Polynomial-Time Approximation Schemes for Subset-Connectivity Problems in Bounded-Genus Graphs

Glencora Borradaile, Erik D. Demaine, Siamak Tazari

Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 171–182
www.stacs-conf.org

POLYNOMIAL-TIME APPROXIMATION SCHEMES FOR SUBSET-CONNECTIVITY PROBLEMS IN BOUNDED-GENUS GRAPHS

GLENCORA BORRADAILE [1] AND ERIK D. DEMAINE [2] AND SIAMAK TAZARI [3]

[1] University of Waterloo
E-mail address: glencora@uwaterloo.ca

[2] Massachusetts Institute of Technology
E-mail address: edemaine@mit.edu

[3] Technische Universität Darmstadt
E-mail address: tazari@cs.tu-darmstadt.de

Abstract. We present the first polynomial-time approximation schemes (PTASes) for the following subset-connectivity problems in edge-weighted graphs of bounded genus: Steiner tree, low-connectivity survivable-network design, and subset TSP. The schemes run in $O(n \log n)$ time for graphs embedded on both orientable and non-orientable surfaces. This work generalizes the PTAS frameworks of Borradaile, Klein, and Mathieu [BMK07, Kle06] from planar graphs to bounded-genus graphs: any future problems shown to admit the required structure theorem for planar graphs will similarly extend to bounded-genus graphs.

1. Introduction

In many practical scenarios of network design, input graphs have a natural drawing on the sphere or equivalently the plane. In most cases, these embeddings have few crossings, either to avoid digging multiple levels of tunnels for fiber or cable or to avoid building overpasses in road networks. But a few crossings are common, and can easily come in bunches where one tunnel or overpass might carry several links or roads. Thus we naturally arrive at graphs of small (bounded) genus, which is the topic of this paper.

We develop a PTAS framework for subset-connectivity problems on edge-weighted graphs of bounded genus. In general, we are given a subset of the nodes, called *terminals*, and the goal is to connect the terminals together with some substructure of the graph by using cost within $1+\varepsilon$ of the minimum possible cost. Our framework applies to three well-studied problems in this framework. In *Steiner tree*, the substructure must be connected, and thus forms a tree. In *subset TSP (Traveling Salesman Problem)*, the substructure must be a cycle; to guarantee existence, the cycle may traverse vertices and edges multiple

Key words and phrases: polynomial-time approximation scheme, bounded-genus graph, embedded graph, Steiner tree, survivable-network design, subset TSP.

Glencora Borradaile was supported by an NSERC Postdoctoral Fellowship; Siamak Tazari was supported by the Deutsche Forschungsgemeinschaft (DFG), grant MU1482/3-1.

times, but pays for each traversal. In $\{0,1,2\}$-*edge-connectivity survivable network design*, the substructure must have $\min\{c_x, c_y\}$ edge-disjoint paths connecting terminals x and y, where each $c_x \in \{0,1,2\}$; we allow the substructure to include multiple copies of an edge in the graph, but pay for each copy. In particular, if $c_x = 1$ for all terminals x and y, then we obtain the Steiner tree problem; if $c_x = 2$ for all terminals x and y, then we obtain the *minimum-cost* 2*-edge-connected submultigraph* problem.

Our framework yields the first PTAS for all of these problems in bounded-genus graphs. These PTASs are efficient, running in $O(f(\varepsilon, g)\, n + h(g)\, n \log n)$ time for graphs embedded on orientable surfaces and non-orientable surfaces. (We usually omit the mention of $f(\varepsilon, g)$ and $h(g)$ by assuming ε and g are constant, but we later bound $f(\varepsilon, g)$ as singly exponential in a polynomial in $1/\varepsilon$ and g and $h(g)$ as singly exponential in g.) In contrast, the problems we consider are APX-complete (and $O(1)$-approximable) for general graphs.

We build upon recent PTAS framework of Borradaile, Klein, and Mathieu [BMK07] for subset-connectivity problems on planar graphs. In fact, our result is strictly more general: any problem to which the previous planar-graph framework applies automatically works in our framework as well, resulting in a PTAS for bounded-genus graphs. For example, Borradaile, Klein and Pritchard [BKP] have recently claimed a PTAS for the $\{0,1,\ldots,k\}$-*edge-connectivity survivable network design* problem using the planar framework. This will imply a similar result in bounded genus graphs. In contrast to the planar-graph framework, our PTASes have the attractive feature that they run correctly on all graphs with the performance degrading with genus.

Our techniques for attacking bounded-genus graphs include two recent results from SODA 2007: decompositions into bounded-treewidth graphs via contractions [DHM07] and fast algorithms for finding the shortest noncontractible cycle [CC07]. We also use a simplified version of an algorithm for finding a short sequence of loops on a topological surface from SODA 2005 [EW05], and sophisticated dynamic programming.

2. Basics

All graphs $G = (V, E)$ have n vertices, m edges and are undirected with edge lengths (weights). The length of an edge e, subgraph H, and set of subgraphs $\mathcal{H}$ are denoted $\ell(e)$, $\ell(H)$ and $\ell(\mathcal{H})$, respectively. The shortest distance between vertices x and y in graph G is denoted $\mathrm{dist}_G(x, y)$. The boundary of a graph G embedded in the plane is denoted by ∂G. For an edge $e = uv$, we define the operation of *contracting* e as identifying u and v and removing all loops and duplicate edges.

We use the basic terminology for embeddings as outlined in [MT01]. In this paper, an embedding refers to a *2-cell embedding*, i.e. a drawing of the vertices and faces of the graph as points and arcs on a surface such that every face is homeomorphic to an open disc. Such an embedding can be described purely combinatorially by specifying a *rotation system*, for the cyclic ordering of edges around vertices of the graph, and a *signature* for each edge of the graph; we use this notion of a *combinatorial embedding*. A combinatorial embedding of a graph G naturally induces such a 2-cell embedding on each subgraph of G. We only consider compact surfaces without boundary. When we refer to a planar embedding, we actually mean an embedding in the 2-sphere. If a surface contains a subset homeomorphic to a Möbius strip, it is *non-orientable*; otherwise it is *orientable*. For a 2-cell embedded graph G with f facial walks, the number $g = 2 + m - n - f$ is called the Euler genus of the surface. The Euler genus is equal to twice the usual genus for orientable surfaces and

equals the usual genus for non-orientable surfaces. The *dual* of an embedded graph G is defined as having the set of faces of G as its vertex set and having an edge between two vertices if the corresponding faces of G are adjacent. We denote the dual graph by $G^\star$ and identify each edge of G with its corresponding edge in $G^\star$. A cycle of an embedded graph is *contractible* if it can be continuously deformed to a point; otherwise it is *non-contractible*. The operation of *cutting along a 2-sided cycle* C is essentially: partition the edges adjacent to C into left and right edges and replace C with two copies C_ℓ and C_r, adjacent to the left or right edges, accordingly. The inside of these new cycles is "patched" with two new faces. If the resulting graph is disconnected, the cycle is called *separating*, otherwise *non-separating*. Cutting along a 1-sided cycle C on non-orientable surfaces is defined similarly, only that C is replaced by one bigger cycle C' that contains every edge of C exactly twice.

Next we define the notions related to treewidth as introduced by Robertson and Seymour [RS86]. A *tree decomposition* of a graph G is a pair (T, χ), where $T = (I, F)$ is a tree and $\chi = \{\chi_i | i \in I\}$ is family of subsets of $V(G)$, called *bags*, such that

(1) every vertex of G appears in some bag of χ;
(2) for every edge $e = uv$ of G, there exists a bag that contains both u and v;
(3) for every vertex v of G, the set of bags that contain v form a connected subtree T_v of T.

The *width* of a tree decomposition is the maximum size of a bag in χ minus 1. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width over all possible tree decompositions of G.

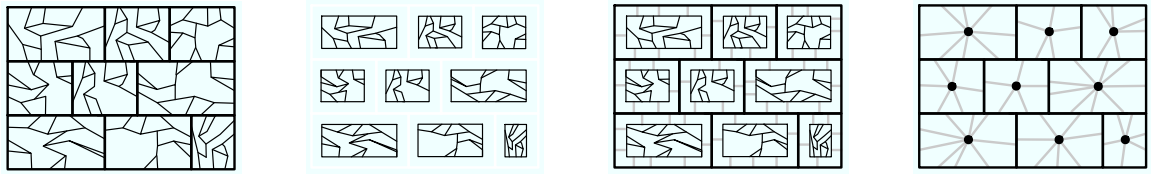
The input graph is $G_0 = (V_0, E_0)$ and has genus g_0; the terminal set is Q. We assume G_0 is equipped with a combinatorial embedding; such an embedding can be found in linear time, if the genus is known to be fixed, see [Moh99]. Let $\mathcal{P}$ be the considered subset-connectivity problem. In Section 5.1, we show how to find a subgraph $G = (V, E)$ of G_0, so that for $0 \leq \varepsilon \leq 1$ any $(1+\varepsilon)$-approximate solution of $\mathcal{P}$ in G_0 also exists in G. Hence, we may use G instead of G_0 in the rest of the paper. Note that as a subgraph of G_0, G is automatically equipped with a combinatorial embedding.

Let OPT denote the length of an optimal Steiner tree spanning terminals Q. We define $\mathrm{OPT}_\mathcal{P}$ to be the length of an optimal solution to problem $\mathcal{P}$. For the problems that we solve, we require that $\mathrm{OPT}_\mathcal{P} = \Theta(\mathrm{OPT})$ and in particular that $\mathrm{OPT} \leq \mathrm{OPT}_\mathcal{P} \leq \xi\,\mathrm{OPT}$. The constant ξ will be used in Section 5 and is equal to 2 for both the subset TSP and $\{0,1,2\}$-edge-connectivity problems. This requirement is also needed for the planar case; see [BK08]. Because $\mathrm{OPT}_\mathcal{P} \geq \mathrm{OPT}$, upper bounds in terms of OPT hold for all the problems herein. As a result, we can safely drop the $\mathcal{P}$ subscript throughout the paper.

We show how to obtain a $(1 + c\varepsilon)\,\mathrm{OPT}_\mathcal{P}$ solution for a fixed constant c. To obtain a $(1+\varepsilon)\,\mathrm{OPT}_\mathcal{P}$ solution, we can simply use $\varepsilon' = \varepsilon/c$ as input to the algorithm.

3. Structure Theorem

In [BMK07] and [BMK07b], Borradaile, Klein and Mathieu developed a PTAS for the Steiner tree problem in planar graphs. The method involves finding a grid-like subgraph called the *mortar graph* that spans the input terminals and has length $O(\mathrm{OPT})$. The set of feasible Steiner trees is restricted to those that cross between adjacent faces of the mortar graph only at a small number (per face of the mortar graph) of pre-designated vertices called *portals*. A Structure Theorem guarantees the existence of a nearly optimal solution (one



(a) (b) (c) (d)

Figure 1: (a) An input graph G with mortar graph MG given by bold edges. (c) The set of bricks corresponding to MG (d) A portal-connected graph, $\mathcal{B}^+(MG, \theta)$. The portal edges are grey. (e) $\mathcal{B}^+(MG, \theta)$ with the bricks contracted, resulting in $\mathcal{B}^\div(MG, \theta)$. The dark vertices are brick vertices.

that has length at most $(1+\varepsilon)\,\mathrm{OPT}$) in this set. We review the details that are relevant to this work and generalize to genus-g graphs.

3.1. Mortar Graph

Here we define the mortar graph in such a way that generalizes to higher genus graphs. A path P in a graph G is *t-short in G* if for every pair of vertices x and y on P, the distance from x to y along P is at most $(1+t)$ times the distance from x to y in G: $\mathrm{dist}_P(x,y) \le (1+t)\mathrm{dist}_G(x,y)$. Given a graph G embedded on a surface and a set of terminals Q, a mortar graph is a subgraph of G with the following properties (where κ, to be defined later, will depend polynomially on ε^{-1} and g):

Definition 3.1 (Mortar Graph and Bricks)**.** Given an embedded graph G and a set of terminals Q, a mortar graph is a subgraph MG of G spanning Q such that each facial walk of MG encloses an area homeomorphic to an open disk. For each face of MG, we construct a *brick* B of G by cutting G along the facial walk. B is the subgraph of G embedded inside the face, including the facial walk. Each brick satisfies the following properties:

(1) B is planar.
(2) The boundary of B is the union of four paths in the clockwise order W, N, E, S.
(3) Every terminal of Q that is in B is on N or on S.
(4) N is 0-short in B, and every proper subpath of S is ε-short in B.
(5) There exists a number $k \le \kappa$ and vertices $s_0, s_1, s_2, \ldots, s_k$ ordered from left to right along S such that, for any vertex x of $S[s_i, s_{i+1})$, the distance from x to s_i along S is less than ε times the distance from x to N in B: $\mathrm{dist}_S(x, s_i) < \varepsilon\,\mathrm{dist}_B(x, N)$.

The mortar graph and the set of bricks are illustrated in Figures 1 (a), (b) and (c). Constructing the mortar graph for planar graphs first involves finding a 2-approximate Steiner tree T and cutting open the graph along T creating a new face with boundary H and then:

(1) Finding shortest paths between certain vertices of H. These paths result in the N and S boundaries of the bricks.
(2) Finding shortest paths between vertices of the paths found in Step 1. These paths are called *columns*, do not cross each other, and have a natural order.
(3) Taking every κth path found in Step 2. These paths are called *supercolumns* and form the E and W boundaries of the bricks.

The mortar graph is composed of the edges of T (equivalently, H) and the edges found in Steps 1 and 3. In [BMK07], it is shown that the total length of the mortar graph edges is $9\varepsilon^{-1}\,\mathrm{OPT}$[1]. For the purposes of this paper, we bound the length of the mortar graph in terms of $\ell(H)$. The following theorem can be easily deduced from [Kle06] and [BMK07]:

Theorem 3.2. [Kle06, BMK07] *Given a planar graph with boundary H containing the terminals and a number κ, there is a mortar graph containing H whose length is at most $(\varepsilon^{-1}+1)^2\ell(H)$ and whose supercolumns have length at most $\frac{1}{\kappa}\varepsilon^{-1}(\varepsilon^{-1}+1)\ell(H)$. The mortar graph can be found in $O(n\log n)$ time.*

We will use this theorem to prove the existence of a mortar graph for genus-g embedded graphs. Section 5 is devoted to proving the following theorem:

Theorem 3.3. *Let G be an embedded graph with fixed genus g. Let Q be a subset of vertices. For $\gamma = 2(8\xi g+2)(\varepsilon^{-1}+1)^2$ and $\kappa = \gamma/\varepsilon$, there is a mortar graph MG of G such that the length of MG is $\le \gamma\,\mathrm{OPT}$ and the supercolumns of MG have length $\le \varepsilon\,\mathrm{OPT}$. The mortar graph can be found in $O(n\log n)$ time.*

3.2. Structural properties

Along with the mortar graph, Borradaile et al. [BMK07b] define an operation $\mathcal{B}^+$ called *brick-copy* that allows a succinct statement of the Structure Theorem. For each brick B, a subset of θ vertices are selected as *portals* such that the distance along ∂B between any vertex and the closest portal is at most $\ell(\partial B)/\theta$. For every brick B, embed B in the corresponding face of MG and connect every portal of B to the corresponding vertex of MG with a zero-length *portal edge*: this defines $\mathcal{B}^+(MG,\theta)$. $\mathcal{B}^+(MG,\theta)$ is illustrated in Figure 1 (d). We denote the set of all portal edges by {portal edges}. The following simple lemma, proved in [BMK07b] holds also for bounded-genus graphs:

Lemma 3.4. [BMK07b] *If A is a connected subgraph of $\mathcal{B}^+(MG,\theta)$, then $A\setminus\{\text{portal edges}\}$ is a connected subgraph of G spanning the same nodes of G.*

The following Structure Theorem is the heart of the correctness of the PTASs.

Theorem 3.5 (Structure Theorem)**.** *Let G be an embedded graph of genus g. There exists a constant $\theta(\gamma,\alpha)$ depending polynomially on γ and α such that*

$$\mathrm{OPT}(\mathcal{B}^+(MG,\theta),Q) \le (1+c\varepsilon)\,\mathrm{OPT}(G,Q).$$

Here $\alpha = o(\varepsilon^{-2.5}\kappa)$ for Steiner tree and $\{0,1,2\}$-edge connectivity and $\alpha = O(\kappa)$ for subset TSP. Recall that κ and γ depend polynomially on ε^{-1} and g (Theorem 3.3).

Since the bricks are always planar, the proof for bounded-genus graphs follows as for the planar cases: see [BMK07b] for the Steiner tree problem and [BK08] for the $\{0,1,2\}$-edge-connectivity problem. In the remainder of this section, we sketch the proof for the subset-TSP problem.

We will use the following lemma due to Arora:

Lemma 3.6 (Patching Lemma [Aro03])**.** *Let S be any line segment of length s and π be a closed path that crosses S at least thrice. Then we can break the path in all but two of these places and add to it line segments lying on S of total length at most $3s$ such that π changes into a closed path π' that crosses S at most twice.*

[1]Actually, they claim that the length is $5\varepsilon^{-1}\,\mathrm{OPT}$. The correction is forthcoming in a journal version.

This lemma applies to embedded graphs as well. Note: the patching process connects paths in the tour that end on a common side of S by a subpath of S.

Let H be a subgraph of a graph G and let P be a path in H. A *joining vertex* of H with P is a vertex of P that is the endpoint of an edge of $H \setminus P$. The proof of Theorem 3.2 for subset-TSP follows from the following TSP Property of Bricks using the same technique as the planar case for, e.g., Steiner tree. See [BMK07b] for details. The condition that E and W are crossed at most twice each is achieved by including two copies of each supercolumn in the solution and rerouting the tour around these copies.

Theorem 3.7 (TSP Property of Bricks)**.** *Let B be a brick of graph G with boundary $N \cup E \cup S \cup W$ (where E and W are supercolumns). Let T be a tour in G such that T crosses E and W at most 2 times each. Let H be the intersection of T with B. Then there is another subgraph of B, H', such that:*
(T1) H' has at most $\alpha(\varepsilon)$ joining vertices with ∂B.
(T2) $\ell(H') \leq (1 + 3\varepsilon)\ell(H)$.
(T3) There is a tour in the edge set $T \setminus H \cup H'$ that spans the vertices in $\partial B \cap T$.
In the above, $\alpha(\varepsilon) = O(\kappa)$.

Proof. Let H be the subgraph of T that is strictly enclosed by B (i.e., H contains no edges of ∂B). We can write H as the union of 4 sets of minimal ∂B-to-∂B paths $\mathcal{P}_S \cup \mathcal{P}_N \cup \mathcal{P}_{EW} \cup \mathcal{P}_{NS}$ where paths in: $\mathcal{P}_S$ start and end on S; $\mathcal{P}_N$ start and end on N; $\mathcal{P}_{EW}$ start on E or W and end on ∂B; $\mathcal{P}_{NS}$ start on S and end on N.

Because T crosses E and W at most 4 times, $|\mathcal{P}_{EW}| \leq 4$ and $\mathcal{P}_{EW}$ results in at most 8 joining vertices with ∂B. For each path $P \in \mathcal{P}_N$, let $\widehat{P}'$ be the minimal subpath of N that spans P's endpoints and let $\widehat{\mathcal{P}}_N$ be the resulting set. Similarly define $\widehat{\mathcal{P}}_S$. Because N is 0-short and S is ε-short,

$$\ell(\widehat{\mathcal{P}}_N) + \ell(\widehat{\mathcal{P}}_S) \leq \ell(\mathcal{P}_N) + (1 + \varepsilon)\ell(\mathcal{P}_S). \tag{3.1}$$

Because $\widehat{\mathcal{P}}_N$ and $\widehat{\mathcal{P}}_S$ are subpaths of ∂B, they result in no joining vertices with ∂B.

It remains to deal with the paths in $\mathcal{P}_{NS}$.

Let $s_0, s_1, s_2, \ldots, s_k$ (where $k \leq \kappa$) be the vertices of S guaranteed by the properties of the bricks (see Section 3.1). Let $\mathcal{X}_i$ be the subset of paths of $\mathcal{P}_{NS}$ that start on S strictly between s_i and s_{i+1}. Let $\mathcal{X}$ be the remaining paths (note: $|\mathcal{X}| \leq \kappa + 1$).

If $|\mathcal{X}_i| > 2$, we do the following: Let P_i be the path in $\mathcal{X}_i$ whose endpoint x on S is closest to s_{i+1}. Let Q_i be the subpath of S from s_i to x. By the properties of the bricks, $\ell(Q_i) \leq \varepsilon\ell(P_i)$. Apply the Patching Lemma to the tour T and path Q_i: as a result, at most two paths of $\mathcal{X}_i$ occur right before (or after) crossings of Q in the new tour T'. Let $\mathcal{X}'_i$ whose endpoints are not crossings in T': $|\mathcal{X}'_i| \leq |\mathcal{X}_i| - 2$. Let $\mathcal{Q}_i$ be the subpaths of Q_i that are added to the tour. By the Patching Lemma,

$$\ell(\mathcal{Q}_i) \leq 3\ell(Q_i) \leq 3\varepsilon\ell(P_i). \tag{3.2}$$

While $|\mathcal{X}'_i| \geq 2$, we do the following: Let P be any path in $\mathcal{X}'_i$. As a result of the patching process P is connected to another path P' in $\mathcal{X}_i$ via a path $Q' \in \mathcal{Q}_i$: that is, $P \cup Q' \cup P'$ is a subpath of the tour. Let L be the minimal subpath of N connecting the endpoints of P and P'. Because N is 0-short, $\ell(L) \leq \ell(P) + \ell(Q') + \ell(P')$. By the patching process, the endpoints of P and P' on S are spanned by a part of the tour "on the other side" of Q_i. Remove Q_i from $\mathcal{Q}_i$ and add L. Remove P and P' from $\mathcal{X}_i$. When we are done, $\mathcal{X}'_i$ is empty.

Let $\widehat{\mathcal{Q}}_i$ be the set resulting from all such replacements. Let $\widehat{\mathcal{P}}_{NS}$ be the union of $\mathcal{X}$, $\mathcal{X}_i \setminus \mathcal{X}_i'$ and $\widehat{\mathcal{Q}}_i$ over all i. Because $\widehat{\mathcal{Q}}_i$ is a set of subpaths of N and S, these paths result in no joining vertices with ∂B. Because $|\mathcal{X}_i \setminus \mathcal{X}_i'| \leq 2$, these paths result in at most 4 joining vertices with ∂B. Because $|\mathcal{X}| \leq \kappa + 1$, these paths have at most $2(\kappa + 1)$ joining vertices with ∂B. Therefore $\widehat{\mathcal{P}}_{NS}$ has at most $6(\kappa + 1)$ joining vertices with ∂B.

Because the only added length is introduced via the patching process, by Equation (3.2),

$$\ell(\widehat{\mathcal{P}}_{NS}) \leq \ell(\mathcal{P}_{NS}) + \sum_i 3\varepsilon\ell(P_i) \leq (1 + 3\varepsilon)\ell(\mathcal{P}_{NS}). \tag{3.3}$$

Let H' be the union of the paths in $\mathcal{P}_{EW}$, $\widehat{\mathcal{P}}_S$, $\widehat{\mathcal{P}}_N$, and $\widehat{\mathcal{P}}_{NS}$. Combining Equations (3.1) and (3.3), we find that $\ell(H') \leq (1 + 3\varepsilon)\ell(H)$. By construction, the edges in $T \setminus H \cup H'$ contains a tour. H' has $6(\kappa + 1) + 8 = 6\kappa + 14$ joining vertices with ∂B. ∎

4. Algorithm

In [Kle06], Klein gave a framework for designing PTASes in planar graphs that is based on finding a *spanner* for a problem, a subgraph containing a nearly optimal solution having length $O(\mathrm{OPT})$. It is possible, using the techniques in this paper and in [DHM07], to find such a spanner for bounded-genus graphs. We omit the details in favour of generalizing the framework of Borradaile et al. [BMK07b]. While both methods result in $O(n \log n)$ algorithms, the first method is doubly exponential in a polynomial in g and ε^{-1} and the second is singly exponential.[2]

The idea in [BMK07b] is to perform dynamic programming over the bricks of the brick decomposition after performing a *thinning* step which groups the bricks into manageable groups. To this end, the operation *brick-contraction* $\mathcal{B}^{\div}$ is defined to be the application of the operation $\mathcal{B}^{+}$ followed by contracting each brick to become a single vertex of degree at most θ (see Figure 1(e)). The thinning algorithm decomposes the mortar graph MG into parts, called *parcels*, of *bounded dual radius* (implying bounded treewidth). Applying $\mathcal{B}^{\div}$ to each parcel maintains bounded dual radius. The dynamic program computes optimal Steiner trees inside the bricks using the method of [EMAFV87] only at the leaves of the dynamic programming tree, thus eliminating the need of an a-priori constructed spanner. The interaction between subproblems of the dynamic programming is restricted to the portals, of which there are a few.

For embedded graphs with genus > 0, the concept of bounded dual radius does not apply in the same way. We deal with treewidth directly by applying the following Contraction Decomposition Theorem due to Demaine et al. [DHM07]:

Theorem 4.1. [DHM07, Theorem 1.1] *For a fixed genus g, and any integer $k \geq 2$ and for every graph G of Euler genus at most g, the edges of G can be partitioned into k sets such that contracting any one of the sets results in a graph of treewidth at most $O(g^2k)$. Furthermore, such a partition can be found in $O(g^{5/2}n^{3/2} \log n)$ time.*

[2]For the subset-TSP problem, it is possible to obtain a singly exponential algorithm by following the spanner construction of Klein [Kle06] after performing the planarizing step (Lemma 5.2). Our presentation here is chosen to unify the methods for all problems studied.

(Recent techniques [CC07] have improved the above running time to $O(n \log n)$.)

We apply the above theorem directly to $\mathcal{B}^{\div}(MG)$ and contract a set of edges $S^\star$ in $\mathcal{B}^{\div}(MG)$. In $\mathcal{B}^{\div}(MG)$, we modify the definition of contraction: after contracting an edge, we do not delete parallel portal edges. Because portal edges connect the mortar graph to the bricks, they are not parallel in the graph in which we find a solution via dynamic programming. With this modified definition, we have the following algorithm:

THINNING(G, MG):

(1) Assign the weight $\ell(\partial F)$ to each portal edge e enclosed in a face F of MG.
(2) Apply the contraction decomposition of Theorem 4.1 to $\mathcal{B}^{\div}(MG)$ with $k := 3\theta\gamma\varepsilon^{-1}$ to obtain edge sets $S_1, \ldots, S_k$; let $S^\star$ be the set of minimum weight.
(3) If $S^\star$ includes a portal edge e of a brick B enclosed in a face F of MG, add ∂F to $S^\star$ and mark B as ignored.
(4) Let $K := \mathcal{B}^{\div}(MG)/S^\star$ and (T, χ) be a tree decomposition of width $O(g^2k)$ of K.
(5) For each vertex b of K that represents an unignored contracted brick with portals $\{p_1, \ldots, p_\theta\}$:
    5.1. Replace every occurence of b in χ with $\{p_1, \ldots, p_\theta\}$;
    5.2. Add a bag $\{b, p_1, \ldots, p_\theta\}$ to χ and connect it to a bag containing $\{p_1, \ldots, p_\theta\}$.
(6) Reset the weight of the portal edges back to zero.
(7) Return (T, χ) and $S^\star$.

Lemma 4.2. *The algorithm* THINNING(G, MG) *returns a set of edges $S^\star$ and a tree decomposition (T, χ) of $\mathcal{B}^{\div}(MG)$, so that*

(i) the treewidth of (T, χ) is at most ω where $\omega(\varepsilon^{-1}, g) = O(g^2k\theta)$;
(ii) every brick is either marked as ignored or none of its portal edges are in $S^\star$;
(iii) $\ell(S^\star) \le \varepsilon\,\mathrm{OPT}$;

Proof. We first verify that (T, χ) is indeed a tree decomposition. For a vertex v and a tree decomposition (T', χ'), let T'_v denote the subtree of T' that contains v in all of its bags. Let us denote the tree decomposition of step (4) by (T^0, χ^0). For each brick vertex b and each of its portals p_i, we know that T^0_b is connected and $T^0_{p_i}$ is connected and that these two subtrees intersect; it follows that after the replacement in step (5.2), we have that $T_{p_i} = T^0_b \cup T^0_{p_i}$ is a connected subtree of T and hence, (T, χ) is a correct tree decomposition. Note that Theorem 4.1 guarantees a tree decomposition of width $O(g^2k)$ if any of $S_1, \ldots, S_k$ are contracted; and in step (3), we only add to the set of edges to be contracted. Hence, the treewidth of (T^0, χ^0) is indeed $O(g^2k)$ and with the construction in line (5.1), the size of each bag will be multiplied by a factor of at most θ. This shows the correctness of claim (i). The correctness of claim (ii) is immediate from the construction in line (3). It remains to verify claim (iii).

Let L denote the weight of $\mathcal{B}^{\div}(MG)$ after setting the weights of the portal edges according to step (1) of the algorithm. We have that

$$L \le \ell(MG) + \sum_F \ell(\partial F)\theta \le \gamma\,\mathrm{OPT} + \theta \sum_F \ell(\partial F) \le \gamma\,\mathrm{OPT} + \theta \cdot 2\gamma\,\mathrm{OPT} \le 3\theta\gamma\,\mathrm{OPT}\ .$$

Hence, the weight of $S^\star$, as selected in step (2), is at most $L/k \le \frac{3\theta\gamma\,\mathrm{OPT}}{3\theta\gamma\varepsilon^{-1}} \le \varepsilon\,\mathrm{OPT}$. The operation in step (3) does not add to the weight of $S^\star$: when the boundary of a face F is

added to $S^\star$, its weight is subtracted again when resetting the weights of the portal edges back to 0 in step (6). ∎

If a brick is "ignored" by THINNING, the boundary of its enclosing mortar graph face is completely added to $S^\star$. Because $S^\star$ can be added to the final solution, every potential connection through that brick can be rerouted through $S^\star$ around the boundary of the brick. The interior of the brick is not needed.

An almost standard dynamic programming algorithm for bounded-treewidth graphs (cf. [AP89, KS90]) can be applied to K and (T, χ). However, for the leaves of the tree decomposition that are added in step (5.2) of the THINNING procedure, the cost of a subset of portal edges is calculated as, e.g., the cost of the minimum Steiner tree interconnecting these portals in the corresponding brick. Because the bricks are planar, this cost can be calculated by the algorithm of Erickson et al. [EMAFV87] for Steiner tree or [BK08] for 2-edge connectivity. Because all the portal edges of this brick are present in this bag (recall that we do not delete parallel portal edges after contractions), all possible solutions restricted to the corresponding brick will be considered. Because the contracted brick vertices only appear in leaves of the dynamic programming tree, the rest of the dynamic programming algorithm can be carried out as in the standard case, considering the portal edges, whenever they occur again, as having zero length.

Analysis of the running time. As was shown in [BMK07b], the total time spent in the leaves of the dynamic programming is $O(4^\theta n)$. The rest of the dynamic programming takes time $O(2^{\mathrm{poly}(\omega)} n)$. The running time of the thinning algorithm is dominated by the contraction decomposition of Theorem 4.1 which is $O(n \log n)$ [CC07]. Hence, the total time is $O(2^{\mathrm{poly}(\omega)} n + n \log n)$ for the general case; particularly, this is singly exponential in ε^{-1} and g, as desired.

5. Constructing a Mortar Graph for Bounded-Genus Graphs

Let $G_0 = (V_0, E_0)$ be the input graph of genus g_0 and Q be the terminal set. In a first preprocessing step, we delete a number of unnecessary vertices and edges of G_0 to obtain a graph $G = (V, E)$ of genus $g \leq g_0$ that still contains every $(1+\varepsilon)$-approximate solution for terminal set Q for all $0 \leq \varepsilon \leq 1$ while fulfilling certain bounds on the length of shortest paths. In the next step, we find a *cut graph* CG of G that contains all terminals and whose length is bounded by a constant times OPT. We cut the graph open along CG, so that it becomes a planar graph with a simple cycle σ as boundary, where the length of σ is twice that of CG. See Figure 2 for an illustration. Afterwards, the remaining steps of building the mortar graph can be the same as in the planar case, by way of Theorem 3.2.

For an edge $e = vw$ in G_0, we let $\mathrm{dist}_{G_0}(r, e) = \min\{\mathrm{dist}_{G_0}(r, v), \mathrm{dist}_{G_0}(r, w)\} + \ell(e)$ and say that e is *at distance* $\mathrm{dist}_{G_0}(r, e)$ from r. If the root vertex represents a contracted graph H, we use the same terminology with respect to H.

5.1. Preprocessing the Input Graph

Our first step is to apply the following preprocessing procedure:

PREPROCESS(G_0, Q, ξ):
(1) Find a 2-approximate Steiner tree T_0 of G_0 and Q; contract T_0 to a vertex r.
(2) Find a shortest-path tree rooted at r.
(3) Delete all vertices v and edges e of G_0 with $\text{dist}_{G_0}(r, v), \text{dist}_{G_0}(r, e) > 2\xi\ell(T_0)$.

Any deleted vertex or edge is at distance $> 2\xi\ell(T_0) > 2\xi\,\text{OPT}$ from any terminal and hence, can not be part of a $(1+\varepsilon)$-approximation for any $0 \le \varepsilon \le 1$. We call the resulting graph $G = (V, E)$ and henceforth use G instead of G_0 in our algorithm. The preprocessing step can be accomplished in linear time: T_0 can be calculated with the recent improvement on Mehlhorn's algorithm [Meh88] by Tazari and Müller-Hannemann [TMH08], and the shortest path tree with Henzinger et al.'s algorithm [HKRS97]. Trivially, we have

Proposition 5.1. *Any vertex and any edge of G is at distance at most $4\xi\,\text{OPT}$ from T_0.*

5.2. Constructing the Cut Graph

Start again with T_0 contracted to a vertex r. We construct our cut graph as a *system of loops* rooted at r: iteratively find short non-separating cycles through r and cut the graph open along each cycle. Erickson and Whittlesey [EW05] showed that, for orientable surfaces, taking the *shortest* applicable cycle at each step results in the shortest system of loops through r. They suggest an implementation of their algorithm using the *tree-cotree decomposition*, introduced by Eppstein [Epp03], that runs in linear time on bounded-genus graphs. A tree-cotree decomposition of an embedded graph G is a triple $(T, T^\star, X)$, so that T is a spanning tree of G rooted at r, $T^\star$ is a spanning tree of the dual of $G \setminus T$, and X is the remaining set of edges (recall that we identify the edges of G and $G^\star$). Eppstein showed that the set of elementary cycles $\{\text{loop}(T, e) : e \in X\}$ is a cut graph of G where $\text{loop}(T, e)$ is the cycle formed by the paths in T from r to the endpoints of e plus the edge e. Eppstein's decomposition also works for non-orientable embeddings. As we only need to bound the length of our cut graph, we present a simpler algorithm:

PLANARIZE(G, R):
(1) Contract T_0 to become a single vertex r.
(2) Find a shortest paths tree SPT rooted at r.
(3) Uncontract r and set $T_1 = T_0 \cup SPT$. (T_1 is a spanning tree of G.)
(4) Find a spanning tree $T_1^\star$ in $(G \setminus T_1)^\star$. ($T_1^\star$ is a spanning tree of $G^\star$.)
(5) Let $X = E \setminus (T_1 \cup T_1^\star)$.
(6) Return $CG = T_0 \cup \{\text{loop}(T_1, e) : e \in X\}$.

Lemma 5.2. *The algorithm* PLANARIZE *returns a cut graph CG, so that cutting G open along CG results in a planar graph G_p with a distinguished face whose facial walk σ*

(1) is a simple cycle;
(2) contains all terminals (some terminals might appear more than once as multiple copies might be created during the cutting process);
(3) has length $\ell(\sigma) \le 2(8\xi g + 2)\,\text{OPT}$.

The algorithm can be implemented in linear time.

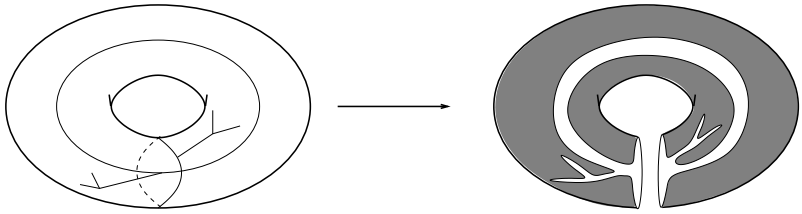


Figure 2: (a) a cut graph of a tree drawn on a torus; (b) the result of cutting the surface open along the cut graph: the shaded area is now homeomorphic to a disc and the white area is the new distinguished face of the planarized surface.

Proof. Clearly, $(T_1, T_1^\star, X)$ is tree-cotree decomposition of G and so, by Eppstein's lemma [Epp03], CG is a cut graph. By Euler's formula, we get that $|X| = g$, the Euler genus of G. Each edge $e = vw \in X$ completes a (non-surface-separating, not-necessarily simple) cycle as follows: a shortest path P_1 from T_0 to v, the edge e, a shortest path P_2 from w to T_0 and possibly a path P_3 in T_0. By Proposition 5.1, we know that e is at distance at most $4\xi\,\mathrm{OPT}$ from T_0 and so, P_1, P_2, and at least one of $P_1 \cup \{e\}$ and $P_2 \cup \{e\}$ have length at most $4\xi\,\mathrm{OPT}$. Hence, we have that $\ell(P_1 \cup \{e\} \cup P_2) \leq 8\xi\,\mathrm{OPT}$. Because there are (exactly) g such cycles in CG, we get that

$$\ell(CG) \leq g \cdot 8\xi\,\mathrm{OPT} + \ell(T_0) \leq (8\xi g + 2)\,\mathrm{OPT}\,.$$

Because CG is a cut graph, it follows that it consists of a single facial walk σ'; this follows easily from Euler's formula, because CG has Euler genus g (because $G \setminus CG$ is planar), with some k vertices and $k + g - 1$ edges. So, σ' contains every edge of CG exactly twice (cf. [MT01]), i.e. $\ell(\sigma') = 2\ell(CG)$. Cutting the graph open along σ' results in a planar graph with a simple cycle $\sigma = \sigma'$ as its boundary, as desired (see Fig. 2).

As mentioned in the previous section, T_0 and SPT can be computed in linear time on bounded-genus graphs [HKRS97, TMH08]. $T_1^\star$ can be obtained, for example, by a simple breadth-first-search in the dual. The remaining steps can also easily be implemented in linear time. ∎

5.3. Proof of Theorem 3.3

We complete the construction of a mortar graph for genus-g embedded graphs.

Let G_p be the result of planarizing G as guaranteed by Lemma 5.2. G_p is a planar graph with boundary σ spanning Q and with length $\leq 2(8\xi g + 2)\,\mathrm{OPT}$. Let MG be the mortar graph guaranteed by Theorem 3.2. Every edge of MG corresponds to an edge of G. Let MG' be the subgraph of G composed of edges corresponding to MG. Every face f of MG (other than σ) corresponds to a face f' of MG' and the interior of f' is homeomorphic to a disk on the surface in which G is embedded. It is easy to verify that MG' is a mortar graph of G satisfying the length bounds of Theorem 3.3.

Acknowledgements. The authors thank Jeff Erickson for pointing out that Theorem 4.1 can be implemented in $O(n \log n)$ time in both orientable and non-orientable surfaces.

References

[AP89] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989.

[Aro03] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. *Mathematical Programming*, 97(1–2):43–69, 2003.

[BK08] G. Borradaile and P. N. Klein. The two-edge connectivity survivable network problem in planar graphs. In *ICALP '08: Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 485–501. Springer, 2008.

[BMK07] G. Borradaile, C. Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1285–1294, 2007.

[BKP] G. Borradaile, P. N. Klein, and D. Pritchard. A polynomial-time approximation scheme for the survivable network problem in planar graphs. In preparation.

[BMK07b] G. Borradaile, C. Mathieu, and P. N. Klein. Steiner tree in planar graphs: An $O(n \log n)$ approximation scheme with singly exponential dependence on epsilon. In *WADS '07: Proceedings of the 10th Workshop on Algorithms and Data Structures*, volume 4619 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 2007.

[CC07] S. Cabello and E. W. Chambers. Multiple source shortest paths in a genus g graph. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 89–97, Philadelphia, PA, USA, 2007.

[DHM07] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–287, Philadelphia, PA, USA, 2007.

[EMAFV87] R. E. Erickson, C. L. Monma, and Jr. A. F. Veinott. Send-and-split method for minimum-concave-cost network flows. *Math. Oper. Res.*, 12(4):634–664, 1987.

[Epp03] D. Eppstein. Dynamic generators of topologically embedded graphs. In *SODA '03: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 599–608, Philadelphia, PA, USA, 2003.

[EW05] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *SODA '05: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1046, Philadelphia, PA, USA, 2005.

[HKRS97] M. Henzinger, P. N. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1):3–23, 1997.

[Kle06] P. N. Klein. A subset spanner for planar graphs, with application to subset TSP. In *STOC '06: Proceedings of the 38th ACM Symposium on Theory of Computing*, pages 749–756, 2006.

[KS90] E. Korach and N. Solel. Linear time algorithm for minimum weight Steiner tree in graphs with bounded treewidth. Manuscript, 1990.

[Meh88] K. Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.

[Moh99] B. Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics*, 12(1):6–26, 1999.

[MT01] B. Mohar and C. Thomassen. *Graphs on Surfaces*. The John Hopkins University Press, 2001.

[RS86] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

[TMH08] S. Tazari and M. Müller-Hannemann. Shortest paths in linear time on minor-closed graph classes, with an application to Steiner tree approximation. *Discrete Applied Mathematics*, in press, 2008. An extended abstract appeared in WG '08: Proceedings of the 34th Workshop on Graph Theoretic Concepts in Computer Science, LNCS 5344, pp. 360–371, Springer, 2008.