Linear Algorithms for Partitioning Embedded Graphs of Bounded Genus

L. Aleksandrov[1] H. Djidjev[1,2]

January 13, 1995

Abstract

This paper develops new techniques for constructing separators for graphs embedded on surfaces of bounded genus. For any arbitrarily small positive ε we show that any n-vertex graph G of genus g can be divided in $O(n+g)$ time into components whose sizes do not exceed εn by removing a set C of $O(\sqrt{(g+1/\varepsilon)n})$ vertices. Our result improves the best previous ones with respect to the size of C and the time complexity of the algorithm. Moreover, we show that one can cut off from G a piece of no more than $(1-\varepsilon)n$ vertices by removing a set of $O(\sqrt{n\varepsilon(g\varepsilon+1})$ vertices. Both results are optimal up to a constant factor.

Keywords: graph separator, graph genus, algorithm, divide-and-conquer, topological graph theory

AMS(MOS) subject classifications: 05C10, 05C85, 68R10

[1]Bulgarian Academy of Sci., CICT, G.Bonchev 25-A, 1113 Sofia, Bulgaria
[2]Department of Comp.Sci.,Rice University, P.O.Box 1892, Houston, Texas 77251, USA

1 Introduction

Let S be a class of graphs closed under the subgraph relation and $f(n)$ be a non-negative function. An $f(n)$*-separator theorem* for S is defined in [18] as a theorem of the following form: *There exist constants* $\alpha < 1$ *and* $\beta > 0$ *such that if* G *is any* n*-vertex graph in* S*, then the vertices of* G *can be divided into three sets* A*,* B*, and* C *such that no edge joins a vertex in* A *with a vertex in* B*,* $\max(|A|, |B|) \leq \alpha n$ *and* $|C| \leq \beta f(n)$*.* The set of vertices C is called an α*-separator.* Separator theorems are known for many classes of graphs, i.e. for forests, grids [17], planar graphs [18], graphs of bounded genus [6, 11], graphs with an excluded minor [2], geometric graphs [20], and others. The $\sqrt{n}$-separator theorem for the class of planar graphs [18] is now a classic result in computational graph theory.

If an $o(n)$-separator theorem exists for S, then one can efficiently apply divide-and-conquer method for solving graph problems on graphs from S. The sets of vertices A and B define subproblems which are independent and essentially smaller than the original problem. For finding solutions to the subproblems defined by A and B one applies the same method until the sizes of the subproblems become so small that they can be easily solved directly. Separators have been successfully applied for solving different computational problems. Such applications include finding a suitable ordering of the equations of very large sparse systems of linear equations in order to minimize the fill-in during a Gaussian elimination [17, 12], finding approximate solutions to NP-complete problems [19], VLSI layout [3, 4, 16], designing efficient sequential and parallel algorithms [8, 15, 9, 13], and many others.

One inefficiency that results when the above method is straightforwardly applied is that the time required to find the decomposition of the original problem can be too large or even dominate the total time needed to solve the problem of interest. For instance, if an algorithm for finding an $\frac{2}{3}$-separator of an n-vertex graph belonging to a certain class of graphs requires $\Theta(n)$ time, then it will take $\Theta(n \log n)$ time to find a decomposition of the graph into subgraphs of $O(1)$ size by using that algorithm.

In our paper we give a solution to the problem of dividing an n-vertex graph into many pieces of small size in optimal linear time. More specifically, we prove that if G is an n-vertex graph with non-negative vertex weights embedded on a surface of orientable genus g, then for any $\varepsilon \in (0, 1)$ there exists a set C of no more than $4\sqrt{(g+1/\varepsilon)n}$ vertices of G whose removal leaves no component of total weight exceeding ε times the total weight of G. Such separators for $\varepsilon = o(n)$ are referred in the literature as *ε-separators*. They have been used for the solutions of various problems, e.g. in constructing approximation algorithms for the maximum independent set and other NP-complete problems [19], pebbling [19], embedding of data or communication structures [19, 4], shortest path problems [8], design of parallel algorithms [13]. We present an algorithm that finds such a separator in $O(n+g)$ time, given the embedding of G. A preliminary version of this result was published in [5]. Similar results for the class of planar graphs (i.e. for the case where g=0) were proved in [19, 8], where the complexity of the algorithms is $O(n \log n)$ and for the class of planar unweighted graphs in [13], where the size of the separator is not explicitly estimated. In [11] the existence of an ε-separator of size $O(\sqrt{ng/\varepsilon})$ for graphs of genus g is derived as a consequence of the iterative 2/3-separation. The complexity of the algorithm that follows from this approach is $O(n \log n)$. Thus our algorithms improve the previous results with respect to the size of the separator and the time complexity of the algorithm. Moreover we show that the size of the separators found by our algorithm is optimal within a constant factor.

Essential to our approach will be a data structure we call a *separation graph* and a technique we develop for manipulating with separation graphs. The separation graphs, which are sparse graphs of degree 3, contain all the relevant information we need in order to construct the separator. Separation graphs are more convenient to be used for purposes of graph separation than the original graphs because of their simple structure and their sparsity.

We demonstrate the use of our new technique also by proving an optimal separator theorem for the class of graphs of bounded genus with a constant $\alpha = 1 - o(1)$. We prove that if G is an n-vertex graph with non-negative vertex weights embedded on a surface of orientable genus g, then for any $\varepsilon \in (0, 1/15)$ there exists an $(1-\varepsilon)$-separator C_1 of G of $O(\sqrt{n\varepsilon(g\varepsilon+1)})$ vertices. The size of C_1 is optimal within a constant factor and C_1 can be found in $O(n+g)$ time given the embedding of G. A similar result was proved by Gilbert in [10] and by Djidjev and Gilbert in [7], however the constants in [10, 7] are much larger and the complexities of the algorithms are $O((n+g)\log n)$. Another application of separation graphs was described in [1].

We view the contribution of our work in the following: (i) We prove separator theorems for graphs embedded on orientable surfaces, where similar previous results concerned separators for planar graphs only; (ii) The complexity of our algorithms is linear; (iii) Our separators are smaller in size than the best previous separators for the planar case and are asymptotically optimal for arbitrary genus; (iv) We present a new algorithm design technique for dividing graphs embedded on surfaces.

The paper is organized as follows. In Section 2 we give a definition of a separation graph and prove some of its properties. Next we give an algorithms for construction of separation graphs and prove that any edge separator of the separation graph induces a set of cycles separating the original graph. In Section 3 we present the basic results of the paper. In Section 3.2 we construct a linear algorithm for ε-separation of graphs embedded onto a surface of genus g in the case when ε is close to zero and in Section 3.3 we construct a linear algorithm for the case when ε is close to one. In the final subsection we establish the optimality of our results.

2 Separation graphs

We begin with a brief description of some basic notions from topological graph theory. More detailed and formal treatment can be found in [14].

A *graph* G is an ordered pair $(V(G), E(G))$ of sets, where $V(G)$ is a set of *vertices* and $E(G)$ is a set of *edges*. Each edge is an unordered pair of different vertices. We will consider graphs embedded on orientable surfaces. A *surface* is a connected, compact, 2-manifold. An *embedding* $I(G)$ of the graph G onto a surface Z is a mapping of the vertices onto different points of Z and of the edges onto arcs along Z, so that the incidences are preserved, and no two arcs intersect at an inner point. We call the vertices and edges of G vertices and edges of $I(G)$ and we call *faces* of $I(G)$ the connected components of $Z \setminus I(G)$. We consider *simplicial* embeddings, i.e. each face is surrounded by at least three edges. An embedding is a 2-*cell embedding* if each of its faces is homeomorphic to an open disk. If no ambiguity arises, we will refer to the 2-cell embeddings as *embeddings* and to $I(G)$ as G. The *orientable genus* (or shortly the *genus*) of a graph G is the minimum genus of an orientable surface, called the *genus surface* of G, onto which G can be embedded. A *region* R of $I(G)$ we call any connected open subset of $I(G)$. The *boundary* of R is $\partial R = \bar{R} \setminus R$, where $\bar{R}$ is the closure of R.

The sets of vertices, edges, and faces of $I(G)$ (or G) will be denoted respectively by $V(G)$, $E(G)$, and $F(G)$ hereafter. An important relation between the numbers of vertices, edges, faces, and the genus of a 2-cell embedding $I(G)$ is given by the Euler's formula

$$|V(G)| - |E(G)| + |F(G)| = 2c - 2g \ , \tag{1}$$

where c is the number of connected components of G, and $|X|$ denotes the cardinality of a set X.

Definition 2.1 *Let G be an embedded graph and T be a spanning tree of G. A* separation graph *of G with respect to the spanning tree T is a graph $S = S(G,T)$, where*
$V(S) := F(G)$,
$E(S) := \{(f_1, f_2) \ : \ f_1, f_2 \in F(G), \ f_1$ and f_2 share a non-tree edge$\}$.

Algorithm 2.1 described below will assign proper weights to the edges of S which are essential for the use of a separation graph.

According to the above definition, the separation graph of a given embedded graph G is a subgraph of the dual graph of G and can be constructed in $O(|V(G)| + |E(G)|)$ time. For an example, see Figure 1.

As the next lemma shows, if the genus g of G is small compared to $|V(G)|$, then S is much simpler (sparser) than G.

Lemma 2.1 *Let G be a connected graph that has a 2-cell embedding on a surface of genus g with a spanning tree T. Then the separation graph $S(G,T)$ is connected and*

$$|E(S)| - |V(S)| = 2g - 1 \ . \tag{2}$$

Proof: Assume that S is disconnected and let K be a component of S. Denote by R the region consisting of all faces that correspond to vertices of K together with their boundaries. The boundary ∂R of R is non-empty and it contains a simple closed curve that is an embedding of a cycle of G. On the other hand, by the definition of S, the boundary ∂R is a subset of the

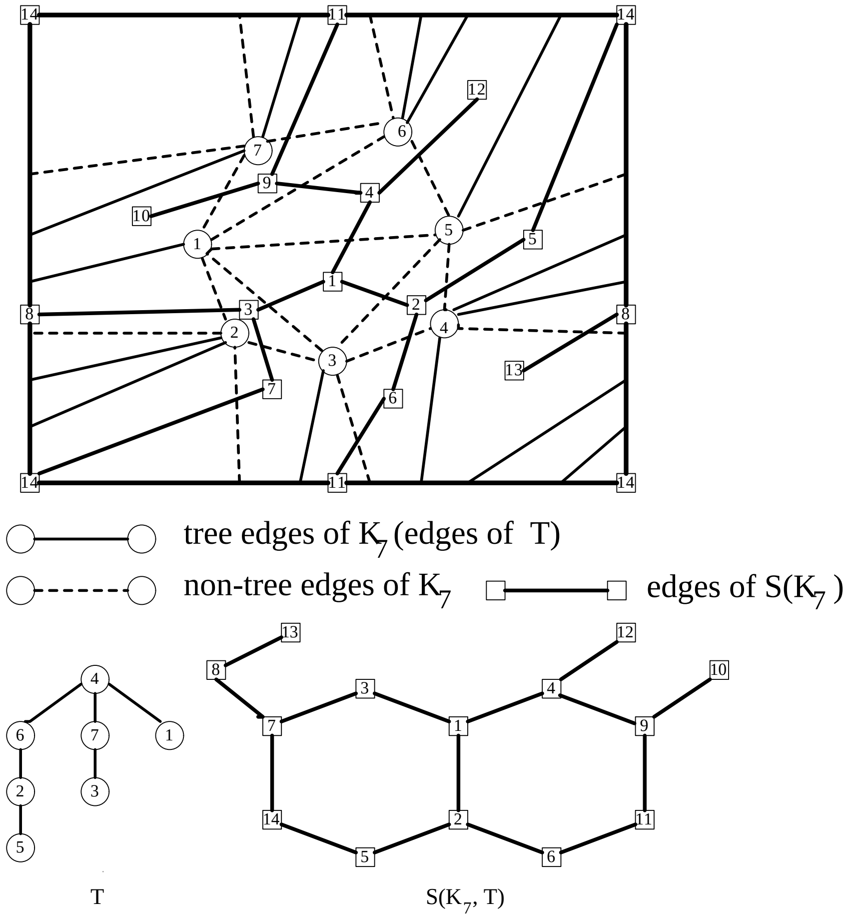


Figure 1: An embedding of K_7 on the torus and its separation graph.

embedding $I(T)$ of T. This means that $I(T)$ contains a closed curve and therefore T contains a cycle, which is a contradiction. Thus S is connected.

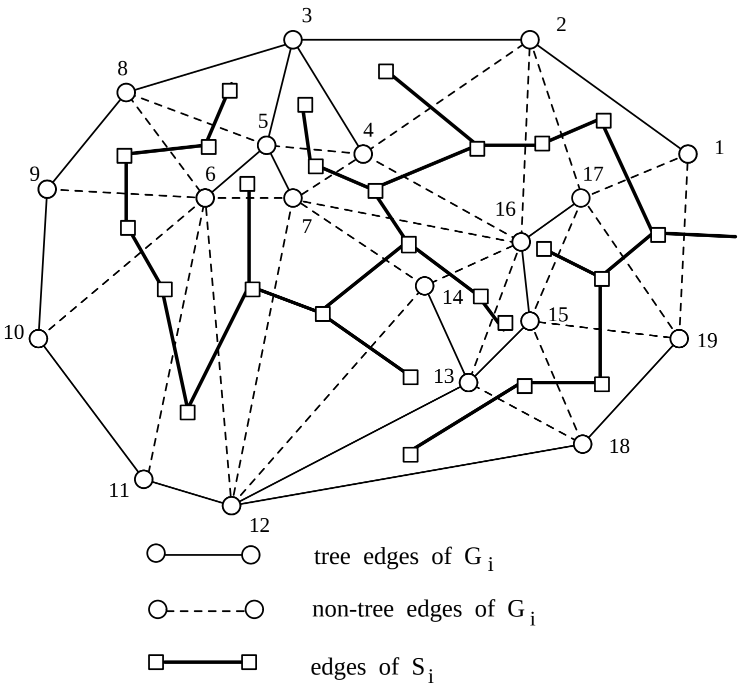
By the definition of S, we have

$$|E(S)| = |E(G)| - |V(G)| + 1 \quad \text{and} \quad |V(S)| = |F(G)|.$$

Applying the Euler's formula (1), we obtain

$$|E(S)| - |V(S)| = |E(G)| - |V(G)| + 1 - |F(G)| = 2g - 1 \ ,$$

which proves (2). $\square$



e	(1,17)	(2,4)	(2,17)	(4,5)	(5,8)	(6,8)	(6,9)	(6,10)	(6,11)	(7,12)	(13,18)	(14,16)	(15,18)	(17,19)	any other
vert(e)	{1}	{4}	{2,17}	{3}	{5}	{6,8}	{9}	{10}	{11}	{7,12}	{13}	{14,16}	{15,18}	{19}	{}

Figure 2: Illustrations to Algorithm 2.1 and Theorem 2.1. The subgraph G_i of G embedded inside a T-cycle of G and the corresponding region Z_i and subtree S_i.

For instance, if $g = 0$ (i.e. if G is planar), then by Lemma 2.1 S is a tree. In general, S can be represented as a tree plus $2g$ additional edges. An example for $g = 1$ is presented on Figure 1.

We call a *T-cycle* of G any simple cycle that contains exactly one edge that is not in T. Any edge of $S(G,T)$ determines an unique T-cycle, the one containing the corresponding non-tree edge of G. In the sequel we will show that, if weights are assigned to the edges of S in a proper way, we can construct separators of G consisting of T-cycles by considering S instead of G.

Hereafter, we consider the case when the embedding of G is a triangulation, i.e. each of its faces is incident to three edges. For our purposes, this is not a restriction since any embedding could be extended to a triangulation by adding new edges. Subsequently any vertex set that separates the resulting triangulation separates the original graph as well. Note that in the case when the embedding of G is a triangulation the vertices of S have maximal degree three, which fact can be used to simplify the algorithms on S.

We assume that the vertices of G have non-negative weights $wt(\cdot)$. The algorithm below assigns weights on the edges of S that depend on the structure of G and on the weight function $wt(\cdot)$. We will use the following notation. For $v \in V(G)$ denote by $E(v,0)$ the set of all non-tree edges of G incident to v and by $E(v,1)$ the set of all non-tree edges of G whose both endpoints are adjacent to v. As the vertices at distance 1 from v define (at least one) simple cycle and there are no simple cycles in any spanning tree of G, then $E(v,1) \neq \emptyset$. The algorithm described below first defines for each non-tree edge of G (or equivalently for each edge of S) a set $vert(e)$ of vertices of G such that the sets $vert(e)$ for $e \in E(G) \setminus E(T)$ form a partition of $V(G)$ (see Figure 2). Then the algorithm assigns to each edge e of S a weight equal to the total weight of the set $vert(e)$.

Algorithm 2.1 ASSIGNMENT OF WEIGHTS ON THE EDGES OF S
Input: An embedded graph G with a spanning tree T and the corresponding separation graph $S(G,T)$; a weight function $wt(\cdot)$ defined on $V(G)$;
Output: A weight function $wt(\cdot)$ on the edges of S;

```
Set vert(e) := ∅ for e ∈ E(G) \ E(T)
For each v ∈ V(G) do
    if  E(v,0) ≠ ∅
            then   e_1 := any edge in E(v,0)
                   vert(e_1) := vert(e_1) ∪ {v}
            else   { since always E(v,1) ≠ ∅ }
                   e_2 := any edge in E(v,1)
                   vert(e_2) := vert(e_2) ∪ {v}
    endif
enddo
```

Set $wt(e') := \sum_{v \in vert(e)} wt(v)$, for $e' \in E(S)$,
where e is the corresponding to e' edge from $E(G) \setminus E(T)$.

Algorithm 2.1 can be implemented in linear time in a straightforward manner. Note that from the definition of $wt(e)$ it follows directly:

$$\sum_{e\in E(S)} wt(e) = \sum_{v\in V(G)} wt(v)\ . \tag{3}$$

Remark: We will need weights on the edges of S as assigned by Algorithm 2.1 even in the case when the original graph is non-weighted. In this case we assume that all vertices of G have weight $1/|V(G)|$.

The next theorem shows that for finding a separator consisting of T-cycles one can use $S(G)$ instead of G (see Figure 2). Observe that the inequalities (4) and (5) below can be used to estimate the weights of the parts into which G is divided.

Theorem 2.1 *Let G be a graph with weights on the vertices embedded on a surface Z such that each face is a triangle and let T be a spanning tree of G. Let M be a set of edges of S whose removal divides S into components $S_1,\ldots,S_k$. Then the set of curves $\tilde{C}(M)$ that are embeddings of the T-cycles corresponding to the edges of M divides Z into connected open regions $Z_1,\ldots,Z_k$ such that:*

(i) $$wt(Z_i) \leq wt(S_i), \quad i = 1,\ldots,k, \tag{4}$$

where we denote by $wt(Z')$, for $Z' \subset Z$, the total weight of the vertices of G embedded in Z';

(ii) *If T is a breadth-first spanning tree of G, then*

$$wt(Z_i \cup \partial Z_i) \geq wt(S_i \cup \partial S_i) - wt(t), \quad i = 1,\ldots,k, \tag{5}$$

where ∂Z_i is the boundary of Z_i, ∂S_i is the set of edges of M adjacent to S_i, and t is the root of T.

Proof: Let the set of curves $\tilde{C}(M)$ divides Z into connected open regions $Z_1,\ldots,Z_{k'}$. For $i=1,\ldots,k'$, denote by S'_i the subgraph of $(V(S),E(S)\setminus M)$ that is embedded inside Z_i. Then the graphs S'_i, $i = 1,\ldots,k'$, define a partition of the graph $(V(S),E(S)\setminus M)$, i.e. we have

$$\bigcup_{i=1}^{k'} V(S'_i) = V(S), \quad \text{and} \quad V(S'_i)\cap V(S'_j) = \emptyset \quad \text{for} \quad i\neq j;$$

$$\bigcup_{i=1}^{k'} E(S'_i) = E(S) \setminus M, \quad \text{and} \quad E(S'_i) \cap E(S'_j) = \emptyset \quad \text{for} \quad i \neq j.$$

We will show that each graph S'_i, $i = 1, \ldots, k'$, is connected. Assume that S'_i is disconnected for some $i \in \{1, \ldots, k'\}$ and let K be one of its components. Denote by R the region of Z consisting of all faces dual to the vertices of K. The boundary ∂R of R contains a simple closed curve that is an embedding of a cycle of G. According to the assumption it follows that $\partial R \subset I(M^* \cup T)$ and $\partial R \cap Z_i \neq \emptyset$, where M^* denotes the set of edges dual to the edges of M and $I(M^* \cup T)$ is the embedding of $M^* \cup T$. Consequently, there exists a simple closed curve $\gamma \subset \partial R$ such that $\gamma \cap Z_i \neq \emptyset$. In other words, we have found an embedding of a cycle $\gamma \subset I(M^* \cup T)$ with $\gamma \not\subset \tilde{C}(M)$. This is a contradiction, since the embedding of each cycle in $T \cup M^*$ belongs to $\tilde{C}(M)$.

We proved that S'_i, $i = 1, \ldots, k'$, are connected components of the graph $(V(S), E(S) \setminus M)$. Therefore, $k = k'$ and $\{S_1, \ldots, S_k\} \equiv \{S'_1, \ldots, S'_k\}$. To simplify the notations we assume $S'_i = S_i$ for $i = 1, \ldots, k$.

Next, we prove that (i) and (ii) hold for Z_i for $i = 1, \ldots, k$. Let $v \in Z_i$ and e be the edge of S such that $v \in vert(e)$. Then, by Algorithm 2.1, the edge from $E(G) \setminus E(T)$ dual to e is either incident, or has both endpoints adjacent to v. By the definition of Z_i $e \in S_i \cup M$. On the other hand it is not possible that $e \in M$, because it implies $E(v, 0) \neq \emptyset$ and therefore $v \in \tilde{C}(M)$, which contradicts to $v \in Z_i$. Thus $e \in S_i$ and (i) follows.

In order to prove (ii), we consider any edge $e \in S_i \cap \partial S_i$ and we will show that $vert(e) \subset \{t\} \cup Z_i \cup \partial Z_i$. If $e \in S_i$, the relation follows immediately. Let $e \in \partial S_i$ and (v_1, v_2) be the edge of G dual to e. If $w \in vert(e)$ and $w \neq v_i, i = 1, 2$, then by Algorithm 2.1, $E(w, 0) = \emptyset$. (The assumption $E(w, 0) \neq \emptyset$, leads to $w \in vert(h)$ for some $h \in E(w, 0)$.) Therefore, any edge incident to w is from $E(T)$. The only vertex with this property in a breadth-first spanning tree of a triangulation is its root. Thus $vert(e) \subset \{v_1, v_2, t\}$ and the inequality (ii) holds. □

We use the result of Theorem 2.1 as a basic tool for constructing separators in the next section. More precisely, if G, T, S, M, and $S_1, \ldots, S_k$ are as in Theorem 2.1, then the removal of the vertices of G that belong to the T-cycles corresponding to the edges in M divides G into components $G_1, \ldots, G_k$ such that $wt(G_i) \leq wt(S_i)$, $i = 1, \ldots, k$. Therefore, we can construct separators for G by constructing edge–separators for the separation graph S. This approach relies on the fact that, as we will show in the next section, we are able to construct edge–separators for sparse graphs in optimal linear time. Note that the length of a T-cycle can be $\Omega(|V(G)|)$ and thus a good bound on the size of the resulting separator of G does not directly follow from the existence of a small separator of S. So, in order to achieve both an optimal running time of our algorithm and an optimal size of the constructed separators we will combine the use of separation graphs with an appropriate radius-reducing technique.

3 ε-separation of graphs of bounded genus

In this section we derive two results on separation of graphs of bounded genus. The first one is related to the problem of dividing a graph into (possibly many) components of small weights, known as the *ε-separation* problem, and the second one concerns the problem of separating a single piece of small weight from a graph.

Definition 3.1 *For any weighted graph G and $\varepsilon \in (0,1)$, the vertex (resp. edge) set C is called* ε-separator *(resp.* ε-edge separator*) of G if the subgraph of G induced by $V(G) \setminus C$ (resp. the subgraph induced by $E(G) \setminus C$) has no component of weight greater than $\varepsilon wt(G)$.*

First we will develop fast algorithms for ε-separation of graphs of degree three.

3.1 Separating trees and sparse graphs

In the next lemma we present an linear algorithm that finds a small ε-edge separator for any graph of degree three.

Lemma 3.1 *Let S be a graph of degree three with n vertices, m edges, and q connected components and non-negative weights $wt(\cdot)$ on its edges. For any $\varepsilon \in (0,1)$ there exists a set M_ε of no more than $m-n+q+\lfloor 2/\varepsilon \rfloor$ edges whose removal divides S into connected components $S_1, \ldots, S_k$ with the properties:*

$$\text{(i)} \qquad wt(S_i) \leq \varepsilon wt(S), \quad i = 1, \ldots, k \tag{6}$$

(ii) *The inequality*

$$wt(S_i \cup \partial S_i) \geq (\varepsilon/2)wt(S), \tag{7}$$

holds for at least $k - q$ *of the components* S_i, $i = 1, \dots, k$, *where* ∂S_i *is the set of edges from* M_ε *adjacent to* S_i. *The set* M_ε *can be found in* $O(m)$ *time.*

Proof: First we consider the case where S is a binary tree, i.e. $m = n-1$ and $q = 1$. Let the set of edges of S be divided into levels $E_0, \dots, E_r$ according to their distance to some fixed vertex of S. For $e \in E(S) \setminus E_0$ we denote by $pr(e)$ the unique edge at the lower level that shares common vertex with e and we create an additional edge to be a predecessor of the edges from E_0. Consider the following procedure:

Procedure TREE SEPARATION

Input: A binary tree S, the set of levels $E_0, \dots, E_r$ with respect to a root t, a parameter $\varepsilon \in (0, 1)$, and the weights $wt(\cdot)$ on the edges of S.
Output: A set M_ε, which is an ε-edge separator of S.

```
For i = r, r − 1, ..., 0, do
   For each e ∈ E_i do
      If wt(e) > (ε/2)wt(S) then insert e in M_ε
         else wt(pr(e)) := wt(pr(e)) + wt(e)
```

This procedure runs in time linear on m, since every edge of S is considered only once. During the i-th iteration of the loop, the current weight of any edge x in E_{r-i+1} is equal to its original weight plus the total weight of the component of S which would be cut if x and all edges already in M_ε are deleted. We include an edge e in M_ε iff its current weight $wt(e)$ is greater than $(\varepsilon/2)wt(S)$. Therefore, the proposition (ii) follows for all components of $S \setminus M_\varepsilon$, except possibly for the one containing the root. The proposition (i) follows by the fact that the total weight of the component that will be cut if e is deleted is equal to the sum of the current weights of its ancestors (no more than two) which were not included in M_ε during the previous iteration.

The estimation $M_\varepsilon \leq \lfloor 2/\varepsilon \rfloor$ follows by the observation that with each edge inserted in M_ε the total weight of the edges of S decreases by at least $(\varepsilon/2)wt(S)$.

In the general case when S is a graph of degree three we apply the following algorithm:

Algorithm 3.1 EDGE SEPARATION OF GRAPHS OF DEGREE THREE

Input: A graph of S degree three, an $\varepsilon \in (0, 1)$, and weights $wt(\cdot)$ on edges of S.
Output: A set M_ε, which is an ε-edge separator of S.

Step 1: Find the set of the connected components of S.

Step 2: For each connected component Q with $wt(Q) > \varepsilon wt(S)$ do:

Step 2.1: Find a breadth-first spanning tree T of Q and divide the set of edges of T into levels with respect to the root of T.

Step 2.2: Insert the set of non-tree edges of Q in M_ε.

Step 2.3: If $wt(T) > \varepsilon wt(S)$ then apply the Tree Separation procedure on T with parameter $\varepsilon_1 = \varepsilon wt(S)/wt(T)$ to find an ε_1-edge separator M_{ε_1} of T. Add M_{ε_1} to the set M_ε.

The proof that the set M_ε constructed by Algorithm 3.1 satisfies requirements of the lemma follows from the correctness of the Tree Separation procedure. $\square$

3.2 Finding ε-separators for ε close to zero

Let G be an n-vertex graph with non-negative vertex weights and $I(G)$ be a 2-cell embedding of G on a surface Z of orientable genus g. We are going to describe an algorithm that for a given $I(G)$ and $\varepsilon \in (0, 1)$ finds an ε-separator C of G of no more than $4\sqrt{n(g + 1/\varepsilon)}$ vertices in $O(n+g)$ time. Let us recall that, without loss of generality, we assume that $I(G)$ is a triangulation, since we can make each face a triangle by adding new edges. Any vertex set that separates the resulting graph will separate the original graph as well.

Let T be a breadth-first spanning tree rooted at a vertex t and let S be the corresponding separation graph as defined in Definition 2.1 and Algorithm 2.1. Denote the radius of T by R. For $r = 0, 1 \ldots, R$ we define level $L(r)$ to be the set of vertices lying at distance r from t.

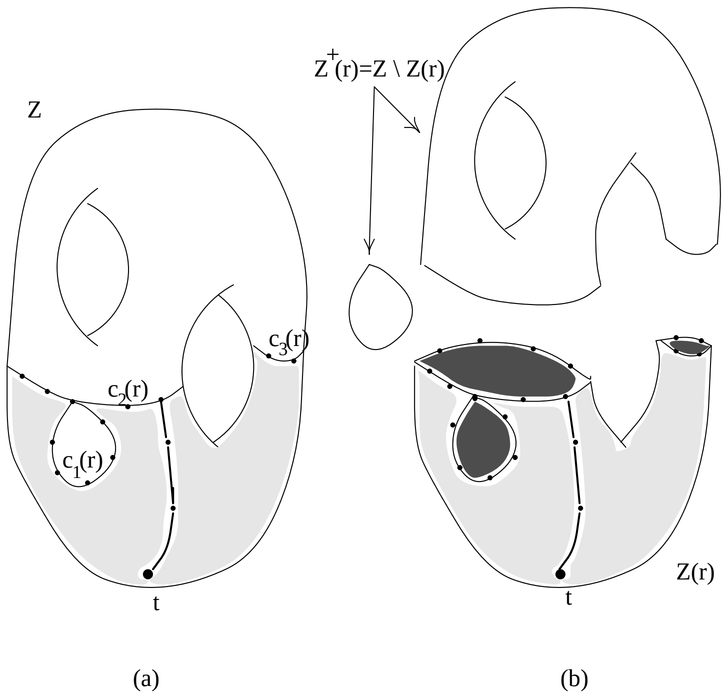


Figure 3: The region $Z(r)$ and its boundary

We denote by $Z(r)$, $0 < r \leq R$, the closed region of Z consisting of all triangles that have at least one vertex lying on a level lower than r. The boundary of $Z(r)$ consists of simple closed curves that are embeddings of the vertices in $L(r)$ and certain non-tree edges between them. They form a set of edge disjoint cycles in G, which we denote by $\tilde{L}(r) = \left\{c_1(r), \ldots, c_{\nu(r)}(r)\right\}$ (see Figure 3). The set $\tilde{L}(r)$ can be constructed by traversing each connected component of the boundary of $Z(r)$ in one direction so that $Z(r)$ remains on the right and by choosing the first non-traversed edge in a counterclockwise order to be traversed next. This procedure requires time proportional to the size of the subgraph of G induced by the vertex set $L(r)$.

Let $Z^+(r)$ be the complement of $Z(r)$ to Z. We denote by $G(r)$, $T(r)$, and $S(r)$ the subgraphs of G, T, and S that are embedded on $Z(r)$. Correspondingly, the subgraphs embedded on $Z^+(r)$ are denoted by $G^+(r)$, $T^+(r)$, and $S^+(r)$. Clearly the level $L(r)$ separates $G(r)$ and $G^+(r)$, and the set of edges of S that are dual to the edges of the cycles in $\tilde{L}(r)$ separates $S(r)$ and

$S^+(r)$.

Our algorithm for finding an ε-separator constructs the required ε-separator iteratively. At each iteration step a fixed level $L(r)$ along which the graph will be cut is considered. The algorithm constructs the set of embedded cycles $\tilde{L}(r)$ and the graph $S^+(r)$ defined above and by using the separation graph finds a set of vertices that divides the graph $G^+(r)$ into components of weights not exceeding $\varepsilon wt(G)$. The constructed vertex set is added to the current separator. Finally the algorithm transforms the remaining (not still separated) subgraph $G(r)$ and the corresponding subgraph of the separation graph $S(r)$ in a form suitable for the next iteration step. Next we describe in more details how this transformation is done and then we will give an outline of the whole algorithm.

The region $Z(r)$ can be considered as a surface with holes determined by the embeddings of the cycles in $\tilde{L}(r)$. Let $\tilde{Z}(r)$ be the surface obtained from $Z(r)$ by pasting open discs (faces) $f(c_1), \ldots, f(c_\nu(r))$ on its holes. In our transformation we supplement the graphs $G(r)$ and $S(r)$ obtaining graphs $\tilde{G}(r)$ and $\tilde{S}(r)$ that possess the following properties: (a) the graph $\tilde{G}(r)$ triangulates the surface $\tilde{Z}(r)$; (b) the tree $T(r)$ is a breadth-first spanning tree of $\tilde{G}(r)$; (c) the graph $\tilde{S}(r)$ is a weighted separation graph for $\tilde{G}(r)$ with respect to the tree $T(r)$. The following procedure describes how such graphs $\tilde{G}(r)$ and $\tilde{S}(r)$ can be constructed:

Procedure: SUBGRAPH SUPPLEMENTATION

Input: Subgraphs $G(r)$, $S(r)$ embedded on $Z(r)$; A set of embedded cycles $\tilde{L}(r)$ and the weights of the edges in S dual to the edges in $\tilde{L}(r)$.
Output: Graphs $\tilde{G}(r)$ and $\tilde{S}(r)$ embedded on $\tilde{Z}(r)$ satisfying properties (a), (b), and (c).

Step 1: For each cycle $c \in \tilde{L}(r)$ add a face $f(c)$ to the embedding of $G(r)$ with a boundary the embedding of c. Triangulate each face $f(c)$ by adding new edges to $G(r)$ embedded onto $f(c)$. Define $\tilde{G}(r)$ to be the resulting graph.

Step 2: For each face $f(c)$ $c \in \tilde{L}(r)$ construct the tree $tr(c)$ dual to the triangulation of $f(c)$ obtained in Step 1. Assign zero weights to the edges of $tr(c)$ (see Figure 4). Add the trees $tr(c)$ for $c \in \tilde{L}(r)$ to the graph $S(r)$. Add to $S(r)$ the edges dual to the edges of the cycles

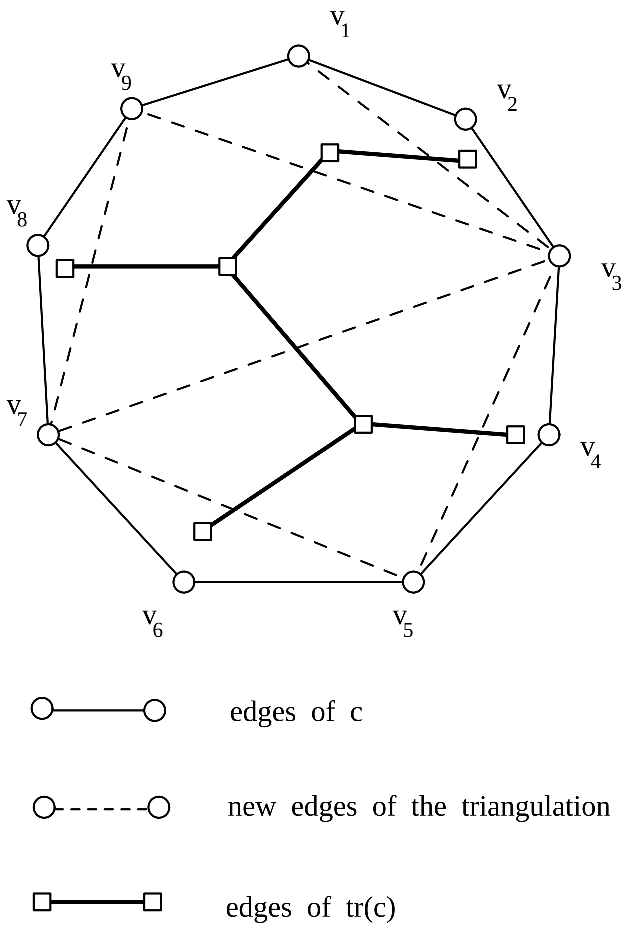


Figure 4: A triangulated face $f(c)$ and the corresponding $tr(c)$.

in $\tilde{L}(r)$ preserving their original weights in S. Define $\tilde{S}(r)$ to be the resulting graph.

The proof that graphs $\tilde{G}(r)$, $T(r)$ and $\tilde{S}(r)$ have properties (a), (b), and (c) follows directly from their definitions. The running time of the procedure is proportional to the size of the set $\tilde{L}(r)$.

Next, we present our main algorithm that constructs an ε-separator for a given graph G embedded onto a surface of genus g. We assume that $g < n/16$, otherwise $4\sqrt{(g + 1/\varepsilon)n} > n$ and the set $V(G)$ is a trivial ε-separator. Additionally, we assume without loss of generality that $I(G)$ is a triangulation, for otherwise we can define additional edges to make each face a triangle.

Algorithm 3.2 CONSTRUCTION OF AN ε-SEPARATOR

Input: A 2-cell embedding of a graph G on a surface of orientable genus g; a non-negative weight function defined on vertices of G; a number $\varepsilon \in (0,1)$.
Output: An ε-separator C of G.

Step 1: Choose any vertex t of G and construct a breadth-first spanning tree T of G rooted at t. Let R be the radius of T. For $j = 0, 1, \ldots, R$, find the levels $L(j)$ consisting of all vertices at distance j from t.

Step 2: Construct the separation graph $S = S(G,T)$ of G with respect to T. Assign weights to the edges of S using Algorithm 2.1.

Step 3: Set $h = \left\lceil \frac{1}{2}\sqrt{n/(g+1/\varepsilon)} \right\rceil$. Find an integer k between 0 and $h-1$, that minimizes the number of vertices in the set $L = \bigcup_{j=0}^{l} L(k+jh)$, where $l = \lfloor \frac{R-k}{h} \rfloor$. (Thus $|L| \le n/h$.) Set initially $C = L$.

Step 4: For $j = l+1, l, \ldots, 0$ do steps 4.1 through 4.7 below:

Step 4.1: Set $r = k + (j-1)h$ if $j \ge 1$, or $r = 0$ if $j = 0$.

Step 4.2: Find the set of cycles $\tilde{L}(r)$.

Step 4.3: Determine the subgraph $S^+(r)$, i.e. the subgraph of S embedded onto $Z^+(r)$.

Step 4.4: If $wt(S^+(r)) > \varepsilon wt(S)$, then find an ε_j-edge separator M_j for $S^+(r)$ by applying Algorithm 3.1 on $S^+(r)$ with parameter $\varepsilon_j = \varepsilon wt(S)/wt(S^+(r))$. Otherwise set $M_j = \emptyset$.

Step 4.5: Add to the current separator C the set C_j of vertices of G on levels above $L(r)$ that belong to T-cycles corresponding to edges in M_j.

Step 4.6: Find subgraphs $G(r)$, $S(r)$, and $T(r)$.

Step 4.7: Apply the Subgraph Supplementation procedure on $G(r)$ and $S(r)$ and obtain graphs $\tilde{G}(r)$ and $\tilde{S}(r)$. Set $G = \tilde{G}(r)$, $S = \tilde{S}(r)$, and $T = T(r)$.

In the next theorem we prove the correctness of Algorithm 3.2 and estimate the size of the constructed ε-separator.

Theorem 3.1 *Let G be an n-vertex graph with non-negative vertex weights and $I(G)$ be a 2-cell embedding of G on an orientable surface of genus g. For any $\varepsilon \in (0,1)$ there exists an ε-separator C of G of no more than $4\sqrt{(g+1/\varepsilon)n}$ vertices. The separator C can be found in $O(n+g)$ time, given $I(G)$.*

Proof: Let C be the set constructed by Algorithm 3.2 for the graph G. The proof will consist of three parts.

A. Proof that C is an ε-separator of G

It will be appropriate to define the following set of curves that contain all vertices of C

$$\tilde{C} = \{\bigcup_{j=0}^{l} \tilde{L}(k+jh)\} \cup \{\bigcup_{j=0}^{l+1} \tilde{C}_j\}, \tag{8}$$

where $\tilde{L}(k+jh)$ is the embedding of the cycles in $L(k+jh)$; $\tilde{C}_j$ denotes the set of the embeddings of the T-cycles corresponding to the edges in the separator M_j, where both endpoints are on levels between $k+(j-1)h$ and $k+jh$.

To simplify our presentation we will introduce more convenient notations. For $j=1,\ldots,l+1$ we denote $X(k+(j-1)h)$ by X_j, where X can be any of the symbols Z, Z^+, $\tilde{Z}$, S, S^+, and $\tilde{S}$. Additionally, let us denote $Z_0 = S_0 = \emptyset$, $Z_0^+ = \tilde{Z}_1$, $\tilde{Z}_{l+2} = Z$, and $\tilde{S}_{l+2} = S$.

Lemma 3.2 *The set of curves $\tilde{C}$ divides Z into regions of weights not exceeding $\varepsilon wt(G)$, where the weight of a region is defined as the total weight of the vertices of G embedded onto it.*

Proof of the lemma: We will show that the set of curves $\tilde{C}_{l+1}$ divides $Z_{l+1}^+ = Z \setminus Z_{l+1}$ into regions of weights not exceeding $\varepsilon wt(G)$. Consider the set M of edges of S containing the edges dual to the edges of the cycles of $\tilde{L}(k+lh)$ plus the edges included in M_{l+1}. According to Theorem 2.1, there is an one-to-one correspondence between the components of $S \setminus M$ and the regions of $Z \setminus \{\tilde{L}(k+lh) \cup \tilde{C}_{l+1}\}$ associated with $I(G)$. In addition, the weight of each region does not exceed the weight of the corresponding component of

$S \setminus M$ (see Theorem 2.1 (i)). The set of curves $\tilde{L}(k+lh)$ separates Z_{l+1} from the regions on Z_{l+1}^{+}. The set of edges M_{l+1} separates S_{l+1}^{+} into components of weight not exceeding $\varepsilon wt(G)$ (note that $wt(S) = wt(G)$) and therefore the set of curves $\tilde{C}_{l+1}$ divides Z_{l+1}^{+} into regions of weights not exceeding $\varepsilon wt(G)$. Applying the same argument we prove that for $j = l, \ldots, 0$ the set of curves $\tilde{C}(M_j)$ divides Z_j^{+} into regions of weights not exceeding $\varepsilon wt(G)$. According to our definitions, the set $\tilde{C}(M_j)$ is a superset of $\tilde{C}_j$, since Z_j^{+} is equal to $Z_{j+1} \setminus Z_j$ plus the set of open discs (faces) pasted onto the holes determined by the curves in (embedded cycles) $\tilde{L}(k+jh)$. But $\tilde{C}_j$ is a restriction of $\tilde{C}(M_j)$ on $Z_{j+1} \setminus Z_j$ and thus $\tilde{C}_j$ divides $Z_{j+1} \setminus Z_j$ into regions of weights not exceeding $\varepsilon wt(G)$. The lemma follows by this fact and the observation that the set of curves $\cup_{j=0}^{l} \tilde{L}(k+jh)$ divides Z into the regions $Z_{j+1} \setminus Z_j,\ j = l+1, \ldots, 0$.$\square$

The vertex set C constructed by the algorithm induces a subgraph of G whose embedding contains the set of curves $\tilde{C}$. By Lemma 3.2 and the definitions it follows that C is an ε-separator for G.

B. Time complexity

Steps 1,2, and 3 run in $O(|I(G)|)$ time. Each of the substeps included in the loop of Step 4 runs in linear time on its input. During the Step 4 the algorithm traverses the graph and therefore this step and the algorithm run in $O(|I(G)|) = O(n+g)$ time.

C. Estimation of the size of C

The set C was constructed as the union

$$C = L \cup \{\bigcup_{j=0}^{l+1} C_j\}, \tag{9}$$

where $L = \cup_{j=0}^{l} L(k+jh)$ is the set of levels determined in Step 3 and sets C_j were constructed in Step 4.5. By the choice of k in Step 3, it follows $|L| \leq n/h$ and thus

$$|C| \leq \frac{n}{h} + \sum_{j=0}^{l+1} |C_j'|,$$

where C_j' denotes the set C_j without the vertices in L. The sets C_j' consist of parts of T-cycles that lie on levels between $k+jh$ and $k+(j-1)h$ for

$j > 0$ and on levels less than k for $j = 0$. Therefore each of these parts could contain no more than $2(h-1)$ vertices and hence from the previous inequality we obtain

$$|C| \leq \frac{n}{h} + 2(h-1)\sum_{j=0}^{l+1}|M_j|. \tag{10}$$

Let us now estimate the sum $\sum_{j=0}^{l+1}|M_j|$. For $j = 0,\dots,l+1$, we denote by $q(S_j^+)$ the number of the connected components of the graph S_j^+. According to the construction of the sets M_j in Step 4.4, we have

$$|M_j| \leq |E(S_j^+)| - |V(S_j^+)| + q(S_j^+) + \left\lfloor \frac{2wt(S_j^+)}{\varepsilon wt(G)} \right\rfloor,$$

which follows by Lemma 3.1 if $wt(S_j^+) > \varepsilon wt(G)$, or by the fact that the right side is nonnegative otherwise (in this case $|M_j| = 0$). Summing these inequalities over $j = 0,\dots,l+1$, we obtain

$$\begin{aligned}\sum_{j=0}^{l+1}|M_j| &\leq \sum_{j=0}^{l+1}(|E(S_j^+)| - |V(S_j^+)| + q(S_j^+) + \left\lfloor \frac{2wt(S_j^+)}{\varepsilon wt(G)} \right\rfloor) \leq \\ &\leq \frac{2}{\varepsilon} + \sum_{j=0}^{l+1} q(S_j^+) + \sum_{j=0}^{l+1}(|E(S_j^+)| - |V(S_j^+)|), \end{aligned} \tag{11}$$

since $\sum_{j=0}^{l+1} wt(S_j^+) \leq wt(G)$. By the construction of graphs $\tilde{S}_j$ in the Subgraph Supplementation procedure it follows , for $j = 0,1,\dots,l+1$, that

$$E(\tilde{S}_{j+1}) = \left\{E(S_j^+) \cup E(\tilde{S}_j)\right\} \setminus \{\bigcup_{c \in \tilde{L}(k+(j-1)h)} E(tr(c))\}$$

$$V(\tilde{S}_{j+1}) = \left\{V(S_j^+) \cup V(\tilde{S}_j)\right\} \setminus \{\bigcup_{c \in \tilde{L}(k+(j-1)h)} V(tr(c))\}$$

and

$$\begin{aligned}&|E(\tilde{S}_{j+1})| - |V(\tilde{S}_{j+1})| = \\ &\quad = |E(S_j^+)| - |V(S_j^+)| + |E(\tilde{S}_j)| - |V(\tilde{S}_j)| + \\ &\qquad + \sum_{c \in \tilde{L}(k+(j-1)h)} (|V(tr(c))| - |E(tr(c))|) = \\ &\quad = |E(S_j^+)| - |V(S_j^+)| + |E(\tilde{S}_j)| - |V(\tilde{S}_j)| + |\tilde{L}(k+(j-1)h)|, \end{aligned} \tag{12}$$

where $|\tilde{L}(k+(j-1)h)|$ denotes the number of the cycles in $\tilde{L}(k+(j-1)h)$. Summing (12) over j and since $\tilde{S}_0 = \emptyset$ and $\tilde{S}_{l+2} = S$ we obtain that

$$\sum_{j=0}^{l+1} \Big(|E(S_j^+)| - |V(S_j^+)|\Big) = |E(S)| - |V(S)| - \sum_{j=0}^{l+1} |\tilde{L}(k+(j-1)h)|. \quad (13)$$

We substitute (13) in (11) and apply (2)

$$\begin{aligned}\sum_{j=0}^{l+1} |M_j| &\leq \frac{2}{\varepsilon} + |E(S)| - |V(S)| + \sum_{j=0}^{l+1} (q(S_j^+) - |\tilde{L}(k+(j-1)h)|) = \\ &= \frac{2}{\varepsilon} + 2g - 1 + \sum_{j=0}^{l+1} (q(S_j^+) - |\tilde{L}(k+(j-1)h)|) \quad (14)\end{aligned}$$

(see Lemma 2.1). For each $j = 0, \ldots, l+1$, the number $|\tilde{L}(k+(j-1)h)|$ is the number of cycles on the boundary of Z_j. Using Theorem 2.1 it is easy to see that the number of components of the graph S_j^+ is equal to the number of connected regions of $\tilde{Z}_{j+1} \setminus Z_j$. Each of the connected regions of $\tilde{Z}_{j+1} \setminus Z_j$ has at least one of the cycles from $\tilde{L}(k+(j-1)h)$ on its boundary and no two of these regions have common cycle on their boundaries. Hence, the number of cycles in $\tilde{L}(k+(j-1)h)$ is grater or equal to the number of connected regions of $\tilde{Z}_{j+1} \setminus Z_j$, which gives the inequality

$$q(S_j^+) \leq |\tilde{L}(k+(j-1)h)| \quad \text{for} \quad j = 0, 1, \ldots, l+1. \quad (15)$$

By (14), and (15) it follows

$$\sum_{j=0}^{l+1} |M_j| \leq 2g + \frac{2}{\varepsilon}. \quad (16)$$

We obtain the desired estimation substituting (16) and the value of h in (10), i.e.

$$|C| \leq \frac{n}{h} + 2(h-1)(2g + \frac{2}{\varepsilon}) \leq 4\sqrt{(g + 1/\varepsilon)n} \ .$$

□

As we will show in the last subsection, the size of C is optimal within a constant factor for any $\varepsilon \leq 2/3$. On the other hand, when ε is close to one, it is evident that the size of C can not be optimal. We investigate such type of separators in the next subsection.

Intuitively, when ε is close to one, we need to cut off a small piece of the given graph in order to divide it into parts of sizes not greater than ε and therefore the separator (the "boundary" of this small piece) should be small too. An algorithm that finds good separators in this case could be useful for designing incremental type algorithms on graphs.

3.3 Finding ε-separators for ε close to one

It will be more convenient to talk about $(1-\varepsilon)$-separators for small values of ε instead of talking about ε-separators for big values of ε. Below we show how Algorithm 3.2 can be extended to cope with $(1-\varepsilon)$-separators when ε is small. More precisely we present an algorithm based on Algorithm 3.2 that, for $\varepsilon \in (0, 1/15)$, finds an $(1-\varepsilon)$-separator C_1 of no more than $2 + 30\sqrt{2}\sqrt{n\varepsilon(2g\varepsilon+1)}$ vertices.

The idea of the algorithm for constructing C_1 is the following. We first run Algorithm 3.2 on the input graph G with parameter 2ε, storing information for the obtained division. Let the constructed separator be $C_{2\varepsilon}$. As discussed above, the set $C_{2\varepsilon}$ is defined as the set of vertices belonging to the set of curves $\tilde{C}_{2\varepsilon}$, see (8). The set of curves $\tilde{C}_{2\varepsilon}$ divides the surface Z into connected open regions, which we denote by $Z_1, \ldots, Z_{p''}$. Let $G_1, \ldots, G_{p''}$ be the subgraphs of G embedded inside these regions and $\partial G_1, \ldots, \partial G_{p''}$ be the subgraphs that form region boundaries. Clearly the vertex set $V(\partial G_i)$ separates the graph G_i from G. From the previous subsection we know that the weight of any Z_i does not exceed $2\varepsilon wt(G)$. For our purposes, however, we will need some *lower* bound on the size of the parts of G. For this end we "group" the sets Z_i as follows. Using the division $Z_1, \ldots, Z_{p''}$ we derive a collection $V_1, \ldots, V_p (p \leq p'')$ of non-intersecting sets of vertices of G. Each of these sets contains the vertices of an appropriately chosen set of graphs G_i so that its weight plus the weight of its boundary exceeds $\varepsilon wt(G)$. The boundary of a given vertex set is defined as the set of vertices that are not in the set but are adjacent to a vertex in the set. So, the boundary ∂V_i of a set V_i, for each $i = 1, \ldots, p$, is the union of the vertices of the boundaries of those of the graphs $G_1, \ldots, G_{p''}$, whose vertex sets are in V_i. We find the required $(1-\varepsilon)$-separator C_1 as the smallest amongst the sets $\partial V_1, \ldots, \partial V_p$.

Next we give some more details about how to find the sets V_i for $i = 1, \ldots, p$. Initially, we define a collection of p' sets $U_i(r)$ ($p \leq p' \leq p''$), where $i = 1, \ldots, p'(r)$ and $r \in R_{k,h} = \{k + lh, k + (l-1)h, \ldots, k, 0\}$, during the implementation of Algorithm 3.2 as follows. Consider any specific iteration of Step 4, i.e. let $r \in R_{k,h}$ be fixed. In Step 4.4 we found an edge separator M for the graph $S^+(r)$. Let the connected components of $S^+(r) \setminus M$ be $S_1^+(r), \ldots, S_{p'(r)}^+(r)$. According to Theorem 2.1, these components correspond to regions $Z_1^+(r), \ldots, Z_{p'(r)}^+(r)$ that satisfy inequalities (4), (5). The relation between the regions $Z_1^+(r), \ldots, Z_{p'(r)}^+(r)$ and the regions $Z_1, \ldots, Z_{p''}$ is that each connected region of the interior of the intersection $Z(r) \cap Z_i^+(r)$ is equal to one of the regions $Z_1, \ldots, Z_{p''}$. (Note that the region $Z_i^+(r)$ could contain parts of faces added by the Subgraph Supplementation procedure that pastes open disks onto the holes of $Z(r)$.) Then, for $i = 1, \ldots, p'(r)$, we define $U_i(r)$ as the set of vertices of G embedded inside $Z \cap Z_i^+(r)$. The boundary $\partial U_i(r)$ consists of the vertices of G lying on the boundary of the region $Z \cap Z_i^+(r)$. Since the separation graphs are associated with the embedding of G, i.e. the vertices of a component $S_i^+(r)$ correspond to the triangles inside the region $Z_i^+(r)$ for $i = 1, \ldots, p'(r)$, then we easily construct sets $U_i(r)$ and their boundaries, for $i = 1, \ldots, p'(r)$, from the components of $S^+(r) \setminus M$ during the corresponding iteration of Step 4 in Algorithm 3.2. According to Lemma 3.1 and Theorem 2.1

$$wt(U_i(r)) \leq 2\varepsilon wt(G), \quad i = 1, \ldots, p'(r). \tag{17}$$

Additionally, since the number of the connected components of $S^+(r)$ is $q(r)$, then for at least $p'(r) - q(r)$ (w.l.o.g. for the first $p'(r) - q(r)$) of the sets $U_i(r)$ we have

$$\varepsilon wt(G) \leq wt(U_i(r) \cup \partial U_i(r)), \quad i = 1, \ldots, p'(r) - q(r). \tag{18}$$

By reordering the sequence of the sets $U_i(r)$ so that the sets satisfying inequality (18) precede the others, we obtain a sequence $U_1, \ldots, U_{p'}$ with $p' = \sum_{r \in R_{k,h}} p'(r)$. Let the sets satisfying (18) be $U_1, \ldots, U_{p_1}$ for some $p_1 \leq p'$. For $i = 1, \ldots, p_1$ we define $V_i = U_i$.

In the second stage we define sets $V_{p_1+1}, \ldots, V_p$ as unions of groups of the "light" sets $U_{p_1+1}, \ldots, U_{p'}$, i.e. we find numbers $i_0 = 0 < i_1 < \ldots < i_{p-p_1} \leq$

$p' - p_1$, so that

$$\varepsilon wt(G) \leq \sum_{i=p_1+i_{j-1}+1}^{p_1+i_j} wt(U_i \cup \partial U_i) \leq 2\varepsilon wt(G), \quad j = 1, \ldots, p - p_1,$$

and

$$\sum_{i=p_1+i_{p-p_1}+1}^{p'} wt(U_i \cup \partial U_i) \leq 2\varepsilon wt(G).$$

Such numbers i_j, for $j = 1, \ldots, p - p_1$, exist since the sets U_i, for $i = p_1, \ldots, p'$, do not satisfy (18). So the sets $V_{p_1+1}, \ldots, V_p, V_{p+1}$ are defined by

$$V_{p_1+j} = \bigcup_{i=p_1+i_{j-1}+1}^{p_1+i_j} U_i, \quad \text{for} \quad j = 1, \ldots, p - p_1,$$

and

$$V_{p+1} = \bigcup_{i=p_1+i_{p-p_1}+1}^{p'} U_i.$$

We define the boundary ∂V_i of a set V_i to be the union of the boundaries of the sets that are included in V_i.

As described above, we obtain vertex sets $V_1, \ldots, V_{p+1}$ with boundaries $\partial V_1, \ldots, \partial V_{p+1}$ satisfying

$$wt(V_i) \leq 2\varepsilon wt(G) \quad \text{for } i = 1, \ldots, p+1, \tag{19}$$

and

$$wt(V_i \cup \partial V_i) \geq \varepsilon wt(G) \quad \text{for } i = 1, \ldots, p. \tag{20}$$

Each of the boundaries $\partial V_1, \ldots, \partial V_p$ is an $(1-\varepsilon)$-separator and we just chose the minimal amongst these separators. Formally, our algorithm is presented below.

Algorithm 3.3 CONSTRUCTION OF AN $(1-\varepsilon)$-SEPARATOR

Input: A 2-cell embedding of a graph G on a surface of orientable genus g; weight function defined on vertices of G; a number $\varepsilon \in (0, 1/15)$.
Output: A set C_1 of vertices that is an $(1-\varepsilon)$-separator of G.

Step 1: Run Algorithm 3.2 on G with parameter 2ε. Find a (2ε)-separator $C_{2\varepsilon}$, vertex sets $U_1, \ldots, U_{p'}$, and their boundaries $\partial U_1, \ldots, \partial U_{p'}$, as defined above.

Step 2: If $wt(C_{2\varepsilon}) \geq (1/15)wt(G)$, then define C_1 to contain the heaviest $\lceil 15\varepsilon |C_{2\varepsilon}| \rceil$ vertices of $C_{2\varepsilon}$ and output C_1.

Step 3: As described before the algorithm construct sets $V_1, \ldots V_{p+1}$ and boundaries $\partial V_1, \ldots \partial V_{p+1}$, so that the inequalities (19), (20) hold.

Step 4: Amongst the sets of vertices $\partial V_1, \ldots, \partial V_p$ find the set containing minimum number of vertices. Output that set to be the required separator C_1.

The correctness and complexity of this algorithm will be formally proved in the next theorem.

Theorem 3.2 *Let G be an n-vertex graph with non-negative vertex weights and $I(G)$ be a 2-cell embedding of G on an orientable surface Z of genus g. For any $\varepsilon \in (0, 1/15)$ there exists an $(1-\varepsilon)$-separator C_1 of G of no more than $2 + 30\sqrt{2}\sqrt{n\varepsilon(2g\varepsilon + 1)}$ vertices. The separator C_1 can be found in $O(n + g)$ time given $I(G)$.*

Proof: Let C_1 be the separator constructed by Algorithm 3.3. We divide our proof into three parts.

A. Proof that C_1 is an $(1-\varepsilon)$-separator
Consider the case where the 2ε-separator $C_{2\varepsilon}$ found in Step 1 has weight greater than $(1/15)wt(G)$. In this case, according to the construction of C_1 in Step 2, we have

$$wt(C_1) \geq \frac{\lceil 15\varepsilon |C_{2\varepsilon}| \rceil}{|C_{2\varepsilon}|} wt(C_{2\varepsilon}) \geq \varepsilon wt(G).$$

This means that C_1 is an $(1-\varepsilon)$-separator of G and the theorem follows, since by Theorem 3.1

$$\lceil 15\varepsilon |C_{2\varepsilon}| \rceil \leq \left\lceil 30\sqrt{2}\sqrt{n\varepsilon(2\varepsilon g + 1)} \right\rceil.$$

Hereafter, we assume that

$$wt(C_{2\varepsilon}) < (1/15)wt(G). \tag{21}$$

In this case the set C_1 is found in Step 4 as the smallest of the boundaries $\partial V_1, \ldots, \partial V_p$. Therefore C_1 is an $(1-\varepsilon)$-separator for G, since the inequality (19) holds for each of the sets $V_1, \ldots, V_p$.

B. Time complexity

The first step of Algorithm 3.3 consists of running Algorithm 3.2 and constructing the sets $U_1, \ldots, U_{p'}$ and their boundaries. Algorithm 3.2 runs in $O(n+g)$ as shown in Theorem 3.1. The additional time for the construction of the sets $U_1, \ldots, U_{p'}$ and $\partial U_1, \ldots, \partial U_{p'}$ during the implementation of Algorithm 3.2 is clearly linear too. Step 2 can be implemented in $O(n)$ time by using a linear algorithm for ordered statistics. Steps 3 and 4 obviously run in $O(n)$ time. Thereby, the complexity of Algorithm 3.3 constructs an $(1-\varepsilon)$-separator for G in $O(n+g)$ time.

C. Estimation of the size of C_1

For the size of C_1, we have by Step 4

$$|C_1| \leq \frac{1}{p}\sum_{i=1}^{p} |\partial V_i|. \tag{22}$$

First, we estimate the sum $\sum_{i=1}^{p} |\partial V_i|$ and then we give an estimation of p. Recall that the union $\cup_{i=1}^{p+1} \partial V_i$ coincides with the separator $C_{2\varepsilon}$. On the other hand Algorithm 3.2 constructs $C_{2\varepsilon}$ as the set of vertices lying onto a set of curves $\tilde{C}_{2\varepsilon}$, see (8). The set of curves $\tilde{C}_{2\varepsilon}$ could be considered as an embedding of a graph with a vertex set $V(\tilde{C}_{2\varepsilon}) = \bigcup_{i=1}^{p''} V(\partial Z_i)$, an edge set $E(\tilde{C}_{2\varepsilon})$ consisting of the edges embedded on $\tilde{C}_{2\varepsilon}$, and faces $Z_1, \ldots, Z_{p''}$. Note, that this embedding may not be a 2-cell embedding. In this case the Euler's formula (1) gives the inequality

$$|E(\tilde{C}_{2\varepsilon})| - |V(\tilde{C}_{2\varepsilon})| \leq 2g + p'' - 2. \tag{23}$$

Each edge of $E(\tilde{C}_{2\varepsilon})$ belongs to the boundary of two of the regions $Z_1 \ldots, Z_{p''}$. The boundaries $\partial Z_1 \ldots, \partial Z_{p''}$ consist of simple closed curves (cycles). Then

$$\sum_{i=1}^{p''} |V(\partial Z_i)| = 2E(\tilde{C}_{2\varepsilon})$$

By the definition of the sets V_i and (23) we obtain

$$\sum_{i=1}^{p+1} |\partial V_i| \leq \sum_{i=1}^{p''} |V(\partial Z_i)| = 2E(\tilde{C}_{2\varepsilon}) \leq 2\left(|C_{2\varepsilon}| + 2g + p'' - 2\right). \tag{24}$$

Next, we find a lower bound for p. By (19),

$$(p+1)2\varepsilon wt(G) \geq \sum_{i=1}^{p+1} wt(V_i) = wt(G \setminus C_{2\varepsilon}).$$

By (21) and since $\varepsilon < 1/15$

$$p \geq \frac{wt(G) - wt(C_{2\varepsilon})}{2\varepsilon wt(G)} - 1 > \frac{2}{5\varepsilon}. \tag{25}$$

Combining (22), (24), and (25) we obtain

$$|C_1| \leq 2 + 5\varepsilon\left(|C_{2\varepsilon}| + 2g + p'' - p - 2\right). \tag{26}$$

To complete the estimation of $|C_1|$ we use the inequality

$$p'' - p \leq 6\sqrt{n(g + \frac{1}{2\varepsilon})} + 6(g-1), \tag{27}$$

that will be proved below. We substitute (27) in (26), and by using $n > 16g$ and Theorem 3.1 we obtain

$$\begin{aligned}|C_1| &\leq 2 + 5\varepsilon\left(|C_{2\varepsilon}| + 6\sqrt{n(g + \frac{1}{2\varepsilon})} + 8(g-1)\right) \\ &\leq 2 + 60\varepsilon\sqrt{n(g + \frac{1}{2\varepsilon})} = 2 + 30\sqrt{2}\sqrt{n\varepsilon(2g\varepsilon + 1)}.\end{aligned}$$

Let us now prove (27). Recall that p'' is the number of the connected open regions $Z_1, \ldots, Z_{p''}$ of $Z \setminus \tilde{C}_{2\varepsilon}$ and p is the number of the vertex sets $V_1, \ldots, V_p$. We will use the parameter p', which is the number of the vertex sets $U_1, \ldots, U_{p'}$. Since by definition $U_1, \ldots, U_{p'}$ form a subpartition of $V_1, \ldots, V_p$ and for $i = 1, \ldots, p'$ the set U_i comprises the vertices embedded into one or more of the regions Z_j for $j = 1, \ldots, p''$, then $p \leq p' \leq p''$. According to our constructions, vertices embedded in two regions Z_{i_1} and Z_{i_2}

are in one $U_i(r)$ for some $i = 1, \ldots, p'(r)$ and $r \in R_{k,h}$, if and only if the regions Z_{i_1} and Z_{i_2} are components of $Z \cap Z^+(r)$. Therefore, both Z_{i_1} and Z_{i_2} neighbor a hole of $Z(r)$, see Figure 3. Thereby, they have at least one edge of the graph $\tilde{L}(r)$ on its boundary, where $\tilde{L}(r)$ was defined as the graph of cycles $\{c_1(r), \ldots, c_{\nu(r)}(r)\}$ forming the boundary of $Z(r)$. Moreover, no two regions corresponding to a fixed r can be adjacent the same edge of $\tilde{L}(r)$. Hence

$$p'' - p' \leq \sum_{r \in R_{k,h}} (|E(\tilde{L}(r)| - \nu(r)). \tag{28}$$

Further, the difference $p' - p$ does not exceed the number $p' - p_1$ of sets U_i, for $i = 1, \ldots, p'$, for which (18) is not valid. As pointed out, for any $r \in R_{k,h}$, at most $q(r)$ of the sets $U_i(r)$ do not satisfy (18), where $q(r)$ denotes the number of connected components of $S^+(r)$. Thus,

$$p' - p_1 \leq \sum_{r \in R_{k,h}} q(r)$$

and by (15), where the numbers $q(r)$ are compared to the numbers $\nu(r)$ denoting the number of cycles in $\tilde{L}(r)$, we obtain

$$p' - p \leq p' - p_1 \leq \sum_{r \in R_{k,h}} q(r) \leq \sum_{r \in R_{k,h}} \nu(r). \tag{29}$$

Adding (28) to (29) gives

$$p'' - p \leq \sum_{r \in R_{k,h}} |E(\tilde{L}(r)|. \tag{30}$$

So, to estimate $p'' - p$ we need an estimation for the number of edges of the graph $\tilde{L} = \bigcup_{r \in R_{k,h}} \tilde{L}(r)$. The graph $\tilde{L}$ is simplicial (has no parallel edges or loops) and has an embedding (not necessarily a 2-cell) onto a surface of genus g. Therefore, using the Euler's formula (1) as inequality and the inequality $2|E(\tilde{L})| \geq 3|F(\tilde{L})|$ between the number of edges and the number of faces in any embedding of a simplicial graph, we obtain

$$|E(\tilde{L})| \leq 3|V(\tilde{L})| + 6(g-1). \tag{31}$$

The set $V(\tilde{L})$ is the union of the vertices lying on the levels $L(r)$ for $r \in R_{k,h}$. According to the choice of h and k (see Step 3 of Algorithm 3.2), we have

$$|V(\tilde{L})| = \sum_{r \in R_{k,h}} |L(r)| \leq 2\sqrt{(g + \frac{1}{2\varepsilon})n},$$

that, together with (31) and (30), implies (27). □

3.4 Tightness of Results

In that subsection we show that the results of Theorems 3.1 and 3.2 are optimal up to a constant factor. More precisely, we are going to prove the following theorem.

Theorem 3.3 *There exist a constant $d > 0$ such that for any $\varepsilon \in (0, 2/3)$, g, and $n > g + 1/\varepsilon$*

(i) *There exists an n-vertex graph G of genus g that has no ε-separator of size less than $d\sqrt{n(g + 1/\varepsilon)}$;*

(ii) *There exists an n-vertex graph G_1 of genus g that has no $(1 - \varepsilon)$-separator of size less than $d\sqrt{n\varepsilon(g\varepsilon + 1)}$.*

Proof: We will base our proof on two special cases of our theorem that are proved in previous works. The case $g = 0$ follows immediately from the result of Theorem 6 in [18]. The case $\varepsilon = 2/3$ is proved in [6, 11].

We consider graphs with equal weights on the vertices. For a graph G and $\varepsilon \in (0, 1)$ we denote by $c(G, \varepsilon)$ the minimum number of vertices in an ε-separator of G. Let $K(n, g)$ be the class of graphs with no more than n vertices and genus not exceeding g. We define the following function

$$\mu(n, g, \varepsilon) = \max_{G \in K(n,g)} c(G, \varepsilon).$$

From the definition it is clear that the function μ is increasing on n and g and decreasing on ε. The validity of the theorem in the cases $g = 0$ and $\varepsilon = 2/3$ means that

$$\mu(n, 0, \varepsilon) = \Omega\left(\sqrt{n/\varepsilon}\right), \tag{32}$$

$$\mu(n, 0, 1 - \varepsilon) = \Omega\left(\sqrt{n\varepsilon}\right), \tag{33}$$

$$\mu(n, g, 2/3) = \Omega\left(\sqrt{ng}\right). \tag{34}$$

By using the monotonicity of μ, (32) and (34) we obtain

$$\begin{aligned} 2\mu(n, g, \varepsilon) \geq \mu(n, 0, \varepsilon) + \mu(n, g, 2/3) &= \\ \Omega\left(\sqrt{n/\varepsilon}\right) + \Omega\left(\sqrt{ng}\right) &= \Omega\left(\sqrt{n(g + 1/\varepsilon)}\right), \end{aligned}$$

which proves (i).

We shall prove (ii) by considering two cases.

1) $\varepsilon g \leq 1$. In this case by the monotonicity of μ and (33) we have

$$\mu(n, g, 1-\varepsilon) \geq \mu(n, 0, 1-\varepsilon) = \Omega\left(\sqrt{n\varepsilon}\right) = \Omega\left(\sqrt{n\varepsilon(g\varepsilon+1)}\right),$$

which gives (ii).

2) $\varepsilon g > 1$. To show that (ii) holds in this case, we shall prove first that

$$\varepsilon\mu(n, g, 2/3) \leq \mu(n, g, 1-\varepsilon). \tag{35}$$

Let G be any graph from $K(n, g)$. We will construct a 2/3-separator of G as an union of $(1-\varepsilon)$-separators. We set $G_0 = G$ and define a sequence of graphs $G_1, \ldots, G_k$ in the following way. Let C_{i-1}, for $i = 1, \ldots, k$, be the minimal $(1-\varepsilon)$-separator of the graph G_{i-1} and G_i be the greatest component of $G_{i-1} \setminus C_{i-1}$. The weights $wt(G_i)$ decrease with i and satisfy the inequalities $wt(G_i) \leq (1-\varepsilon)^i wt(G)$, for $i = 1, \ldots, k$. The integer k is chosen to be the smallest such that $wt(G_k) \leq (2/3)wt(G)$ and thereby $k \leq 1/\varepsilon$. Since C_{i-1}, for $i = 1, \ldots, k$, is an $(1-\varepsilon)$-separator of G_{i-1}, then the only component of $G_{i-1} \setminus C_{i-1}$, for $i = 1, \ldots, k$ that is heavier than $(2/3)wt(G)$ is G_i. Thus the union $C = \cup_{i=1}^{k} C_{i-1}$ is a 2/3-separator of G. Therefore, we have

$$c(G, 2/3) \leq |C| = \sum_{i=1}^{k} |C_{i-1}| = \sum_{i=1}^{k} c(G_i, 1-\varepsilon) \leq k\mu(n, g, 1-\varepsilon),$$

which together with the inequality $k \leq 1/\varepsilon$ gives (35). Finally, we prove (ii) by combining (35), (34), and $\varepsilon g > 1$

$$\mu(n, g, 1-\varepsilon) \geq \varepsilon\ \mu(n, g, 2/3) = \varepsilon\Omega\left(\sqrt{ng}\right) = \Omega\left(\sqrt{n\varepsilon(g\varepsilon+1)}\right).$$

$\square$

References

[1] L. Aleksandrov, H.N. Djidjev, Improved upper and lower bounds on separation of toroidal graphs, Proc. Optimal algorithms'89, in *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo (1989), vol.401, 126-138.

[2] N. Alon, P. Seymour, R. Thomas, A separator theorem for graphs with an excluded minor and its applications, Proceedings of *the 22nd Ann. ACM Simp. on Theory of Computing*, 1990, 293-299.

[3] S.N. Bhatt, F.T. Leighton, A framework for solving VLSI Graph layout problems, *J.Comp. and Sys. Sci.*, 28(2),1984, 300-343.

[4] H.N. Djidjev, K. Diks, O. Sykora, I. Vrto, Edge separators for planar graphs and their applications, *Journal of Algorithms*, vol. 14, (1993), 258-279.

[5] H.N. Djidjev, Linear algorithms for graph separation problems, Proceedings of SWAT'88, *Lecture Notes in Computer Science*, vol. 318, Springer–Verlag, 1988, 216-221.

[6] H.N. Djidjev, A separator theorem, *Compt. rend. Acad. bulg. Sci.*, vol. 34, 1981, 5, 643-645.

[7] H. N. Djidjev, J.R. Gilbert, Separators in graphs with negative or multiple vertex weights, *SIAM Conf. on Discrete Mathematics*, Atlanta, Georgia, 1990.

[8] G.N. Frederickson, Fast algorithms for shortest paths in planar graphs with applications, *SIAM J. Computing*, 16(6), 1987, 1004-1021.

[9] G.N. Frederickson, R. Janardan, Efficient message rooting in planar networks, *SIAM J. Computing*, 18, 1989, 843-857.

[10] J.R. Gilbert, Graph separator theorem and sparse Gaussian elimination, *Ph. D. Thesis*, Stanford University, 1980.

[11] J.R. Gilbert, J.P. Hutchinson, R.E. Tarjan, A Separator theorem for graphs of bounded genus, *Journal of Algorithms*, 5, 1984, 391-407.

[12] J.R. Gilbert, R.E. Tarjan, The analysis of a nested disection algorithm, *Numerische Mathematik*, 50(4), 1987, 377-404.

[13] M.T. Goodrich, Planar Separators and Parallel Polygon Triangulation, Proceedings of *the 24-th Symp. on Theory of Computing*, 1992, 507-516.

[14] J.L. Gross, T.W. Tucker, *Topologycal Graph Theory*, John Wiley & Sons, 1987.

[15] X. He, Y. Yesha, A nearly optimal parallel algorithm for constructing depth-first spanning trees in planar graphs. *SIAM J. Computing*, 17(3), 1988, 486-491.

[16] C.E. Leiserson, *Area efficient VLSI computation*, Foundations of Computing, MIT Press, Cambridge, MA, 1983.

[17] R.J. Lipton, D.J. Rose, R.E. Tarjan, Generalized nested dissection, *SIAM J. Numer. Analysis*, Vol. 16, 1979, 346-358.

[18] R.J. Lipton, R.E. Tarjan, A separator theorem for planar graphs, *SIAM J. Appl. Math.*, Vol. 36, 1979, 177-189.

[19] R.J. Lipton, R.E. Tarjan, Applications of a planar separator theorem, *SIAM J. Comput.*, Vol. 9, 1980, 615-627.

[20] G.L. Miller, S.-H. Teng, S.A. Vavasis, A unified geometric approach to graph separators, Proceedings of *the 23rd Ann. ACM Simp. on Theory of Computing*, 1991, 538-547.