Shortest Non-trivial Cycles in Directed Surface Graphs*

Jeff Erickson
Department of Computer Science
University of Illinois, Urbana-Champaign

ABSTRACT

Let G be a directed graph embedded on a surface of genus g. We describe an algorithm to compute the shortest non-separating cycle in G in $O(g^2 n \log n)$ time, exactly matching the fastest algorithm known for undirected graphs. We also describe an algorithm to compute the shortest non-contractible cycle in G in $g^{O(g)} n \log n$ time, matching the fastest algorithm for undirected graphs of constant genus.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [Discrete Mathematics]: Graph Theory—*Graph algorithms, Path and circuit problems*

General Terms: Algorithms, Performance, Theory

Keywords: topological graph theory, computational topology

1. INTRODUCTION

A key step in several algorithms for surface-embedded graphs is cutting a surface along a topologically interesting cycle to reduce its topological complexity. Examples include algorithms for probabilistically embedding high-genus graphs into planar graphs [3, 30], drawing abstract graphs in the plane with few crossings [37], testing isomorphism between graphs of fixed genus [36], approximating optimal traveling salesman tours [18] and Steiner trees [1], and removing topological noise from surface models [27, 47]. In all these applications, cutting along the shortest possible cycle is preferred or even required. These and other applications have motivated a long series of results on finding shortest non-trivial cycles in surface-embedded graphs.

Itai and Shiloach [31] observed that the minimum (s,t)-cut in an undirected planar graph G^* is dual to the shortest cycle in the dual graph that separates the dual faces s^* and t^*. Thus, computing minimum cuts in planar graphs is equivalent to finding the shortest non-trivial cycle in a graph embedded on an annulus. Itai and Shiloach described a simple algorithm to find this cycle in $O(n^2 \log n)$ time. Their algorithm has been improved several times [42, 25], most recently by Italiano *et al.* [32, 33, 46], who describe an algorithm that runs in $O(n \log\log n)$ time.

Thomassen [44] developed the first efficient algorithm for graphs on arbitrary surfaces, which runs in $O(n^3)$ time and exploits his so-called *3-path condition*; see also Mohar and Thomassen [40, Sect. 4.3]. Erickson and Har-Peled described a faster algorithm that runs in $O(n^2 \log n)$ time [21]. This is the best algorithm known for arbitrary surface-embedded graphs, but faster algorithms are known when the genus g of the underlying surface is small [4, 6, 12, 39], the fastest of which runs in $O(g^2 n \log n)$ time, where g is the genus of the underlying surface [7]. All of these faster algorithms exploit the observation by Cabello and Mohar [12] that the shortest non-trivial cycle crosses any shortest path at most once. For related results and extensions, see [5, 9, 10, 11, 13, 15, 24].

Both Thomassen's 3-path condition [44] and Cabello and Mohar's crossing condition [12] are consequences of the following easy observation: For any four vertices s, t, u, v in an *undirected* surface graph, there is a shortest path from s to t and a shortest path from u to v that cross at most once. In directed surface graphs, however, shortest paths may cross an arbitrary number of times, as long as the two paths visit the crossings in opposite orders. As a result, the history of algorithms for finding shortest non-trivial cycles in directed graphs is much shorter.

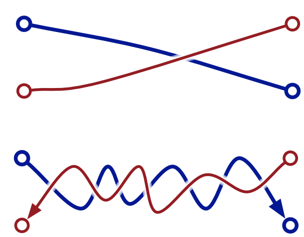

Figure 1. Undirected shortest paths cross at most once, but directed shortest paths may cross arbitrarily many times.

The shortest non-trivial *directed* cycle in an annular graph is dual to either the minimum (s,t)-cut or the minimum (t,s)-cut in the directed planar dual graph, whichever has smaller capacity, where s and t are the dual vertices corresponding to the boundaries of the annulus. Janiga and Koubek [34] describe

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

an adaptation of Reif's algorithm that can find this cycle[1] in $O(n\log^2 n/\log\log n)$ time; the running time of their algorithm can be improved to $O(n\log n)$ using more recent algorithms for shortest paths [29, 38]. The same running time can also be achieved by recent planar maximum flow algorithms [2, 20, 45].

The first results for directed graphs on surfaces of higher genus were only recently obtained by Cabello, Colin de Verdière, and Lazarus [8], who describe two algorithms. Their first algorithm, which runs in $O(n^2\log n)$ time, relies on a subtle generalization of both Thomassen's 3-path condition to directed graphs. Their second algorithm, which runs in $O(g^{1/2}n^{3/2}\log n)$ time, uses a divide-and-conquer strategy based on balanced separators. Even more recently, Erickson and Nayyeri describe an algorithm to compute the shortest *non-separating* cycle in $2^{O(g)}n\log n$ time [22]. However, their approach does not imply a faster algorithm for computing shortest *non-contractible* cycles.

This paper describes faster algorithms for computing shortest non-separating and non-contractible cycles in directed surface graphs. Specifically, in Section 3, we describe an algorithm to compute the shortest non-separating cycle in G in $O(g^2 n\log n)$ time, exactly matching the fastest algorithm known for undirected graphs. Like Cabello and Mohar [12], we reduce the problem to finding the shortest cycle γ that crosses a given non-separating cycle λ composed of two shortest paths an odd number of times. Following Janiga and Koubek [34], we solve this subproblem by reducing it to a shortest-path problem in a certain double cover of the surface.

In Section 4, we describe an algorithm to compute the shortest non-contractible cycle in G in $g^{O(g)}n\log n$ time, matching the fastest algorithm for undirected graphs of constant genus. Our algorithm follows the same high-level strategy as Kutz's algorithm for undirected graphs [39]. Specifically, our algorithm searches a finite portion of the universal cover of the surface, by cutting the surface into a disk along shortest paths and then pasting together several copies of this disk along corresponding paths. Our key observation is that although the shortest non-contractible cycle γ may intersect each of these shortest paths arbitrarily many times, at most one intersection with any shortest path is topologically non-trivial.

2. BACKGROUND

We begin by recalling several standard definitions and results related to surface-embedded graphs. For further background, we refer the reader to Gross and Tucker [26] or Mohar and Thomassen [40] for topological graph theory, and to Hatcher [28] or Stillwell [43] for algebraic topology.

2.1 Surfaces and Curves

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

[1] Janiga and Koubek actually claim an algorithm to compute the minimum (s,t)-cut, but their algorithm has a subtle error [35], which may lead to an incorrect result when the minimum (t,s)-cut is smaller than the minimum (s,t)-cut.

A *path* in a surface Σ is a continuous function $p\colon[0,1]\to\Sigma$. A *loop* is a path whose endpoints $p(0)$ and $p(1)$ coincide; we refer to this common endpoint as the *basepoint* of the loop. An *arc* is a path whose endpoints lie on the boundary of Σ. A *cycle* is a continuous function $\gamma\colon S^1\to\Sigma$, where $S^1=\mathbb{R}/\mathbb{Z}$ is the circle; the only difference between a cycle and a loop is that a loop has a distinguished basepoint. We say that a loop ℓ and a cycle γ are *equivalent* if, for some real number δ, we have $\ell(t)=\gamma(t+\delta)$ for all $t\in[0,1]$. We collectively refer to paths, loops, arcs, and cycles as *curves*. A curve is *simple* if it is injective, except for the basepoint in the case of loops; we usually do not distinguish between simple curves and their images in Σ.

The *reversal* $\mathrm{rev}(p)$ of a path p is defined by setting $rev(p)(t)=p(1-t)$. The *concatenation* $p\cdot q$ of two paths p and q with $p(1)=q(0)$ is the path created by setting $(p\cdot q)(t)=p(2t)$ for all $t\leq 1/2$ and $(p\cdot q)(t)=q(2t-1)$ for all $t\geq 1/2$. Finally, let $p[x,y]$ denote the subpath of a path p from point x to point y.

The *genus* of a surface Σ is the maximum number of disjoint simple cycles in Σ whose complement is connected. We will consider only compact, connected, orientable surfaces. Unless explicitly stated otherwise, all surfaces in this paper are compact, connected, orientable, and without boundary. Up to homeomorphism, there is exactly one such surface with any genus $g\geq 0$. (We briefly consider surfaces with boundary in Section 4.3.)

2.2 Graphs and Embeddings

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

Any undirected graph G embedded on a surface Σ has a *dual graph* G^*, which has a vertex f^* for each face f of G, and an edge e^* for each edge e in G joining the vertices dual to the faces of G that e separates. The dual graph G^* has a natural cellular embedding in Σ, whose faces corresponds to the vertices of G. For any subgraph $F=(U,D)$ of $G=(V,E)$, we write $G\setminus F$ to denote the edge-complement $(V,E\setminus D)$. We also abuse notation by writing F^* to denote the subgraph of G^* corresponding to any subgraph F of G.

A *tree-cotree decomposition* (T,L,C) of an undirected surface-embedded graph G is a partition of its edges into three disjoint subsets: a spanning tree T of G, a spanning cotree C (the dual of a spanning tree C^* of G^*), and leftover edges $L=G\setminus(T\cup C)$. Euler's formula implies that in any tree-cotree decomposition, the set L contains exactly $2g$ edges [19].

For the problems we consider, the input is actually a *directed* edge-weighted graph G with a cellular embedding on some surface. We use the notation $u{\to}v$ to denote the directed edge from vertex u to vertex v. Without loss of generality, we consider only *symmetric* directed graphs, in which the reversal $v{\to}u$ of any edge $u{\to}v$ is another edge, possibly with infinite weight. We also assume that in the cellular embedding, the images of any edge in G and its reversal coincide (but with opposite orientations). Thus, like Cabello *et al.* [8, Section 2.3], we implicitly model directed graphs as *undirected graphs with asymmetric edge weights*. Duality can be extended to directed graphs [14], but the results in this paper do not require this extension.

To simplify our presentation and analysis, we assume that any two vertices x and y in G are connected by a unique shortest directed path, denoted $\sigma(x,y)$. The Isolation Lemma [41] implies that this assumption can be enforced (with high probability) by perturbing the edge weights with random infinitesimal values [21].

Our algorithms rely the following seminal result of Klein [38] for planar graphs, and its generalization to higher-genus graphs by Cabello *et al.* [6, 7].

Lemma 2.1 (Klein [38]). *Let G be a directed plane graph with non-negative edge weights and let f be an arbitrary face of G. We can preprocess G in $O(n \log n)$ time and $O(n)$ space, so that the length of the shortest path from any vertex incident to f to any other vertex can be retrieved in $O(\log n)$ time.*

Lemma 2.2 (Cabello *et al.* [6, 7]). *Let G be a directed graph with non-negative edge weights, cellularly embedded on a surface Σ of genus g, and let f be an arbitrary face of G. We can preprocess G in $O(gn \log n)$ time[2] and $O(n)$ space, so that the length of the shortest path from any vertex incident to f to any other vertex can be retrieved in $O(\log n)$ time.*

2.3 Homotopy and Homology

Two paths p and q in a surface Σ are *homotopic* if one can be continuously deformed into the other without changing their endpoints. More formally, a *homotopy* between p and q is a continuous map $h\colon [0,1]\times[0,1] \to \Sigma$ such that $h(0,\cdot) = p$, $h(1,\cdot) = q$, $h(\cdot,0) = p(0) = q(0)$, and $h(\cdot,1) = p(1) = q(1)$. Homotopy defines an equivalence relation over the set of paths with any fixed pair of endpoints. The set of homotopy classes of loops in Σ with basepoint x_0 defines a group $\pi_i(\Sigma, x_0)$ under concatenation, called the *fundamental group* of Σ. (For all basepoints x_0 and x_1, the groups $\pi_i(\Sigma, x_0)$ and $\pi_i(\Sigma, x_1)$ are isomorphic.) A cycle is *contractible* if it is homotopic to a constant map.

Homology is a coarser equivalence relation than homotopy, with nicer algebraic properties; intuitively, two cycles are homologous if together they define the boundary of some subset of the surface. Like several earlier papers [12, 15, 22], we consider only one-dimensional cellular homology with coefficients in the finite field $\mathbb{Z}_2$; this restriction allows us to radically simplify our definitions.

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

[2]The published version of this algorithm [6] has a weaker time bound of $O(g^2 n \log n)$. Using this version increases the running time of our algorithm for computing the shortest non-separating cycle by a factor of g.

2.4 Covering Spaces

A continuous map $\pi\colon \Sigma' \to \Sigma$ between two surfaces is called a *covering map* if each point $x \in \Sigma$ lies in an open neighborhood U such that (1) $\pi^{-1}(U)$ is a countable union of disjoint open sets $U_1 \cup U_2 \cup \cdots$ and (2) for each i, the restriction $\pi|_{U_i} : U_i \to U$ is a homeomorphism. If there is a covering map π from Σ' to Σ, we call Σ' a *covering space* of Σ. The *universal cover* $\widetilde{\Sigma}$ is the unique simply-connected covering space of Σ (up to homeomorphism). The universal cover is so named because it covers every path-connected covering space of Σ.

For any path $p\colon [0,1] \to \Sigma$ such that $\pi(x') = p(0)$ for some point $x' \in \Sigma'$, there is a unique path p' in Σ', called a *lift* of p, such that $p'(0) = x'$ and $\pi \circ p' = p$. We also say that p *lifts to* p'. Conversely, for any path p' in Σ', the path $\pi \circ p'$ is called a *projection* of p'.

We define a lift of a cycle $\gamma\colon S^1 \to \Sigma$ to be the infinite path $\gamma'\colon \mathbb{R} \to \Sigma'$ such that $\pi(\gamma'(t)) = \gamma(t \bmod 1)$ for all real t. We call the path obtained by restricting γ' to any unit interval a *single-period lift* of γ; equivalently, a single-period lift of γ is a lift of any loop equivalent to γ. We informally say that a cycle is the *projection* of any of its single-period lifts.

3. NON-SEPARATING CYCLES

Let G be a symmetric directed graph with non-negative edge weights, cellularly embedded on an orientable surface Σ of genus g. In this section, we develop our algorithm to compute the shortest non-separating directed cycle in G. Without loss of generality, we assume that Σ has no boundary, since pasting a disk onto a boundary cycle of Σ does not change the set of non-separating cycles.

3.1 The Cyclic Double Cover

Let λ be an arbitrary simple non-separating cycle in Σ. For any other directed cycle γ, we define the *crossing parity* $\varepsilon_\lambda(\gamma)$ to be 1 if γ crosses λ an odd number of times, and 0 otherwise. Equivalently, if λ and γ are both cycles in the graph G, we have

$$\varepsilon_\lambda(\gamma) = \bigoplus_{u\to v \in \gamma} \varepsilon_\lambda(u\to v),$$

where for any directed edge $u\to v$, we define $\varepsilon_\lambda(u\to v)$ to be 1 if $u\to v$ *either* enters λ from the left *or* leaves λ to the left, but not both, and 0 otherwise. Here $\oplus$ denotes addition modulo 2.

We define a covering space Σ^2_λ, called a *cyclic double cover*, as follows. Cutting the surface Σ along λ gives us a new surface Σ' with exactly two boundary cycles λ^+ and λ^-. Let $(\Sigma', 0)$ and $(\Sigma', 1)$ denote two distinct copies of Σ'. For any point $p \in \Sigma'$, let $(p, 0)$ and $(p, 1)$ denote the corresponding points in $(\Sigma', 0)$ and $(\Sigma', 1)$, respectively. In particular, let $(\lambda^+, 0)$ and $(\lambda^-, 0)$ denote the copies of λ^+ and λ^- in $(\Sigma, 0)$. Finally, let Σ^2_λ be the surface obtained by identifying $(\lambda^+, 0)$ and $(\lambda^-, 1)$ to a single cycle, denoted $(\lambda, 0)$, and identifying $(\lambda^+, 1)$ and $(\lambda^-, 0)$ to a single cycle, denoted $(\lambda, 1)$. Any graph G that is cellularly embedded in Σ lifts to a graph G^2_λ, with twice as many vertices and edges, that is cellularly embedded in Σ^2_λ. See Figure 2.

For combinatorial surfaces, we can equivalently define the double cyclic cover using the following standard *voltage construction* [26, Chapter 4]. Let G^2_λ be the graph whose vertices are the pairs (v, b), where v is a vertex of G and b is a bit, and whose edges are the ordered pairs

$$(u\to v, b) \;:=\; (u, b)\to(v, b \oplus \varepsilon_\lambda(u\to v))$$

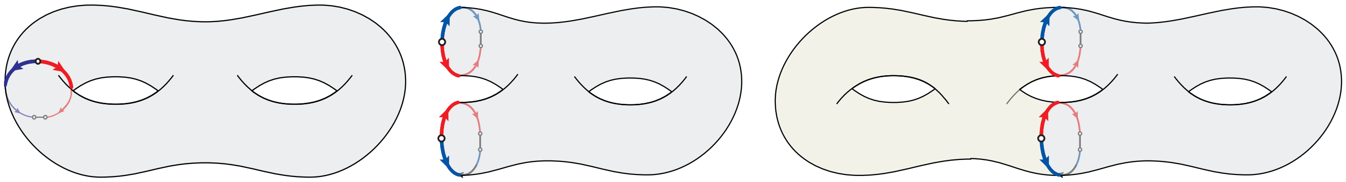

Figure 2. Left: A non-separating cycle λ on a surface Σ of genus 2. Middle: the cut surface Σ'. Right: the cyclic double cover Σ^2_λ.

for all edges $u \rightarrow v$ of G and both bits b. Let $\pi\colon G^2_\lambda \to G$ denote the obvious covering map $\pi(v,b) = v$. We declare that a cycle in G^2_λ bounds a face of G^2_λ if and only if its projection to G bounds a face of G. The resulting embedding of G^2_λ defines the cyclic double cover Σ^2_λ.

The following lemmas are now immediate.

Lemma 3.1. *Let λ be any simple non-separating cycle in Σ; let γ be any cycle in Σ; and let s be any vertex of γ. Then γ is the projection of a unique path in Σ^2_λ from $(s,0)$ to $(s,\varepsilon_\lambda(\gamma))$.*

Lemma 3.2. *Let λ be any simple non-separating cycle in Σ. Every lift of a shortest directed path in G is a shortest directed path in G^2_λ.*

Lemma 3.3. *Let λ be any simple non-separating cycle in Σ; let γ be the shortest cycle in Σ that crosses λ an odd number of times; and let s be any vertex of γ. Then γ is the projection of a shortest path in Σ^2_λ from $(s,0)$ to $(s,1)$.*

For any simple non-separating cycles λ and λ', the cyclic double covers Σ^2_λ and $\Sigma^2_{\lambda'}$ are homeomorphic surfaces; however, the graphs G^2_λ and $G^2_{\lambda'}$ are *not* necessarily isomorphic.

3.2 Algorithm

Our algorithm begins by constructing a greedy tree-cotree decomposition (T,L,C) of G, where T is a shortest-path tree rooted at some arbitrary vertex of G. Euler's formula implies that L contains exactly $2g$ edges; label these edges arbitrarily as u_1v_1, $u_2v_2, \dots, u_{2g}v_{2g}$. For each index i, let λ_i denote the unique cycle in the *undirected* graph $T \cup u_iv_i$, oriented so that it contains the directed edge $u_i \rightarrow v_i$. The set of cycles $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{2g}\}$ is a basis for the first homology group of Σ [19], which implies that every non-separating cycle in Σ crosses at least one cycle in λ an odd number of times [12, Lemma 3]. The greedy tree-cotree decomposition (T,L,C) can be constructed in $O(n\log n)$ time via Dijkstra's algorithm, after the greedy homology basis Λ can be computed easily in $O(gn)$ time.

Lemma 3.4. *Let λ be any cycle in the greedy homology basis Λ. The shortest cycle γ that crosses λ an odd number of times can be computed in $O(gn\log n)$ time.*

Proof: We can construct the covering space Σ^2_λ in $O(n)$ time, either by pasting together two copies of the cut surface Σ', or by constructing the voltage-induced graph G^2_λ.

Lemmas 3.3 implies that the target cycle γ is the projection of a shortest path from $(s,0)$ to $(s,1)$, for some vertex s of λ. Thus, we could compute γ by computing a shortest-path tree in Σ' at every vertex of $(\lambda,0)$; however, this approach would require $O(n^2\log n)$ time in the worst case. We improve the running time using the multiple-source shortest path algorithm of Cabello *et al.* [6, 7], described in Lemma 2.2.

The cycle λ is obtained by adding an edge uv to a shortest path tree T. Thus, we can write $\lambda = \sigma \cdot (u \rightarrow v) \cdot rev(\tau)$, where $\sigma = \sigma(t,u)$ and $\tau = \sigma(t,v)$ and t is the lowest common ancestor of u and v in T.

Suppose the target cycle γ contains a vertex of σ. Write $\sigma = s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_k$, where $s_1 = t$ and $s_k = u$, and let s_i be the minimum-index vertex of σ that also lies on γ. Then γ is the projection of the shortest path in Σ^2_λ from $(s_i,0)$ to $(s_1,1)$; call this shortest path $\hat\gamma$. If $\hat\gamma$ passes through any other vertex $(s_j,0)$, then $\hat\gamma$ contains the entire shortest path from $(s_i,0)$ to $(s_j,0)$, which is a subpath of $(\sigma,0)$. Thus, $\hat\gamma$ begins with a subpath of $(\sigma,0)$ and is otherwise completely disjoint from $(\sigma,0)$.

Now consider the surface $\Sigma^2_\lambda \,\mbox{✂}\, (\sigma,0)$ obtained by cutting the cyclic double cover along the path $(\sigma,0)$. Let $(\sigma,0)^+$ and $(\sigma,0)^-$ denote the two copies of $(\sigma,0)$ on this surface; these two paths share endpoints $s_1 = t$ and $s_k = u$ but are otherwise disjoint. The previous paragraph implies that if γ and σ intersect, then the lifted path $\hat\gamma$ is the shortest path in $\Sigma^2_\lambda \,\mbox{✂}\, (\sigma,0)$ between some vertex $(s_i,0)^\pm$ and some vertex $(s_i,1)$. Lemma 2.2 implies that we can compute all such shortest paths in $O(gn\log n)$ time.

A similar search of $\Sigma^2_\lambda \,\mbox{✂}\, (\tau,0)$ finds γ if it intersects τ but does not intersect σ. $\square$

Running the previous algorithm once for each cycle $\lambda \in \Lambda$ gives us our main result.

Theorem 3.5. *The shortest non-separating cycle in a directed graph embedded on an orientable surface of genus g can be computed in $O(g^2 n\log n)$ time.*

4. NON-CONTRACTIBLE CYCLES

Our algorithm for computing shortest non-contractible cycles in directed surface graphs directly generalizes of the earlier algorithms of Cabello and Mohar [12] and Kutz [39] for *undirected* surface graphs. Like these earlier algorithms, our algorithm constructs and searches a finite portion of the universal cover of the surface. Cabello and Mohar's observation that the shortest non-contractible cycle crosses any shortest path at most once no longer holds in the directed setting; we use a more subtle crossing condition to limit the size of the universal cover we must search.

As in the previous section, let G be a directed graph with non-negative edge weights, cellularly embedded on an orientable surface Σ of genus g and no boundary. We describe an straightforward extension of our algorithm to surfaces with boundary at the end of this section.

4.1 Signed Crossing Sequences

Fix an arbitrary basepoint x_0 in Σ. A *system of loops* based at x_0 is a set $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{2g}\}$, where each element λ_i is a loop with basepoint x_0, such that $\Sigma \,\mbox{✂}\, \Lambda$ is a topological disk.

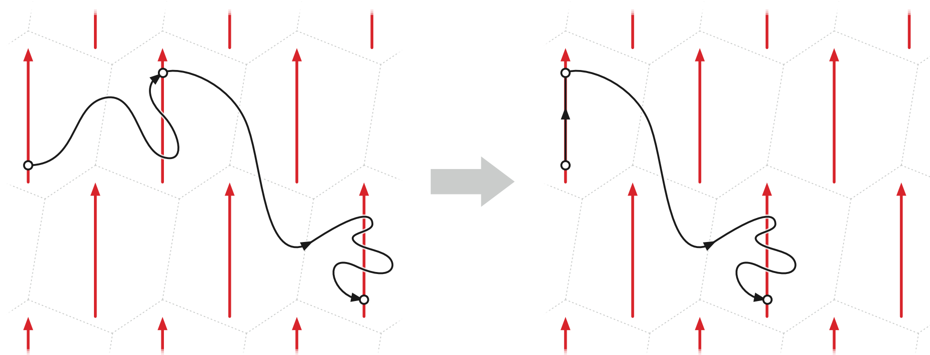

Figure 3. Proof of Lemma 4.6. Left: A lift of an non-contractible loop γ that intersects more than two lifts of a shortest path in the universal cover. Right: A lift of a non-contractible loop that is shorter than γ.

Euler's formula implies that every system of loops has exactly $2g$ elements. Any system of loops is also a basis for the fundamental group $\pi_1(\Sigma, x_0)$ [23]. We refer to the disk $D = \Sigma \setminus\!\!\!\!\times \Lambda$ as a *fundamental domain*. Each loop λ_i appears as two directed paths λ_i^+ and λ_i^- on the boundary of D; in particular, the basepoint x appears as $4g$ different vertices on the boundary of D.

Let Λ be an arbitrary system of loops, and let γ be a loop in Σ with basepoint s.[3] The *signed crossing sequence* $X_\Lambda(\gamma)$ records the orientation of each crossing between γ and the loops in Λ, in order along γ. We take the elements of the signed crossing sequence to be the indices $1, 2, \ldots, 2g$ and their negations; each occurrence of an index i indicates γ crossing λ_i from left to right, and each occurrence of $-i$ indicates γ crossing λ_i from right to left. For technical reasons, if the cycle equivalent to γ crosses some loop in Λ at the basepoint s, we include this crossing at the end of the crossing sequence. Thus, if two loops γ and γ' are equivalent to the same cycle, their crossing sequences differ only by a cyclic shift.

We call two signed crossing sequences $X_\Lambda(\gamma)$ and $X_\Lambda(\gamma')$ *equivalent* if and only if they are generated by homotopic loops γ and γ'. We say that a crossing sequence is *trivial* if it is equivalent to the empty sequence ε and *non-trivial* otherwise. A loop is contractible if and only if its signed crossing sequence is trivial. Thus, equivalence classes of signed crossing sequences correspond bijectively to elements of the fundamental group; in particular, the equivalence class of trivial sequences corresponds to the identity element.

Let $\overline{x}$ denote the signed reversal of a signed crossing sequence x, obtained by writing the symbols of x in reverse order and changing all their signs. Let $x \cdot y$ denote the concatenation of two signed crossing sequences x and y. Finally, let $[x]$ denote the equivalence class of a signed crossing sequence x; in particular, $[\varepsilon]$ is the equivalence class of trivial sequences.

The universal cover $\widetilde{\Sigma}$ is obtained by pasting together an infinite number of copies of the fundamental domain D along corresponding boundary paths $\lambda_i^\pm$. Specifically, we have a copy $(D, [x])$ of the fundamental domain for each equivalence class $[x]$ of signed crossing sequences. For each index i, let $(\lambda_i^+, [x])$ and $(\lambda_i^-, [x])$ denote the copies of λ_i^+ and λ_i^- in $(D, [x])$. The universal cover $\widetilde{\Sigma}$ is defined by identifying the paths $(\lambda_i^+, [x])$ and $(\lambda_i^-, [x \cdot i])$, for every index i and equivalence class $[x]$. Any graph G cellularly embedded in Σ lifts to an infinite graph $\widetilde{G}$ that is cellularly embedded in $\widetilde{\Sigma}$.

The following lemmas follow immediately from our definitions.

Lemma 4.1. *Let γ be any loop in Σ with basepoint s. Then γ is the projection of a unique path in $\widetilde{\Sigma}$ from $(s, [\varepsilon])$ to $(s, [X_\Lambda(\gamma)])$.*

Corollary 4.2. *A loop in Σ is contractible if and only if it lifts to a loop in $\widetilde{\Sigma}$.*

Lemma 4.3. *Every lift of a shortest directed path in G is a shortest directed path in $\widetilde{G}$.*

Lemma 4.4. *Let x be any crossing sequence, let γ be the shortest loop with basepoint s such that $X_\Lambda(\gamma) \in [x]$. Then γ lifts to a shortest path in $\widetilde{\Sigma}$ from $(s, [\varepsilon])$ to $(s, [x])$.*

Corollary 4.5. *The shortest non-contractible cycle γ in Σ lifts to the shortest path in $\widetilde{\Sigma}$ from $(s, [\varepsilon])$ to $(s, [x])$, for some vertex s of γ and some non-trivial crossing sequence x.*

4.2 Algorithm

Following Cabello and Mohar [12] and Kutz [39], our algorithm begins by constructing a *greedy system of loops* as follows. Let (T, L, C) be a tree-cotree decomposition of G, where T is a shortest-path tree rooted at an arbitrary vertex x_0. Arbitrarily label the edges in L as $u_1v_1, u_2v_2, \ldots, u_{2g}v_{2g}$, and for each index i, let λ_i denote the loop $\sigma(x_0, u_i) \cdot (u_i{\to}v_i) \cdot rev(\sigma(v_i, x_0))$. Finally, let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{2g}\}$. It is straightforward to compute Λ in $O(n \log n + gn)$ time via Dijkstra's algorithm.

If G is undirected, Thomassen's 3-path condition implies that the shortest non-separating cycle γ in G crosses any shortest path at most once, which implies that γ crosses each cycle in Λ at most twice [12, Lemma 4]. However, when G is directed, the shortest non-separating cycle could cross any cycle in Λ arbitrarily many times. Our main observation is all but a constant number of these crossings are topologically trivial, in a sense made precise by the following lemma. Again, recall that a *single-period lift* of a cycle γ is any lift of a loop equivalent to γ.

Lemma 4.6. *Let γ be the shortest non-contractible cycle in Σ, and let σ be any shortest path in Σ. Any single-period lift of γ to $\widetilde{\Sigma}$ intersects at most two lifts of σ.*

[3] Formally, we must assume that γ does not intersect the basepoint of Λ. However, this assumption can be guaranteed by a simple modification of the graph G [39].

Proof: Assume σ and γ intersect, since otherwise the lemma is trivial. Write $\sigma = s_1 \to s_2 \to \cdots \to s_k$, and let s_i be the minimum-index vertex that also lies in γ. Consider γ to be a loop based at s_i. Lemmas 4.1 and 4.4 imply that the shortest path $\hat{\gamma}$ from $(s_i, [\varepsilon])$ to $(s_i, [X_\Lambda(\gamma)])$ is a lift of γ.

For the sake of argument, suppose $\hat{\gamma}$ contains a vertex (s_j, y) such that $[y] \neq [\varepsilon]$ and $[y] \neq [X_\Lambda(\gamma)]$. Then we can create a shorter non-contractible loop based at s_i, contradicting the definition of γ, by replacing the subpath of γ from s_i to s_j with the shortest path $\sigma(s_i, s_j)$. Specifically, let ω denote the loop $\sigma(s_i, s_j) \cdot \gamma[s_j, s_i]$. Uniqueness of shortest paths in G implies that $\sigma(s_i, s_j)$ is shorter than $\gamma[s_i, s_j]$, and therefore that ω is shorter than γ. Uniqueness of shortest paths in G also implies that $\sigma(s_i, s_j)$ is a subpath of σ. It follows by Lemma 4.1 that $[X_\Lambda(\omega)] = [\overline{y} \cdot X_\Lambda(\gamma)] \neq [\varepsilon]$. Thus, ω is non-contractible, and we have a contradiction. See Figure 3.

If $\hat{\gamma}$ passes through any other vertex $(s_j, [\varepsilon])$, then $\hat{\gamma}$ contains the entire shortest path from $(s_i, [\varepsilon])$ to $(s_j, [\varepsilon])$, which is a subpath of $(\sigma, [\varepsilon])$. Thus, $\hat{\gamma}$ begins with a subpath of $(\sigma, [\varepsilon])$ and is otherwise disjoint from $(\sigma, [\varepsilon])$.

It follows that γ can be decomposed into two paths $\alpha = \gamma[s_j, s_k]$ and $\beta = \gamma[s_k, s_j]$, where α is disjoint from σ except at its endpoints s_j and s_k, every lift of α joins two different lifts of σ, and every lift of β intersects exactly one lift of σ.

Finally, let u be any vertex of γ, and let $(u, [x])$ be any lift of u. The single-period lift of γ that starts at $(u, [x])$ intersects exactly two lifts of σ if u is a vertex of β, and intersects exactly one lift of σ otherwise. □

Corollary 4.7. *Let Λ be a greedy system of loops in Σ, and let γ be the shortest non-contractible cycle in Σ. Any single-period lift of γ to $\widetilde{\Sigma}$ intersects at most four lifts of any cycle in Λ.*

Call a signed crossing sequence *valid* if it contains at most four instances of any index i or its negation. Every valid crossing sequence has length at most $8g$; thus, there are trivially at most $4g^{8g} = g^{O(g)}$ valid crossing sequences.

Lemma 4.8. *Let Λ be a greedy system of loops in Σ, and let λ be any loop in Λ. The shortest non-contractible cycle in Σ that intersects λ can be computed in $g^{O(g)} n \log n$ time.*

Proof: Let Σ'' denote the finite portion of the universal cover $\widetilde{\Sigma}$ containing each fundamental domain $(D, [x])$ where x is a valid signed crossing sequence, and let G'' denote the corresponding subgraph of $\widetilde{G}$. The surface Σ'' is a topological disk; the graph G'' is planar and has complexity $n'' = g^{O(g)} n$. We can construct Σ'' and G'' in $O(n'') = g^{O(g)} n$ time, by enumerating all valid signed crossing sequences and either pasting copies of the fundamental domain D [12, 39] or using a voltage construction.

Recall that λ is defined by a leftover edge uv from the greedy tree-cotree decomposition. Let $\sigma = \sigma(x_0, u)$ and $\tau = \sigma(v, x_0)$, so that $\lambda = \sigma \cdot (u \to v) \cdot rev(\tau)$. Write $\sigma = s_1 \to s_2 \to \cdots \to s_k$, and let s_i be the minimum-index vertex of σ that also lies on γ. Consider γ to be a loop based at s_i, and let $\hat{\gamma}$ be the lift of γ that starts at $(s_i, [\varepsilon])$. As argued in the proof of Lemma 4.6, γ is the projection of a shortest path $\hat{\gamma}$ that begins with a subpath of $(\sigma, [\varepsilon])$ and is otherwise disjoint from $(\sigma, [\varepsilon])$.

Now consider the annulus $\Sigma'' \,✂\, (\sigma, [\varepsilon])$ obtained by cutting Σ'' along $(\sigma, 0)$. Let $(\sigma, [\varepsilon])^+$ and $(\sigma, [\varepsilon])^-$ denote the two copies of $(\sigma, 0)$ on this annulus. The previous paragraph implies that $\hat{\gamma}$ is also the shortest path in $\Sigma'' \,✂\, (\sigma, [\varepsilon])$ from $(s_i, [\varepsilon])^\pm$ to $(s_i, [x])$, for some vertex s_i of σ. We can compute *all* such shortest paths in $O(n'' \log n'') = g^{O(g)} n \log n$ time using Klein's multiple-source shortest path algorithm [38]; see Lemma 2.1.

A similar search of $\Sigma' \,✂\, (\tau, [\varepsilon])$ finds γ if it intersects τ but does not intersect σ. □

Running the previous algorithm for each loop λ in Λ gives us our final result.

Theorem 4.9. *The shortest non-contractible cycle in a directed graph embedded on an orientable surface of genus g without boundary can be computed in $g^{O(g)} n \log n$ time.*

4.3 Surfaces with Boundary

Finally, we observe that the previous algorithm generalizes easily to surfaces with boundary. The only significant difference is that instead of a system of *loops*, we define crossing sequences with respect to a system of *arcs*: a set of boundary-to-boundary paths that cut the surface Σ into a disk. For a surface of genus g with b boundaries, any system of arcs contains exactly $2g + b - 1$ arcs. A simple variant of the greedy tree-cotree algorithm described in this paper constructs a greedy system of arcs in $O(n \log n + (g + b)n)$ time [13, 16, 17, 22]; each arc in this system is the concatenation of a directed shortest path, a directed edge, and a reversed shortest path. The rest of the algorithm and its analysis is essentially unchanged.

Theorem 4.10. *The shortest non-contractible cycle in a directed graph embedded on an orientable surface of genus g with b boundaries can be computed in $(g + b)^{O(g+b)} n \log n$ time.*

Acknowledgements. Thanks to Yahav Nussbaum for communicating the error in Janiga and Koubek's algorithm [34, 35] and to the anonymous reviewers for their extraordinarily careful reading and suggestions for improvement.

References

[1] G. Borradaile, E. D. Demaine, and S. Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Proc. 26th Int. Symp. Theoretical Aspects Comput. Sci.*, 171–182, 2009. Leibniz Int. Proc. Informatics 3, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ⟨http://drops.dagstuhl.de/opus/volltexte/2009/1835⟩.

[2] G. Borradaile and P. Klein. An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph. *J. ACM* 56(2): 9:1–30, 2009.

[3] G. Borradaile, J. R. Lee, and A. Sidiropoulos. Randomly removing g handles at once. *Proc. 25th Ann. ACM Symp. Comput. Geom.*, 371–376, 2009.

[4] S. Cabello. Many distances in planar graphs. *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1213–1220, 2006.

[5] S. Cabello. Finding shortest contractible and shortest separating cycles in embedded graphs. *ACM Trans. Algorithms* 6(2):24:1–24:18, 2010.

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

[7] S. Cabello, E. W. Chambers, and J. Erickson. Multiple-source shortest paths in embedded graphs. Full version of [6], in preparation.

[8] S. Cabello, É. Colin de Verdière, and F. Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *Proc. 26th Ann. ACM Symp. Comput. Geom.*, 156–165, 2010.

[9] S. Cabello, É. Colin de Verdière, and F. Lazarus. Output-sensitive algorithm for the edge-width of an embedded graph. *Proc. 26th Ann. ACM Symp. Comput. Geom.*, 147–155, 2010.

[10] S. Cabello, É. Colin de Verdière, and F. Lazarus. Finding cycles with topological properties in embedded graphs. Preprint, October 2010. ⟨http://www.di.ens.fr/~colin/textes/09truecycle.pdf⟩.

[11] S. Cabello, M. DeVos, J. Erickson, and B. Mohar. Finding one tight cycle. *ACM Trans. Algorithms* 6(4):article 61, 2010.

[12] S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete Comput. Geom.* 37:213–235, 2007.

[13] E. W. Chambers, É. Colin de Verdière, J. Erickson, F. Lazarus, and K. Whittlesey. Splitting (complicated) surfaces is hard. *Comput. Geom. Theory Appl.* 41(1–2):94–110, 2008.

[14] E. W. Chambers, J. Erickson, and A. Nayyeri. Homology flows, cohomology cuts. *Proc. 42nd Ann. ACM Symp. Theory Comput.*, 273–282, 2009. Full version available at http://www.cs.uiuc.edu/~jeffe/pubs/surflow.html.

[15] E. W. Chambers, J. Erickson, and A. Nayyeri. Minimum cuts and shortest homologous cycles. *Proc. 25th Ann. ACM Symp. Comput. Geom.*, 377–385, 2009.

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

[17] É. Colin de Verdière and J. Erickson. Tightening non-simple paths and cycles on surfaces. *SIAM J. Comput.* 39(8):3784–3813, 2010.

[18] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 278–287, 2007.

[19] D. Eppstein. Dynamic generators of topologically embedded graphs. *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, 599–608, 2003. ArXiv:cs.DS/0207082.

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

[21] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31:37–59, 2004.

[22] J. Erickson and A. Nayyeri. Shortest homologous cycles and minimum cuts via homology covers. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 1166–1176, 2011.

[23] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1038–1046, 2005.

[24] J. Erickson and P. Worah. Computing the shortest essential cycle. *Discrete Comput. Geom.* 44(4):912–930, 2010.

[25] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM J. Comput.* 16(6):1004–1004, 1987.

[26] J. L. Gross and T. W. Tucker. *Topological graph theory*. Dover Publications, 2001.

[27] I. Guskov and Z. Wood. Topological noise removal. *Proc. Graphics Interface*, 19–26, 2001.

[28] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001. ⟨http://www.math.cornell.edu/~hatcher/⟩.

[29] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.* 55(1):3–23, 1997.

[30] P. Indyk and A. Sidiropoulos. Probabilistic embeddings of bounded genus graphs into planar graphs. *Proc. 23rd Ann. ACM Symp. Comput. Geom.*, 204–209, 2007.

[31] A. Itai and Y. Shiloach. Maximum flow in planar networks. *SIAM J. Comput.* 8:135–150, 1979.

[32] G. F. Italiano, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen. Improved minimum cuts and maximum flows in undirected planar graphs. *Proc. 43rd Ann. ACM Symp. Theory Comput.*, to appear, 2011.

[33] G. F. Italiano and P. Sankowski. Improved minimum cuts and maximum flows in undirected planar graphs. Preprint, November 2010. ArXiv:1011.2843.

[34] L. Janiga and V. Koubek. Minimum cut in directed planar networks. *Kybernetika* 28(1):37–49, 1992.

[35] H. Kaplan and Y. Nussbaum. Minimum $s-t$ cut in undirected planar graphs when the source and the sink are close. *Proc. 28th Int. Symp. Theoretical Aspects Comput. Sci.*, 117–128, 2011. Leibniz Int. Proc. Informatics 9, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ⟨http://drops.dagstuhl.de/opus/volltexte/2011/3004⟩.

[36] K. Kawarabayashi and B. Mohar. Graph and map isomorphism and all polyhedral embeddings in linear time. *Proc. 40th Ann. ACM Symp. Theory Comput.*, 471–480, 2008.

[37] K. Kawarabayashi and B. Reed. Computing crossing number in linear time. *Proc. 39th Ann ACM Symp. Theory Comput.*, 382–390, 2007.

[38] P. N. Klein. Multiple-source shortest paths in planar graphs. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 146–155, 2005.

[39] M. Kutz. Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time. *Proc. 22nd Ann. ACM Symp. Comput. Geom.*, 430–438, 2006.

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

[41] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* 7:105–113, 1987.

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

[43] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, 2nd edition. Graduate Texts in Mathematics 72. Springer-Verlag, 1993.

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

[45] K. Weihe. Maximum (s,t)-flows in planar networks in $O(|V| \log |V|)$-time. *J. Comput. Syst. Sci.* 55(3):454–476, 1997.

[46] C. Wolff-Nilsen. Min st-cut of a planar graph in $O(n \log \log n)$ time. Preprint, July 2010. ArXiv:1007.3609.

[47] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder. Removing excess topology from isosurfaces. *ACM Trans. Graphics* 23(2):190–208, 2004.