Tightening Non-Simple Paths and Cycles on Surfaces*

Éric Colin de Verdière[†] Jeff Erickson[‡]

Abstract

We describe algorithms to compute the shortest path homotopic to a given path, or the shortest cycle freely homotopic to a given cycle, on an orientable combinatorial surface. Unlike earlier results, our algorithms do not require the input path or cycle to be simple. Given a surface with complexity n, genus $g \geq 2$, and no boundary, we construct in $O(n^2 \log n)$ time a *tight octagonal decomposition* of the surface—a set of simple cycles, each as short as possible in its free homotopy class, that decompose the surface into a complex of octagons meeting four at a vertex. After the surface is preprocessed, we can compute the shortest path homotopic to a given path of complexity k in $O(gnk)$ time, or the shortest cycle homotopic to a given cycle of complexity k in $O(gnk \log(nk))$ time. A similar algorithm computes shortest homotopic curves on surfaces with boundary or with genus 1. We also prove that the recent algorithms of Colin de Verdière and Lazarus for shortening embedded graphs and sets of cycles have running times polynomial in the complexity of the surface and the input curves, regardless of the surface geometry.

1 Introduction

We consider the following topological version of the shortest path problem in geometric spaces: Given a path or cycle γ on an arbitrary topological surface, find the shortest path or cycle that can be obtained from γ by continuous deformation, keeping the endpoints fixed if γ is a path. Except in very special cases (such as hyperbolic surfaces), local improvement algorithms do not always converge to the true shortest path, but only to a local minimum. A more global approach is required.

Versions of this problem have been studied by several authors during the last decade. Hershberger and Snoeyink [18] find the shortest path or cycle homotopic to a given path or cycle in a triangulated piecewise-linear surface where every vertex lies on the boundary—for example, a triangulated polygon with holes in the plane. Using techniques developed by Cabello et al. [2], Efrat et al. [10] and Bespamyatnikh [1] describe algorithms to find homotopic shortest paths in the plane minus a finite set of points.

Building on our earlier work [4, 5, 6, 13, 14], we formulate the shortest homotopic curve problem in the *combinatorial surface* model. A combinatorial surface is an abstract 2-manifold provided with a weighted embedded graph satisfying certain properties; the input and output curves are required to be walks on this graph. For example, a polyhedral surface where the curves are drawn on its 1-skeleton falls into this model. Vegter and Yap [23] and Lazarus et al. [19] describe how to build canonical polygonal schemata on combinatorial surfaces. Dey and Guha [8] describe an algorithm to determine whether two paths are homotopic in linear time. Several algorithms have been developed recently for computing shortest families of curves with certain topological properties on combinatorial surfaces: Examples include the shortest cut graph [13], the shortest fundamental system of loops [14], and the shortest non-contractible or non-separating cycle [3, 13]. Colin de Verdière and Lazarus [4, 5, 6] describe algorithms to compute the shortest *simple* loop homotopic to a given *simple* loop, or the shortest cycle homotopic to a given *simple* cycle, in time polynomial in the complexity of the surface, the complexity of the input curve, and the ratio between the largest and smallest edge lengths.

In this paper, a curve is *tight* if it is as short as possible in its homotopy class.[1] We describe efficient algorithms to tighten *possibly non-simple* paths and cycles in polynomial time, regardless of the surface geometry.

*See http://www.cs.uiuc.edu/~jeffe/pubs/octagons.html for the most recent version of this paper.

†CNRS, Laboratoire d'informatique de l'École normale supérieure, Paris, France; Eric.Colin.de.Verdiere@ens.fr; http://www.di.ens.fr/~colin/.

‡Department of Computer Science, University of Illinois at Urbana-Champaign; jeffe@cs.uiuc.edu; http://www.cs.uiuc.edu/~jeffe/. Partially supported by NSF under CAREER award CCR-0093348 and grants DMR-012169, CCR-0219594, and DMS-0528086. Portions of this work were done while this author was visiting l'École normale supérieure, LORIA/INRIA Lorraine, Polytechnic University, and Freie Universität Berlin.

[1] Erickson and Whittlesey [14] define a cycle to be tight if it contains the global shortest path between any two of its points. Our notion of tightness is weaker.

As in earlier papers [4, 5, 6, 8], we decompose the surface with a set of (tight) curves C such that (a) the homotopy class of a given curve is encoded by the way it crosses the curves in C, and (b) the way the output curve crosses C depends only on the way the input curve crosses C. In the case of a surface with boundary, we can employ a so-called *tight system of arcs*, generalizing the greedy system of loops constructed by Erickson and Whittlesey [14]. Surfaces without boundary are considerably more difficult; for this case, we introduce the notion of *tight regular decomposition* of the surface: an arrangement of tight cycles where every vertex of the arrangement has degree four and every face is a disk with the same number of sides. The efficiency of our algorithm essentially follows from classical results in combinatorial group theory and hyperbolic geometry. Our results also imply that the algorithms of Colin de Verdière and Lazarus [4, 5, 6] for shortening simple curves run in polynomial time.

Due to space constraints, this extended abstract focuses on our algorithm for tightening *paths* on surfaces *without* boundary. We will only briefly discuss our algorithm for tightening cycles, our algorithms for surfaces with boundary, and our improved analysis of the algorithms of Colin de Verdière and Lazarus [4, 5, 6].

2 Background

2.1 Topology. We begin by recalling several standard definitions from manifold topology. Further background can be found in textbooks by Hatcher [16] and Stillwell [22].

A *surface* (or 2-manifold with boundary) $\mathcal{M}$ is a topological Hausdorff space where each point has a neighborhood homeomorphic to either the plane or the closed half-plane. The points without neighborhood homeomorphic to the plane comprise the *boundary* of $\mathcal{M}$. A (g, b)-*surface* is any surface homeomorphic to a sphere with g handles attached and b disks removed. Every compact, connected, orientable surface $\mathcal{M}$ is a (g, b)-surface for unique integers g (its *genus*) and b (its *number of boundaries*). A *sphere* is a $(0, 0)$-surface; a *disk* is a $(0, 1)$-surface; an *annulus* is a $(0, 2)$-surface; a *pair of pants* is a $(0, 3)$-surface; a *torus* is a $(1, 0)$-surface.

We distinguish between four different types of *curves*. A *path* on a surface $\mathcal{M}$ is (the image of) a continuous map $p : [0, 1] \to \mathcal{M}$; its *endpoints* are $p(0)$ and $p(1)$. A *loop* is a path p whose endpoints coincide. An *arc* is a path intersecting the boundary of a surface exactly at its endpoints. A cycle is (the image of) a continuous map $\gamma : S^1 \to \mathcal{M}$ where $S^1 = \mathbb{R}/\mathbb{Z}$ is the standard circle. A curve is *simple* if does not self-intersect (except, for a loop, at its basepoint).

Two paths p and p' are *homotopic* if there is a continuous map $h : [0, 1] \times [0, 1] \to \mathcal{M}$ such that $h(0, t) = p(t)$ and $h(1, t) = p'(t)$ for all t, and $h(\cdot, 0)$ and $h(\cdot, 1)$ are constant maps. Two cycles γ and γ' are *(freely) homotopic* if there is a continuous map $h : [0, 1] \times S^1 \to \mathcal{M}$ such that $h(0, t) = \gamma(t)$ and $h(1, t) = \gamma'(t)$ for all t. A loop or cycle is *contractible* if it is homotopic to a constant loop or cycle; an arc is contractible if it is homotopic to a boundary path. A simple loop, arc, or cycle is *separating* if $\mathcal{M}$ minus (the image of) this curve is disconnected. In particular, every simple contractible curve is separating. Any cycle homotopic to the boundaries of an annulus is called a *generating cycle*.

A map $\pi : \mathcal{M}' \to \mathcal{M}$ between two surfaces is called a *covering map* if each point $x \in \mathcal{M}$ 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 $\mathcal{M}'$ to $\mathcal{M}$, we call $\mathcal{M}'$ a *covering space* of $\mathcal{M}$.

If p is a path in $\mathcal{M}$ and $\pi(x') = p(0)$ for some point $x' \in \mathcal{M}'$, there is a unique path p' in $\mathcal{M}'$ such that $p'(0) = x'$ and $\pi \circ p' = p$. This path p' is called a *lift* of p. Two paths are homotopic in $\mathcal{M}$ if and only if they have homotopic lifts in $\mathcal{M}'$. Similarly, a *lift* of a cycle γ is either a cycle γ' on $\mathcal{M}'$ such that $\pi \circ \gamma' = \gamma$, or a continuous 'open arc' $p' : \mathbb{R} \to \mathcal{M}$ such that $\pi(p'(t)) = \gamma(t \bmod 1)$ for all t. Any lift of a contractible cycle is itself a contractible cycle. Every surface $\mathcal{M}$ has a unique covering space $\widetilde{\mathcal{M}}$ in which every cycle is contractible, called the *universal cover* of $\mathcal{M}$.

All surfaces considered in this paper are connected, compact, and orientable, although their covering spaces are of course not necessarily compact.

2.2 Combinatorial and Cross-Metric Surfaces. A *combinatorial surface* is an abstract surface $\mathcal{M}$ together with a weighted undirected graph $G(\mathcal{M})$, embedded on $\mathcal{M}$ so that each open face is a disk and every boundary is a simple circuit in $G(\mathcal{M})$. (We will simply write G if the surface $\mathcal{M}$ is clear from context.) In this model, the only allowed paths are walks in G; the length of a path is the sum of the weights of the edges traversed by the path, counted with multiplicity. The *complexity* of a combinatorial surface is the total number of vertices, edges, and faces of G.

Most of our results are developed in an equivalent dual formulation of this model. A *cross-metric* surface is an abstract surface $\mathcal{M}$ together with an undirected weighted graph $G^* = G^*(\mathcal{M})$, embedded so that every open face is a disk and every boundary of $\mathcal{M}$ is a simple circuit in G^*. We consider only *regular* paths and cycles on $\mathcal{M}$, which intersect the edges of G^* only transversely and away from the vertices. The *length* of a regular curve p is defined to be the sum of the weights of the dual edges that p *crosses*, counted with multiplicity. To emphasize this usage, we refer to the weight of a dual edge as its *crossing weight*. The crossing weight of any boundary edge is ∞, since no path can cross the boundary.

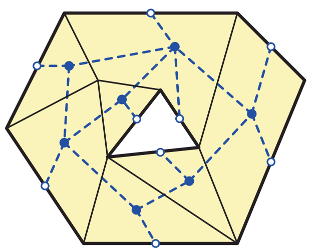

Figure 1. Primal (solid) and dual (dashed) graphs on a combinatorial annulus.

The equivalence of these two models is easy to establish. Given a combinatorial surface $(\mathcal{M}, G)$, we construct the *dual graph* G^* as follows. G^* has a vertex f^* in the interior of each face f of G, and a vertex $\bar{e}^*$ in the relative interior of each boundary edge e of G. For each non-boundary edge e of G separating faces f_1 and f_2, there is a dual edge e^* between f_1^* and f_2^*; and for each boundary edge e with incident face f, there is a dual edge e^* between f^* and $\bar{e}^*$. Each dual edge e^* in G^* intersects only its corresponding primal edge e in G. Each pair of consecutive vertices of G^* on the boundary of $\mathcal{M}$ is connected by an edge of G^* on the boundary of $\mathcal{M}$. Each interior edge e^* has the same weight as the corresponding primal edge e, and each boundary edge of G^* has infinite weight. The faces of G^* correspond to the vertices of G. Any walk in G is regular with respect to G^*, and the two notions of length coincide. Conversely, any curve that is regular with respect to G^* is homotopic to a walk in G of the same length. We can easily construct shortest paths on a cross-metric surface (using Dijkstra's algorithm, for example) by restating the usual algorithms on G in terms of the dual graph G^*.

We can represent an arbitrary arrangement of possibly (self-)intersecting curves on a cross-metric surface $\mathcal{M}$ by modifying the underlying graph G^* inductively as follows. The initial empty arrangement is just the graph G^*. We embed each new curve *generically*: every crossing point of the new curve and the existing arrangement, and every self-crossing of the new curve, creates a vertex of degree four; endpoints of arcs become vertices of degree three; and endpoints of other paths become vertices of degree one. Whenever we split an edge e^* of G^*, we give both sub-edges the same crossing weight as e^*. Each segment of the curve between two intersection points becomes a new edge, whose crossing weight is a fixed *formal infinitesimal* $\varepsilon > 0$.[2] This ensures that later shortest-path computations always prefer paths with fewer crossings when the lengths (with respect to the original G^*) are equal. These modifications change the length of any regular curve in $\mathcal{M}$ by at most a multiple of ε; in particular, any path that is tight with respect to the refined graph is tight with respect to the original graph. The *multiplicity* of a set of curves in $\mathcal{M}$ is the maximum, over all edges e^* of $G^*(\mathcal{M})$, of the number of times e^* is crossed by the curves.

Our algorithms sometimes cut a cross-metric surface $\mathcal{M}$ along some embedded curve α. The resulting cross-metric surface $\mathcal{M} \setminus \alpha$ can be represented simply by assigning infinite crossing weight to the edges that comprise α, indicating that these edges cannot be crossed by other curves. We also sometimes need to glue surfaces together along a common boundary; again, our representation easily supports this operation. Finally, for any (even infinite) covering space $\mathcal{M}'$ of $\mathcal{M}$, the graph $G^*(\mathcal{M}')$ is simply a lift of $G^*(\mathcal{M})$.

We emphasize that combinatorial and cross-metric surfaces do not have any 'geometry' in the usual sense; only the combinatorial structure is important. Eppstein's *gem representation* [11] is a convenient data structure for maintaining this structure.

3 Tight Curves

Our algorithms rely on the two following technical lemmas, whose proofs are omitted from this extended

[2] Equivalently, we can take the crossing weight of any edge in G^* to be a vector $(\ell, 0)$ for some non-negative real number ℓ, and the crossing weight of any edge of any curve embedded on $\mathcal{M}$ to be $(0, 1)$. Crossing weights are now vectors, which are added normally and compared lexicographically.

abstract. A path p *wraps around* a cycle γ if $p(t) = \gamma((at + b) \bmod 1)$ for some real numbers a and b.

Lemma 3.1. *Any path that wraps around a tight cycle is tight.*

Lemma 3.2. *Let α be a simple tight arc or cycle in a cross-metric surface $\mathcal{M}$, and let β be a simple path or cycle disjoint from α. Then β is tight in $\mathcal{M} \setminus \alpha$ if and only if β is also tight in $\mathcal{M}$.*

We introduce five elementary constructions of tight arcs and cycles on a cross-metric surface.

Lemma 3.3. *Let $\mathcal{M}$ be a cross-metric surface with complexity n, genus g, and b boundary components.*

(a) If $b \geq 2$, we can compute in $O(n \log n)$ time a tight simple arc joining two specified boundaries, with multiplicity one, and with multiplicity zero at each edge adjacent to the two specified boundaries.

(b) If $b = 1$ and $g \neq 0$, we can compute in $O(n \log n)$ time a tight simple non-contractible or non-separating arc with multiplicity at most two, and with multiplicity zero at each edge adjacent to the boundary.

(c) If $b = 0$, we can compute in $O(n^2 \log n)$ time a tight simple non-separating cycle with multiplicity one.

(d) If $g = 0$ and $b = 2$, we can compute in $O(n \log n)$ time a tight simple generating cycle with multiplicity one.

(e) If $g = 0$ and $b = 3$, we can compute in $O(n \log n)$ time a tight simple cycle with multiplicity at most two that is homotopic to a chosen boundary cycle.

Proof: (a) Shrink the two specified boundaries to points a and b and compute a shortest path from a to b using Dijkstra's algorithm. This path corresponds to the desired arc on $\mathcal{M}$.

(b) Shrink the boundary to a point p, and compute a shortest non-contractible or non-separating loop with basepoint p, using an algorithm of Erickson and Har-Peled [13, Lemma 5.4]. This loop corresponds to the desired arc on $\mathcal{M}$.

(c) Compute a shortest non-separating cycle, using an algorithm of Erickson and Har-Peled [13, Lemma 5.4].

(d) Compute a minimum cut in $G^*(\mathcal{M})$ that separates the two boundary cycles [20, Propositions 1 and 2]. Since $G^*(\mathcal{M})$ is a planar graph, we can compute such a cut in $O(n \log n)$ time using an algorithm of Frederickson [15, Theorem 7]. Because we can separate the surface by cutting each edge in the cut once, the cycle has multiplicity one.

(e) Colin de Verdière and Lazarus describe an algorithm for this problem [5]. Let δ_1, δ_2, and δ_3 denote the boundaries of $\mathcal{M}$, and suppose we want a cycle homotopic to δ_1. Compute a shortest arc α between δ_2 and δ_3 (by part (a)) and compute the shortest generating cycle γ in the annulus $\mathcal{M} \setminus \alpha$ (by part (d)). Lemma 3.2 implies that γ is tight in $\mathcal{M}$. □

Our algorithms combine these elementary constructions by computing a tight simple arc or cycle on $\mathcal{M}$, cutting $\mathcal{M}$ along it, and iterating on the resulting surface. Lemma 3.2 implies that the arcs and cycles computed in this fashion are always tight *on the original surface* $\mathcal{M}$. Until the end of the proof of Theorem 4.1, *all the curves considered are simple, tight arcs or cycles* whose tightness in $\mathcal{M}$ follows from Lemma 3.2.

We can ensure that *the tight curves we construct intersect each other as few times as possible* using the perturbation scheme already described: The crossing weight of the edges of each created curve is a fixed formal infinitesimal ε. Thus, the number of curve crossings is used only as a tie-breaking measure when two curves have the same length with respect to G^*.

The following technical lemma, whose proof is omitted, will be used to prove that our later algorithms construct only cycles with constant multiplicity.

Lemma 3.4. *Any tight cycle homotopic to a boundary of a cross-metric surface has multiplicity at most two.*

4 Tight Octagonal Decompositions

In this section, we describe the preprocessing phase of our path- and cycle-shortening algorithm. A *tight octagonal decomposition* of a surface is an arrangement of tight simple cycles, each with constant multiplicity, in which every vertex has degree four and every face has eight sides. See Figure 2.

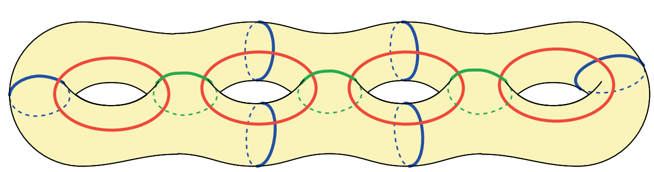

Figure 2. An octagonal decomposition built by our algorithm.

The universal cover of a tight octagonal decomposition is combinatorially isomorphic to a tiling of the hyperbolic plane by regular right-angled octagons. Thus, our decomposition imposes a crude regular hyperbolic structure on any combinatorial surface, thereby allowing us to exploit classical results in hyperbolic geometry and combinatorial group theory, primarily in Sections 4.3 and 5.1.

4.1 Construction.

Theorem 4.1. *Let $\mathcal{M}$ be a cross-metric surface with complexity n, genus $g \geq 2$, and no boundary. In $O(n^2 \log n)$ time, we can construct a tight octagonal decomposition of $\mathcal{M}$.*

Proof: Our construction algorithm has four phases.

Phase 1: Unzipping. We begin by 'unzipping' the surface into a disk using one tight cycle and $2g - 1$ tight arcs. Let τ_1 be the shortest non-separating cycle in $\mathcal{M}$; let β_1 be the shortest arc βetween the two boundary components of $\mathcal{M} \setminus \tau_1$; and let $\mathcal{M}_1 = \mathcal{M} \setminus (\tau_1 \cup \beta_1)$. For each i from 1 to $g - 1$, let α_{i+1} be the shortest non-separating αrc in $\mathcal{M}_i$; let β_{i+1} be the shortest arc βetween the two boundaries of $\mathcal{M}_i \setminus \alpha_{i+1}$; and let $\mathcal{M}_{i+1} = \mathcal{M}_i \setminus (\alpha_{i+1} \cup \beta_{i+1})$. For each i, the surface $\mathcal{M}_i$ has genus $g - i$ and one boundary cycle. See Figure 3(a). Lemma 3.3 implies that the *union* of the cycle τ_1 and the arcs α_i and β_i has multiplicity at most two on $\mathcal{M}$ (since each curve can have multiplicity two, but no arc intersects edges of G^* crossed by the cycle and the arcs created before) and that we can compute these curves in $O(n^2 \log n)$ total time.

Phase 2: Pants decomposition. Next, we use the arcs in the previous phase to help construct a set of $3g - 3$ tight simple cycles, each with multiplicity $O(1)$, that decompose $\mathcal{M}$ into $2g - 2$ pairs of pants (Figure 3).

Let τ_g denote the shortest generating cycle in the annulus $\mathcal{M}_{g-1} \setminus \alpha_g$. Let σ_{g-1} be the shortest cycle in $\mathcal{M}_{g-1} \setminus \tau_g$ homotopic to the boundary of $\mathcal{M}_{g-1}$. For each i from $g - 1$ down to 2, let τ_i^+ be the shortest cycle in $\mathcal{M}_{i-1} \setminus (\alpha_i \cup \sigma_i)$ homotopic to a boundary of $\mathcal{M}_{i-1} \setminus \alpha_i$; let τ_i^- be the shortest cycle, in the component of $\mathcal{M}_{i-1} \setminus (\alpha_i \cup \sigma_i \cup \tau_i^+)$ that is a pair of pants, homotopic to a boundary of $\mathcal{M}_{i-1} \setminus \alpha_i$; and let σ_{i-1} be the shortest cycle in $\mathcal{M}_{i-1} \setminus (\tau_i^+ \cup \tau_i^-)$ homotopic to the boundary of $\mathcal{M}_{i-1}$. Recall that τ_1 is our original starting cycle.

In each case, we are computing the shortest cycle homotopic to a boundary of a pair of pants. For each i, the boundaries of $\mathcal{M}_i$ and $\mathcal{M}_{i-1} \setminus \alpha_i$ have multiplicity at most 4 in $\mathcal{M}$; since (by Lemma 3.2) each cycle σ_i and $\tau_i^\pm$ is a shortest cycle homotopic, on $\mathcal{M}$, to a boundary of $\mathcal{M}$ cut along τ_1 and some α_i's and β_i's, Lemma 3.4 implies that each such cycle has multiplicity at most 8 in $\mathcal{M}$; thus each of them can be computed in $O(n \log n)$ time by Lemma 3.3(e).

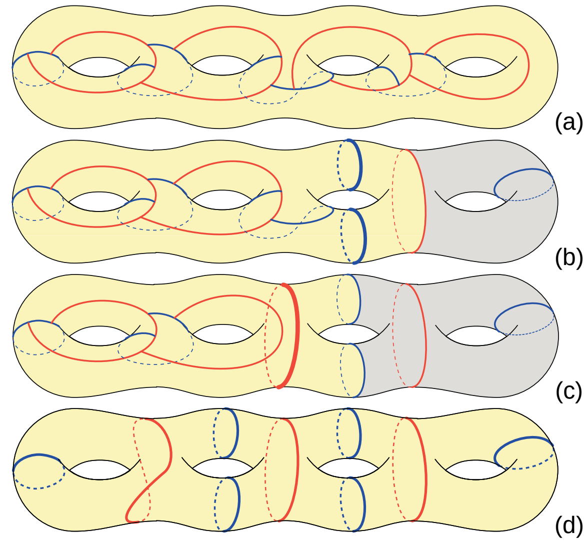

Figure 3. (a) The surface $\mathcal{M}$ unzipped. (b) Computing τ_3^+ and τ_3^-. (c) Computing σ_2. (d) The final pants decomposition.

The $3g - 3$ cycles $\tau_1, \sigma_1, \tau_2^+, \tau_2^-, \sigma_2, \ldots, \tau_{g-1}^+, \tau_{g-1}^-, \sigma_{g-1}, \tau_g$ split $\mathcal{M}$ into $2g - 2$ pairs of pants. Specifically, the cycles σ_i partition $\mathcal{M}$ into a chain of punctured tori $T_1 \cup T_2 \cup \cdots \cup T_g$, where T_1 and T_g each have one boundary (σ_1 and σ_{g-1}, respectively), and every other T_i has two boundaries (σ_{i-1} and σ_i). The first torus T_1 (resp. the last torus T_g) is cut into a pair of pants by τ_1 (resp. τ_g), and each intermediate torus T_i is cut into two pairs of pants by the cycles τ_i^+ and τ_i^-.

Phase 3: Around the holes. In the next phase, we find tight simple cycles that go 'around the hole' of each punctured torus T_i, crossing the cycle(s) $\tau_i^\pm$ exactly once.

First consider the torus T_1. Let α be the shortest non-contractible arc in $T_1 \setminus \tau_1$ with both endpoints on the boundary σ_1, and let β be the shortest non-contractible arc in $T_1 \setminus \alpha$. Finally, let ϕ_1 be the shortest generating cycle in the annulus $T \setminus \beta$. Because τ_1 is homotopic to the boundary of $T_1 \setminus \alpha$, the arc β crosses τ_1 exactly once, so ϕ_1 also crosses τ_1 exactly once. See Figure 4. Since τ_1 and σ_1 each have constant multiplicity, so does ϕ_1.

A symmetric construction finds a tight cycle ϕ_g in the torus T_g that crosses τ_g exactly once.

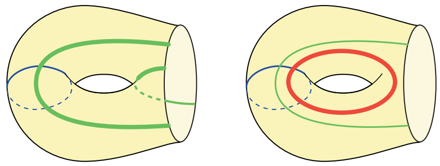

Figure 4. Left: α and β. Right: ϕ_1.

Now, for some $2 \leq i \leq g-1$, consider the torus T_i, whose boundary consists of σ_{i-1} and σ_i. Let α^- be the shortest non-contractible arc with endpoints on σ_{i-1} in $T_i \setminus (\tau_i^+ \cup \tau_i^-)$. Similarly, let α^+ be the shortest non-contractible arc with endpoints on σ_i in $T_i \setminus (\tau_i^+ \cup \tau_i^-)$. These arcs α^- and α^+ split T_i into two annuli, one containing τ_i^+ and the other τ_i^-. Let β^+ and β^- be shortest arcs, one on each of these annuli, joining a point of σ_{i-1} and a point of σ_i. The arc β_i^+ crosses τ_i^+ once and does not cross τ_i^-; symmetrically, β_i^- crosses τ_i^- once and does not cross τ_i^+. Finally, let ϕ_i be the shortest generating cycle in the annulus $T_i \backslash (\beta^+ \cup \beta^-)$; this cycle crosses τ_i^+ and τ_i^- each exactly once. See Figure 5. Since $\tau_i^\pm$ and σ_i each have constant multiplicity, so does ϕ_i.

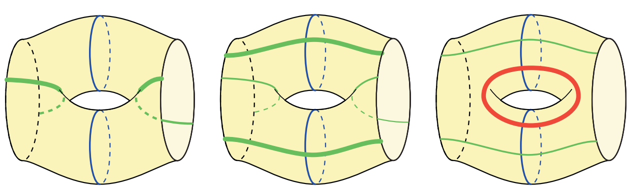

Figure 5. Left: α^+ and α^-. Middle: β^+ and β^-. Right: ϕ_i.

Lemma 3.3 implies that each curve ϕ_i is computed in time $O(n_i \log n_i)$, where n_i denotes the complexity of T_i. Since $\sum_i n_i = O(n)$, the overall running time of this phase is $O(n \log n)$.

Phase 4: Around the handles. Finally, for each i between 1 and $g-1$, let M_i be the pair of 'monkey pants' formed by gluing together the two pairs of pants with σ_i as their common boundary. The boundaries of M_i are $\tau_i^+, \tau_i^-, \tau_{i+1}^+$, and τ_{i+1}^-. (Here τ_1^+ and τ_1^- denote the two copies of τ_1 in M_1, and τ_g^+ and τ_g^- denote the two copies of τ_g in M_{g-1}.) Let β^+ be the shortest arc in $M_i \setminus (\phi_i \cup \phi_{i+1})$ between τ_i^+ and τ_{i+1}^+, and let β^- be the shortest arc in $M_i \setminus (\phi_i \cup \phi_{i+1} \cup \beta^+)$ between τ_i^- and τ_{i+1}^-. The arcs β^+ and β^- are disjoint and cross σ_i exactly once. Finally, let θ_i be the shortest generating cycle in the annulus $M_i \setminus (\beta^+ \cup \beta^-)$. We easily verify that θ_i crosses σ_i exactly twice and ϕ_i and ϕ_{i+1} each exactly once. See Figure 6. Finally, since all the earlier cycles have constant multiplicity, so does θ_i. As in the previous phase, Lemma 3.3 implies that this phase of the algorithm runs in $O(n \log n)$ time.

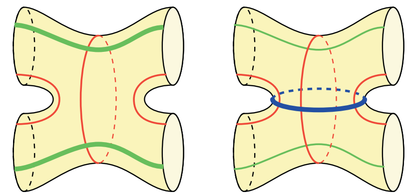

Figure 6. Left: β^+ and β^-. Right: θ_i.

To summarize, the tight simple cycles $\tau_i^\pm$, ϕ_i, and θ_i decompose the surface $\mathcal{M}$ into octagons exactly as shown in Figure 2. Lemma 3.2 implies that each of these cycles is tight in $\mathcal{M}$. □

If we do not discard the cycles σ_i, we obtain a tight *hexagonal* decomposition of $\mathcal{M}$. Our remaining results use an octagonal decomposition, but this hexagonal decomposition could be used as well, applying exactly the same arguments.

4.2 Limiting Crossings. In the actual curve-shortening algorithm, we need to bound the number of times an input curve crosses the cycles in our octagonal decomposition. To that end, we will actually construct our decomposition on a refinement of the input surface $\mathcal{M}$.

Let $G^+ = G^+(\mathcal{M})$ be the graph obtained by overlaying the primal graph $G(\mathcal{M})$ and the dual graph $G^*(\mathcal{M})$. The vertices of G^+ are either vertices of G, vertices of G^*, or intersections between an edge e of G and its dual edge e^* in G^*. Each edge of G and dual edge in G^* is partitioned into two edges in G^+. Finally, each face of G^+ is a quadrilateral.

To treat $\mathcal{M}$ as a cross-metric surface with 'dual' graph G^+, we assign a crossing weight to each edge e^+ of G^+ as follows. If e^+ is on the boundary of $\mathcal{M}$, it has crossing weight ∞. If e^+ is contained in a dual edge e^* or $\bar{e}^*$, it has the same crossing weight as that dual edge. If e^+ is contained in an edge of G, its crossing weight is a fixed formal infinitesimal $\varepsilon' < \varepsilon$. Any curves that are tight with respect to G^+ are also tight with respect to G^*. Among all tight curves that cross each other as few times as possible, our algorithms choose curves that cross the edges of G as few times as possible.

We actually apply Theorem 4.1 in this augmented cross-metric surface. In $O(n^2 \log n)$ time, we obtain an octagonal decomposition $\mathcal{O}$ of $\mathcal{M}$ where each cycle is tight, each edge of G^* is crossed $O(1)$ times by each cycle in $\mathcal{O}$, and each edge of G is crossed $O(1)$ times by each cycle in $\mathcal{O}$. In particular, any walk in G of length k crosses the cycles in $\mathcal{O}$ a total of $O(gk)$ times.

4.3 The Universal Cover. Let $\mathcal{O}$ be a tight octagonal decomposition of a surface $\mathcal{M}$ without boundary. As we mentioned earlier, the universal cover of this decomposition is isomorphic to the regular tiling of the hyperbolic disk by right-angled octagons; see Figure 7. This regular tiling can also be viewed as an infinite arrangement of hyperbolic lines. Building on this intuition, we call any lift of a cycle in $\mathcal{O}$ to the universal cover $\widetilde{\mathcal{M}}$ a *line*. The set of lines is denoted by $\widetilde{\mathcal{O}}$. This terminology is further motivated by Lemmas 4.2 and 4.5.

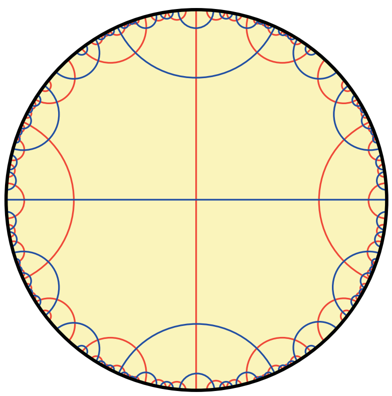

Figure 7. Universal cover of a tight octagonal decomposition.

Lemma 4.2. *No two lines in $\widetilde{\mathcal{O}}$ cross more than once.*

Proof (sketch): Let $\tilde{\alpha}$ and $\tilde{\beta}$ be arbitrary lines in $\widetilde{\mathcal{O}}$, obtained by lifting two intersecting cycles α and β in $\mathcal{O}$. (Any two lifts of disjoint cycles, or of the same cycle, are disjoint, so this is the only interesting case.) Cycles α and β intersect exactly once; the lines $\tilde{\alpha}$ and $\tilde{\beta}$ intersect only at lifts of $\alpha \cap \beta$. If $\tilde{\alpha}$ and $\tilde{\beta}$ intersect more than once, the segments of $\tilde{\alpha}$ and $\tilde{\beta}$ between any two intersection points are homotopic paths. Projecting these paths back down to $\mathcal{M}$ gives us two homotopic loops based at $\alpha \cap \beta$, one wrapping around α and the other wrapping around β. But this implies that α and β are homotopic, which is impossible. □

Lemma 4.3 (Dehn [7], [22, p. 188]). *Let S be the bounded union of at least two octagons in the regular right-angled octagon tiling. At least two octagons in S have five consecutive sides on the boundary of S.*

Lemma 4.4. *Any union of N octagons in the regular right-angled octagon tiling has perimeter at least $2N+6$.*

Proof: Removing an octagon with at least five consecutive sides on the boundary of the union reduces the perimeter by at least two. The base case is a single octagon. □

Lemma 4.5. *Between any two points in $\widetilde{\mathcal{M}}$, there is a shortest path that crosses each line in $\widetilde{\mathcal{O}}$ at most once.*

Proof: Fix two points w and z in $\widetilde{\mathcal{M}}$, and let $\tilde{p}$ be a shortest path from w to z that intersects the lines $\widetilde{\mathcal{O}}$ as few times as possible. If $\tilde{p}$ crosses some line ℓ twice, at points x and y, then $\tilde{p}$ and ℓ form a bigon. Since ℓ is a lift of a tight cycle, every subpath of ℓ is a shortest path, by Lemma 3.1. Thus, we can remove the bigon from $\tilde{p}$ by replacing the subpath from x to y with the shortest path in ℓ. Since any pair of lines in $\widetilde{\mathcal{O}}$ intersect at most once, this exchange results in a path with fewer line crossings (and possibly shorter length), which is impossible. □

5 Shortening Paths and Cycles

In this section, we describe how to compute a tight path or cycle homotopic to a given path or cycle in polynomial time. Our algorithm is much faster than previous results [4, 5, 6], and unlike those results, our algorithm does not require the input curve to be simple. We focus on surfaces of genus at least two, without boundaries; this is the most difficult case.

Consider an arbitrary path p on a surface $\mathcal{M}$. Let $\widetilde{p}$ be a lift of p to the universal cover $\widetilde{\mathcal{M}}$, and let $\widetilde{p}'$ be a shortest path in $\widetilde{\mathcal{M}}$ between the endpoints of $\widetilde{p}$. Projecting $\widetilde{p}'$ back down to $\mathcal{M}$ gives us a shortest path homotopic to p. Our algorithm exploits this characterization by constructing a subset of $\widetilde{\mathcal{M}}$ of small complexity that contains both $\widetilde{p}$ and some shortest path $\widetilde{p}'$. Compared to previous approaches [9, 21], the construction of this part of the universal cover is very simple once we have computed an octagonal decomposition of the surface.

5.1 Building the Relevant Region. Let $\mathcal{O}$ be a tight octagonal decomposition of a surface $\mathcal{M}$ with genus $g \geq 2$ and no boundary. Consider a path p in $\mathcal{M}$, and let $\widetilde{p}$ be a lift of p to the universal cover $\widetilde{\mathcal{M}}$. For any line ℓ in $\widetilde{\mathcal{O}}$, let ℓ^+ denote the component of $\widetilde{\mathcal{M}} \setminus \ell$ that contains the starting point $\widetilde{p}(0)$.

Let $\ell_1, \ell_2, \ldots, \ell_z$ be the sequence of lines in $\widetilde{\mathcal{O}}$ crossed by $\widetilde{p}$, in order of their first crossing. Let $\mathcal{L}_0 = \varnothing$, and for any positive integer i, let $\mathcal{L}_i = \mathcal{L}_{i-1} \cup \{\ell_i\}$. For each i between 0 and z, let $\widetilde{\mathcal{M}}_i$ be the subset of $\widetilde{\mathcal{M}}$ reachable from $\widetilde{p}(0)$ by crossing only (a subset of) lines in $\mathcal{L}_i$, in any order. Combinatorially, the region

$\widetilde{\mathcal{M}}_i$ is a convex polygon formed by intersecting the half-planes ℓ^+ for all lines ℓ not in the set $\mathcal{L}_i$. By Lemma 4.5, and since each line is separating, there exists a shortest path $\widetilde{p}'$ between the endpoints of $\widetilde{p}$ that crosses only a subset of the lines that $\widetilde{p}$ crosses; so $\widetilde{p}'$ is contained in $\widetilde{\mathcal{M}}_z$. For this reason, $\widetilde{\mathcal{M}}_z$ is called the *relevant region* of $\widetilde{\mathcal{M}}$ (with respect to $\widetilde{p}$).

Lemma 5.1. *For all $i \geq 1$, $\ell_i \cap \widetilde{\mathcal{M}}_{i-1}$ is a connected subset of the boundary of $\widetilde{\mathcal{M}}_{i-1}$.*

Proof: Since $\widetilde{\mathcal{M}} \setminus \ell_i$ is disconnected, $\ell_i \cap \widetilde{\mathcal{M}}_{i-1}$ is a subset of $\partial\widetilde{\mathcal{M}}_{i-1}$. Let $\ell_i[x, y]$ be the segment of ℓ_i between two points x and y in $\ell_i \cap \widetilde{\mathcal{M}}_{i-1}$, and suppose some line ℓ crosses $\ell_i[x, y]$. Since two lines cross at most once, ℓ intersects the interior of $\widetilde{\mathcal{M}}_{i-1}$, so ℓ must be in the set $\mathcal{L}_{i-1}$. It follows that the entire segment $\ell_i[x, y]$ is a subset of $\partial\widetilde{\mathcal{M}}_{i-1}$. □

Lemma 5.1 implies that ℓ_i intersects $\partial\widetilde{\mathcal{M}}_{i-1}$ along a connected set of octagons $O_1, O_2, \ldots, O_u$. For each k between 1 and u, let O'_k be the reflection of O_k across ℓ_i. See Figure 8. The octagons O'_k do not belong to $\widetilde{\mathcal{M}}_{i-1}$.

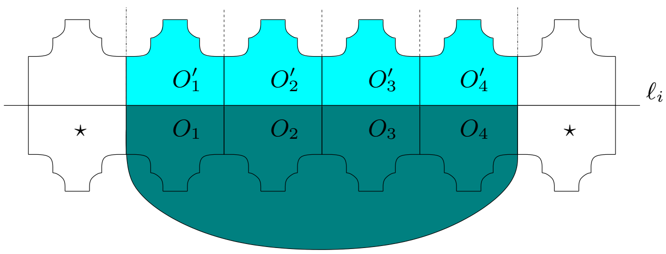

Figure 8. From $\widetilde{\mathcal{M}}_{i-1}$ (dark shaded) to $\widetilde{\mathcal{M}}_i$ (all shaded).

Lemma 5.2. $\widetilde{\mathcal{M}}_i = \widetilde{\mathcal{M}}_{i-1} \cup O'_1 \cup \cdots \cup O'_u$.

Proof: Let $\widetilde{\mathcal{N}} = O'_1 \cup \cdots \cup O'_u$ (the lightly shaded region in Figure 8). To prove the lemma, it suffices to show that none of the lines bounding $\widetilde{\mathcal{N}}$ are in the set $\mathcal{L} = \{\ell_1, \ldots, \ell_{i-1}\}$. Obviously ℓ_i is not in this set. Each octagon O'_k is bounded by eight lines: ℓ_i, two *inner* lines that cross ℓ_i at a vertex of O'_k, and five *outer* lines. If some outer line intersected ℓ_i, then we could construct a disk in the tiling $\widetilde{\mathcal{O}}$ bounded by at most five lines, which would contradict Lemma 4.3. Since every line in $\mathcal{L}$ has a point in ℓ_i^+, no outer line can be in $\mathcal{L}$. Only the first and last inner lines contribute a side to the boundary of $\widetilde{\mathcal{N}}$. Neither of these two lines is in $\mathcal{L}$, for otherwise one of the starred octagons in Figure 8 would also belong to $\widetilde{\mathcal{M}}_{i-1}$. □

Lemma 5.3. *$\widetilde{\mathcal{M}}_z$ contains at most $7z - 3$ octagons.*

Proof: Each vertex on the boundary of $\widetilde{\mathcal{M}}_z$ has degree 2 or 3. Every degree-3 boundary vertex is the intersection of some line ℓ_i with the boundary of $\widetilde{\mathcal{M}}_z$. Because $\widetilde{\mathcal{M}}_z$ is convex, there are at most $2z$ such vertices. Between any pair of degree-3 boundary vertices, there are trivially at most 6 degree-2 boundary vertices, all on the boundary of the same octagon. Thus, the perimeter of $\widetilde{\mathcal{M}}_z$ is at most $14z$. The lemma now follows directly from Lemma 4.4. □

Constructing the relevant region $\widetilde{\mathcal{M}}_z$ is now straightforward. $\widetilde{\mathcal{M}}_0$ is a copy of the octagon containing the source of p. To compute $\widetilde{\mathcal{M}}_i$, we follow $\widetilde{p}$ until it exits the previous region $\widetilde{\mathcal{M}}_{i-1}$. At the exit point, the path is crossing ℓ_i into some octagon O'_k (with the notation of Figure 8). To complete $\widetilde{\mathcal{M}}_i$, we append the octagons $O'_1, \ldots, O'_u$.

5.2 Tightening Paths.

Theorem 5.4. *Let $\mathcal{M}$ be a combinatorial surface with complexity n, genus $g \geq 2$, and no boundary. Let p be a (not necessarily simple) path on $\mathcal{M}$, represented as a walk in $G(\mathcal{M})$ with complexity k. After preprocessing the surface in $O(n^2 \log n)$ time, we can compute a shortest path p' homotopic to p with complexity $k' = O(gnk)$ in time $O(gk + gn\bar{k})$, where $\bar{k} = \min\{k, k'\}$.*

Proof: The preprocessing consists of building the tight octagonal decomposition $\mathcal{O}$ on the cross-metric surface defined by $G^+(\mathcal{M})$. Let x be the number of crossings of p with $\mathcal{O}$; as we argued earlier, $x = O(gk)$. Let $\widetilde{p}$ be an arbitrary lift of p to the universal cover $\widetilde{\mathcal{M}}$. We first compute the relevant region of the universal cover with respect to $\widetilde{p}$, ignoring the internal structure of the surface within each octagon of $\widetilde{\mathcal{O}}$. In other words, we construct a subset of the abstract regular right-angled octagon tiling. The construction is described by Lemma 5.2. We require only constant time for every edge of p, every crossing between p and $\mathcal{O}$, and every relevant octagon; Lemma 5.3 shows that this phase takes time $O(k + x)$.

Now let L be the set of lines crossed an odd number of times by $\widetilde{p}$. Since each line separates $\widetilde{\mathcal{M}}$, then, by Lemma 4.5, there is a shortest path $\widetilde{q}$ with the same endpoints as $\widetilde{p}$ that crosses each line in L exactly once and no other line. Let x' be the number of lines in L. We now compute the set of octagons accessible from $\widetilde{p}(0)$ by crossing only these lines, this time building also the internal surface structure of the octagons. There are $O(x')$ such octagons by Lemma 5.3; since each cycle in $\mathcal{O}$ has constant multiplicity, each octagon has complexity $O(n)$. Thus, the relevant region has total complexity $O(x'n)$ and can be constructed in time $O(x'n)$.

Finally, we compute a shortest path $\widetilde{p}'$ between the endpoints of $\widetilde{p}$ in this relevant region, in time $O(x'n)$ using the planar shortest path algorithm by Henzinger et al. [17]. By giving the lines L infinitesimal crossing weight, we guarantee that $\widetilde{p}'$ crosses each line in L exactly once. The projection $p' = \pi(\widetilde{p}')$ onto $G(\mathcal{M})$ is the desired output path.

The complexity k' of p' is at most $O(x'n) = O(xn) = O(gnk)$. To complete the time analysis, we observe that $x' = O(g \min\{k, k'\})$. □

For the torus, we can apply a similar algorithm, only using a tight *square* decomposition comprised of two tight non-separating cycles that cross exactly once, which we can compute in $O(n^2 \log n)$ time. However, the relevant region has $O(x^2)$ squares; Lemma 5.3 no longer holds in this setting. Otherwise, the algorithm is unchanged. The running time is $O(k^2 + n\bar{k}^2)$, where $\bar{k} = \min\{k, k'\}$ and $k' = O(nk^2)$ is the complexity of the output path.

5.3 Tightening Cycles. Our algorithm for tightening cycles uses similar ideas as for paths. The major difference is that we lift the input cycle γ to the *cylindric cover* generated by γ; this is a covering space in which every simple cycle is either contractible or homotopic to a lift of γ [12, Lemma 2.5], [16, Proposition 1.36]. Topologically, the cylindric cover is an annulus, from which we can extract a relevant (annular) region of small complexity. We compute the output cycle using Lemma 3.3(d).

Theorem 5.5. *Let $\mathcal{M}$ be a combinatorial surface with complexity n, genus $g \geq 2$, and no boundary. Let γ be a (not necessarily simple) cycle on $\mathcal{M}$, represented as a closed walk in $G(\mathcal{M})$ with complexity k. After preprocessing the surface in $O(n^2 \log n)$ time, we can compute a shortest cycle γ' homotopic to γ with complexity $k' = O(gnk)$ in time $O(gk + gn\bar{k}\log(n\bar{k}))$, where $\bar{k} = \min\{k, k'\}$.*

In the case of the torus, again a tight square decomposition can be used. After the usual $O(n^2 \log n)$-time preprocessing, the algorithm runs in time $O(k^2 + n\bar{k}^2 \log(n\bar{k}))$.

6 Surfaces with Boundary

The algorithms for shortening paths and cycles on surfaces with boundary are much simpler than our corresponding results for boundary-free surfaces, although they follow roughly the same high-level approach. Specifically, the existence of at least one boundary allows to build a decomposition of the surface using *disjoint* simple *arcs*. Let $\mathcal{M}$ be a (g, b)-surface. Using a variation of an algorithm by Erickson and Whittlesey [14], we compute a set of $O(g + b)$ disjoint simple tight arcs, each of multiplicity at most two, cutting the surface into disks that are incident to three arcs, in $O((g + b)n + n \log n)$ time. This *tight triangulated system of arcs* is the analogue of the tight octagonal decomposition; given an input path or cycle, we similarly (but more easily) define the relevant region of the universal cover in which the output path or cycle can be searched. Our algorithms essentially follow the technique of Hershberger and Snoeyink [18], only using a certain collection of tight arcs in place of a triangulation. We obtain:

Theorem 6.1. *Let $\mathcal{M}$ be a combinatorial surface with complexity n, genus g, and b boundaries. Let c be a (not necessarily simple) path or cycle on $\mathcal{M}$, represented as a walk in $G(\mathcal{M})$ with complexity k. After preprocessing the surface in $O((g + b)n + n \log n)$ time, we can compute a shortest curve c' homotopic to c with complexity $k' = O((g + b)nk)$ in time $O((g + b)k + (g + b)n\bar{k})$ if c is a path or $O((g + b)k + (g + b)n\bar{k}\log(n\bar{k}))$ if c is a cycle, where $\bar{k} = \min\{k, k'\}$.*

7 Better Analysis of Loop and Cycle Shortening

We can prove that the algorithms of Colin de Verdière and Lazarus [4, 5, 6] run in time polynomial in the complexity of their input: the number of vertices, edges, and faces of the surface, plus the total number of edges of the input curves. Previously, these algorithms were only known to work in time polynomial in the size of the input and in the ratio between the largest and smallest length of an edge of the input surface.

These algorithms shorten a set of curves by iteratively replacing each curve by a tight curve in some simple space (a disk or a pair of pants). The number of iterations depends on the number of crossings between the input curves and a tight curve homotopic to one of the input curves. The existence of a tight octagonal decomposition (or of a tight triangulated system of arcs) with low multiplicity allows to bound the complexity of a shortest curve homotopic to a given curve, and hence the number of iterations. An independent argument shows that each iteration actually does not increase too much the complexity of the curves. Our results are:

Theorem 7.1. *On a combinatorial $(g, 0)$-surface with complexity n, the algorithm of Colin de Verdière and Lazarus [6] for shortening a fundamental system of loops of complexity k has running-time $O(g^4nk^4)$.*

Theorem 7.2. *On a combinatorial (g, b)-surface with complexity n, the algorithm of Colin de Verdière [4] for shortening a cut system by graph of complexity k has running-time $O((g + b)^4nk^4)$.*

Theorem 7.3. *On a combinatorial (g, b)-surface with complexity n, the algorithm of Colin de Verdière and Lazarus [5] for shortening a pants decomposition of complexity k has running-time $O((g+b)^4nk^4 \log(nk))$.*

Acknowledgments. The authors would like to thank Kim Whittlesey for helpful comments on topology and combinatorial group theory, and Francis Lazarus for several discussions (notably on the use of the cylindric cover and on a part of the proof of Theorem 7.1) that arose when writing previous papers [5, 6]. Jeff would also like to thank Michel Pocchiola for his invitation to visit ENS, where this work was initiated.

References

[1] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. *J. Algorithms* 49(2):284–303, 2003.

[2] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete Comput. Geom.* 31:61–81, 2004.

[3] S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Proc. 13th Annu. European Sympos. Algorithms*, 2005. To appear.

[4] É. Colin de Verdière. *Raccourcissement de courbes et décomposition de surfaces [Shortening of Curves and Decomposition of Surfaces]*. Ph.D. thesis, University of Paris 7, Dec. 2003. ⟨http://www.di.ens.fr/~colin/textes/these.html⟩.

[5] É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. *Proc. 11th Sympos. Graph Drawing*, 478–490, 2003. Lecture Notes Comput. Sci. 2912.

[6] É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete Comput. Geom.* 33(3):507–534, 2005.

[7] M. Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Math. Ann.* 72:413–421, 1912.

[8] T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Computer Syst. Sci.* 58:297–325, 1999.

[9] T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete Comput. Geom.* 14:93–110, 1995.

[10] A. Efrat, S. G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. *Proc. 10th Annu. European Sympos. Algorithms*, 411–423, 2002. Lecture Notes Comput. Sci. 2461, Springer-Verlag.

[11] D. Eppstein. Dynamic generators of topologically embedded graphs. *Proc. 14th Annu ACM-SIAM Sympos. Discrete Algorithms*, 599–608, 2003.

[12] D. B. A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica* 115:83–107, 1966.

[13] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31(1):37–59, 2004.

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

[15] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.* 16(6):1004–1022, 1987.

[16] A. Hatcher. *Algebraic topology*. Cambridge University Press, 2002. ⟨http://www.math.cornell.edu/~hatcher/⟩.

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

[18] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.* 4:63–98, 1994.

[19] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 80–89, 2001.

[20] J. H. Reif. Minimum s-t cut of a planar undirected network in $O(n \log^2(n))$ time. *SIAM J. Comput.* 12(1):71–81, 1983.

[21] H. Schipper. Determining contractibility of curves. *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, 358–367, 1992.

[22] J. Stillwell. *Classical Topology and Combinatorial Group Theory*. Springer-Verlag, New York, 1993.

[23] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 102–111, 1990.