Discrete Comput Geom 37:213–235 (2007)
DOI: 10.1007/s00454-006-1292-5

Discrete & Computational
Geometry

Finding Shortest Non-Separating and Non-Contractible Cycles for Topologically Embedded Graphs*

Sergio Cabello[1] and Bojan Mohar[2]

[1]Department of Mathematics, Institute for Mathematics, Physics and Mechanics,
University of Ljubljana, 1000 Ljubljana, Slovenia
sergio.cabello@imfm.uni-lj.si

[2]Department of Mathematics, Faculty of Mathematics and Physics,
University of Ljubljana, 1000 Ljubljana, Slovenia
bojan.mohar@fmf.uni-lj.si

Abstract. We present an algorithm for finding shortest surface non-separating cycles in graphs embedded on surfaces in $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$ time, where V is the number of vertices in the graph and g is the genus of the surface. If $g = o(V^{1/3})$, this represents an improvement over previous results by Thomassen, and Erickson and Har-Peled. We also give algorithms to find a shortest non-contractible cycle in $O(g^{O(g)}V^{3/2})$ time, which improves previous results for fixed genus.

This result can be applied for computing the face-width and the non-separating face-width of embedded graphs. Using similar ideas we provide the first near-linear running time algorithm for computing the face-width of a graph embedded on the projective plane, and an algorithm to find the face-width of embedded toroidal graphs in $O(V^{5/4}\log V)$ time.

1. Introduction

Cutting a surface for reducing its topological complexity is a common technique used in geometric computing and topological graph theory. Erickson and Har-Peled [10] discuss the relevance of cutting a surface to get a topological disk in computer graphics. Colin de Verdière [3] describes applications that algorithmical problems involving curves on topological surfaces have in other fields.

* A preliminary version was presented at the 13th Annual European Symposium on Algorithms (ESA 2005) [2]. Sergio Cabello was partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship. Bojan Mohar was supported in part by the Ministry of Higher Education, Science and Technology of Slovenia, Research Project L1-5014 and Research Program P1-0297.

Many results in topological graph theory rely on the concept of face-width, sometimes called representativity, which is a parameter that quantifies local planarity and density of embeddings. The face-width is closely related to the edge-width, the minimum number of vertices of any shortest non-contractible cycle of an embedded graph [20]. Among some relevant applications, face-width plays a fundamental role in the graph minors theory of Robertson and Seymour, and large face-width implies that there exists a collection of cycles that are far apart from each other, and after cutting along them, a planar graph is obtained. By doing so, many computational problems for locally planar graphs on general surfaces can be reduced to corresponding problems on planar graphs. See Chapter 5 of [20] for further details. The efficiency of algorithmical counterparts of several of these results passes through the efficient computation of face-width. The same can be said for the non-separating counterparts of the width parameters, where the surface non-separating cycles are considered instead of non-contractible ones.

In this work we focus on what may be considered the most natural problem for graphs embedded on surfaces: finding a shortest non-contractible and a shortest surface non-separating cycle. Our results give polynomial-time improvements over previous algorithms for low-genus embeddings of graphs (in the non-separating case) or for embeddings of graphs in a fixed surface (in the non-contractible case). In particular, we improve previous algorithms for computing the face-width and the edge-width of embedded graphs. In our approach we reduce the problem to that of computing the distance between a few pairs of vertices, what some authors have called the *k-pairs shortest path problem*.

1.1. *Overview of the Results*

Let G be a graph with V vertices that is 2-cell embedded on a (possibly non-orientable) surface Σ of genus g, and with non-negative weights on the edges, representing edge-lengths. We further assume that the embedding does not have faces consisting of one or two edges. Our main contributions are the following:

- We find a shortest surface non-separating cycle of G in $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$ time. This result relies on a characterization of the surface non-separating cycles given in Section 4. The algorithmical implications of this characterization are described in Section 5.
- We find a shortest non-contractible cycle in $O(g^{O(g)}V^{3/2})$ time. Then $g = O(1)$, the running time becomes $O(V^{3/2})$, which represents an improvement over previous results. This is achieved by considering a small portion of the universal cover. See Section 6.
- We compute the non-separating face-width and edge-width of G in $O(g^{3/2}V^{3/2} + g^{5/2}V^{1/2})$ time. We can also compute the face-width and edge-width of G in $O(g^{O(g)}V^{3/2})$ time. For the non-separating face-width, this is a particular case of the result mentioned above where a log factor can be shaved off. See Section 7.
- For graphs embedded on the projective plane or the torus we can compute the face-width in $O(V^2\log^2 V)$ or $O(V^{5/4}\log V)$ time, respectively. This is described in Sections 7.2 and 7.3.

Although the general approach is common in all our results, the details are quite different for each case. The overview of the technique is as follows. We find a set of generators either for the first homology group (in the non-separating case) or the fundamental group (in the non-contractible case) that is made of a few shortest paths. It is then possible to show that the shortest cycles we are interested in (non-separating or non-contractible ones) intersect these generators according to certain patterns, and this allows us to reduce the problem to computing distances between pairs of vertices in associated graphs.

The paper is organized as follows. The remainder of this section describes the most relevant related work, and in Section 2 we introduce the basic background. In Section 3 we describe basic tools that are used through the paper. The rest of the sections are as described above; we conclude in Section 8.

1.2. *Related Previous Work*

Thomassen [23] was the first to give a polynomial-time algorithm for finding a shortest non-separating and a shortest non-contractible cycle in a graph on a surface; see also Chapter 4 of [20]. Although Thomassen does not claim any specific running time, his algorithm tries a quadratic number of cycles, and for each one it has to decide if it is non-separating or non-contractible. This yields a rough estimate $O(V(V+g)^2)$ for its running time. More generally, his algorithm can be used for computing in polynomial time a shortest cycle in any class $\mathcal{C}$ of cycles that satisfy the so-called 3-*path-condition*: if u, v are vertices of G and P_1, P_2, P_3 are internally disjoint paths joining u and v, and if two of the three cycles $C_{i,j} = P_i \cup P_j$ $(i \neq j)$ are not in $\mathcal{C}$, then the third one is also not in $\mathcal{C}$. The class of one-sided cycles for embedded graphs is another relevant family of cycles that satisfy the 3-path-condition.

Erickson and Har-Peled [10] considered the problem of computing a *cutset* of minimum length, that is, a subgraph $C \subseteq G$ of minimum length such that $\Sigma \backslash C$ is a topological disk. They show that the problem is NP-hard when the genus is not fixed, provide a polynomial-time algorithm for fixed surfaces, and provide efficient approximation algorithms.

More relevant for our work, Erickson and Har-Peled [10] also give the currently best algorithms for the problems we consider: they find a shortest non-contractible cycle in $O(V^2 \log V + Vg)$ time, and a shortest non-separating cycle in $O(V^2 \log V + Vg \log g)$ time. Regarding the edge-width or the face-width, we are not aware of any specific research to compute them, and therefore these general algorithms were also the asymptotically fastest for these problems. Their approach is based on showing that a shortest non-contractible (resp. non-separating) loop through a fixed vertex can be computed in $O(V \log V + g)$ (resp. $O(V \log V + g \log g)$) time, and then iterate this procedure over all vertices of the graph. Erickson and Har-Peled also provide an algorithm that in $O(g(V+g) \log V)$ time finds a non-separating (or non-contractible) cycle whose length is at most twice the length of a shortest one.

Several other algorithmical problems for graphs embedded on surfaces have been considered. Colin de Verdière and Lazarus [5] considered the problem of finding a shortest cycle with a fixed base point within a given homotopy class, as well as a system of loops in a given homotopy class. Under some realistic assumption on the edge-lengths, they provide polynomial-time algorithms for both problems. The same authors have also studied algorithms for finding optimal pants decompositions [4].

Eppstein [9] discusses how to use the tree–cotree partition for dynamically maintaining properties from a graph under several operations. For example, he can maintain the minimum and maximum spanning tree under edge insertions, edge deletions, and edge reweightings.

Very recently, Erickson and Whittlesey [11] have shown that the greedy homotopy generators through a fixed basepoint determine a shortest set of loops generating the fundamental group. They can compute this optimal system of loops and represent it implicitly in $O((V+g)\log V)$ time; an explicit representation may need $\Theta(gV)$ space. Other known results for curves embedded on topological surfaces include [6], [7], [19], and [24]; see also [21], [22], and the references therein.

2. Background

We describe most of the topological and graph-theoretical background that is used through the paper.

Topology. We consider *surfaces* Σ that are connected, compact, Hausdorff topological spaces in which each point has a neighborhood that is homeomorphic to $\mathbb{R}^2$. In particular, we only consider surfaces without boundary. A *closed curve* is a continuous function of the circle S^1 in Σ. A *simple closed curve* is an injective closed curve. Two closed curves are *homotopic* if there is a continuous deformation of one onto the other, that is, if there is a continuous function from the cylinder $S^1 \times [0,1]$ to Σ such that each boundary of the cylinder is mapped to one of the curves.

A closed curve is *contractible* if it is homotopic to a constant (a curve whose image is a single point); otherwise it is *non-contractible*. A closed curve is (*surface*) *non-separating* if the removal of its image keeps the surface connected. Another definition of a non-separating closed curve is that it is a non-trivial element of the ($\mathbb{Z}_2$-)homology group. Although these two definitions can be seen to be equivalent for *simple* closed curves (see Lemma 4), they are not equivalent in general, as can be seen by concatenating two copies of the same non-separating cycle. Although the first definition is more intuitive, it has the drawback problem of not being stable under small perturbations of the curves. However, as we will see later, it turns out that both concepts agree when it comes to finding a shortest non-separating closed curve.

If c is a simple closed curve, a small band around c is either a cylinder or a Möbius band. In the former case, we say that c is two-sided, otherwise it is one-sided. In an orientable surface all the closed curves are two-sided. We refer to Chapter 1 of [14] and to Chapter 4 of [20] for additional details.

Every graph, viewed as a one-dimensional cell complex, determines a topological space and we can speak of embeddings (continuous 1–1 maps) in surfaces. It is customary to consider only 2-*cell embeddings*, in which every *face* (i.e., a connected component of the surface after we remove the image of the graph) is homeomorphic to an open disk. In this paper we also make such a restriction and, henceforth, *every embedding will be*

2-*cell*. As shown in Chapter 3 of [20], such embeddings admit a simple combinatorial description whose development is usually attributed to Heffter, Edmonds, and Ringel. See also a paragraph below.

If G is a graph with V vertices, E edges, and is embedded in a surface Σ with F faces, then *Euler's formula* holds:

$$V - E + F = 2 - g,$$

where g is a non-negative integer, called the (*Euler*) *genus* of Σ.

If Σ is orientable, then $\bar{g} = \frac{1}{2}g$ is also an integer, known as the *genus* of Σ. Since we will meet the genus only in the O-notation, there is no need to distinguish between g and $\bar{g}$.

Walks in Graphs. In a graph G, a *walk* is an alternating sequence of vertices and edges $v_1 e_1 v_2 e_2 \cdots v_k e_k v_{k+1}$, starting and ending with a vertex, and such that each edge e_i connects the vertices v_i and v_{i+1}; a *path* is a walk where all vertices are distinct; a *segment* is a subwalk; a *loop* is a walk where the first and last vertex are the same; a *cycle* is a loop without repeated vertices. In a loop the first and last vertex is a distinguished base vertex. If G is on a surface Σ, the loops and cycles of G correspond to closed curves and simple closed curves in Σ, respectively, and therefore we can refer to their topological properties.

In general, we consider graphs with *non-negative* edge-lengths, that is, we have a function $w\colon E \to \mathbb{R}^+$ describing the length of the edges. The *length* of a walk is the sum of the weights of its edges, counted with multiplicity if they occur on the walk more than once. We also assume that no cycle has length zero.

It is not difficult to see that any shortest non-contractible loop in G must be a cycle. Indeed, it holds that the concatenation of two contractible loops is a contractible loop. Therefore, if a loop is non-contractible and it is not a cycle, then it must be the concatenation of two shorter loops, and it cannot be that both are contractible.

Similarly, any shortest non-separating loop in G must be a cycle. Indeed, if a loop ℓ is non-separating and it is not a cycle, then ℓ must be the concatenation of two loops ℓ', ℓ''. However, the image of ℓ' or ℓ'' is a subset of the image of ℓ, and therefore they cannot disconnect the surface.

Cutset, *System of Cycles*, *and Universal Cover.* In G a *cutset* is a set of edges whose complement is a topological disk. We reserve the term *system of cycles* for a set of cycles in G through a common vertex x, and pairwise disjoint otherwise, whose edges form a cutset. It is known that a system of cycles consists of exactly g cycles. Note that, in general, a system of cycles does not need to exist in G, since it requires vertices with high degrees. However, after a suitable transformation in the graph, we may assume its existence.

Let D be the topological disk obtained from a cutset. Several (possibly an infinite number of) copies of D can be glued together to obtain a so-called *covering space* X, which locally looks like Σ. In this case the projection $\pi\colon\ X \to \Sigma$ that sends each point of X to its original point in D is locally a homeomorphism. When X has the property that it does not contain any non-contractible cycle, then X is the so-called *universal cover*, which can be shown to be unique up to homeomorphism. It is known that the universal cover of the projective plane or the sphere is the sphere, while the universal cover of any other surface is the plane. Since the surface Σ and its universal cover U look locally the same, any curve c in Σ can be routed locally in the same way in U, giving the so-called lift $\tilde{c}$ of c in U. It turns out that a closed curve in Σ is contractible if and only if its lift finishes at the same point where it starts. See Chapter 1 of [14] for a comprehensive treatment of covering spaces.

Representation of Embedded Graphs. For computational purposes, an embedded graph can be represented as described by Eppstein [9]. However, for our purposes, the Heffter–Edmonds–Ringel representation is used. It is enough to specify for each vertex v the circular ordering of the edges emanating from v, where the ordering coincides with that on the surface in a small disk neighborhood of v. Additionally we need the *signature* $\lambda(e) \in \{+1, -1\}$ for each edge $e \in E(G)$. The negative signature of e tells us that the selected circular ordering around vertices changes from clockwise to anti-clockwise when passing from one end of the edge to the other. If the embedding is in an orientable surface, all the signatures can be made positive, and there is no need to specify them. A cycle in the embedding is two-sided if the product of the signatures of its edges is positive; otherwise, it is a one-sided cycle. It is known that this representation uniquely determines the embedding of G, up to homeomorphism. Knowing the circular ordering at each vertex and the signatures, one can compute the set of facial walks in linear time. See Chapter 3 of [20] if more detail is needed.

In an embedded graph, a 1-*gon* is a face defined by a single edge of the graph, and a 2-*gon* is a face defined by two (parallel) edges. For the problems that we consider, any edge defining a 1-gon is not relevant, since it cannot appear in a shortest non-contractible or non-separating cycle. The same applies to the longest edge of a 2-gon. Since a simple preprocessing taking $O(E)$ time allows us to obtain an embedding without 1- or 2-gons, we assume, henceforth, that the graph is *embedded without* 1- *or* 2-*gons*.

We use V to denote the number of vertices in G, and g for the (Euler) genus of the surface Σ. That the graph G has $\Theta(V + g)$ edges follows from Euler's formula and the fact that the number of faces is at most $\frac{2}{3}E$. Asymptotically, we may consider $V + g$ as the measure of the size of the input. Observe that this is different from the approach followed by some authors, where $n = V + E$ is used for the size of the graph.

We use the notation $G\text{✂}C$ for the graph obtained by cutting the embedded graph G along a cycle C. Each vertex $v \in C$ gives rise to two vertices v', v'' in $G\text{✂}C$. If C is a two-sided cycle, then it gives rise to two cycles C' and C'' in $G\text{✂}C$ whose vertices are $\{v' \mid v \in V(C)\}$ and $\{v'' \mid v \in V(C)\}$, respectively. If C is one-sided, then it gives rise to a cycle C' in $G\text{✂}C$ whose length is twice the length of C, in which each vertex v of C corresponds to two diagonally opposite vertices v', v'' on C'. Observe that if G is embedded in a surface Σ of Euler genus g, then $G\text{✂}C$ is naturally embedded in the surface obtained after cutting Σ along C and then pasting disks to each of the boundaries that were created. Therefore, $G\text{✂}C$ is embedded in a surface of Euler genus $g - 1$ (if C is a one-sided curve in Σ) or $g - 2$ (if C is two-sided). When $\mathcal{C} = \{C_1, \ldots, C_t)$ is a system of cycles, we use $G\text{✂}\mathcal{C}$ to denote the graph obtained by cutting G along the cycles $C_1, \ldots, C_t$, in this order, and without pasting disks. Note that for $i \geq 2$, the cycle

C_i may be a path connecting two vertices of the boundary of $G\text{✄}\{C_1, \ldots, C_{i-1}\}$. If $\mathcal{C}$ is a system of cycles, then $G\text{✄}\mathcal{C}$ is a planar graph.

Distances in Embedded Graphs. For two vertices $u, v \in V(G)$, the *distance* in G, denoted $d_G(u, v)$, is the minimum length of a path in G from u to v. A *shortest-path tree* from a vertex v is a tree T such that for any vertex u we have $d_G(v, u) = d_T(v, u)$. Since $E = O(V + g)$, a shortest-path tree from any given vertex can be computed in $O(V \log V + E) = O(V \log V + g)$ time using Fibonacci heaps [13]. When $g = O(V^{1-\varepsilon})$ for any positive, fixed ε, then a shortest path tree can be constructed in $O(V)$ time.[1]

In the special case that all the edge-lengths are equal to one, any breadth-first-search tree is a shortest-path tree from the starting vertex, and can be computed in $O(V + g)$ time.

Width of Embeddings. The *edge-width* $\mathtt{ew}(G)$ (resp. *non-separating edge-width* $\mathtt{ew}_0(G)$) of a graph G embedded in a surface is defined as the minimum number of vertices in a non-contractible (resp. surface non-separating) cycle. The *face-width* $\mathtt{fw}(G)$ (resp. *non-separating face-width* $\mathtt{fw}_0(G)$) is the minimum number of faces (without multiplicity) that any non-contractible (surface non-separating) closed curve in the surface is going to intersect.

Algorithmically, the following characterization of face-width is more suitable. For an embedded graph G, consider its *vertex–face incidence graph* Γ: a bipartite graph, naturally embedded on the surface, whose vertices are faces and vertices of G, and there is an edge between face f and vertex v if and only if v is on the face f. (If v and f have multiple incidence, then Γ also has multiple edges joining v and f.) It is not difficult to see that $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma)$ and $\mathtt{fw}_0(G) = \frac{1}{2}\mathtt{ew}_0(\Gamma)$ [20, page 148]. Since the construction of Γ takes linear time from an embedding of G, this reduces the problem of computing the face-width of G to that of computing the edge-width of Γ.

Model of Computation. We assume non-negative real edge-lengths, and our algorithms run in the comparison-based model of computation, that is, we only add and compare (sums of) edge weights. For integer weights and the word-RAM model of computation, some logarithmic improvements may be possible. See the survey by Zwick [25] for a discussion.

3. Basic Tools

We first describe some of the topological tools that are used in the paper, and then present algorithms for computing distances in embedded graphs.

[1] Eppstein [9] shows how to compute in linear time a separator S of size $O(\sqrt{gV}) = O(V^{1-\varepsilon/2})$ for G such that $G - S$ is planar. The recursive subdivision that Henzinger et al. [15] require can then be obtained using the division by Eppstein in the first level and then continue in each planar subpiece using their approach.

3.1. *Topological Tools*

3-Path-Condition. Let $\mathcal{C}$ be a family of cycles. We say that $\mathcal{C}$ satisfies the 3-*path-condition* if it has the following property: if u, v are vertices of G and P_1, P_2, P_3 are internally disjoint paths joining u and v, and if the cycle $P_1 \cup P_2$ is in $\mathcal{C}$, then (at least) one of the cycles $P_1 \cup P_3$ or $P_2 \cup P_3$ is in $\mathcal{C}$. This concept was introduced by Thomassen [23] (see also Section 4.3 of [20]), who showed that the family of non-contractible cycles and the family of non-separating cycles satisfy the 3-path-condition. For these families, there are shortest cycles with the following property.

Lemma 1. *Let $\mathcal{C}$ be a family of cycles satisfying the* 3*-path property, and let $\mathcal{P}$ be a family of pairwise disjoint shortest paths, except, possibly, at common endpoints. Then there is a shortest cycle in $\mathcal{C}$ that contains at most one segment of each $P \in \mathcal{P}$.*

Proof. Let $C \in \mathcal{C}$. For any two vertices $u, v \in V(C)$, let $C_1[u, v], C_2[u, v]$ be the two segments of C connecting u, v. For any vertices u, v in a path P, let $P[u, v]$ be the segment of P between u and v. For a cycle C and a (simple) path P, let $s(C, P)$ be the number of maximal common segments that C and P have. Define $s(C, \mathcal{P}) = \max_{P \in \mathcal{P}} s(C, P)$ and $\sigma(C, \mathcal{P}) = \sum_{P \in \mathcal{P}} s(C, P)$.

Let $C \in \mathcal{C}$ be a shortest cycle in $\mathcal{C}$ that lexicographically minimizes the pair $(s(C, \mathcal{P}), \sigma(C, \mathcal{P}))$. If $s(C, \mathcal{P}) \leq 1$, then the statement holds. If $s(C, \mathcal{P}) \geq 2$, then there is some path $P \in \mathcal{P}$ such that $s(C, P) = s(C, \mathcal{P}) \geq 2$. In this case there are vertices $u, v \in V(C) \cap V(P)$ such that $P[u, v]$, $C_1[u, v]$, $C_2[u, v]$ are internally disjoint. Observe that the cycles $C_1 = P' \cup C_1[u', v']$ and $C_2 = P' \cup C_2[u', v']$ are no longer than C, have strictly fewer maximal common segments with P, have no more maximal common segments with any other $P' \in \mathcal{P}$ because $\mathcal{P}$ are pairwise disjoint (except at possible common endpoints), and by the 3-path-condition, one of them, say C_1, must be in the family $\mathcal{C}$. However, then we have $\sigma(C_1, \mathcal{P}) < \sigma(C, \mathcal{P})$ while $s(C_1, \mathcal{P}) \leq s(C, \mathcal{P})$, arriving at a contradiction with how C was chosen. Therefore, it holds that $s(C, \mathcal{P}) \leq 1$. □

Tree–Cotree Decomposition. We use the tree–cotree decomposition for embedded graphs introduced by Eppstein [9]. Let T be a spanning tree of G rooted at $x \in V(G)$. For any edge $e = uv \in E(G) \setminus T$, we denote by *loop*$(T, e)$ the closed walk in G obtained by following the path in T from x to u, the edge uv, and the path in T from v to x; we use *cycle*(T, e) for the cycle obtained by removing the repeated edges in *loop*(T, e). A subset of edges $U \subseteq E(G)$ is a *cotree* of G if $U^* = \{e^* \in E(G^*) \mid e \in C\}$ is a spanning tree of the dual graph G^*. A *tree–cotree* partition of G is a triple (T, U, X) of disjoint subsets of $E(G)$ such that T forms a spanning tree of G, U is cotree of G, and $E(G) = T \cup U \cup X$. As shown by Eppstein, if (T, U, X) is a tree–cotree partition, then $L_{(T,U,X)} = \{loop(T, e) \mid e \in X\}$ consists of g loops, and it is a cutset.

Orientable Double Cover. In Sections 7.1 and 7.2, for a graph G embedded on a non-orientable surface, we use its *orientable double cover* D_G, constructed as follows: for each vertex $v \in V(G)$ we place two vertices v, v' in $V(D_G)$, for each edge $uv \in E(G)$ we place edges $uv, u'v'$ in $E(D_G)$ if the signature of uv is $\lambda(uv) = +1$, and edges $uv', u'v$

in $E(D_G)$ if $\lambda(uv) = -1$; and the circular order of edges around vertices $u, u' \in V(D_G)$ is the same as around $u \in V(G)$. Let $\pi\colon\ D_G \to G$ be the natural projection. If G has genus g, then D_G has genus at most $2g$.

There are two properties of the orientable double cover D_G that will be used; both follow from the homotopy lifting property [14, pages 60ff.]. Firstly, if P is a path in D_G between a vertex v and its copy v', then its projection $\pi(P)$ is a non-contractible loop in G. Secondly, if $\tilde{C}$ is a non-contractible cycle in D_G, its projection $\pi(C)$ is a non-contractible loop in G. Moreover, any non-contractible loop in G comes from the projection of one of these two types of walks in D_G. As a consequence, if G is on the projective plane, then D_G is a planar graph, and only the first type of non-contractible cycles can occur in G, that is, any non-contractible cycle in G through vertex v corresponds to a path in D_G between v and v'.

3.2. *k-Pairs Distance Problem*

Consider the *k-pairs distance problem*:

> Given a graph G with positive edge-weights and k pairs $(s_1, t_1), \dots, (s_k, t_k)$ of vertices of G, compute the distances $d_G(s_i, t_i)$ for $i = 1, \dots, k$.

Djidjev [8] and Fakcharoenphol and Rao [12] (slightly improved by Klein [18] for non-negative edge-lengths) describe data structures for shortest path queries in planar graphs. We will need the following special case.

Lemma 2. *For a planar graph of order V, the k-pairs distance problem can be solved in*

(i) $O(V^{3/2} + k\sqrt{V})$ *time, and in*
(ii) $O(V \log^2 V + k\sqrt{V} \log^2 V)$ *time.*

Furthermore, the shortest path between the closest pair can be found in the same amount of time.

Proof. For (i), use the data structure by Djidjev [8]: after $O(V^{3/2})$ time for preprocessing, a distance query can be answered in $O(\sqrt{V})$ time.

For (ii), use the data structure by Fakcharoenphol and Rao [12] with the quicker construction by Klein [18]: after $O(V \log^2 V)$ preprocessing time, a distance query can be answered in $O(\sqrt{V} \log^2 V)$ time. Once we know the closest pair, the shortest path between that pair can be found in $O(V)$ time [15]. □

For surfaces, we have the following result.

Lemma 3. *Let G be a graph embedded on a surface of genus g. The k-pairs distance problem can be solved in $O(\sqrt{gV}(V \log V + g + k))$ time, and in $O(\sqrt{gV}(V + k))$ time if $g = O(V^{1-\varepsilon})$ for some $\varepsilon > 0$. Furthermore, the shortest path between the closest pair can be found in the same amount of time.*

Proof. Let $(s_1, t_1), \dots, (s_k, t_k)$ be k pairs of vertices. The general idea is as follows. We compute first a small vertex set S such that $G - S$ is a planar graph. To compute the distance between s_i and t_i, it then suffices to take the minimum of the following two values:

(i) The length of a shortest path between s_i and t_i that passes through at least one vertex in S, that is,

$$d_G^S(s_i, t_i) = \min_{s \in S}\{d_G(s_i, s) + d_G(s, t_i)\}.$$

(ii) The length of a shortest path between s_i and t_i that avoids all the vertices in S. In other words, if s_i and t_i belong to the same (planar) component G_j of $G - S$, we compute $d_{G_j}(s_i, t_i)$.

We now move onto the details. For a graph G embedded on a surface of genus g, there exist a set $S \subset V(G)$ of size $O(\sqrt{gV})$ such that $G - S$ is planar. It can be computed in time linear in the size of the graph, that is, $O(V + g)$ time [9].

Making a shortest path tree from each vertex $s \in S$, we compute all the values $d_G(s, v)$ for $s \in S$, $v \in V(G)$. Each shortest path tree takes $O(V \log V + g)$ time in general, and $O(V)$ time if $g = O(V^{1-\varepsilon})$. Therefore, we need $O(\sqrt{gV}(V \log V + g))$ time in total, or $O(\sqrt{gV}V)$ if $g = O(V^{1-\varepsilon})$.

For each pair (s_i, t_i), we can now compute the value $d_G^S(s_i, t_i)$ of item (i) in $O(\sqrt{gV})$ time. For all pairs, this amounts to $O(k\sqrt{gV})$ time.

For item (ii), let $G_1, \dots, G_t$ be the connected components of $G - S$, let k_j be the number of pairs (s_i, t_i) that are in component G_j and let V_j be the number of vertices of the graph G_j. Because each G_j is planar, the values $d_{G_j}(s_i, t_i)$ for the k_j pairs in the component G_j can be computed in $O(V_j^{3/2} + k_j\sqrt{V_j})$ time as shown in Lemma 2. Since each pair (s_i, t_i) is in at most one component of $G - S$, we have $\sum k_j \le k$. Therefore, we can compute the distances $d_{G_j}(s_i, t_i)$ for all pairs using $O\big(\sum_j (V_j^{3/2} + k_j\sqrt{V_j})\big) \le O\big(\sum_j V_j^{3/2} + \sqrt{V}\sum_j k_j\big) \le O(V^{3/2} + \sqrt{V}k)$ time. This completes the proof. □

4. Separating versus Non-Separating Cycles

In this section we characterize the surface non-separating cycles using the concept of crossing; see Fig. 1. For two simple closed curves that cross transversally, the number of crossings is the cardinality of the intersection of their images. For two arbitrary simple

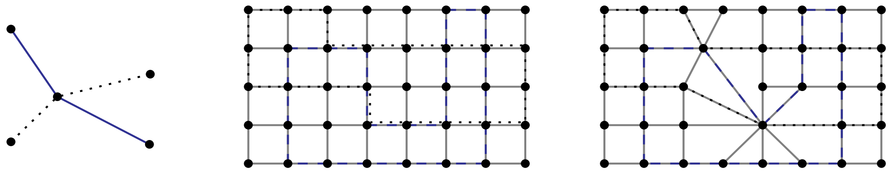

Fig. 1. Left: a crossing without shared edges. Center and right: cycles with shared edges and four crossings; we count crossings after contracting the common edges.

closed curves, the number of crossings is the minimum number of crossings over all small perturbations of the curves that make all the intersections transversal. When the simple curves are cycles in an embedded graph G, any common segment can be perturbed to two parallel curves that are arbitrarily close, and defer all the crossings to the endpoints of the common segment. This leads to the following equivalent, combinatorial definition of number of crossings for cycles $C = u_0 e_0 \cdots u_k e_k u_0$ and $C' = u'_0 e'_0 \cdots u'_l e'_l u'_0$ in an embedded graph G. If C, C' do not have any common edge, for each pair of common vertices $u_i = u'_j$ we count a crossing if the edges e_{i-1}, e_i of C and the edges e_{j-1}, e_j of C' alternate in the local rotation around $u_i = u'_j$ (where the indices are taken modulo $k+1$ and $l+1$, respectively); the number of all crossings is denoted by $cr(C, C')$. If C, C' are distinct and have a set of edges E' in common, then $cr(C, C')$ is the number of crossings after contracting G along E'. Finally, if $C = C'$, then we define $cr(C, C') = 0$ if C is two-sided, and $cr(C, C') = 1$ if C is one-sided.

Consider a tree–cotree decomposition (T, U, X) of G, and consider the set of cycles $\mathcal{C}_{(T,U,X)} = \{cycle(T, e) \mid e \in X\} = \{C_1, \ldots, C_g\}$. Note that if C is a surface separating cycle, then it splits the surface into two components. Thus, every other cycle crosses C an even number of times. The following result tells that this parity property with the cycles in $\mathcal{C}_{(T,U,X)}$ is enough to characterize the separating cycles.

Lemma 4. *A cycle C in G is surface non-separating if and only if there is some cycle $C_i \in \mathcal{C}_{(T,U,X)}$ such that C and C_i cross an odd number of times, that is, $cr(C, C_i) \equiv 1 \pmod 2$.*

Proof. For the proof, we need the concept of ($\mathbb{Z}_2$-)homology, which we introduce next; see Chapter 2 of [14] for a comprehensive treatment of homology. A set of edges E' is a 1-*chain*; it is a 1-*cycle* if each vertex has even degree in E'; in particular, every cycle in the graph is a 1-cycle, and also the symmetric difference of 1-cycles is a 1-cycle. The set of 1-cycles with the symmetric difference operation $+$ is an Abelian group, denoted by $\mathcal{C}_1(G)$. This group can also be viewed as a vector space over $\mathbb{Z}_2$ and is henceforth called the *cycle space* of the graph G. If f is a closed walk in G, the edges that appear an odd number of times in f form a 1-cycle. For convenience, we denote the 1-cycle corresponding to f by the same symbol f.

Two 1-chains E_1, E_2 are *homologically equivalent* if there is a family of facial walks $f_1, \ldots, f_t$ of the embedded graph G such that $E_1 + f_1 + \cdots + f_t = E_2$. Being homologically equivalent is an equivalence relation compatible with the symmetric difference of sets. The 1-cycles that are homologically equivalent to the empty set, form a subgroup $\mathcal{B}_1(G)$ of $\mathcal{C}_1(G)$. The quotient group $H_1(G) = \mathcal{C}_1(G)/\mathcal{B}_1(G)$ is called the homology group of the embedded graph G.

A set L of 1-chains generates the homology group if for any cycle C' in G, there is a subset $L' \subset L$ such that C' is homologically equivalent with $\sum_{E_i \in L'} E_i$. There are sets of generators consisting of g 1-chains. It is known that any set of loops that gives a cutset generates the homology group $H_1(G)$. Since the repeated edges in a loop disappear, we conclude that the set of cycles $\mathcal{C}_{(T,U,X)} = \{C_1, \ldots C_g\}$ generates $H_1(G)$.

We next argue that a cycle C is separating if and only if it corresponds to the trivial element of $H_1(G)$. Indeed, if C is a separating cycle, then $G \setminus\!\!\setminus C$ has two connected components. Let I be the set of indices such that f_i, $i \in I$, are the faces in one of the components. Then we have $C = \sum_{i \in I} f_i$, and therefore C is homologically equivalent to the empty set. Conversely, if C is a trivial element of the homology group, then there are indexes I such that $C = \sum_{i \in I} f_i$. Since C is a cycle, it follows that the boundary of the union of the faces f_i, $i \in I$, is C. Therefore, C separates f_i, $i \in I$, from f_i, $i \notin I$. For the rest of the proof, we use that non-separating cycles are non-trivial elements of the homology group.

Let $f_0, \dots, f_r$ be the 1-cycles that correspond to the facial walks. Then $f_0 = f_1 + \cdots + f_r$ and $\mathcal{C}_{(T,U,X)} \cup \{f_1, \dots, f_r\}$ is a generating set of $\mathcal{C}_1(G)$. If E_i is a 1-cycle, then $E_i = \sum_{j \in J} C_j + \sum_{i \in I} f_i$ for certain indices I, J. We define $cr_{E_i}(C)$ as the modulo 2 value of

$$\sum_{j \in J} cr(C, C_j) + \sum_{i \in I} cr(C, f_i) \equiv \sum_{j \in J} cr(C, C_j) \bmod 2.$$

It is easy to see that $cr_{E_i} \colon \mathcal{C}_1(G) \to \mathbb{Z}_2$ is a homomorphism. Since $cr(C, f_i) = 0$ for every facial walk f_i, cr_{E_i} also determines a homomorphism $H_1(G) \to \mathbb{Z}_2$.

If C is a surface separating cycle, then it corresponds to the trivial element of $H_1(G)$, so every homomorphism maps it to zero. In particular, for every j, $cr(C, C_j) \equiv cr_{C_j}(C) \equiv 0 \pmod 2$.

Let C be a non-separating cycle and consider $\tilde{G} = G \mathbin{\not\!\curlywedge} C$. Take a vertex $v \in C$, which gives rise to two vertices $v', v'' \in V(\tilde{G})$. Since C is non-separating, there is a simple path P in $\tilde{G}$ connecting v', v''. The path P is a loop in G (not necessarily a cycle), but it contains a cycle C' that crosses C exactly once.

Since $\mathcal{C}_{(T,U,X)}$ generates the homology group, there is a subset $\mathcal{C}' \subset \mathcal{C}_{(T,U,X)}$ such that the cycle C' and $\sum_{C_i \in \mathcal{C}'} C_i$ are homologically equivalent. However, then $1 \equiv cr_{C'}(C) \equiv \sum_{C_i \in \mathcal{C}'} cr(C, C_i) \pmod 2$, which means that for some $C_i \in \mathcal{C}'$, it holds that $cr(C, C_i) \equiv 1 \pmod 2$. □

5. Shortest Non-Separating Cycle

Let T_x be a shortest-path tree from vertex $x \in V(G)$. Let us fix any tree–cotree partition (T_x, U_x, X_x), and let $\mathcal{C}_x = \{cycle(T_x, e) \mid e \in X_x\}$. For a cycle $C \in \mathcal{C}_x$, let $\mathrm{Odd}(C)$ be the set of cycles that cross C an odd number of times. It follows from Lemma 4 that $\bigcup_{C \in \mathcal{C}_x} \mathrm{Odd}(C)$ is precisely the set of non-separating cycles. We will compute a shortest cycle in $\mathrm{Odd}(C)$, for each $C \in \mathcal{C}_x$, and take the shortest cycle among all them; this will be a shortest non-separating cycle.

We next show how to compute a shortest cycle in $\mathrm{Odd}(C)$ for $C \in \mathcal{C}_x$. Firstly, we use that T_x is a shortest-path tree to argue that we only need to consider cycles that intersect C exactly once; a similar idea is used by Erickson and Har-Peled [10] for their 2-approximation algorithm. Secondly, we reduce the problem of finding a shortest cycle in $\mathrm{Odd}(C)$ to an $O(V)$-pairs distance problem.

Lemma 5. *Among the shortest cycles in* $\mathrm{Odd}(C)$, *where* $C \in \mathcal{C}_x$, *there is one that crosses* C *exactly once.*

Proof. Let P, P' be the two shortest paths that, together with an edge, define C. The family of cycles $\mathrm{Odd}(C)$ satisfies the 3-path-condition, and the paths P, P' are disjoint, except at x. Therefore, from Lemma 1 we conclude that there is a shortest cycle C_0 in $\mathrm{Odd}(C)$ that contains at most one segment of P, and one segment of P'. It follows that C and C_0 can cross at most twice, and since $C_0 \in \mathrm{Odd}(C)$, they must cross exactly once. □

Lemma 6. *For any $C \in \mathcal{C}_x$, we can compute a shortest cycle in* $\mathrm{Odd}(C)$ *in $O((V \log V + g)\sqrt{gV})$ time, or $O(V\sqrt{gV})$ time if $g = O(V^{1-\varepsilon})$.*

Proof. Consider the graph $\tilde{G} = G \text{✂} C$. Each vertex v on C gives rise to two copies v', v'' of v in $\tilde{G}$. In G a cycle that crosses C exactly once (at vertex v, say) gives rise to a path in $\tilde{G}$ from v' to v'' (and vice versa). Therefore, finding a shortest cycle in $\mathrm{Odd}(C)$ is equivalent to finding a shortest path in $\tilde{G}$ between pairs of the form (v', v'') with v on C. In $\tilde{G}$, we have $O(V)$ pairs (v', v'') with v on C, and using Lemma 3 we can find the shortest path connecting a closest pair (v_0', v_0'') in $O((V \log V + g)\sqrt{gV})$ time, or $O(V\sqrt{gV})$ if $g = O(V^{1-\varepsilon})$. □

Theorem 7. *Let G be a graph with V vertices 2-cell embedded on a surface of gbzs g, without* 1- *or* 2-*gons. We can find a shortest surface non-separating cycle in $O(g^{3/2}V^{3/2}\log V + g^{5/2}V^{1/2})$ time, or $O((gV)^{3/2})$ time if $g = O(V^{1-\varepsilon})$.*

Proof. Since $\bigcup_{C \in \mathcal{C}_x} \mathrm{Odd}(C)$ is precisely the set of non-separating cycles, we find a shortest non-separating cycle by using the previous lemma for each $C \in \mathcal{C}_x$, and taking the shortest among them. The running time follows because $\mathcal{C}_x$ contains $O(g)$ loops. □

Observe that the algorithm by Erickson and Har-Peled [10] outperforms our result for $g = \Omega(V^{1/3}\log^{2/3} V)$. Therefore, we can recap concluding that a shortest non-separating cycle can be computed in $O(\min\{(gV)^{3/2}, V(V + g)\log V\})$ time.

6. Shortest Non-Contractible Cycle

Like in the previous section, we consider a shortest-path tree T_x from vertex $x \in V(G)$, and we fix a tree–cotree partition (T_x, U_x, X_x). Consider the cutset given by the loops $L_x = \{loop(T_x, e) \mid e \in X_x\}$. By increasing the number of vertices to $O(gV)$, we can assume that L_x becomes a system of cycles (instead of loops) whose pairwise intersection is x. This can be shown by slightly modifying G in such a way that L_x can be transformed without harm; we give the precise modification in the proof of the following result.

Lemma 8. *The problem is reduced to finding a shortest non-contractible cycle in an embedded graph $\tilde{G}$ of $O(gV)$ vertices with a given system of cycles $\mathcal{C}_x$ such that each cycle from $\mathcal{C}_x$ consists of two shortest paths from x plus an edge. This reduction can be done in $O(gV)$ time.*

Proof. Throughout this proof we assume that the given embedding is represented in such a way that signatures of edges in T_x are all positive. For vertices $u, u' \in V(G)$,

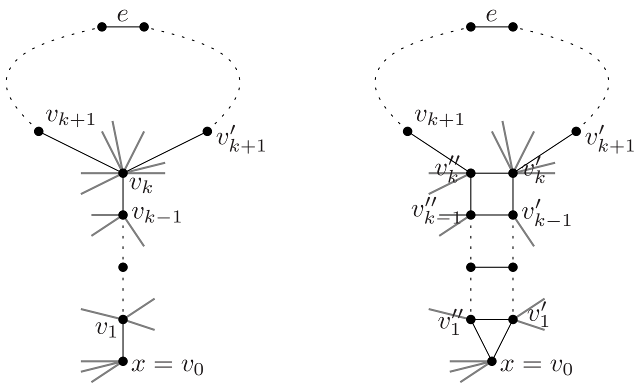

Fig. 2. How to modify the graph such that a loop $l_0 = loop(T_x, e) \in L_x$ becomes a cycle. The edges $v'_i v''_i$ have length 0.

we use $P[u, u']$ for the (unique) path in T_x from u to u'. For a loop $l \in L_x$, we define *split*(l) as the vertex v on l such that the part that appears twice in l is equal to $P[x, v]$. In particular, *split*$(l) = x$ if and only if l is a cycle.

Our first goal is to change the graph G and the spanning tree T_x in such a way that the loops in L_x will all become cycles whose pairwise intersection is only the vertex x. To achieve this goal, we proceed as follows.

There is nothing to do if all loops in T_x are cycles. Otherwise, consider a loop l_0 in L_x that is not a cycle and whose repeated part $P_0 = P[x, split(l_0)]$ is shortest. Let $x = v_0, v_1, \dots, v_k = split(l_0)$ be the consecutive vertices on P_0. Let v_{k+1} and v'_{k+1} be the neighbors of v_k that are on the loop l_0 and are distinct from v_{k-1}. Assume, moreover, that the edges of l_0 around v_k have local rotation $v_{k-1}, v_{k+1}, v'_{k+1}$ (in clockwise order on Fig. 2). The edges incident with each vertex v_i $(1 \le i \le k)$ that are not on the path $v_0, v_1, \dots, v_{k+1}$ can be classified as those on the *left* or on the *right* of that path. Now we replace the path $x, v_1, \dots, v_k$ in G with two paths $P'_0 = x, v'_1, \dots, v'_k$ and $P''_0 = x, v''_1, \dots, v''_k$, and add the edges $v'_i v''_i$ $(1 \le i \le k)$ between them. All edges incident with v_i that are on the right of v_i in G are now incident with v'_i, while the rest are incident with v''_i, $i = 1, \dots, k$. In particular, the edge $v_k v'_{k+1}$ which is on the right of v_k, is replaced by the edge $v'_k v'_{k+1}$, while the edge $v_k v_{k+1}$ is replaced by $v''_k v_{k+1}$. See Fig. 2. The edges that correspond to previous edges have the same length in the new graph, while all new edges $v'_i v''_i$ have length 0.

The new graph is naturally embedded in Σ as well. We replace the loop l_0 in L_x by the cycle $P_0, l_0 \backslash P[v_0, v_k], P'_0$. The rest of loops (or cycles) in L_x remain the same except that their segment common with P_0 is replaced with the corresponding new segments. They keep being loops (or cycles) because we have chosen l_0 such that *length*$(P[x, split(l_0)])$ is shortest among non-simple loops in T_x.

We repeat the procedure until L_x consists of only cycles; we need $O(g)$ repetitions. Each cycle in L_x may have two paths from x in common with other cycles. Consider a *longest* path P that two cycles $C, C' \in L_x$ have in common. Using the same technique as above, we can modify the graph and the cycles C, C' in such a way that they share one path less; details are similar and omitted. We keep repeating this step until any pair of cycles intersects only at x. We have to repeat this step at most $2|L_x| = 2g$ times because, at each step, a new edge adjacent to x that is used by some cycle is created. Therefore, the cycles in the resulting set pairwise intersect only in vertex x.

Let $\tilde{G}$ be the final graph that is obtained and let $\tilde{L}_x$ be the final set of cycles. Observe that each cycle in $\tilde{L}_x$ is composed of two shortest paths from x plus an edge. It is clear that a shortest non-contractible cycle in $\tilde{G}$ corresponds to a shortest non-contractible cycle in the original graph G, as well as shortest paths in $\tilde{G}$ correspond to shortest paths in G. The problem reduces then to finding a shortest non-contractible cycle in $\tilde{G}$. However, observe that the number of vertices has increased; each step may add $O(V)$ vertices, and therefore $\tilde{G}$ consists of $O(gV)$ vertices. This reduction can easily be done in $O(gV)$ time. □

The problem that remains is to find a shortest non-contractible cycle in a new graph $\tilde{G}$ with $O(gV)$ vertices where we are given a system of cycles $\mathcal{C}_x$. Moreover, each cycle of $\mathcal{C}_x$ consists of two shortest paths from x plus an edge. Let $\mathcal{C}^*$ be the set of shortest non-contractible cycles in $\tilde{G}$. Using Lemma 1, we obtain the following property.

Lemma 9. *Among the cycles in $\mathcal{C}^*$, there is one that crosses each cycle in $\mathcal{C}_x$ at most twice.*

Proof. Consider the $2g$ shortest paths $\mathcal{P}$ that define the cycles in $\mathcal{C}_x$. They are pairwise disjoint, except at x. Since the set of shortest non-contractible cycles satisfies the 3-path-condition, Lemma 1 implies that there is a shortest non-contractible cycle $C_0 \in \mathcal{C}^*$ that contains at most one segment from each $P \in \mathcal{P}$. It follows that each cycle $C \in \mathcal{C}_x$ can be crossed at most twice by C_0. □

Since $\mathcal{C}_x$ is a system of cycles, the set $D = \Sigma \text{✂} \mathcal{C}_x$ is a topological disk, and the corresponding graph $G_D = \tilde{G} \text{✂} \mathcal{C}_x$ is planar. In G_D each cycle from $\mathcal{C}_x$ corresponds to two paths on the boundary of D. We can glue an infinite number of copies of D to construct the universal cover of Σ. However, because of Lemma 9, we can find a shortest non-contractible cycle by constructing only a portion of this universal cover. These are the main ideas to prove the following result.

Theorem 10. *Let G be a graph with V vertices 2-cell embedded on a surface of genus g, without 1- or 2-gons. We can find a shortest non-contractible cycle in $O(g^{O(g)}V^{3/2})$ time.*

Proof. According to Lemma 8, we assume that $\tilde{G}$ has $O(gV)$ vertices and we are given a system of cycles $\mathcal{C}_x$ such that each cycle of $\mathcal{C}_x$ consists of two shortest paths plus an edge. Moreover, because of Lemma 9, there is a shortest non-contractible cycle crossing each cycle of $\mathcal{C}_x$ at most twice.

Consider the topological disk $D = \Sigma \text{✂} \mathcal{C}_x$ and the planar graph $G_D = \tilde{G} \text{✂} \mathcal{C}_x$. Let U be the universal cover that is obtained by gluing copies of D along the cycles in $\mathcal{C}_x$, and let G_U be the graph that is obtained by gluing copies of G_D according to the same pattern. The graph G_U is naturally embedded in U, and it is an infinite planar graph, unless Σ is the projective plane $\mathbb{P}^2$, in which case G_U is finite.

Let us fix a copy D_0 of D, and let U_0 be the portion of the universal cover U which is reachable from D_0 by visiting at most $2g$ different copies of D. Since each copy of D is adjacent to $2|\mathcal{C}_x| = 2g$ copies of D, U_0 consists of $(2g)^{2g} = g^{O(g)}$ copies of D. The portion G_{U_0} of the graph G_U that is contained in U_0 can be constructed in $O(g^{O(g)}gV) = O(g^{O(g)}V)$ time. We assign to the edges in G_{U_0} the same weights they have in G.

A cycle is non-contractible if and only if its lift in U finishes in different copies of the same vertex. Each time that we pass from a copy of D to another copy we must intersect a cycle in $\mathcal{C}_x$. Using the previous lemma, we conclude that there is a shortest non-contractible cycle whose lift intersects at most $2|\mathcal{C}_x| = O(g)$ copies of D. That is, there exists a shortest non-contractible cycle in G whose lifting to U starts in D_0 and is contained in G_{U_0}.

We can then find a shortest non-contractible cycle by computing, for each vertex $v \in D_0$, the distance in G_{U_0} from the vertex v to all the other copies of v that are in G_{U_0}. Each vertex $v \in D_0$ has $O(g^{O(g)})$ copies in G_{U_0}. Therefore, the problem reduces to computing the shortest distance in G_{U_0} between $O(g^{O(g)}V)$ pairs of vertices. Since G_{U_0} is a planar graph with $O(g^{O(g)}V)$ vertices, the result follows from Lemma 2(i). □

Observe that, for a fixed surface, the genus is constant and the running time of the algorithm is $O(V^{3/2})$. However, for most values of g as a function of V (when $g \geq c\log V/\log\log V$ for a certain constant c), the near-quadratic time algorithm by Erickson and Har-Peled [10] is better.

7. Edge-Width and Face-Width

When edge-lengths are all equal to 1, shortest non-contractible and surface non-separating cycles determine combinatorial width parameters (see Chapter 5 of [20]). Since their computation is of considerable interest in topological graph theory, it makes sense to consider this special case in more details.

7.1. *Arbitrary Embedded Graphs*

Recall the parameters measuring the width of embedded graphs that were introduced in Section 2. The (*non-separating*) *edge-width* $\mathrm{ew}(G)$ ($\mathrm{ew}_0(G)$, respectively) of an embedded graph G is the minimum number of vertices in a non-contractible (surface non-separating) cycle. Therefore, the non-separating edge-width can be computed by setting $w(e) = 1$ for all edges e in G and running the algorithms from previous sections. The (*non-separating*) *face-width* $\mathrm{fw}(G)$ ($\mathrm{fw}_0(G)$, respectively) of an embedded graph G is one half of the (non-separating) edge-width of its *vertex–face incidence graph* Γ: a bipartite graph whose vertices are faces and vertices of G, and there is an edge between face f and vertex v if and only if v is on the face f.

Observe that, for this special case, a breadth-first-search tree is indeed a shortest-path tree, and it can be computed in $O(V+g)$ time, instead of $O(V \log V+g)$ that is required for arbitrary lengths if the genus is high. This improves slightly the running time for this special case.

Theorem 11. *Let G be a graph with V vertices 2-cell embedded on a surface of genus g, without 1- or 2-gons. Its non-separating edge-width and face-width can be computed in $O(g^{3/2}V^{3/2} + g^{5/2}V^{1/2})$ time. Its edge-width and face-width can be computed in $O(g^{O(g)}V^{3/2})$ time.*

It can happen that $\mathtt{ew}(G) = \Omega(V)$. The situation is different for the face-width $\mathtt{fw}(G)$ for which there exist non-trivial bounds. The key tool is the following result by Hutchinson [16], which improves a previous bound by Albertson and Hutchinson [1].

Theorem 12 [16]. *Let T be a triangulation in an orientable surface of genus g. Then $\mathtt{ew}(T) = O(\sqrt{V/g}\log g)$ if $g \le V$, and $\mathtt{ew}(T) = O(\log g)$ if $g > V$.*

We now extend this result to non-orientable surfaces, and then provide a bound on the face-width of graphs.

Lemma 13. *Let T be a triangulation in an arbitrary surface of genus g. Then $\mathtt{ew}(T) = O(\sqrt{V/g}\log g)$ if $g \le V$, and $\mathtt{ew}(T) = O(\log g)$ if $g > V$.*

Proof. If T is orientable, then the previous theorem suffices. For non-orientable surfaces we form its orientable double cover D_T, as explained in Section 3. D_T is also a triangulation and has genus $\tilde{g} \le 2g$. By Theorem 12, D_T contains a non-contractible cycle $\tilde{C}$ of length $O(\sqrt{2V/\tilde{g}}\log \tilde{g}) = O(\sqrt{V/g}\log g)$ if $\tilde{g} \le 2V$, and $O(\log \tilde{g}) = O(\log g)$ if $\tilde{g} > 2V$. The projection C of $\tilde{C}$ to T is a closed walk in T which is non-contractible. This walk contains a non-contractible cycle of length at most $|V(\tilde{C})| = \mathtt{ew}(D_T)$, so $\mathtt{ew}(T) \le \mathtt{ew}(D_T)$. Since $\tilde{g} \le 2g$, the bound also follows for the non-orientable case. □

Theorem 14. *Let G be a graph of order V embedded in a surface of genus g. Then $\mathtt{fw}(G) = O(\sqrt{V/g}\log g)$ if $g \le V$, and $\mathtt{fw}(G) = O(\log g)$ if $g > V$.*

Proof. Consider the vertex–face incidence graph Γ, which has a natural embedding as a quadrangulation in the same surface as G, that is, all facial walks consist of exactly four edges. Let $T = \Gamma + E(G)$ be the triangulation obtained from Γ by adding all edges of G in the quadrangular faces of Γ. Then $\mathtt{fw}(G) = \frac{1}{2}\mathtt{ew}(\Gamma) \le \mathtt{ew}(T)$.

From the previous lemma, we have $\mathtt{ew}(T) = O(\sqrt{|V(T)|/g}\log g)$ if $g \le |V(T)|$, and $O(\log g)$ if $g > |V(T)|$. Since $|V(T)| = V + |F(G)| = O(V+g)$, the first bound reduces to $\mathtt{ew}(T) = O(\sqrt{V/g}\log g)$. This completes the proof. □

7.2. *Face-Width in the Projective Plane*

For the special case when G is embedded in the projective plane $\mathbb{P}^2$, we can improve the running time for computing the face-width. The idea is to use an algorithm for computing the edge-width whose running time depends on the value $\mathtt{ew}(G)$. This is achieved by combining three ideas: an algorithm of Erickson and Har-Peled [10] to compute a non-contractible cycle which is at most twice as long as a shortest non-contractible cycle; a double cover of G which is planar; and Lemma 2(ii) for distance queries on planar graphs. The result is as follows.

Lemma 15. *If G is embedded in $\mathbb{P}^2$ and $\mathtt{ew}(G) \leq t$, then we can compute $\mathtt{ew}(G)$ and find a shortest non-contractible cycle in $O(V \log^2 V + t\sqrt{V} \log^2 V)$ time.*

Proof. Consider the orientable double cover D_G of G, as explained in Section 3. D_G is a planar graph, and each vertex v of G gives rise to two vertices v, v'. Moreover, any shortest non-contractible loop passing through a vertex $v \in V(G)$ is equivalent to a shortest path in D_G between the vertices v and v'.

We can compute in $O(V \log V)$ time a non-contractible cycle C of G of length at most $2\mathtt{ew}(G) \leq 2t$ using the results of [10]. Actually, this can be done in $O(V)$ time by using a breadth-first-search tree. Any non-contractible cycle in G has to intersect C at some vertex. In particular, every shortest non-contractible cycle intersects C. Consider in D_G the $|C| \leq 2t$ pairs of vertices $\Pi = \{(v, v') \mid v \in V(C)\}$, and let P be a shortest path in D_G between a closest pair of Π. The path P can be found in $O(V \log^2 V + t\sqrt{V} \log^2 V)$ time using Lemma 2, and it corresponds to a shortest non-contractible cycle in G. □

Like before, consider the vertex–face incidence graph Γ which can be constructed in linear time. We have $\mathtt{ew}(\Gamma) = 2\mathtt{fw}(G)$, and from Theorem 14 we obtain $\mathtt{ew}(\Gamma) = O(\sqrt{V})$. Therefore, the problem reduces to that of computing the edge-width of a graph knowing a priori that $\mathtt{ew}(\Gamma) = 2\mathtt{fw}(G) = O(\sqrt{V})$. Using the previous lemma with $t = O(\sqrt{V})$, we conclude the following.

Theorem 16. *Let G be a graph with V vertices 2-cell embedded in $\mathbb{P}^2$, without 1- or 2-gons. We can compute its face-width in $O(V \log^2 V)$ time.*

Juvan and Mohar [17] obtained a linear-time algorithm for deciding if $\mathtt{fw}(G) \leq k$, where k is a fixed constant. They needed the special case when $k = 4$ in an algorithm for testing embeddability in the torus.

7.3. *Face-Width in the Torus*

We next describe an algorithm for computing the face-width for a graph G embedded on the torus $\mathbb{T}$. Consider the vertex–face incidence graph Γ; we will compute $\mathtt{ew}(\Gamma) = 2\mathtt{fw}(G)$. Observe that on a fixed surface, $|V(\Gamma)| = \Theta(V)$.

We compute in $O(V \log V)$ time a non-contractible cycle C of Γ of length at most $2\mathtt{ew}(\Gamma) = \mathtt{fw}(G)$ using the results of [10]. We know that $|C| = O(\sqrt{V})$ by Theorem 14. See also Lemma 17 below.

Fix a vertex $v_0 \in C$ and find in $O(V \log V)$ time a shortest non-contractible loop l from v_0. Let A be the non-contractible cycle obtained by removing the repeated edges from l. We know that A consists of two shortest paths from a vertex $u_0 \in V(\Gamma)$ plus an edge; $u_0 = v_0$ if l was a cycle, and $u_0 \neq v_0$ otherwise. Moreover, $|A| \leq |l| \leq |C| = O(\sqrt{V})$.

The graph $\Gamma_A = \Gamma ✂ A$ is planar. Let v', v'' be the two copies of $v \in A$ in Γ_A. Consider the $O(\sqrt{V})$ pairs $\Pi = \{(v', v'') \mid v \in V(A)\}$ and find a pair $(v_1', v_1'') \in \Pi$ such that $d_{\Gamma_A}(v_1', v_1'') = \min\{d_{\Gamma_A}(v', v'') \mid (v', v'') \in \Pi\}$ in $O(V \log^2 V)$ time using Lemma 2. Let P_{v_1} be a shortest path in Γ_A from v_1' to v_1'', which is a non-contractible cycle B in Γ. The graph $\Gamma_B = \Gamma ✂ B$ is planar as well.

We may assume that

$$|B| \geq \frac{\sqrt{2}}{2}|A|. \tag{1}$$

If not, then we repeat the above procedure by starting with B playing the role of the cycle C. Then we find new cycles A and B. If (1) is violated again, then the length of the new cycle B would be strictly smaller than one-half of the length of the former cycle A. This would imply that $|B| < \mathtt{ew}(\Gamma)$, a contradiction.

Using Menger's theorem for vertex-disjoint paths, we can prove the following bound.

Lemma 17. *It holds that* $|A| \cdot |B| = O(V)$.

Proof. Let M be the maximum number of vertex-disjoint paths from A' to A'', the copies of A in Γ_A. By Menger's theorem, M is equal to the cardinality of a minimum (A', A'')-separator S in γ_A. Since Γ_A is embedded in a cylinder with A' and A'' being the cycles on the boundary, the separator S gives rise to a closed curve γ in the torus homotopic to A that intersects Γ precisely in the vertices in S. Since Γ is a quadrangulation, the curve γ determines a non-contractible cycle C in Γ of length at most $2|S| = 2M$. In particular, $M \geq \frac{1}{2}\mathtt{ew}(\Gamma) \geq \frac{1}{4}|A|$.

Let $R_1, \ldots, R_M$ be disjoint (A', A'')-paths. Each R_i together with a segment on A'' determines a path from a vertex $v' \in V(A')$ to its mate v'' in A'' of length at most $|R_i| + \frac{1}{2}|A|$. Consequently, $|R_i| + \frac{1}{2}|A| \geq |B|$. Since $R_1, \ldots, R_M$ are vertex disjoint and (1) holds, we get

$$V \geq \sum_{i=1}^{M} |R_i| \geq M(|B| - \tfrac{1}{2}|A|) \geq \tfrac{1}{4}|A| \left(1 - \frac{\sqrt{2}}{2}\right)|B|.$$

This completes the proof. □

Cycles A and B constructed above can be used for a fast computation of the face-width of G.

Theorem 18. *Let G be a graph with V vertices 2-cell embedded in the torus, without 1- or 2-gons. We can compute its face-width in $O(V^{5/4} \log V)$ time.*

Proof. Every non-contractible cycle in the torus is surface non-separating. Since the cycles A, B are a cutset, the proof of Lemma 4 tells us that we can assume that a shortest non-contractible cycle crosses either A or B an odd number of times.

We first assume that a shortest non-contractible cycle crosses A an odd number of times. Since A is composed of two shortest paths (plus an edge), the argument used in the proof of Lemma 5 shows that there is a shortest non-contractible cycle that crosses A exactly once. In this case, B is a shortest non-contractible cycle, and we are done.

We now consider the case where every shortest non-contractible cycle crosses B an odd number of times. The argument used in Lemma 5 shows that we can assume that a shortest non-contractible cycle crosses A at most twice and, therefore, it crosses A twice or not at all. Let C be one such shortest non-contractible cycle crossing B a *minimum* number of times and A twice or not at all. Then C crosses B exactly once. To see this, consider the crossings C has with A and B as you walk along C. A segment s of C between two crossings of C with B must contain a crossing of C with A, as otherwise s is contained in a copy of $\Gamma \text{✂} A$ and it could be replaced by a homotopic segment of B to reduce the number of crossings with B. Since between two crossings of C with B there must be a crossing of C with A, then C and B can cross at most twice. However, we were assuming that C crosses B an odd number of times, and we conclude that they cross exactly once.

We distinguish two cases depending on the length of A:

- If $|A| \geq V^{1/4} \log V$, then $|B| = O(V^{3/4}/\log V)$ because of Lemma 17. We compute the distance from v' to v'' for any $v \in B$ as follows. Consider the graph $\Gamma_B = \Gamma \text{✂} B$, that is planar, and the $|B| = O(V^{3/4}/\log V)$ pairs $\Pi = \{(v', v'') \mid v \in B\}$. We can then use Lemma 2 with Γ_B and Π to find in $O(V^{5/4} \log V)$ time a shortest path between the closest pair of Π, which corresponds to a shortest non-contractible cycle in Γ.
- If $|A| < V^{1/4} \log V$, consider the topological disk $D = \mathbb{T} \backslash (A \cup B)$, make its copy D_0, and construct the portion U_0 of the universal cover reachable from D_0 by crossing A at most twice and without crossing B. This needs five copies of D; see Fig. 3. Let Γ_0 be the cover of Γ naturally embedded in U_0 by gluing copies of D.

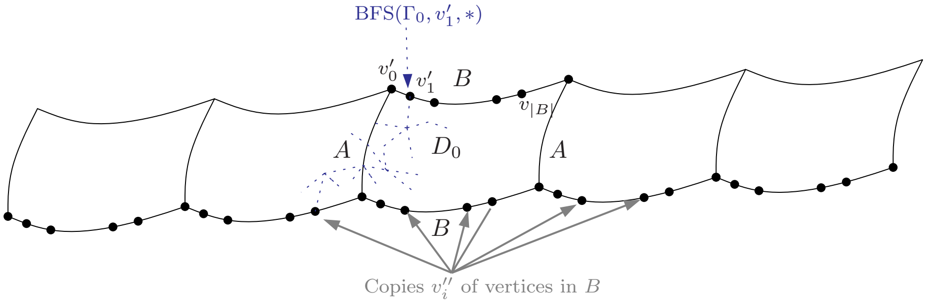

Fig. 3. Portion of the universal cover reachable by crossing A at most twice and B not at all. We show some of the concepts used in the proof.

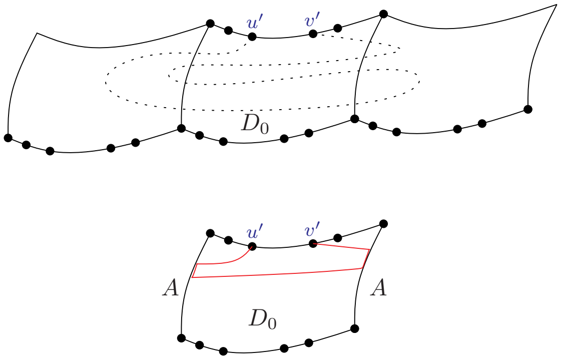

Fig. 4. A path of length l in Γ_0 yields a path of length at most $l + 2|A|$ in D_0.

In D_0, let $v'_1, v'_2, \ldots, v'_{|B|}$ be the consecutive vertices of one copy of B. The shortest cycle we are seeking corresponds to a shortest path connecting v'_i with some copy v''_i of v'_i.

We make for each v'_i a $\mathrm{BFS}(\Gamma_0, v'_i, |A| - 1)$, where $\mathrm{BFS}(\Gamma_0, u, t)$ is a breadth-first-search tree from vertex u in the graph Γ_0 clipped at depth t, that is, only including vertices up to depth t. If a shortest non-contractible cycle has length strictly smaller than $|A|$, then its lift in Γ_0 has to be contained in $\mathrm{BFS}(\Gamma_0, v'_i, |A|-1)$ for some $v'_i \in B$. Therefore, once we have $\mathrm{BFS}(\Gamma_0, v'_i, |A| - 1)$ for all $v'_i \in B$ we can easily find a shortest non-contractible cycle.

We claim that it takes $O(V^{5/4} \log V)$ time to construct the trees $\mathrm{BFS}(\Gamma_0, v'_i, |A|-1)$ for all $i = 1, \ldots, |B|$. Observe that the proof of the claim will finish the proof of the theorem. The proof is as follows. Define the sets $B_k = \{v'_{k+4i|A|} \mid i = 0, \ldots, \lfloor |B|/4|A| \rfloor\}$ for $k = 1, \ldots, 4|A|$; that is, B_k consists of v'_k and each $4|A|$th vertex along B. Observe that $B \setminus \bigcup_k B_k = \{v'_{|B|-(|B| \bmod 4|A|)}, \ldots, v'_{|B|-1}\}$ consists of $O(|A|) = O(V^{1/4} \log V)$ vertices. Therefore, the trees $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for $u \in B \setminus \bigcup_k B_k$ can be computed in $O(V^{1/4} \log V) \cdot O(V) = O(V^{5/4} \log V)$ time.

If $u', v' \in B_k$, then $d_{\Gamma ✂ A}(u', v') \geq 4|A|$ because B is a shortest path in $\Gamma ✂ A$. This implies that $d_{\Gamma_0}(u', v') \geq 2|A|$ because any shortest path of length l in Γ_0 can be clipped by D_0 to obtain a path of length at most $l + 2|A|$; see Fig. 4. Therefore, if $u', v' \in B_k$, we have $\mathrm{BFS}(\Gamma_0, u', |A| - 1) \cap \mathrm{BFS}(\Gamma_0, v', |A| - 1) = \emptyset$. Since for any fixed k, each edge of Γ_0 appears at most once in the trees $\{\mathrm{BFS}(\Gamma_0, u, |A|-1) \mid u \in B_k\}$, we can compute $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for all $u \in B_k$ in $O(V)$ time. The parameter k takes the values $1, \ldots, 4|A| = O(V^{1/4} \log V)$, and therefore we need $O(V^{1/4} \log V) \cdot O(V) = O(V^{5/4} \log V)$ time to compute $\mathrm{BFS}(\Gamma_0, u, |A| - 1)$ for all $u \in \bigcup_k B_k$. This finishes the proof of the claim and of the theorem. □

8. Conclusions

We have presented algorithms for finding shortest non-contractible and surface non-separating cycles for graphs embedded on a surface. For a fixed surface, our algorithms run in $O(V^{3/2})$ time, which is a considerable improvement over previous results. Our algorithms can be used to compute the (non-separating) edge-width and the (non-separating) face-width of embedded graphs.

Our algorithms work for undirected graphs with non-negative edge-lengths. Similar results for directed graphs seem much harder because non-contractible or non-separating cycles do not satisfy the 3-path-condition anymore. Finding shortest cycles with properties that do not satisfy the 3-path-condition remains an elusive problem.

We have also given a near-linear running time algorithm for computing the face-width in the projective plane; for the torus, we show how to compute the face-width in $O(V^{5/4} \log V)$ time. We feel that one of the most appealing open questions is finding near-linear running time algorithms for computing the face-width of graphs embedded on a (possibly fixed) surface. Our approach when dealing with the projective plane and the torus does not seem to extend to surfaces of higher genera.

Acknowledgments

The authors thank an anonymous referee for several valuable suggestions.

References

1. M. O. Albertson and J. P. Hutchinson. On the independence ratio of a graph. *J. Graph Theory*, 2:1–8, 1978.
2. S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. In G. S. Brodal and S. Leonardi, editors, *ESA* '05, volume 3669 of LNCS, pages 131–142. Springer, Berlin, 2005.
3. É. Colin de Verdière. Shortening of curves and decomposition of surfaces. Ph.D. thesis, Université Paris 7, December 2003. Available at http://www.di.ens.fr/users/colin/textes/these.html.en.
4. É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. In *Proc*. 11*th Int*. *Symp*. *Graph Drawing*, volume 2912 of LNCS, pages 478–490. Springer, Berlin, 2004.
5. É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete Comput*. *Geom*., 33:507–534, 2005. Preliminary version in *FOCS* '02.
6. T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Comput*. *System Sci*., 58:297–325, 1999. Preliminary version in *FOCS* '95.
7. T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete Comput*. *Geom*., 14:93–110, 1995.
8. H. Djidjev. Efficient algorithms for shortest path problems on planar digraphs. In F. d'Amore, P. G. Franciosa, and A. Marchetti-Spaccamela, editors, *WG* '96, volume 1197 of LNCS, pages 151–165. Springer, Berlin, 1997.
9. D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proc*. 14*th Symp*. *Discrete Algorithms*, pages 599–608. ACM, New York, and SIAM, Philadelphia, PA, January 2003.
10. J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput*. *Geom*., 31:37–59, 2004. Preliminary version in *SoCG* '02.
11. J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proc*. 16*th Symp*. *Discrete Algorithms*, pages 1038–1046, 2005.
12. J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. In *FOCS* 2001, pages 232–242, 2001.
13. M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34:596–615, 1987. Preliminary version in *FOCS* '84.

14. A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2001. Available at http://www.math.cornell.edu/∼hatcher/.
15. M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. System Sci.*, 55(1):3–23, 1997.
16. J. P. Hutchinson. On short noncontractible cycles in embedded graphs. *SIAM J. Discrete Math.*, 1:185–192, 1988.
17. M. Juvan and B. Mohar. An algorithm for embedding graphs in the torus. Submitted.
18. P. N. Klein. Multiple-source shortest paths in planar graphs. In *Proc.* 16*th ACM–SIAM Symp. Discrete Algebra* (*SODA*), pages 146–155, 2005.
19. F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proc.* 17*th Symp. Computational Geometry*, pages 80–89, 2001.
20. B. Mohar and C. Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, Baltimore, MD, 2001.
21. A. Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *J. Combin. Theory Ser. B*, 51:127–159, 1991.
22. A. Schrijver. Paths in graphs and curves on surfaces. In *Proc. First European Congress of Mathematics, Vol. II* (*Paris*, 1992), volume 120 of Progr. Math., pages 381–406. Birkhäuser, Basel, 1994.
23. C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Combin. Theory, Ser. B*, 48:155–177, 1990.
24. G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proc.* 6*th Ann. ACM Symp. Computational Geometry*, pages 102–111, 1990.
25. U. Zwick. Exact and approximate distances in graphs—a survey. In *ESA* 2001, volume 2161 of LNCS, pages 33–48. Springer, Berlin, 2001.

Received October 28, 2005, *and in revised form June* 1, 2006, *and June* 8, 2006.
Online publication February 5, 2007.