ for all $r \geq 0$.

Proof: Every node at distance r from the basepoint has either zero or one neighbors at distance $r - 1$ from the basepoint. Thus, we can write $n(r) = n_1(r) + n_2(r)$, where each $n_i(r)$ is the number of nodes at distance r that have i neighbors at distance $r - 1$. We immediately have $n_1(1) = 4g$ and $n_2(1) = 0$. Floyd and Plotnick [36] prove that these functions satisfy the following recurrence:

$$\begin{bmatrix} n_1(r) \\ n_2(r) \end{bmatrix} = \begin{bmatrix} 4g - 3 & 4g - 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} n_1(r - 1) \\ n_2(r - 1) \end{bmatrix} = \begin{bmatrix} 4g - 3 & 4g - 4 \\ 1 & 1 \end{bmatrix}^{r-1} \begin{bmatrix} 4g \\ 0 \end{bmatrix}$$

(Moran [55] derives similar recurrences for a slightly different growth function.) Standard techniques now imply that $n(r) = \alpha(\lambda^r - \lambda^{-r})$ for some constant α , where λ and λ^{-1} are the roots of the polynomial $z^2 - (4g - 2)z + 1$; specifically, we have

$$\alpha = \sqrt{g/(g - 1)}, \quad \lambda = 2g - 1 + 2\sqrt{g(g - 1)}, \quad \lambda^{-1} = 2g - 1 - 2\sqrt{g(g - 1)}.$$

Straightforward calculation implies $4g - 3 < \lambda < 4g - 2$. It follows that $N(r) \sim \beta \lambda^r$, where $1 < \beta = \alpha/(1 - \lambda^{-1}) < 2$.

Moreover, a straightforward but tedious induction argument implies that $n_1(r) \geq \lambda^r$ and $n_2(r + 1) \geq \lambda^r$ for all $r \geq 1$. We conclude that $N(r) \geq n_1(r) \geq \lambda^r$ for all $r \geq 1$; the remaining case $r = 0$ is trivial. \square

Lemma 7.2. Any simple cycle of length L in \widehat{Q} has less than $3L/2$ faces of \widehat{Q} in its interior.

Proof: Our proof follows the “spur and bracket” analysis of Erickson and Whittlesey [34, Section 4.2], which is based in turn on results of Gersten and Short [38].

Let γ be a nontrivial simple cycle in \widehat{Q} , without loss of generality oriented counterclockwise around its interior faces (if any). The *turn* of any vertex of γ is the number of interior tiles incident to v that locally lie to the left of γ . A *(left) bracket* is a subpath of γ whose first and last vertices have turn 1 and whose intermediate vertices have turn 2. A discrete curvature argument, ultimately based on Euler’s

formula, implies that any γ has at least four brackets [38, Corollary 5.1] [34, Lemma 4.1]. In particular, γ has at least four distinct vertices with turn 1.

Let $L(\gamma)$ denote the number of edges in γ , let $A(\gamma)$ denote the number of interior tiles enclosed by γ , and let $\Phi(\gamma) = 2t_1 + t_2$, where t_i denotes the number of vertices of γ with turn i . Trivially, $\Phi(\gamma) \leq 2L(\gamma)$.

We can reduce the length of γ by *sliding* a bracket, as shown in Figure 7.4. Intuitively, sliding a bracket of length ℓ replaces a subpath of length $\ell + 2$ around three sides of a $1 \times \ell$ rectangle of tiles with a path of length ℓ around the fourth side of that rectangle. This replacement removes the turn-1 vertices at the ends of the bracket, changes the turns of the vertices in the interior bracket from 2 to $4g - 2$, and decreases the turns of the vertices adjacent to the bracket by 1.



Figure 7.4. Removing a bracket; vertex labels are turns (modulo $4g$).

Sliding a bracket creates a spur (an edge followed by its reversal) if and only if that bracket is adjacent to a vertex with turn 1, and thus another bracket with length 1. The only simple cycle that contains two consecutive length-1 brackets is the boundary of a single tile. Thus, either γ is a single tile boundary, or removing the *shortest* bracket yields a shorter simple cycle γ' . In the latter case, we immediately have

$$L(\gamma') = L(\gamma) - 2, \quad A(\gamma') = A(\gamma) - \ell, \quad \text{and} \quad \Phi(\gamma') \leq \Phi(\gamma) - \ell - 1,$$

where ℓ is the length of the bracket. It follows by induction that γ can be reduced to a single tile boundary by a sequence of $L(\gamma)/2 - 2$ bracket slides, and

$$A(\gamma) = 1 + \sum_i \ell_i \quad \text{and} \quad \Phi(\gamma) \geq 4 + \sum_i (\ell_i + 1)$$

where ℓ_i is the length of the i th bracket. We conclude that $1 + A(\gamma) + L(\gamma)/2 \leq \Phi(\gamma) \leq 2L(\gamma)$, and therefore $A(\gamma) \leq 3L(\gamma)/2 - 1$. \square

Lemma 7.3. *Any simple cycle of length L in \widehat{Q} has less than $2L/g$ vertices of \widehat{Q} in its interior.*

Proof: The dual of \widehat{Q} is another regular tiling \widehat{Q}^* of the hyperbolic plane by $4g$ -gons, meeting 4 at each vertex. A classical lemma of Dehn [21, 22] (see also Lyndon and Schupp [52]) implies that any closed walk γ^* in the dual tiling \widehat{Q}^* contains either a spur or at least $4g - 2$ consecutive edges of some tile. It follows by induction that γ^* encloses at most $L(\gamma^*)/(2g - 1)$ tiles in \widehat{Q}^* .

Now let γ be a simple cycle in the original tiling \widehat{Q} . Offsetting γ slightly into its interior yields a closed curve γ^* that visits every face in the interior of γ at most twice; we can regard γ^* as a closed walk in the dual tiling \widehat{Q}^* . The previous lemma implies $L(\gamma^*) < 3L(\gamma)$. Each vertex of \widehat{Q} in the interior of γ is dual to a unique face of \widehat{Q}^* enclosed by γ^* . We conclude that γ encloses at most $3L(\gamma)/(2g - 1) < 2L/g$ interior vertices. \square

The main result of this section now follows from Lemma 7.3 and the exact lower bound in Lemma 7.1.

Lemma 7.4. *Let γ be a simple cycle in \widehat{Q} with length at most L . Every tiling vertex in the interior of γ has distance at most $\rho = \lceil \log_\lambda(2L/g) \rceil$ from at least one vertex of γ .*

7.3 Triangulation

Now let ω be a closed walk in \widehat{Q} with a distinguished vertex \widehat{a} called its **basepoint**. Following Muller and Schupp [57], we define a **k -triangulation** of ω to be a continuous map \mathcal{T} from a *reference* triangulation T of a simple polygon P to \widehat{Q} with the following properties:

- \mathcal{T} maps vertices of P to vertices of ω .
- \mathcal{T} maps a distinguished edge of P , called the **root edge**, to the basepoint \widehat{a} .
- \mathcal{T} maps the boundary of P continuously onto the walk ω ; in particular, every edge of P except the root edge is mapped to a single edge of ω .
- Finally, \mathcal{T} maps each diagonal edge in T to a walk of length at most k in \widehat{Q} .

This map is not necessarily an embedding, even locally. The diagonal walks may intersect each other or ω , even if ω is a simple cycle; moreover, some diagonal walks may have length zero if ω is not simple.

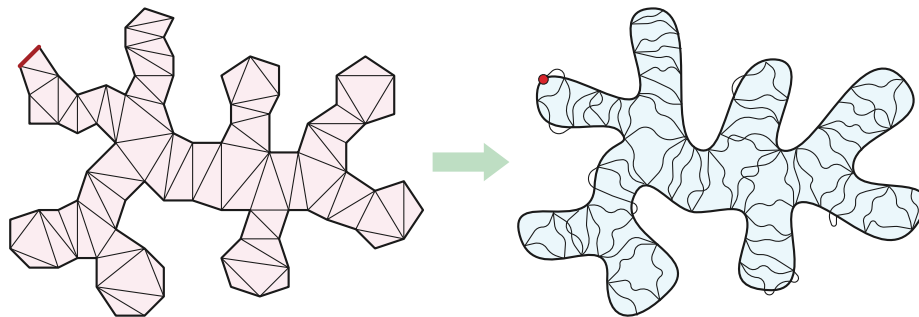


Figure 7.5. Any closed walk in a hyperbolic tiling can be “triangulated” by paths of logarithmic length.

Lemma 7.5. *Every simple cycle of length L in \widehat{Q} has a $(2\rho + 2)$ -triangulation, where $\rho = \lceil \log_\lambda(2L/g) \rceil$.*

Proof: For the moment, let γ be a *simple* cycle of length L in \widehat{Q} , and let Γ denote the closed disk bounded by γ . Intuitively, we construct a “Delaunay triangulation” of the vertices of γ inside the disk Γ .

For each tiling vertex x in Γ , let $nn(x)$ denote its nearest neighbor on γ , measuring distance by counting edges in \widehat{Q} and breaking ties arbitrarily. In particular, if $x \in \gamma$, then $nn(x) = x$. Lemma 7.4 implies that the distance from any vertex x to its nearest neighbor $nn(x)$ is at most ρ .

Arbitrarily triangulate each tile of Γ to obtain a triangulation Δ , and let M denote the set of all edges xy in Δ such that $nn(x) \neq nn(y)$. (Intuitively, the duals of edges in M form a “discrete medial axis” of Γ [5].) For each edge xy in M , let $\sigma(x, y)$ denote any shortest path in \widehat{Q} (not in Δ !) from $nn(x)$ to $nn(y)$. (In particular, if xy is an edge of γ , then $\sigma(x, y) = xy$.) The triangle inequality implies that each shortest path $\sigma(x, y)$ has length at most $2\rho + 2$.⁵

The shortest paths $\{\sigma(x, y) \mid xy \in M(\gamma)\}$ are the diagonal paths of a $(2\rho + 2)$ -triangulation of γ . Specifically, for every triangle xyz in Δ whose vertices have three distinct nearest neighbors, the shortest paths $\sigma(x, y)$, $\sigma(y, z)$, and $\sigma(x, z)$ define a triangle $\Delta(xyz)$ in the $(2\rho + 2)$ -triangulation. \square

Lemma 7.6. *Every closed walk of length L in \widehat{Q} has a $(2\rho + 2)$ -triangulation, where $\rho = \lceil \log_\lambda(2L/g) \rceil$.*

⁵A nearly identical argument implies that for any $t > 1$, any simple cycle of length L in a regular hyperbolic tiling by $2t$ -gons has a $(2\rho + t)$ -triangulation, for some appropriate inradius function $\rho = O(\log_\lambda L)$. Unfortunately, the $+t$ term in the triangulation constant implies a $g^{O(t)}$ factor in the complexity of our context-free grammar, and therefore in the running time of our algorithm. In particular, using a system of loops would lead to an extra $g^{O(s)}$ factor in the running time.

Proof: The proof proceeds by induction on L . Let ω be any closed walk of length L . If ω is a simple cycle, we can defer to the previous lemma, so assume otherwise. There are two cases to consider.

First, suppose ω contains a spur $x \rightarrow y \rightarrow x$, and let ω' be the walk obtained by removing that spur. The induction hypothesis implies that ω' has a $(2\rho + 2)$ -triangulation \mathcal{T}' . There are two subcases to consider; see the top row of Figure 7.6.

- If \mathcal{T}' contains the path $\sigma(xz)$, we can transform \mathcal{T}' into a $(2\rho + 2)$ -triangulation of ω by expanding $\sigma(yz)$ into two triangles $\Delta(yzy)$ and $\Delta(yxy)$.
- Otherwise, the triangulation \mathcal{T}' must contain the path $\sigma(wz)$ between the neighbors of x on ω' , either as an edge of ω or as a diagonal path. We can transform \mathcal{T}' into a $(2\rho + 2)$ -triangulation of ω by replacing the triangle $\Delta(wxz)$ with three triangles $\Delta(wxz)$, $\Delta(wx x)$, and $\Delta(xxy)$.

In either case, each new diagonal path is either empty or a duplicate of some path in \mathcal{T}' , and thus has length at most $2\rho + 2$.

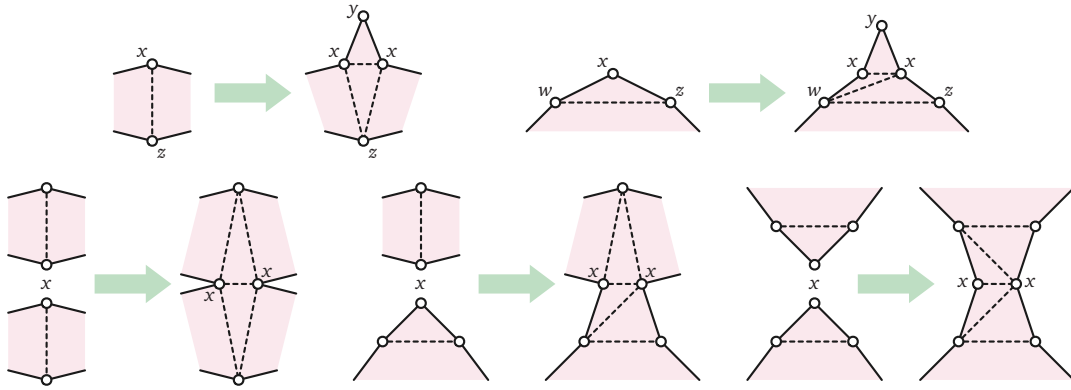


Figure 7.6. Top: Extending a k -triangulation across a spur. Bottom: Merging two k -triangulations across a repeated vertex.

Now suppose ω does not contain a spur, but does contain a repeated vertex x . We can partition ω into two smaller closed walks ω_1 and ω_2 with a common vertex x . The induction hypothesis implies that ω_1 and ω_2 have $(2\rho + 2)$ -triangulations \mathcal{T}_1 and \mathcal{T}_2 . We can merge \mathcal{T}_1 and \mathcal{T}_2 into a $(2\rho + 2)$ -triangulation \mathcal{T} of ω by duplicating x and inserting new diagonal paths. There are three subcases to consider, depending on whether x is the endpoint of a diagonal path in both, only one, or neither of \mathcal{T}_1 and \mathcal{T}_2 ; see the bottom row of Figure 7.6. Again, each new diagonal path is either empty or a duplicate of some path in \mathcal{T}_1 or \mathcal{T}_2 , and thus has length at most $2\rho + 2$.

In all cases, we obtain a $(2\rho + 2)$ -triangulation of the original walk ω . \square

7.4 Context-Free Grammar

Now finally we reach the key component of our algorithm: the construction of a small context-free grammar \mathcal{G} such that the shortest contractible closed walk in G is also the shortest walk whose label is in the language generated by \mathcal{G} . Our construction is essentially the same as that of Muller and Schupp [57, Theorem 1], but with more careful analysis.

Lemma 7.7. *For any positive integer L , there is a context-free grammar \mathcal{G} in Chomsky normal form with $O(g^2 L^2)$ non-terminals, such that (1) Every string generated by \mathcal{G} describes a contractible closed walk in Q , and (2) every contractible closed walk in Q of length at most L is generated by \mathcal{G} .*

Proof: As in the previous proofs, let $\rho = \lceil \log_\lambda(2L/g) \rceil$.

The alphabet Σ of \mathcal{G} consists of all $8g$ darts in the system of quads Q .

Call a homotopy class of walks in Q *short* if it contains a walk of length at most $2\rho + 2$. The non-terminals of \mathcal{G} correspond to short homotopy classes of walks in Q . More formally, fix an arbitrary lift \widehat{a} of a and an arbitrary lift \widehat{z} of z . Each short homotopy class corresponds to a unique ordered pair $(\widehat{s}, \widehat{t})$ of vertices in \widehat{Q} , where $\widehat{s} \in \{\widehat{a}, \widehat{z}\}$ and the shortest-path distance from \widehat{s} to \widehat{t} in \widehat{Q} is at most $2\rho + 2$. Thus, the number of short homotopy classes, and therefore the number of non-terminals in \mathcal{G} , is at most

$$2N(2\rho + 2) = O(\lambda^{2\rho+2}) \leq O(\lambda^{2\log_\lambda(2L/g)+4}) = O((2L/g)^2 \lambda^4) = O(g^2 L^2)$$

by Lemma 7.1 and the definition of ρ . (Moreover, Lemma 7.1 implies that the constant hidden in the final $O()$ bound is at most 8.)

The starting non-terminal of \mathcal{G} is the homotopy class of contractible walks in G that start and end at a , represented by the vertex pair $(\widehat{a}, \widehat{a})$.

Non-terminal productions in \mathcal{G} correspond to pairs of short homotopy classes whose concatenation is also a short homotopy class. Specifically, if α and β are walks in short homotopy classes A and B , and their concatenation $\alpha \cdot \beta$ is in short homotopy class C , then \mathcal{G} contains the production $C \rightarrow AB$. There are trivially at most $|N|^2 = O(g^4 L^4)$ non-terminal productions.⁶

There are $8g$ terminal productions $A \rightarrow d$ in \mathcal{G} , one for (the homotopy class of) each dart d in Q . We deliberately exclude ε -productions from \mathcal{G} to avoid generating the empty string.

Every string generated by the nonterminal $(\widehat{s}, \widehat{t})$ is the projection of a non-empty walk from \widehat{s} to \widehat{t} in \widehat{Q} ; all such walks are in the same (short) homotopy class. In particular, every string generated by the starting non-terminal $(\widehat{a}, \widehat{a})$ is a non-empty contractible walk in Q from a to a .

On the other hand, let ω be any non-empty closed walk of length at most L in \widehat{Q} . Lemma 7.6 implies that γ has a $(2\rho + 2)$ -triangulation \mathcal{T} . The dual tree of this triangulation is a parse tree for ω . Specifically, the (weak) dual of the reference triangulation T is a binary tree T^* . Direct the dual of the root edge of T into the polygon, and direct every other edge of T^* away from the root. For each edge of T , label the *head* of its directed dual edge with the homotopy class of its image in \widehat{Q} . With this labeling, T^* is a parse tree for $\langle \omega \rangle$ with respect to \mathcal{G} . \square

Theorem 7.8. *Let G be a directed graph with m non-negatively weighted edges, embedded on the orientable surface of genus g with no boundary. We can compute the shortest contractible closed walk in G in $O(g^6 m^9)$ time.*

Proof: We construct the system of quads Q and label the edges of G in $O(m)$ time. If necessary, subdivide edges of G so that each edge is labeled with either the empty walk or a single dart in Q . After subdivision, G has at most $2m$ edges, and therefore (because G is symmetric) at most $2m$ vertices.

Then we build a context-free grammar \mathcal{G} that generates only contractible walks in Q and generates all non-empty contractible walks in Q of length at most $2m$, as described in Lemma 7.7. This grammar has $N = O(g^2 m^2)$ nonterminals and $P = O(g^4 m^4)$ productions. Corollary 3.6 implies that the shortest contractible walk in G has length at most $2m$, and therefore has a label of length at most $2m$. It follows that \mathcal{G} generates the label of the shortest contractible walk in G .

Finally, we compute the shortest closed walk in G whose label is generated by \mathcal{G} , using the CFG-shortest-path algorithm of Barrett *et al.* [3], in $O(NP(n')^3) = O(g^6 m^9)$ time. \square

Acknowledgements. The first author would like thank Tillmann Miltzow for asking an annoying question that led to this work, and to apologize for still being unable to answer it.

⁶We believe the number of non-terminal productions is actually $\Theta(g^3 L^3)$; however, this refinement would only improve our final running time to $O(g^5 m^8)$, which doesn't seem worth the trouble.

References

- [1] James W. Alexander. Topological invariants of knots and links. *Trans. Amer. Math. Soc.* 30(2):275–306, 1928.
- [2] Lars Arge, Laura Toma, and Norbert Zeh. I/O-efficient topological sorting of planar DAGs. *Proc. 15th Ann. ACM Symp. Parallel Algorithms Arch.*, 85–93, 2003.
- [3] Chris Barrett, Riko Jacob, and Madhav Marathe. Formal-language-constrained path problems. *SIAM J. Comput.* 30(3):809–837, 2000.
- [4] Phillip G. Bradford and David A. Thomas. Labeled shortest paths in digraphs with negative and positive edge weights. *RAIRO-Theor. Inf. Appl.* 43(3):567–583, 2009.
- [5] Jonathan W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CVGIP: Image Understanding* 59(1):116–124, 1994.
- [6] Sergio Cabello. Finding shortest contractible and shortest separating cycles in embedded graphs. *ACM Trans. Algorithms* 6(2):24:1–24:18, 2010.
- [7] Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-source shortest paths in embedded graphs. *SIAM J. Comput.* 42(4):1542–1571, 2013. [arXiv:1202.0314](https://arxiv.org/abs/1202.0314).
- [8] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *Proc. 26th Ann. Symp. Comput. Geom.*, 156–165, 2010.
- [9] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding cycles with topological properties in embedded graphs. *SIAM J. Discrete Math.* 25:1600–1614, 2011.
- [10] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *J. Comput. Geom.* 7(1):123–148, 2016.
- [11] Sergio Cabello, Matt DeVos, Jeff Erickson, and Bojan Mohar. Finding one tight cycle. *ACM Trans. Algorithms* 6(4):article 61, 2010.
- [12] Sergio Cabello and Bojan Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete Comput. Geom.* 37:213–235, 2007.
- [13] Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Francis Lazarus, and Kim Whittlesey. Splitting (complicated) surfaces is hard. *Comput. Geom. Theory Appl.* 41(1–2):94–110, 2008.
- [14] Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. *Proc. 25th Ann. Symp. Comput. Geom.*, 377–385, 2009.
- [15] Hsien-Chih Chang, Jeff Erickson, Arnaud de Mesmay, David Letscher, Saul Schleimer, Eric Sedgwick, Dylan Thurston, and Stephan Tillmann. Tightening curves on surfaces via local moves. *Proc. 29th Ann. ACM-SIAM Symp. Discrete Algorithms*, 121–135, 2017.
- [16] Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. *Proc. 26th ACM-SIAM Symp. Discrete Algorithms*, 1655–1670, 2015.
- [17] Victor Chepoi, Feodor Dragan, Bertrand Estellon, Michel Habib, and Yann Vaxès. Diameters, centers, and approximating trees of δ -hyperbolic geodesic spaces and graphs. *Proc. 24th Ann. Symp. Comput. Geom.*, 59–68, 2008.

- [18] Edith Cohen and Nimrod Megiddo. Strongly polynomial-time and NC algorithms for detecting cycles in periodic graphs. *J. Assoc. Comput. Mach.* 40(4):791–830, 1993.
- [19] Éric Colin de Verdière and Jeff Erickson. Tightening non-simple paths and cycles on surfaces. *SIAM J. Comput.* 39(8):3784–3813, 2010.
- [20] Max Dehn. Über unendliche diskontinuierliche Gruppen. *Math. Ann.* 71(1):116–144, 1911.
- [21] Max Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Math. Ann.* 72(3):413–421, 1912.
- [22] Max Dehn. *Papers on Group Theory and Topology*. Springer, 1987. Translated by John Stillwell.
- [23] Vincent Despré and Francis Lazarus. Computing the geometric intersection number of curves. *Proc. 33rd Int. Symp. Comput. Geom.*, 35:1–35:15, 2017. Leibniz Int. Proc. Informatics 77. arXiv:[1511.09327](https://arxiv.org/abs/1511.09327).
- [24] Tamal K. Dey and Sumanta Guha. Transforming curves on surfaces. *J. Comput. System Sci.* 58:297–325, 1999.
- [25] Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., 2010.
- [26] David Eppstein. Dynamic generators of topologically embedded graphs. *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, 599–608, 2003. arXiv:[cs.DS/0207082](https://arxiv.org/abs/cs.DS/0207082).
- [27] David B. A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica* 115:83–107, 1966.
- [28] Jeff Erickson. Shortest non-trivial cycles in directed surface graphs. *Proc. 27th Ann. Symp. Comput. Geom.*, 236–243, 2011.
- [29] Jeff Erickson, Kyle Fox, and Luvsandondov Lkhamsuren. Holiest minimum-cost paths and flows in surface graphs. *Proc. 50th Ann. ACM Symp. Theory Comput.*, 1319–1332, 2018. arXiv:[1804.01045](https://arxiv.org/abs/1804.01045).
- [30] Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31:37–59, 2004.
- [31] Jeff Erickson and Amir Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 1166–1176, 2011.
- [32] Jeff Erickson and Amir Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 1166–1176, 2011.
- [33] Jeff Erickson and Yipu Wang. Topologically trivial closed walks in directed surface graphs. *Proc. 33rd Int. Symp. Comput. Geom.*, 34:1–34:17, 2019. Leibniz International Proceedings in Informatics 129.
- [34] Jeff Erickson and Kim Whittlesey. Transforming curves on surfaces redux. *Proc. 24th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1646–1655, 2013.
- [35] Jeff Erickson and Kim Whittlesey. Transforming curves on surfaces redux. *Proc. 24th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1646–1655, 2013.
- [36] William J. Floyd and Steven J. Plotnick. Growth functions on Fuchsian groups and the Euler characteristic. *Invent. Math.* 88(1):1–29, 1987. (<http://eudml.org/doc/143441>).

- [37] Kyle Fox. Shortest non-trivial cycles in directed and undirected surface graphs. *Proc. 24th Ann. ACM-SIAM Symp. Discrete Algorithms*, 352–364, 2013. arXiv:[1111.6990](https://arxiv.org/abs/1111.6990).
- [38] Steve M. Gersten and Hamish B. Short. Small cancellation theory and automatic groups. *Invent. Math.* 102:305–334, 1990.
- [39] Peter Gabilin. *Graphs, Surfaces and Homology*, 3rd edition. Cambridge Univ. Press, 2010.
- [40] Rostislav Grigorchuk and Pierre de la Harpe. On problems related to growth, entropy and spectrum in group theory. *J. Dynam. Control Systems* 3(1):51–89, 1997.
- [41] Mikhail Gromov. Hyperbolic groups. *Essays in Group Theory*, 75–265, 1987. Math. Sci. Res. Inst. Pub. 8, Springer-Verlag.
- [42] Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel J. Math.* 51:90–120, 1985.
- [43] Allen Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2002. (<http://www.math.cornell.edu/~hatcher/AT/ATpage.html>).
- [44] Kazuo Iwano and Kenneth Steiglitz. A semiring on convex polygons and zero-sum cycle problems. *SIAM J. Comput.* 19(5):883–901, 1990.
- [45] Ming-Yang Kao. Linear-processor NC algorithms for planar directed graphs I: Strongly connected components. *SIAM J. Comput.* 22(3):431–459, 1993.
- [46] Ming-Yang Kao and Philip N. Klein. Toward overcoming the transitive-closure bottleneck: Efficient parallel algorithms for planar digraphs. *J. Comput. Syst. Sci.* 47(3):459–500, 1993.
- [47] Ming-Yang Kao and Gregory E. Shannon. Local reorientation, global order, and planar topology. *Proc. 21st Ann. ACM Symp. Theory Comput.*, 286–296, 1989.
- [48] Ming-Yang Kao and Gregory E. Shannon. Linear-processor NC algorithms for planar directed graphs II: Directed spanning trees. *SIAM J. Comput.* 22(3):460–481, 1993.
- [49] S. Rao Kosaraju and Gregory F. Sullivan. Detecting cycles in dynamic graphs in polynomial time (preliminary version). *Proc. 20th ACM Symp. Theory Comput.*, 398–406, 1988.
- [50] Martin Kutz. Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time. *Proc. 22nd Ann. Symp. Comput. Geom.*, 430–438, 2006.
- [51] Francis Lazarus and Julien Rivaud. On the homotopy test on surfaces. *Proc. 53rd Ann. IEEE Symp. Foundations Comput. Sci.*, 440–449, 2012. arXiv:[1110.4573](https://arxiv.org/abs/1110.4573).
- [52] Roger C. Lyndon and Paul E. Schupp. *Combinatorial Group Theory*. Classics in Mathematics. Springer-Verlag, 2001.
- [53] John Milnor. A note on curvature and the fundamental group. *J. Diff. Geom.* 2(1):1–7, 1968.
- [54] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins Univ. Press, 2001.
- [55] Judith Flagg Moran. The growth rate and balance of homogeneous tilings in the hyperbolic plane. *Discrete Math.* 173(1–3):151–186, 1997.

- [56] Shay Mozes, Kirill Nikolaev, Yahav Nussbaum, and Oren Weimann. Minimum cut of directed planar graphs in $O(n \log \log n)$ time. *Proc. 29th Ann. ACM-SIAM Symp. Discrete Algorithms*, 477–494, 2018. arXiv:[1512.02068](https://arxiv.org/abs/1512.02068).
- [57] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.* 26(3):295–310, 1983.
- [58] August F. Möbius. Über der Bestimmung des Inhaltes eines Polyeders. *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl.* 17:31–68, 1865. *Gesammelte Werke* 2:473–512, Leipzig, 1886.
- [59] Max Neumann-Coto. A characterization of shortest geodesics on surfaces. *Algebr. Geom. Topol.* 1:349–368, 2001. arXiv:[math.GT/0106200](https://arxiv.org/abs/math/0106200).
- [60] Ján Plesník. The NP-completeness of the Hamiltonian cycle problem in planar digraphs with degree bound two. *Inf. Proc. Letters* 8(4):199–201, 1979.
- [61] Vijaya Ramachandran and Honghua Yang. Finding the closed partition of a planar graph. *Algorithmica* 11(5):443–468, 1994.
- [62] L. B. Smikun. Connection between context-free groups and groups with decidable problems of automata equivalence. *Cybernetics* 12(5):687–691, 1976. Translated from *Kibernetika (Kiev)* (5):33–37, 1976.
- [63] Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory Ser. B* 48(2):155–177, 1990.
- [64] Mihalis Yannakakis. Graph-theoretic methods in database theory. *Proc. 9th ACM SIGACT-SIGMOD-SIGART Symp. Principles Database Syst.*, 230–242, 1990.