

Global Minimum Cuts in Surface Embedded Graphs

Jeff Erickson*

Kyle Fox†

Amir Nayyeri‡

Department of Computer Science
University of Illinois at Urbana-Champaign
{jeffe,kylefox2,nayyeri2}@cs.uiuc.edu

Abstract

We give a deterministic algorithm to find the minimum cut in a surface-embedded graph in near-linear time. Given an undirected graph embedded on an orientable surface of genus g , our algorithm computes the minimum cut in $g^{O(g)}n \log \log n$ time, matching the running time of the fastest algorithm known for planar graphs, due to Łącki and Sankowski, for any constant g . Indeed, our algorithm calls Łącki and Sankowski's recent $O(n \log \log n)$ time planar algorithm as a subroutine.

Previously, the best time bounds known for this problem followed from two algorithms for general sparse graphs: a randomized algorithm of Karger that runs in $O(n \log^3 n)$ time and succeeds with high probability, and a deterministic algorithm of Nagamochi and Ibaraki that runs in $O(n^2 \log n)$ time. We can also achieve a deterministic $g^{O(g)}n^2 \log \log n$ time bound by repeatedly applying the best known algorithm for minimum (s, t) -cuts in surface graphs. The bulk of our work focuses on the case where the dual of the minimum cut splits the underlying surface into multiple components with positive genus.

1 Introduction

Recently there has been a new-found interest in flow and cut problems in planar graphs. While flows in planar graphs have been studied since Ford and Fulkerson's seminal work in the 1950s [37], it has only been a few years since Borradaile and Klein announced the first algorithm to find maximum flows in directed planar graphs in near-linear time [1, 5]. Borradaile and Klein's result came after a long line of work on finding maximum flows and minimum (s, t) -cuts in undirected planar graphs [38, 41, 45, 62]. Frederickson's $O(n \log n)$ -

time algorithm [38] stood as the fastest algorithm for these problem for almost 25 years, until the recent $O(n \log \log n)$ -time algorithm of Italiano *et al.* [46].

There has also been recent work on finding *global* minimum cuts, where the goal is to partition the vertices of a weighted undirected graph into two non-empty sets, without a specified pair of terminals that must be separated. (Unless specified otherwise, any references to minimum cuts in this paper refer to this version of the problem.) Chalermsook, Fakcharoenphol, and Nanongkai described the first deterministic near-linear time algorithm for computing minimum cuts in planar graphs [13]. This result was recently improved by Italiano *et al.* [46] and again by Łącki and Sankowski [53], who describe a deterministic algorithm that runs in $O(n \log \log n)$ time. These algorithms are significantly faster than the best known algorithms for general sparse graphs: a randomized algorithm of Karger that runs in $O(n \log^3 n)$ time and succeeds with high probability [49], and a deterministic algorithm of Nagamochi and Ibaraki that runs in $O(n^2 \log n)$ time [60].

These results and many others support the thesis that planarity aids in finding efficient algorithms. Moreover, most techniques for efficient computation on planar graphs generalize directly to graphs embedded on surfaces of higher genus, perhaps the most natural generalization of planar graphs. Examples include minimum spanning trees [61, 55]; single-source shortest paths [15, 43, 51, 54, 58, 65]; multiple-source shortest paths [8, 50]; replacement paths [33, 67]; graph and subgraph isomorphism [39, 44, 56, 29, 30]; and approximation of several NP-hard problems [2, 3, 4, 24, 30]. However, the first significant progress on flow and cut problems in surface graphs was made only recently [15, 16, 34]. It appears that all our intuitive notions for solving these problems in planar graphs require substantial modification, specifically a significant infusion of topology, when more complicated surfaces are involved.

In this paper, we describe the first deterministic near-linear time algorithm to find global minimum cuts in graphs embedded on orientable surfaces of fixed genus. Specifically, given an n -vertex graph embedded on a

*Portions of this work were done while this author was visiting IST Austria. Research by this author was partially supported by NSF grant CCF 09-15519.

†Portions of this work were done while this author was visiting Google, Inc. Research by this author is supported in part by the Department of Energy Office of Science Graduate Fellowship Program (DOE SCGF), made possible in part by the American Recovery and Reinvestment Act of 2009, administered by ORISE-ORAU under contract no. DE-AC05-06OR23100.

‡Portions of this work were done while this author was visiting Toyota Technological Institute at Chicago and Disney Animation Studio. Research by this author was partially supported by NSF grant CCF 09-15519.

surface of genus g , our algorithm runs in $g^{O(g)}n \log \log n$ time. When the genus g is constant, our algorithm’s running time matches the recent algorithm of Łącki and Sankowski for planar graphs [53]. In fact, our algorithm invokes both Łącki and Sankowski’s algorithm and the planar minimum (s, t) -cut algorithm of Italiano *et al.* [46]. Further improvements in both of these algorithms would immediately improve our algorithm as well.

Prior to our work, the fastest algorithms known for computing global minimum cuts in arbitrary surface graphs were the algorithms for general sparse graphs mentioned earlier: Karger’s randomized $O(n \log^3 n)$ -time algorithm [49] and Nagamochi and Ibaraki’s $O(n^2 \log n)$ -time deterministic algorithm [60]. For graphs of fixed genus, global minimum cuts can also be computed in $g^{O(g)}n^2 \log \log n$ time by invoking the fastest known algorithm for minimum (s, t) -cuts in surface graphs n times [16, 46].

At a very high level, the planar-graph algorithms of Chalermsook *et al.* [13] and Łącki and Sankowski use a divide-and-conquer strategy in the dual graph. In the dual graph, vertices and faces exchange roles; the minimum cut in the primal graph corresponds to the minimum-length cycle in the dual graph. Their algorithms split the dual graph into two pieces of roughly equal size with a simple cycle, and then recursively compute the minimum length cycle inside each piece. For the conquering step, Chalermsook *et al.* prove that the minimum-length cycle crosses the boundary between the two pieces at more twice. They use this fact to argue that a minimum-length cycle crossing the boundary must separate two specific faces and can therefore be found using an algorithm for minimum (s, t) -cuts (in the original primal graph).

Unfortunately, this divide-and-conquer approach does not immediately generalize to surfaces with positive genus. First, the minimum cut is no longer necessarily dual to a single cycle; second, because not all cycles in surface graphs bound disks, we cannot so easily divide the graph into equal-sized pieces that maintain the necessary crossing properties. To work around these obstacles, as in previous work on (s, t) -minimum cuts [16, 34], we rely on properties of subgraphs that have minimum weight in their \mathbb{Z}_2 -homology class. (Two subgraphs of G are in the same \mathbb{Z}_2 -homology class if their symmetric difference is the boundary of a subset of faces of G ; see Section 2 for further details.) In particular, in Sections 5 and 6, we prove several properties of \mathbb{Z}_2 -minimal subgraphs that can possibly cross the dual of the minimum cut; these properties may be of independent interest.

1.1 Related work

Minimum (s, t) -cuts. The first algorithm to compute minimum cuts in planar graphs appears in the seminal work of Ford and Fulkerson [37]. Itai and Shiloach [45] proposed an algorithm to compute minimum (s, t) -cuts in *undirected* planar graphs in $O(n \log^2 n)$ time. Their algorithm was later improved by Reif [62], Frederickson [38], and most recently Italiano *et al.* [46]; the last of these improvements runs in $O(n \log \log n)$ time. The fastest algorithms known for computing minimum (s, t) -cuts in *directed* planar graphs require the computation of maximum (s, t) -flows [1, 5].¹ Chalermsook *et al.* [13] described the first algorithm to compute *global* minimum cuts in undirected planar graphs in near-linear time. Their algorithm was recently improved by Łącki and Sankowski [53] to run in $O(n \log \log n)$ time.

Until recently, the best algorithms to compute minimum cuts in surface graphs were those for general sparse graphs. Chambers *et al.* described an algorithm to compute minimum (s, t) -cuts in undirected surface graphs in $g^{O(g)}n \log n$ time [16]. This time bound was improved to $g^{O(g)}n \log \log n$ Italiano *et al.* [46] and to $2^{O(g)}n \log n$ by Erickson and Nayyeri [34]. For minimum (s, t) -cuts in *directed* surface graphs, the fastest algorithm known is the maximum-flow algorithm of Chambers *et al.* [15], which runs in near-linear time for graphs of fixed genus and polynomially-bounded integer capacities.

Shortest interesting cycles. Itai and Shiloach [45] proved that the minimum (s, t) -cut in an undirected planar graph G corresponds to the shortest generating cycle in an annulus, obtained by removing the faces s^* and t^* from the dual graph G^* . All the planar minimum-cut algorithms mentioned earlier actually work in this dual formulation. Thomassen [66] described the first polynomial-time algorithm to find the shortest nontrivial cycle in undirected surface graphs; Thomassen’s algorithm runs in $O(n^3)$ time. Erickson and Har-Peled improved the running time to $O(n^2 \log n)$ [32]. Their algorithm is currently the best when the genus is unbounded, but there are several faster algorithms when the genus is sufficiently small [12, 7, 52, 8, 46]. Cabello [6] describes an algorithm to find the shortest *simple* contractible cycle (without repeated vertices) in an undirected surface graph in $O(n^2 \log n)$ time; he also proves that finding the shortest simple *separating* cycle is NP-hard. For related results, see [10, 11, 14, 36].

Cabello *et al.* [9] described the first non-trivial algorithms to compute shortest non-contractible or non-

¹Janiga and Koubek [47] claimed an algorithm to compute minimum (s, t) -cuts in directed planar graphs in near-linear time, but their algorithm has a subtle flaw [48].

separating cycles in *directed* surface graphs; the fastest of their algorithms runs in $O(\sqrt{g}n^{3/2}\log n)$ time. Erickson and Nayyeri [34] showed that a modification of their algorithm for minimum (s, t) -cuts finds shortest non-separating cycles in $2^{O(g)}n\log n$ time. Most recently, Erickson [31] proposed algorithms to find shortest non-separating cycles in $O(g^2n\log n)$ time and to find shortest non-contractible cycles in $g^{O(g)}n\log n$ time.

Shortest equivalent cycles and subgraphs. There are several results on finding shortest cycles in surface graphs that are topologically equivalent to a given cycle. Colin de Verdière and Erickson [22] described an algorithm to compute the shortest cycle *homotopic* to a given cycle in $O(gnk\log nk)$ time, where k is the number of edges in the input cycle, generalizing and improving an earlier result of Colin de Verdière and Lazarus for *simple* cycles [23].

Chambers *et al.* [14] showed that finding the shortest cycle (simple or not) in any given *homology* class is NP-hard, but fixed-parameter tractable with respect to genus. The minimum (s, t) -cut algorithms of Chambers *et al.* [16, 34] work by computing the minimum-cost *even subgraph* in a given \mathbb{Z}_2 -homology class, a problem which is also NP-hard. Chen and Friedman [18, 19] proved that this problem becomes NP-hard to approximate within any constant factor when generalized to simplicial complexes. Following a strategy first suggested by Sullivan [64], Chambers *et al.* [15] describe an algorithm to quickly find minimum-cost *circulations* in any given real or integer homology class in a directed surface graph. Their result was subsequently generalized by Dey *et al.* [25] to work with arbitrary chains of arbitrary dimension in arbitrary simplicial complexes. For more related results, see [20, 28, 26, 35].

2 Notation and Terminology

We begin by recalling several useful definitions related to surface-embedded graphs. For further background, we refer the reader to Gross and Tucker [40] or Mohar and Thomassen [57] for topological graph theory, and to Hatcher [42] or Stillwell [63] for surface topology and homology. We adopt most of our terminology and notation from previous works [16, 34].

2.1 Surfaces and Curves

A *surface* (more formally, a *2-manifold with boundary*) 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 the surface; every component of the boundary is homeomorphic to a circle.

A *path* in a surface Σ is a continuous function $p: [0, 1] \rightarrow \Sigma$. An *arc* is a path whose endpoints lie on the boundary of Σ . A *cycle* is a continuous function $\gamma: S^1 \rightarrow \Sigma$. Paths and cycles are *simple* if they are injective. Following standard practice, we do not distinguish between a simple path or cycle and its image in Σ .

The *genus* of a surface Σ is the maximum number of disjoint simple cycles $\gamma_1, \gamma_2, \dots, \gamma_g$ in Σ whose complement $\Sigma \setminus (\gamma_1 \cup \dots \cup \gamma_g)$ is connected. We will consider only compact, connected, orientable surfaces. Up to homeomorphism, there is exactly one such surface with any genus $g \geq 0$ and any number of boundaries $b \geq 0$; the *Euler characteristic* χ of this surface is $\chi := 2 - 2g - b$.

2.2 Graph Embeddings

An *embedding* of an undirected graph G on a surface Σ maps vertices to distinct points and edges to interior-disjoint curves. 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 $O(n)$.

Cellular graph embeddings are equivalent to the *combinatorial surfaces* introduced by Colin de Verdière [21] and used by several authors to formulate optimization problems for surface-embedded graphs. A combinatorial surface consists of an abstract surface Σ together with a cellularly embedded graph G with (possibly asymmetrically) weighted edges. Paths and cycles in a combinatorial surface are directed walks in its graph; the length of any such walk is the sum of its (directed) edge weights, counted with appropriate multiplicity.

Two paths or cycles in a combinatorial surface *cross* if no continuous infinitesimal perturbation makes them disjoint; if such a perturbation exists, then the paths are *non-crossing*. We say that a cycle γ is *non-self-crossing* if no two sub-paths of γ cross, *weakly simple* if γ is non-self-crossing and traverses each edge at most once, and *(strictly) simple* if γ visits each vertex at most once.

We redundantly use the term *arc* to refer to a walk in the graph whose endpoints are boundary vertices. Likewise, we use the term *cycle* to refer to a closed walk in the graph. *Cutting* a combinatorial surface along a cycle or arc modifies both the surface and the embedded

graph. For any combinatorial surface $S = (\Sigma, G)$ and any cycle or arc γ in G , we define a new combinatorial surface $S \setminus \gamma$ by taking the topological closure of $\Sigma \setminus \gamma$ as the new underlying surface; the new embedded graph contains two copies of each vertex and edge of γ , each bordering a new boundary. In contrast, a **cut** in a graph is a set of edges whose removal partitions the graph into two non-empty connected components. The weight of a cut is the total weight of its edges.

2.3 Homotopy and Homology

Two cycles γ and δ in Σ are *homotopic*, or in the same *homotopy class*, if there is a continuous map $h: [0, 1] \times S^1 \rightarrow \Sigma$ such that $h(0, \cdot) = \gamma$ and $h(1, \cdot) = \delta$. The function h is called a *homotopy* between γ and δ . A cycle γ is **contractible** if it is homotopic to a constant map, **separating** if $\Sigma \setminus \gamma$ is disconnected, and **splitting** if $\Sigma \setminus \gamma$ is disconnected and both components have positive genus. Similarly, two arcs α and β are homotopic if there is a continuous map $h: [0, 1] \times [0, 1] \rightarrow \Sigma$ such that $h(0, \cdot) = \alpha$, $h(1, \cdot) = \beta$, and $h(t, \cdot)$ is an arc for all $t \in [0, 1]$.

An arc or a cycle is **tight** if it is as short as possible in its homotopy class. A **system of tight arcs** is a minimal set $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ of arcs such that $\Sigma \setminus P$ is a disk and each arc α_i is tight. Euler's formula implies that any system of tight arcs contains exactly $2g + b - 1$ arcs.

Homology is a coarser equivalence relation than homotopy, with nicer algebraic properties. Like several earlier papers [17, 18, 26, 27, 16, 34], 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 . 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 . In this paper, the term **separating subgraph** always refers to a null-homologous even subgraph with at least one edge. 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 even subgraphs of G fall into exactly 4^g \mathbb{Z}_2 -homology classes.

We define the **carrier** of a (not necessarily simple) cycle γ in G to be the even subgraph of edges that γ traverses an odd number of times; two cycles are homologous if their carriers are homologous. A cycle γ or even subgraph η is **\mathbb{Z}_2 -minimal** if it has minimum total length among all cycles or even subgraphs homologous with γ or η .

2.4 Graph Duality

For any graph G on a surface without boundary, we can define a canonical **dual graph** G^* . The vertices of G^* correspond to the faces of G , and two vertices in G^* are joined by a (dual) edge if and only if the corresponding faces of G are separated by an edge of G . Thus, every edge e in G has a corresponding dual edge in G^* , denoted e^* . For any face f of G , we let f^* denote the corresponding vertex of G^* . The dual graph G^* has a cellular embedding on Σ , whose faces correspond exactly to the vertices of G . For any vertex v of G , we let v^* denote the corresponding face of G^* . Duality is an involution—the dual of G^* is isomorphic to the original graph G .

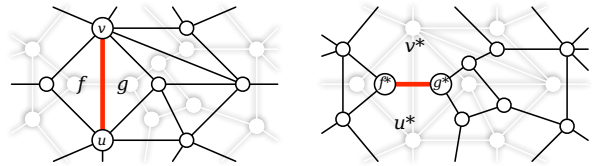


Figure 1. Graph duality. One edge uv and its dual $(uv)^* = f^*g^*$ are emphasized.

For any subgraph $F = (U, D)$ of $G = (V, E)$, we write $G \setminus F$ to denote the edge-complement of F in G or $(V, E \setminus D)$. Also, when the graph G is fixed, we abuse notation by writing F^* to denote the subgraph of G^* corresponding to a subgraph F of G ; each edge in F^* is the dual of a unique edge in F . In particular, we have the identity $(G \setminus F)^* = G^* \setminus F^*$.

3 Homology Cuts

We are now ready to describe our minimum cut algorithm. To work with the topology of Σ in computing a minimum cut, we use the following modification of a lemma of Chambers *et al.* [16, Lemma 3.1]. Recall that a **separating subgraph** is a null-homologous even subgraph with at least one edge.

Lemma 3.1. *Let G be an undirected graph with non-negative edge capacities, cellularly embedded on a surface Σ without boundary, and let C be a minimum-capacity cut in G . Then C^* is a minimum-weight separating subgraph of G^* .*

Proof: Let C be an arbitrary cut in G . The cut partitions the vertices of G into two disjoint subsets S and T . Therefore, the dual subgraph C^* partitions the faces of G^* into two disjoint subsets S^* and T^* . Further, C^* is the boundary of the union of faces in S^* , implying that C^* is null-homologous in Σ and therefore separating.

Conversely, let C^* be an arbitrary separating subgraph of G^* . As C^* is null-homologous, it is the boundary of a

subset of the faces of G^* . Moreover, because C^* is non-empty, it must be the boundary of a *proper, non-empty* subset of faces. Let s^* and t^* be faces of G^* on either side of C^* . Any path from s to t in the primal graph G must traverse at least one edge of C . We conclude that C is a cut (in particular, an (s, t) -cut). \square

Fix an undirected graph $G = (V, E)$, a non-negative weight function $w: E \rightarrow \mathbb{R}$, and a cellular embedding of G on a surface Σ of genus g with at least two faces. In light of Lemma 3.1, we focus our attention on finding a minimum-weight separating subgraph of G .

To simplify our exposition, we assume without loss of generality that G contains a unique shortest path between any pair of vertices; this assumption can be enforced with high probability using standard perturbation techniques [59].

Our algorithm separately considers three (overlapping) cases, illustrated in Figure 2:

1. Some minimum-weight separating subgraph is the union of two nonempty edge-disjoint even subgraphs.
2. Some minimum-weight separating subgraph consists of a single contractible simple cycle.
3. Every minimum-weight separating subgraph splits the surface into two components of positive genus.

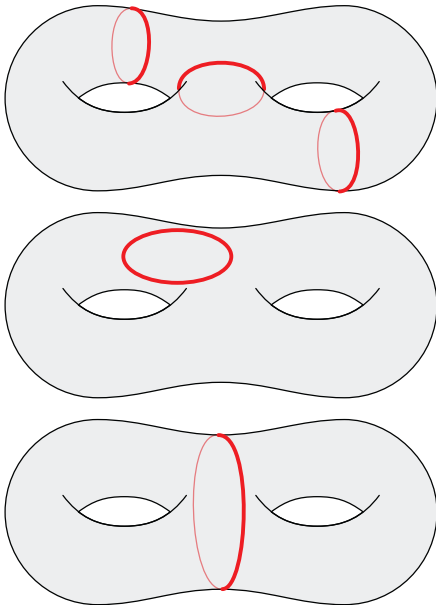


Figure 2. Three types of minimum-weight separating subgraphs: decomposable, contractible, and splitting.

We emphasize that only the first two cases are disjoint. In the following sections, we describe subroutines to

find a minimum-weight separating subgraph if the corresponding condition holds. If one of these conditions does not hold, the corresponding subroutine returns either an error or a separating subgraph that is not minimal. By running all three subroutines and returning the best result, we find a minimum separating subgraph no matter which category it falls into.

4 Multiple Even Subgraphs

We begin with the easiest of the three cases. If the minimum separating subgraph σ is the union of multiple non-empty even subgraphs, the following algorithm finds it. Otherwise, this algorithm may return an even subgraph more expensive than σ .

Lemma 4.1. *Let σ be a minimum-weight separating subgraph of G . If σ is the union of two or more non-empty even subgraphs, then σ is also the union of a non-separating \mathbb{Z}_2 -minimal subgraph η and a minimum-weight even subgraph that is edge-disjoint from η and homologous with η .*

Proof: Suppose σ is the union of two non-empty edge-disjoint even subgraphs γ and δ . Because σ is null-homologous, γ and δ must be homologous. Let η be the minimum-weight even subgraph homologous to δ and γ . If either $\eta = \delta$ or $\eta = \gamma$, we are done. Otherwise, $\eta \oplus \gamma$ is a separating subgraph of smaller total weight than σ , which is impossible. \square

The minimum even subgraph in any homology class can be found in $g^{O(g)}n \log \log n$ time using Italiano *et al.*'s modification of an algorithm by Chambers, Erickson, and Nayyeri [16, 46]. Likewise, we can find the smallest subgraph of any homology class that avoids the minimum even subgraph η in that same class in $g^{O(g)}n \log \log n$ time by assigning infinite weight to every edge of η and applying the same algorithm. We can enumerate \mathbb{Z}_2 -homology classes and apply Lemma 4.1 to show the following:

Lemma 4.2. *We can compute a minimum-weight separating subgraph in $g^{O(g)}n \log \log n$ time, if any such subgraph is the union of two non-empty edge-disjoint even subgraphs.*

5 Contractible Cycle

Next we describe an algorithm to handle the case where some minimum-weight separating subgraph is a contractible simple cycle. We emphasize that our algorithm may fail to return a contractible simple cycle if no such cycle is a minimum separating subgraph. We begin by borrowing a result of Cabello [6, Lemma 4.1].

Lemma 5.1 (Cabello [6]). *Let α be a tight arc or tight cycle on G . The shortest contractible simple cycle does not cross α .*

Corollary 5.2. *The shortest contractible simple cycle and the shortest non-separating cycle in G do not cross.*

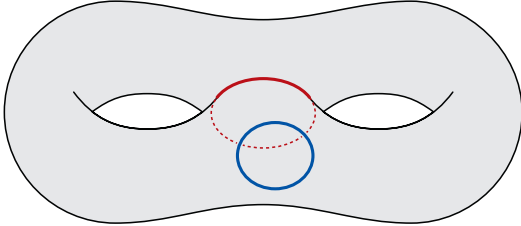


Figure 3. The shortest contractible simple cycle does not cross the shortest non-separating cycle.

Lemma 5.3. *We can compute a minimum-weight separating subgraph in $g^{O(g)}n \log \log n$ time, if any such subgraph is a contractible simple cycle.*

Proof: We begin by computing the shortest non-separating cycle α in G in $g^{O(g)}n \log \log n$ time, using a recent modification of an algorithm of Kutz [52] by Italiano *et al.* [46]. The surface $\Sigma \setminus \alpha$ has two boundary cycles α' and α'' . We compute a system P of tight arcs connecting α' and α'' in $O(n)$ time using the shortest-path algorithm of Henzinger *et al.* [43], as described by Erickson and Nayyeri [34]. Let $G_\#$ denote the planar graph $G \setminus (\alpha \cup P)$; this graph has $O(gn)$ vertices.

Pick an arbitrary edge e of α , and let e_1 and e_2 be the copies of e in $G_\#$. Let γ_1 and γ_2 be the shortest simple cycles in the subgraphs $G_\# \setminus e_1$ and $G_\# \setminus e_2$, respectively. We compute both γ_1 and γ_2 in $O(gn \log \log n)$ time using the algorithm of Łącki and Sankowski [53].

Let γ the shorter of the cycles γ_1 and γ_2 . The cycle γ projects to a null-homologous closed walk γ' in the original graph G , which may or may not be simple. The outer face of $G_\#$ is the only face that is not also a face of G . It follows that the only separating cycle in $G_\#$ that is not a separating subgraph in G is the boundary of outer face. Because γ avoids at least one edge of the outer face, the carrier of γ' must be non-empty.

Corollary 5.2 and Lemma 5.1 imply that some shortest contractible simple cycle σ in G crosses neither P nor α . (We emphasize that our algorithm does not actually compute σ .) This cycle σ appears as a simple cycle in $G_\#$ that avoids at least one of the edges e_1 or e_2 . Thus, σ cannot be shorter than γ .

If the walk γ' is a simple cycle, it must be contractible, and therefore cannot be shorter than σ . In this case, γ'

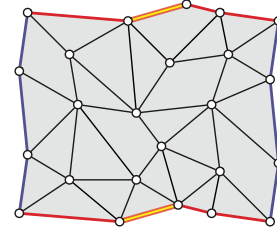


Figure 4. At least one copy of e is forbidden in the planarized graph.

must be a shortest contractible simple cycle in G , and our algorithm returns this cycle. On the other hand, if γ' is not a simple cycle, its carrier is composed of more than one cycle; in this case, our algorithm reports correctly that no minimum-weight separating subgraph of G is a contractible simple cycle. \square

6 Splitting Subgraph

Finally, we consider the case where every minimum-weight separating subgraph splits the surface into two components with genus. Our algorithm may not return the minimum-weight splitting subgraph if it is not the minimum-weight separating subgraph, but it always returns some separating subgraph if it returns anything at all.

Lemma 6.1. *Suppose every minimum-weight separating subgraph is splitting; let σ be a minimum-weight separating subgraph. Let γ be a closed walk on G such that σ does not cross γ , and let η be the shortest even subgraph homologous to γ . Then η lies in the closure of the same component of $\Sigma \setminus \sigma$ as γ .*

Proof: Assume for the sake of deriving a contradiction that η does not lie in the closure of the same component as γ . The subgraph σ separates the faces of G into two non-empty sets. Call the faces in the component of $\Sigma \setminus \sigma$ containing γ the *near* faces and call the rest of the faces *far*. Similarly, the even subgraph $\eta \oplus \gamma$ is null-homologous and separates the faces of G into two subsets; call the faces in one subset *black* and the others *white*.

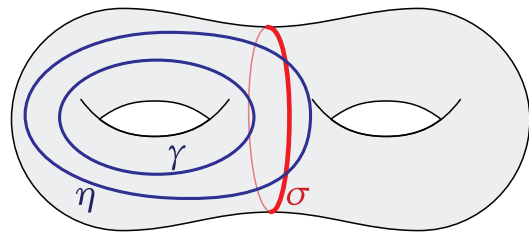


Figure 5. An impossible crossing: the shortest separating cycle σ , a cycle γ that avoids σ , and the shortest even subgraph η homologous to γ .

Let F be the boundary of the union of the far black faces in G . By definition, F is a null-homologous even subgraph. By assumption, η has edges that are incident to two far faces, but γ does not; thus, there is at least one far black face. Since there is also at least one white face, F is non-empty. Because both F and σ are null-homologous, the even subgraph $\eta' = \eta \oplus F \oplus \sigma$ is homologous to η , and therefore to γ .

For any subgraph H of G , let $w(H)$ denote the sum of the weights of the edges of H . We now prove that $w(F) + w(\eta') \leq w(\eta) + w(\sigma)$ by bounding the contribution of each edge $e \in E(G)$ to both sides of the inequality. Note that both F and η' are subgraphs of $\sigma \cap \eta$; moreover, $F \oplus \eta' = \sigma \oplus \eta$. There are three cases to consider.

- If $e \notin \eta \cup \sigma$, then $e \notin F$. In this case, e contributes 0 to both sides of the inequality.
- If $e \in \sigma \oplus \eta$, then $e \in F \oplus \eta'$. In this case, e contributes $w(e)$ to both sides of the inequality.
- If $e \in \sigma \cap \eta$, then e contributes exactly $2w(e)$ to the right side of the inequality. Trivially, e contributes at most $2w(e)$ to the left side.

On the other hand, because F is a separating subgraph, we must have $w(F) > w(\sigma)$. It immediately follows that $w(\eta') < w(\eta)$, which contradicts the minimality of η . \square

Corollary 6.2. *Suppose every minimum-weight separating subgraph is splitting; let σ be a minimum-weight separating subgraph. Then G contains two even subgraphs η_1 and η_2 that lie in the closures of different components of $\Sigma \setminus \sigma$, such that η_1 and η_2 are minimal in their respective nontrivial homology classes.*

Proof: Let Σ_1 and Σ_2 denote the closures of the components of $\Sigma \setminus \sigma$. Let γ_1 be any closed walk in Σ_1 whose homology class is nontrivial; such a closed walk exists because Σ_1 has positive genus. Lemma 6.1 implies that the shortest even subgraph η_1 in G that is homologous to γ_1 lies in Σ_1 . Similarly, Σ_2 contains the shortest even subgraph of G that is homologous to any nontrivial closed walk in Σ_2 . \square

Lemma 6.3. *We can compute a minimum-weight separating subgraph in $g^{O(g)}n \log \log n$ time, if every such subgraph is splitting.*

Before we prove the lemma, we describe the following subroutine. Given two non-crossing closed walks α_1 and α_2 in distinct \mathbb{Z}_2 -homology classes, we show how to compute in $g^{O(g)}n \log \log n$ time the minimum non-empty even subgraph σ that separates α_1 and α_2 if it exists.

We compute the graph $G_{\setminus} = G \setminus (\alpha_1 \cup \alpha_2)$ in linear time. Let Σ_{\setminus} be its underlying surface. The graph G_{\setminus} has genus $g - 2$ and four boundary cycles $\alpha'_1, \alpha''_1, \alpha'_2,$ and α''_2 . An even subgraph separating α_1 and α_2 in G corresponds to an even subgraph in G_{\setminus} that is homologous to $\alpha'_1 \oplus \alpha''_1$. Unfortunately, we cannot simply find the smallest even subgraph of G_{\setminus} homologous to $\alpha'_1 \oplus \alpha''_1$ as this subgraph might correspond to an empty subgraph of G . Instead, we employ a technique similar to the one used in Lemma 5.3.

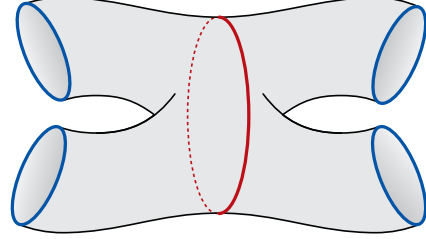


Figure 6. The surface obtained after cutting a double torus along α_1 and α_2 .

Let $e_1, e_2 \in E(G)$ be two distinct edges in α_1 and α_2 respectively. Let $e'_1, e''_1, e'_2,$ and e''_2 be their corresponding edges in G_{\setminus} . Let $\eta_{11}, \eta_{12}, \eta_{21},$ and η_{22} be the shortest even subgraphs homologous to $\alpha'_1 \oplus \alpha''_1$ in $G_{\setminus} \setminus \{e'_1, e'_2\}$, $G_{\setminus} \setminus \{e''_1, e''_2\}$, $G_{\setminus} \setminus \{e'_1, e'_2\}$, and $G_{\setminus} \setminus \{e''_1, e''_2\}$ respectively if they exist. We can find these even subgraphs in $g^{O(g)}n \log \log n$ time using the modification of Italiano *et al.* [46] to the algorithm of Chambers, Erickson, and Nayyeri [16]. Let η' be the shortest subgraph that exists of these four. If no such subgraph exists, then we report there is no non-empty even subgraph that separates α_1 and α_2 in G .

Suppose without loss of generality that $\eta' = \eta_{11}$. Both e'_1 and e'_2 lie on boundary cycles of G_{\setminus} so they are incident to unique faces f_1 and f_2 respectively. Face f_1 must lie in the same component of $\Sigma_{\setminus} \setminus \eta'$ as $\alpha'_1 \oplus \alpha''_1$, because η' must avoid e'_1 . Likewise, f_2 must lie in the opposite component of $\Sigma_{\setminus} \setminus \eta'$. Therefore, η' corresponds to an even subgraph η of G that separates the surface so that f_1 lies in the same component of $\Sigma \setminus \eta$ as α_1 and f_2 lies in the same component of $\Sigma \setminus \eta$ as α_2 . The subgraph η must be non-empty.

Since σ is weakly simple in G , it must avoid one of e'_1 or e''_1 and one of e'_2 or e''_2 in G_{\setminus} . Consequently, σ is not shorter than η , making η a valid output to our subroutine. We may now proceed to prove Lemma 6.3.

Proof: We begin by enumerating the $2^{O(g)}$ pairs of distinct nontrivial \mathbb{Z}_2 -homology classes in G . For each pair of classes, we find the minimum even subgraphs η_1 and η_2 in those classes in $g^{O(g)}n \log \log n$ time using the modification of the algorithm of Chambers *et al.* [16] by of Italiano *et al.* [46]. Both even subgraphs η_1 and η_2

can be decomposed into edge-disjoint, non-crossing, weakly simple cycles [16, Lemma 3.2]. Let α_1 and α_2 be arbitrary cycles from some cycle decompositions of η_1 and η_2 , respectively. If these cycles cross, then we disregard η_1 and η_2 and continue with the next pair of homology classes. Otherwise, we find the minimum non-empty even subgraph γ that separates α_1 and α_2 in $g^{O(g)}n \log \log n$ time using the subroutine above. After enumerating all pairs of even subgraphs, return the smallest subgraph γ found. All of the subgraphs returned by the subroutine are splitting subgraphs.

Corollary 6.2 implies that at least one pair of even subgraphs η_1 and η_2 from the enumeration lie on opposite sides of some smallest splitting subgraph σ . It follows that σ also separates the cycles α_1 and α_2 taken from the cycle decompositions of η_1 and η_2 respectively. As α_1 and α_2 lie in different components of $\Sigma \setminus \sigma$, they cannot cross. \square

7 Conclusion and Open Problems

By returning the smallest result from the algorithms described by Lemmas 4.1, 5.3, and 6.3, we immediately obtain our main results.

Theorem 7.1. *A minimum-weight separating subgraph of an undirected n -vertex graph embedded on an orientable surface of genus g can be computed in $g^{O(g)}n \log \log n$ time.*

Corollary 7.2. *A global minimum-cut in an undirected n -vertex graph embedded on an orientable surface of genus g can be computed in $g^{O(g)}n \log \log n$ time.*

Our algorithm repeatedly applies two recent $O(n \log \log n)$ -time algorithms for planar graphs as black boxes: one due to Łącki and Sankowski for global minimum cuts [53] and another due to Italiano *et al.* for minimum (s, t) -cuts [46]. Indeed, these are the only subroutines in our algorithm that require more than linear time when the genus is fixed. Thus, improvements to both of these algorithms would immediately improve our algorithm as well.

Although our algorithm works in near-linear time for graphs of constant genus, the complexity dependence on the genus is exponential. This exponential dependence is unavoidable with our current technique, as our algorithm calls a subroutine that solves an NP-hard problem: finding the minimum-weight subgraph in a given \mathbb{Z}_2 -homology class [16]. We optimistically conjecture that global minimum cuts in surface graphs can be computed in $O(g^k n \log \log n)$ time for some small constant k , using different techniques.

Acknowledgment. The authors would like to thank the anonymous reviewers for their helpful suggestions on improving this paper.

References

- [1] G. Borradaile. *Exploiting Planarity for Network Flow and Connectivity Problems*. Ph.D. thesis, Brown University, May 2008. (<http://www.cs.brown.edu/research/pubs/theses/phd/2008/glencora.pdf>).
- [2] 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>).
- [3] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1285–1294, 2007.
- [4] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. Steiner tree in planar graphs: An $O(n \log n)$ approximation scheme with singly-exponential dependence on epsilon. *Proc. 10th Workshop on Algorithms and Data Structures*, 275–286, 2007.
- [5] 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.
- [6] S. Cabello. Finding shortest contractible and shortest separating cycles in embedded graphs. *ACM Trans. Algorithms* 6(2):24:1–24:18, 2010.
- [7] S. Cabello. Many distances in planar graphs. *Algorithmica*, in press, 2010. (<http://dx.doi.org/10.1007/s00453-010-9459-0>).
- [8] 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.
- [9] S. Cabello, É. Colin de Verdière, and F. Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *Proc. 26th Ann. Symp. Comput. Geom.*, 156–165, 2010.
- [10] S. Cabello, É. Colin de Verdière, and F. Lazarus. Output-sensitive algorithm for the edge-width of an embedded graph. *Proc. 26th Ann. Symp. Comput. Geom.*, 147–155, 2010.
- [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] P. Chalermsook, J. Fakcharoenphol, and D. Nanongkai. A deterministic near-linear time algorithm for finding minimum cuts in planar graphs. *Proc. 15th Ann. ACM-SIAM Symp. Discrete Algorithms*, 828–829, 2004.
- [14] 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.
- [15] E. W. Chambers, J. Erickson, and A. Nayyeri. Homology flows, cohomology cuts. *Proc. 42nd Ann. ACM Symp. Theory Comput.*, 273–282, 2009. Full version to appear in *SIAM J. Comput.*
- [16] E. W. Chambers, J. Erickson, and A. Nayyeri. Minimum cuts and shortest homologous cycles. *Proc. 25th Ann. Symp. Comput. Geom.*, 377–385, 2009.
- [17] C. Chen and D. Freedman. Quantifying homology classes II: Localization and stability. Preprint, 2007. (<http://arxiv.org/abs/0709.2512v2>).
- [18] C. Chen and D. Freedman. Quantifying homology classes. *Proc. 25th Ann. Symp. Theoretical Aspects Comput. Sci.*, 169–180, 2008. Dagstuhl Seminar Proceedings 08001. (<http://drops.dagstuhl.de/opus/volltexte/2008/1343/>).
- [19] C. Chen and D. Freedman. Hardness results for homology localization. *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, 1594–1604, 2010.
- [20] C. Chen and D. Freedman. Measuring and computing natural generators for homology groups. *Comput. Geom. Theory Appl.* 43(2):169–181, 2010.
- [21] 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>).
- [22] É. Colin de Verdière and J. Erickson. Tightening non-simple paths and cycles on surfaces. *SIAM J. Comput.* 39(8):3784–3813, 2010.
- [23] É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. *J. ACM* 54(4), 2007.
- [24] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, 278–287, 2007.
- [25] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM J. Comput.* 40(4):1026–1044, 2011.
- [26] T. K. Dey, K. Li, and J. Sun. On computing handle and tunnel loops. *IEEE Proc. Int. Conf. Cyberworlds*, 357–366, 2007.
- [27] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3D models. *ACM Trans. Graphics* 27(3):1–9, 2008. Proc. SIGGRAPH 2008.
- [28] N. M. Dunfield and A. N. Hirani. The least spanning area of a knot and the optimal bounding chain problem. *Proc. 27th Ann. Symp. Comput. Geom.*, 135–144, 2011.
- [29] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.* 3(3):1–27, 1999.
- [30] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica* 27:275–291, 2000.
- [31] J. Erickson. Shortest non-trivial cycles in directed surface graphs. *Proc. 27th Ann. Symp. Comput. Geom.*, 236–243, 2011.
- [32] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.* 31(1):37–59, 2004.
- [33] J. Erickson and A. Nayyeri. Computing replacement paths in surface graphs. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 1347–1354, 2011.
- [34] J. Erickson and A. Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 1166–1176, 2011.
- [35] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1038–1046, 2005.
- [36] J. Erickson and P. Worah. Computing the shortest essential cycle. *Discrete Comput. Geom.* 44(4):912–930, 2010.
- [37] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian J. Math.* 8(399–404), 1956. First published as Research Memorandum RM-1400, The RAND Corporation, Santa Monica, California, November 19, 1954.
- [38] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM J. Comput.* 16(6):1004–1004, 1987.
- [39] M. Grohe. Isomorphism testing for embeddable graphs through definability. *Proc. 32nd ACM Symp. Theory Comput.*, 63–72, 2000.
- [40] J. L. Gross and T. W. Tucker. *Topological graph theory*. Dover Publications, 2001.
- [41] R. Hassin and D. B. Johnson. An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks. *SIAM J. Comput.* 14(3):612–624, 1985.
- [42] A. Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2002. (<http://www.math.cornell.edu/~hatcher/AT/ATpage.html>).
- [43] 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.
- [44] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). *Proc. 6th Ann. ACM Symp. Theory Comput.*, 172–184, 1974.
- [45] A. Itai and Y. Shiloach. Maximum flow in planar networks. *SIAM J. Comput.* 8:135–150, 1979.
- [46] G. F. Italiano, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. *Proc. 43rd Ann. ACM Symp. Theory Comput.*, 313–322, 2011.
- [47] L. Janiga and V. Koubek. Minimum cut in directed planar networks. *Kybernetika* 28(1):37–49, 1992.
- [48] 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>).

- [49] D. R. Karger. Minimum cuts in near-linear time. *J. ACM* 47(1):46–76, 2000.
- [50] P. Klein. Multiple-source shortest paths in planar graphs. *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, 146–155, 2005.
- [51] P. Klein, S. Mozes, and O. Weimann. Shortest paths in directed planar graphs with negative lengths: A linear-space $O(n \log^2 n)$ -time algorithm. *ACM Trans. Algorithms* 6(2):article 30, 2010.
- [52] M. Kutz. Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time. *Proc. 22nd Ann. Symp. Comput. Geom.*, 430–438, 2006.
- [53] J. Łącki and P. Sankowski. Min-cuts and shortest cycles in planar graphs in $O(n \log \log n)$ time. *Proc. 19th Ann. Europ. Symp. Algorithms*, 155–166, 2011. Lecture Notes Comput. Sci. 6942, Springer.
- [54] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.* 16:346–358, 1979.
- [55] M. Mareš. Two linear time algorithms for MST on minor closed graph classes. *Archivum Mathematicum* 40(3):315–320, 2004.
- [56] G. L. Miller. Isomorphism testing for graphs of bounded genus. *Proc. 12th Ann. ACM Symp. Theory Comput.*, 225–235, 1980.
- [57] B. Mohar and C. Thomassen. *Graphs on Surfaces*. Johns Hopkins Univ. Press, 2001.
- [58] S. Mozes and C. Wulff-Nilsen. Shortest paths in planar graphs with real lengths in $O(n \log^2 n / \log \log n)$ time. *Proc. 18th Ann. Europ. Symp. Algorithms*, 206–217, 2010. Lecture Notes Comput. Sci. 6347, Springer-Verlag.
- [59] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* 7:105–113, 1987.
- [60] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math.* 5(1):54–66, 1992.
- [61] D. Pe’er. On minimum spanning trees. Master’s thesis, Hebrew University, 1998. (<http://www.math.ias.edu/~avi/STUDENTS/dpthesis.pdf>).
- [62] 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.
- [63] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, 2nd edition. Graduate Texts in Mathematics 72. Springer-Verlag, 1993.
- [64] J. M. Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. Ph.D. thesis, Princeton Univ., October 1990. (<http://torus.math.uiuc.edu/jms/Papers/thesis/thesis.pdf>).
- [65] S. Tazari and M. Müller-Hannemann. Shortest paths in linear time on minor-closed graph classes, with an application to Steiner tree approximation. *Discrete Appl. Math.* 157:673–684, 2009.
- [66] C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory Ser. B* 48(2):155–177, 1990.
- [67] C. Wulff-Nilsen. Solving the replacement paths problem for planar directed graphs in $O(n \log n)$ time. *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, 756–765, 2010.