

A tie is a noose, and inverted though it is,
it will hang a man nonetheless if he's not careful.

— Yann Martel, *Life of Pi* (2001)

(written:) Yakka foob mog. Grug pubbawup zink wattoo m gazork. Chumble špuzz.
(spoken:) I love loopholes.

— Calvin, explaining Newton's First Law of Motion 'in his own words'
Bill Watterson, *Calvin and Hobbes* (January 9, 1995)

8 Shortest Noncontractible Cycles

In this lecture, I'll describe several algorithms to compute the shortest noncontractible cycle in a combinatorial 2-manifold. The input is a cellularly embedded graph G on a 2-manifold Σ (or equivalently, a polygonal schema for Σ) with non-negatively weighted edges, and our goal is to compute a cycle in G of minimum length that is not contractible in Σ .

This problem has several natural motivations. the length of the shortest non-contractible cycle. In topological graph theory, the length of the shortest non-contractible cycle in an embedded graph is called the **handle girth** [1] or **edge-width** of the embedding [15, 18]; graphs that have embeddings with large edge-width share many useful combinatorial properties with planar graphs. A closely related concept is the **face-width** or **representativity** of a graph embedding, which is the minimum number of faces of G intersecting a noncontractible cycle on the surface, or equivalently, half the edge-width of the radial graph G° [17]. The length of the shortest noncontractible cycle in a Riemannian manifold is called the **systole** (to be more specific, the *homotopy 1-systole*) of the manifold [2]. Shortest non-contractible cycles are good indicators of **topological noise** of geometric surface models reconstructed from point clouds or volume data.

The algorithms described below all require that edge weights are non-negative. We treat the graph as a continuous metric space, where the edge weights represent distance. I will assume throughout the lecture that the shortest path between any two vertices of the graph is unique; this assumption can be enforced if necessary using standard perturbation schemes. However, we do *not* assume that the edge weights satisfy the triangle inequality.¹

8.1 Thomassen's 3-Path Property

The first efficient algorithm to compute shortest non-contractible cycles is an easy consequence of the following observation of Thomassen [18].

Lemma 8.1. *Let x and y be two points on a surface Σ , and let α, β, γ be paths in Σ from x to y . If the loops $\alpha \cdot \bar{\beta}$ and $\beta \cdot \bar{\gamma}$ are contractible, then the loop $\alpha \cdot \bar{\gamma}$ is also contractible.*

Proof: The concatenation of any two contractible loops is contractible. □

Corollary 8.2. *Let σ be the shortest non-contractible cycle in a surface Σ . Any pair of antipodal points partition σ into two equal-length shortest paths.*

Proof: Fix a pair x and \bar{x} of antipodal points in σ . These points clearly partition σ into two equal-length paths; call these paths α and β . Suppose α and β are not shortest paths, and let γ be a shortest path from x to \bar{x} . The cycles $\alpha \cdot \bar{\gamma}$ and $\beta \cdot \bar{\gamma}$ are both shorter than $\alpha \cdot \bar{\beta} = \sigma$ and thus are contractible. But then the 3-path property implies that σ is contractible, contradicting its definition. □

¹Alternately, we can enforce the triangle inequality by splitting every edge of weight w into two edges of weight $w/2$.

Thomassen's 3-path property implies the following $O(n^3)$ -time algorithm to find the shortest non-contractible cycle. The cycle must consist of a shortest path from a vertex x to another vertex y , the edge yz , and the shortest path from z back to x . There are only $O(n^2)$ cycles of this form. We compute the distance between every pair of vertices in $O(n^2 \log n)$ time by running Dijkstra's algorithm n times, after which we can compute the length of each candidate cycle in $O(1)$ time. Finally, we for candidate loop ℓ , we test whether ℓ is contractible in $O(n)$ time, as described in the next paragraph. Finally, we return the shortest candidate cycle that is noncontractible.

Each candidate loop ℓ has the form $\pi \cdot \ell' \cdot \bar{\pi}$, for some (possibly empty) path π and *simple* loop ℓ' . Thus, we only need to test *simple* loops for contractibility; this simplifies the algorithm considerably. First we perform a whatever-first search² of the dual subgraph $G^* \setminus \ell^*$. If this dual subgraph is connected, then the surface $\Sigma \setminus \ell$ is also connected, so ℓ is non-contractible. If $\Sigma \setminus \ell$ has two components, we can compute the Euler characteristic of each component in $O(n)$ time. If either component has Euler characteristic 1, that component must be a disk, and ℓ is contractible. Otherwise, ℓ is noncontractible.

8.2 Mixing Dijkstra with Contractibility Tests

Sariel Har-Peled and I [9] found a faster algorithm to compute the shortest non-contractible cycle, by interleaving the shortest-path computations and the tests for contractibility. (We were unaware of Thomassen's results.)

For any vertex x , we can compute the shortest non-contractible *loop* with basepoint x in $O(n \log n)$ time, as follows. We grow a shortest-path tree from x via Dijkstra's algorithm. Whenever the wavefront meets itself (without loss of generality) in the interior of an edge e , we check whether the loop γ defined by e is contractible, by performing simultaneous whatever-first searches on either side of γ . If the two searches meet, then γ is nonseparating and therefore noncontractible. If one search ends before the other, then γ is separating. In this case γ is contractible if and only if the Euler characteristic of the component we just searched is either 1 or $\chi - 1$. If γ is contractible, we can discard the contractible component of $\Sigma \setminus \gamma$. The complexity of the discarded subgraph is more than the running time of the whatever-first search. Thus, the total time spent in *all* such searches is only $O(n)$; this is dominated by the $O(n \log n)$ time required to grow the shortest-path tree.

Theorem 8.3. *The shortest non-contractible cycle in a combinatorial surface with complexity n can be computed $O(n^2 \log n)$ time.*

8.3 The Annulus

The algorithm just described is the fastest known for arbitrary surfaces. However, faster algorithms are known for surfaces with small genus; these are all based on a second straightforward corollary of Thomassen's 3-path property.

Corollary 8.4. *Let σ be the shortest non-contractible cycle in a surface Σ . Any shortest path in Σ crosses σ at most once.*

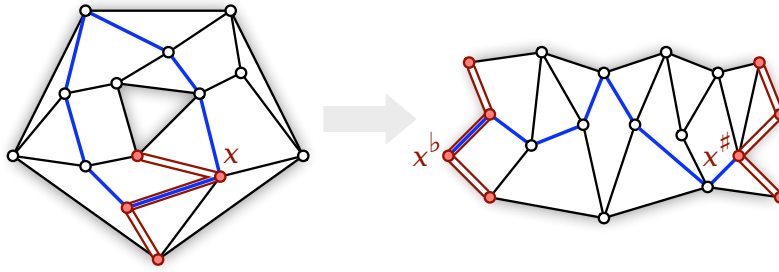
Proof: Suppose some shortest path π crosses σ more than once. Fix two vertices x and y in different components of $\pi \cap \sigma$. These vertices partition σ into two paths; call these paths α and β . Finally, let γ denote the subpath of π from x to y ; this is also the shortest path from x to y . The cycles $\alpha \cdot \bar{\gamma}$ and $\beta \cdot \bar{\gamma}$ are both shorter than $\alpha \cdot \bar{\beta} = \sigma$ and thus are contractible. But then the 3-path property implies that σ is contractible, contradicting its definition. \square

²Depth-first, breadth-first, random-first, whatever.

Itai and Shiloach [12] first observed this corollary for the special case of an **annulus**, which is a disk minus a smaller disk. Specifically, let π be the shortest path between a vertex on one boundary cycle of the annulus to a vertex on the other boundary cycle. Itai and Shiloach proved that the shortest noncontractible cycle σ crosses π exactly once. It follows that σ is homotopic to both boundary cycles of the annulus.

Let $G \setminus\!\!\setminus \pi$ denote the planar graph obtained from G by **cutting** along the path π , transforming the underlying annulus Σ into a disk, the closure of $\Sigma \setminus \pi$.³ The path π in G becomes two paths π^b and π^\sharp in $G \setminus\!\!\setminus \pi$, and every vertex and edge of π appears twice in $G \setminus\!\!\setminus \pi$. (In our proof of the surface classification theorem, we cut the surface along a loop.)

Itai and Shiloach observe that the shortest noncontractible cycle σ appears as the shortest path from x^b to x^\sharp , for any vertex x in the subpath $\pi \cap \sigma$. Thus, to find σ , it suffices to find the shortest path between every corresponding pair of nodes in $G \setminus\!\!\setminus \pi$. Each shortest path can be found in $O(n \log n)$ time via Dijkstra's algorithm, so the overall running time of Itai and Shiloach's algorithm is $O(n^2 \log n)$.



The shortest noncontractible cycle in G becomes a shortest path between clones in $G \setminus\!\!\setminus \pi$.

Reif [16] improved Itai and Shiloach's algorithm by adopting the following divide-and-conquer strategy. In an initialization phase, we compute the shortest paths in $G \setminus\!\!\setminus \pi$ from each endpoint of π^b to the corresponding endpoint of π^\sharp . Call these paths τ and β (for 'top' and 'bottom'); no other shortest path from π^b to π^\sharp crosses τ or β , so we can safely consider only the component of $(G \setminus\!\!\setminus \pi) \setminus\!\!\setminus \tau \setminus\!\!\setminus \beta$ that contains the paths π^b and π^\sharp . This subgraph H is the initial input to Reif's recursive algorithm.

The recursive algorithm has two cases:

- Suppose H has an **articulation vertex**: a vertex z that lies on every path from π^b to π^\sharp . In this case, we compute a shortest path tree rooted at z using Dijkstra's algorithm. Then for every vertex x^b in π^b , we can compute the length of the shortest path from x^b to x^\sharp in $O(1)$ time. This case requires $O(n \log n)$ time altogether.
- Suppose H has no articulation vertex. In this case, we find the *median* vertex x^b in π^b , and compute the shortest path π_x from x^b to x^\sharp in H using Dijkstra's algorithm. We then recursively consider each component of $H \setminus\!\!\setminus \pi_x$ as an independent subproblem.

To help determine whether H has an articulation vertex, we mark every vertex of τ as a 'top' vertex and every vertex of β as a 'bottom' vertex. A vertex is marked both 'top' and 'bottom' if and only if it is an articulation vertex. In the second case, we mark each copy of each vertex of π_x as 'top' or 'bottom' in the appropriate subproblem. This marking requires at most $O(n)$ time, and allows us to distinguish the two cases in later recursive calls in $O(1)$ time.

Assume that every face of H (except possibly the outer face) is a triangle; add infinite-weight edges if necessary. If H has an articulation vertex, the running time is clearly $O(n \log n)$. Otherwise, H must have at least $n/3 = \Omega(n)$ faces. Let $T(n, k)$ denote the running time of Reif's algorithm in this case,

³Cabello and Mohar [4, 6] introduced the notation $G \setminus\!\!\setminus \pi$, which is more evocative but too hard to write on the blackboard.

given a combinatorial disk with n faces, where the boundary paths π^b and π^\sharp each have k edges. This function satisfies the recurrence

$$T(n, k) \leq O(n \log n) + T(n_1, k/2) + T(n_2, k/2)$$

for some non-negative integers n_1 and n_2 with $n_1 + n_2 = n$. (The $O(n \log n)$ term absorbs the possibility that one or both recursive subproblems has an articulation point.) It is easy to prove by induction that $T(n, k) = O(n \log n \log k)$. Because $k = O(n)$ initially, the overall running time of Reif's algorithm is $O(n \log^2 n)$.

Theorem 8.5. *The shortest non-contractible cycle in a combinatorial annulus with complexity n can be computed $O(n \log^2 n)$ time.*

Frederickson [10] improved the running time of Reif's algorithm to $O(n \log n)$ using balanced separator hierarchies (which we will see later in the semester) and very careful analysis. Several other $O(n \log n)$ -time algorithms are now known for this problem [11, 13, 3, 8].

8.4 Cutting and Gluing with Shortest Paths

Corollary 8.4 was used by Cabello and Mohar [4, 6] to derive algorithms to find shortest non-contractible cycles in any surface of fixed genus in subquadratic time. In this section, I'll describe an algorithm of Kutz [14] to find the shortest non-contractible cycle in time $\bar{g}^{O(\bar{g})} n \log n$, where $\bar{g} = 2 - \chi$ is the Euler genus of the underlying surface Σ .

Recall that a **tree-cotree decomposition** (T, L, C) of a surface graph G is a partition of the edges into three disjoint subsets: a spanning tree T of G , a spanning cotree C of G / T (the dual of a spanning tree C^* of $G^* \setminus T^*$), and the leftover edges $L = G \setminus (C \cup T)$. For any tree-cotree decomposition (T, L, C) , the subgraph $T \cup L = G \setminus C$ inherits a cellular embedding from G that has exactly one face. Equivalently, if Σ is the underlying surface, the complement $\Sigma \setminus (T \cup L)$ is homeomorphic to an open disk. We call any subgraph with these properties a **cut graph**.

Now suppose T is a *shortest-path* tree rooted at some vertex x in G . For each leftover edge $e \in L$, let γ_e denote the unique cycle in the graph $T \cup \{e\}$. Finally, let $R = \bigcup_{e \in L} \gamma_e$. The subgraph R is also a cut graph; moreover, because R has no vertices of degree less than 1, we call R a **reduced cut graph**. R can also be defined as the result of repeatedly 'shaving' vertices of degree 1 from the cut graph $T \cup L$.

Lemma 8.6. *R is the union of $O(\bar{g})$ edge-disjoint shortest paths in G (possibly ending in the interiors of edges), where \bar{g} is the Euler genus of Σ .*

Proof: Recall that the number of edges in L is equal to the Euler genus \bar{g} . Within each edge e in L (in fact, each edge in $G \setminus T$), there is an interior point p_e that has two shortest paths to x . Split every edge e of L at the corresponding point p_e ; this transforms the reduced cut graph R into a tree T_R . This tree has at most $2\bar{g} + 1$ leaves (possibly including x), and therefore at most $2\bar{g}$ nodes with degree greater than 2. These nodes (and x) subdivide T_R into at most $4\bar{g} + 1$ shortest paths. \square

Now we can think of $\Sigma \setminus R$ as a polygonal schema for the surface Σ , whose $O(\bar{g})$ sides are shortest paths. Give each side of this schema a unique label. The **crossing sequence** of the shortest noncontractible cycle σ is the (circular) sequence of labels of shortest paths in R that σ crosses. Corollary 8.4 implies that each label appears in this crossing sequence at most once.

Kutz's algorithm enumerates all crossing sequences that use each label at most once and finds the shortest cycle with each crossing sequence. There are clearly at most $\bar{g}^{O(\bar{g})}$ candidate crossing sequences,

which can be enumerated in constant time each. For each candidate crossing sequence X , we assemble an annulus A_X by gluing together k copies of $\Sigma \setminus R$ along the corresponding shortest paths. This annulus has complexity $O(\bar{g}n)$, and we can construct it in $O(\bar{g}n)$ time. The shortest cycle σ_X with crossing sequence x is the projection of the shortest noncontractible cycle in A_X back into G . Thus, we can compute σ_X in $O(\bar{g}n \log n)$ time using the Reif-Frederickson algorithm.

However, not all crossing sequences induce noncontractible cycles; consider, for example, the crossing sequence of a cycle around a high-degree vertex of R . Fortunately, this is easy; we can use Dey and Guha's modification of Dehn's algorithm [7] to test whether *any* (and thus *every*) cycle with crossing sequence X is contractible in $O(\bar{g})$ time. (Crossing sequences are just the dual of traversal sequences!)

Theorem 8.7. *The shortest non-contractible cycle in a combinatorial surface with complexity n and Euler genus \bar{g} can be computed $\bar{g}^{O(\bar{g})}n \log n$ time.*

Cabello and Chambers described an algorithm that finds the shortest noncontractible cycle in $O(g^3 n \log n)$ time, using more advanced data structures [5]. This is the fastest published algorithm for surfaces of small genus.

References

- [1] M. O. Albertson and J. P. Hutchinson. The independence ratio and genus of a graph. *Trans. Amer. Math. Soc.* 226:161–173, 1977.
- [2] M. Berger. What is... a systole? *Notices Amer. Math. Soc.* 55(3):374–376, 2008.
- [3] G. Borradaile and P. Klein. An $O(n \log n)$ algorithm for maximum st -flow in a directed planar graph. *J. ACM* 56(2), 2009.
- [4] S. Cabello. Many distances in planar graphs. *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1213–1220, 2006.
- [5] 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.
- [6] S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete Comput. Geom.* 37:213–235, 2007.
- [7] T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Comput. System Sci.* 58:297–325, 1999.
- [8] J. Erickson. Maximum flows and parametric shortest paths in planar graphs. *Proc. 21st ACM-SIAM Symp. Discrete Algorithms*, p. to appear, 2010. (<http://compgeom.cs.uiuc.edu/~jeffe/pubs/parshort.html>).
- [9] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31(1):37–59, 2004.
- [10] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM J. Comput.* 16(6):1004–1004, 1987.
- [11] 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.
- [12] A. Itai and Y. Shiloach. Maximum flow in planar networks. *SIAM J. Comput.* 8:135–150, 1979.

- [13] P. Klein. Multiple-source shortest paths in planar graphs. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 146–155, 2005.
- [14] 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.
- [15] B. Mohar and C. Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, 2001.
- [16] 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.
- [17] N. Robertson and P. D. Seymour. Graph minors. VII. Disjoint paths on a surface. *J. Comb. Theory Ser. B* 45(2):212–254, 1988.
- [18] C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory Ser. B* 48(2):155–177, 1990.