Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 32, No. 3, June 1986
Printed in Belgium

Finding Small Simple Cycle Separators for 2-Connected Planar Graphs

GARY L. MILLER*

*Department of Computer Science, University of Southern California
Los Angeles, California 90089*

Received October 15, 1985; revised January 14, 1986

We show that every 2-connected triangulated planar graph with n vertices has a simple cycle C of length at most $2\sqrt{2\cdot n}$ which separates the interior vertices A from the exterior vertices B such that neither A nor B contain more than $2/3n$ vertices. The method also gives a linear time sequential algorithm for finding this simple cycle and an NC parallel algorithm. In general, if the maximum face size is d then we exhibit a cycle C as above of size at most $2\sqrt{d\cdot n}$. © 1986 Academic Press, Inc.

1. INTRODUCTION

Many computationally efficient algorithms are known for both trees and planar graphs which do not seem to generalize to all graphs. One basic technique in the design of these algorithms is "divide-and-conquer." For both trees and planar graphs one uses the fact that there exists a small subset of vertices which separates the graph into roughly two equal pieces. For trees one can easily see that there must exist a single vertex which separates the remaining vertices into at worst a "$\frac{1}{3}$–$\frac{2}{3}$" split.

Formally, we say G with n vertices has an $f(n)$-*vertex separator* if there exists a partition of the vertices into three sets A, B, and C such that the size of $C \leqslant f(n)$, the size of A and B are $\leqslant 2n/3$, and no edge exists between A and B. Thus, a tree has 1-vertex separator. Lipton and Tarjan [16] showed that planar graphs have $2\sqrt{2\cdot n}$ vertex separators and Djidjev [6] improved this to $\sqrt{6}\sqrt{n}$ vertex separators. Two now classic applications of the vertex separator theorems for planar graphs are planar graph layouts for VLSI [23, 20] and nested dissection in numerical analysis [15].

It can be easily shown that a simple cycle in an embedded planar graph will partition the faces and the remaining vertices and edges into two sets, those in the interior and those in the exterior. In this paper we shall construct a small separator which forms a simple cycle, called, a *simple cycle separator*. In the special case when

* This research was supported in part by National Science Foundation Grant NSF MCS 80-07756 and Air Force Office of Scientific Research AFOSR-82-0326.

0022-0000/86 $3.00

the embedded graph is triangulated we shall find a simple cycle separator of size $2\sqrt{2 \cdot n}$. We next motivate a stronger theorem.

Some applications of the "divide-and-conquer" technique for planar graphs require separators with more structure. As an example, the planar flow algorithm of Johnson and Venhatesan [12] first constructs a planar separator which consists of a collection of nonnesting cycles. Their algorithm can be simplified if a small simple cycle separator were first found as described in this paper.

A subset C of edges is an *edge separator* if there exists a partition A, B of the vertices such that any edge between A and B is in C and the size of A and B are at most $2n/3$. Note that even trees may have only large edge separators. As an example, the star graph which consists of a single vertex connected to n other vertices requires an edge separator to have $n/3$ edges. We may want to capture the intuitive notion that an edge separator corresponds to a simple incision. Formally, we ask for a simple cycle in the geometric dual of the embedded planar graph which separates faces of the embedded dual graph. Recall that the *geometric dual* of an embedded graph consists of a vertex for every face and two dual vertices share a dual edge for each edge they share as faces, see Fig. 1. This motivates a natural generalization.

DEFINITION. Let G be an embedded planar graph and $\#$ an assignment of non-negative weights to the vertices, edges and faces of G which sums to 1. We say that a simple cycle C of G is a *weighted separator* if both the weight of the interior of C and the weight of the exterior $\leqslant \frac{2}{3}$.

As another example suppose we want a simple cycle separator for the graph consisting of a simple cycle. Since we must return with a simple cycle and there is but one cycle we are forced to pick this cycle. Thus, any estimate for the minimum size of a simple cycle separator must include some other parameter. Let the *size* of a face be the number of vertices on its boundary counting multiple visits when traversing the boundary. Further, let d be the maximum face size for an embedded graph G then infinitely often $c\sqrt{d \cdot n}$ edges will be required for a simple cycle separator for some constant c. One can see this fact by taking any triangulated planar graph which requires $\Omega(\sqrt{n})$ size vertex separators and replacing each edge by a path of length d. We next show that this bound is also achievable for some constant, c', the main theorem of the paper.

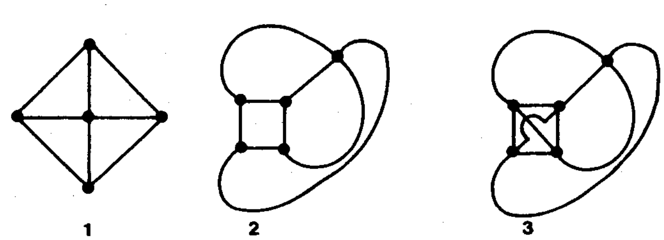

FIG. 1. An embedded graph, its geometric dual, and its face incidence graph.

THEOREM 1. *If G is an embedded 2-connected planar graph, $\#$ is an assignment of weights which sums to* 1, *and no face has weight* $> \frac{2}{3}$ *then there exists a simple cycle weighted separator of size* $2\sqrt{2\lfloor d/2 \rfloor n}$, *where d is the maximum face size. Further, this cycle is constructible in linear sequential time or polylogarithmic parallel time with a polynomial number of processors.*

Note that in the special case when the graph is triangulated, i.e., $d = 3$, we construct separators of size $\sqrt{8n}$ which agrees with Lipton–Tarjan result but is slightly larger than the separators of Djidjev whose size are $\sqrt{6n}$.[1] If we want a vertex separator which forms a simple incision for a graph which may not be triangulated we simply triangulate without adding new vertices and ask for a simple cycle separator of size at most $\sqrt{8n}$.

Theorem 1 is false if the hypothesis that G is 2-connected is dropped. A tree is a simple example of a planar graph with no simple cycle separator. We next observe a simple generalization of the previous theorem for the case when the face weights are zero but the 2-connected hypothesis is dropped.

THEOREM 2. *If G is an embedded planar graph and $\#$ is an assignment of weights to the edges and vertices which sums to* 1, *then either there exists a vertex which is a weighted separator or there exists a simple cycle separator of size at most* $2\sqrt{2\lfloor d/2 \rfloor n}$.

The theorem is only true if we are allowed to modify the embedding of G by rearranging the 2-connected components. See the example in Fig. 2. As in Theorem 1 the separator is constructible in linear time and polylog parallel time. We first show that Theorem 1 implies Theorem 2. Let G be an embedded graph as in the hypothesis of Theorem 2. We first construct a tree T from G, see [7], consisting of the cut vertices of G plus a new vertex for each 2-connected component of G. We connect a cut vertex to a component vertex if it is contained in the corresponding component. The weight of a cut vertex will be its weight in G. While a component vertex will have the weight of the component minus the weight of the cut vertices it contains.

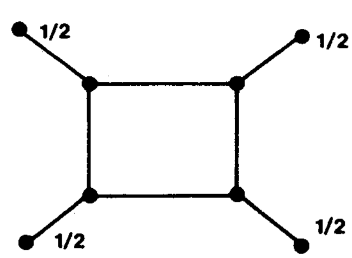

FIG. 2. A graph with no weighted simple cycle separator as embedded.

[1] This bound has recently been improved to $7/3\sqrt{n} \leq \sqrt{5.5n}$ [8].

We now find the weighted vertex separator x for T. If x is a cut vertex of G then x is a weighted separator of G. Thus we must only deal with the case when a vertex separaor x of T is a component vertex. In this case the weight of each subtree obtained from T by removing x has weight $\leqslant \frac{1}{3}$. Let $G_1,..., G_k$ be the subgraphs of G corresponding to these subtrees of T and let H be the 2-connected component of G corresponding to x. Further, let w_i be the weight of G_i minus the weight of the attachment vertex to H. For each G_i we add the weight w_i to the face of H which G_i is embedded. Except for the case when some face of H now has weight $>\frac{2}{3}$, we simply use Theorem 1 to find a simple cycle separator of H. This cycle will be a separator of G as required in Theorem 2. In the case when some face F of H has weight $>\frac{2}{3}$ we pick F as our simple cycle separator and pick a subset of G_i embedded in F whose combined weight is between $\frac{1}{3}$ and $\frac{2}{3}$ to be one of the pieces. Note that in either case the maximum face size of H is at most that of G.

In Section 5 we discuss in detail both sequential and parallel implementation of the simple cycle separator theorem. For sequential implementation we will use the random access machine model (RAM) as described in [1]. The parallel implementation will use a computation model consisting of a collection of synchronized processors (RAMs) with a common memory. The processors are allowed to access this memory using concurrent reads and writes in unit time, see [18]. In the parallel model we will also estimate the number of processors used since we feel this is an important parameter in the design of an efficient parallel algorithm. At the present time breath-first search on a graph can be implemented in $O(\log n)$ time but the number of processors is very large, $O(n^3)$ using classic matrix multiplication methods. All other steps will be performed in $O(\log n)$ time using $O(n)$ processors. Thus Theorem 1 can be implemented in $O(\log n)$ time using $O(n^3)$ processors when the planar embedding is also given. In the reduction of Theorem 2 to theorem 1 we needed to compute the 2-connected components. Hopcroft gives a linear time sequential algorithm [1] and Tarjan and Vishken give an $O(\log n)$ time $O(n)$ processor parallel algorithm [19] for computing the 2-connected components.[2] Thus the time and processor count for Theorem 1 also applies to Theorem 2.

2. Preliminaries

There are many formal definitions and many intuitive definitions of graphs "drawn" or embedded in or on the plane. Following Edmonds, Lehman, Tutte, and many others, we make the following formal definition; Let G be an undirected graph. We view each edge of G as two directed edges or *darts*. An embedding will simply be a description of the cyclic orderings of the darts radiating from each vertex. Formally, let $Sym(E)$ denote all permutations of the darts of G. An embedding is defined as follows:

[2] For randomized algorithms working on embedded graphs all steps except breath first search can be performed in $O(\log n)$ steps using only $O(n/\log n)$ processors [17].

DEFINITION. The permutation $\phi \in \mathrm{Sym}(E)$ is an embedding of G if:

1. $\mathrm{Tail}(e) = \mathrm{Tail}(\phi(e))$ for each dart $e \in E$.
2. ϕ restricted to the darts with tail at $v \in V$ is a cyclic permutation.

To specify the faces of this embedding consider the permutation R such that $R(e)$ is the reflection of the dart e. Now, successive application of ϕ will traverse the darts radiating from a vertex, in say, a clockwise order. On the other hand, the permutation $\phi^* = \phi \cdot R$ will traverse the darts forming the boundary of a face in counterclockwise order. We say that ϕ is a planar embedding if the number of faces f of the embedding satisfies Euler's formula:

$$f - e + v = 2$$

where e is the number of edges and v is the number of vertices of the graph.

We shall not distinguish between a face and its boundary of counterclockwise oriented darts. In general a region, the union of a collection of faces, will also be represented by its darts traversing the boundary in a counterclockwise order. The boundary may not be connected so in general the boundary will be a collection of traversals. We shall view all regions as closed. That is, a region contains the counterclockwise oriented darts appearing on its boundary. Thus, e and $R(e)$ cannot both belong to the boundary of the same region. In particular, if F and F' are the boundaries of two distinct faces then the darts on the boundary of their union will be equal to $F + F'$, where $e + R(e) = 0$. It will be understood that when adding boundaries, sets of darts, that $e + R(e) = 0$. By a path (cycle) C we shall mean the darts on the path (cycle). Since the indegree and the outdegree are equal at any vertex of $F + F'$, the set of darts $F + F'$ can be decomposed into a dart disjoint simple cycles.

In this model, the natural unique decomposition of $F + F'$ into cycles which corresponds to the counterclockwise traversal of the boundary of $F + F'$ can be defined as follows: If e is a dart in $F + F'$ with its head at vertex x then we leave x on the first dart after $R(e)$ given by the ordering $\phi(R(e))$, $\phi^2(R(e))$,.... On the other hand, any simple cycle C has a well-defined interior $\mathrm{int}(C)$ (the faces, vertices, and edges to the left of C) and a well-defined exterior $\mathrm{ext}(C)$ (the faces, vertices, and edges to the right of C). The $\mathrm{int}(C)$ and the $\mathrm{ext}(C)$ do not contain the darts of C or C's reverse, $R(C)$.

3. Breath-First Search for Planar Graphs

We need a breath-first search for planar graphs such that the frontier of the search has a nice form. We would like the frontier to be a simple cycle but this is not always possible. But we can describe a search such that the frontier is a collection of edge disjoint nonnesting simple cycles and the exterior of each simple cycle is a disjoint unsearched region of the graph.

We have given a very formal definition of an embedding of a graph in the plane. This definition or model is very useful for coding up the algorithms to follow but may make it hard to understand the intuition behind them. One normally defines an embedding of a planar graph as a drawing without crossing on the plane or a sphere. These drawings in general will have the property that distance on the plane or sphere will not correspond to distance in the graph. We have, at least for intuition, considered drawing graphs on 2-dimensional surfaces which are topologically equivalent to the sphere. Further, we have assumed that they have been drawn in such a way that distance on the surface is roughly proportional to distance in the graph. One can also think of surfaces which have been triangulated to a very good approximation to the surface. We have found that planar graphs which have been drawn in a "dense" distance preserving way on a "real" tree seems to best capture the spirit of the algorithms and exhibit the pitfalls in other approaches. Most of the terminology has been motivated by this class of embeddings.

Suppose we execute breath first search on the surface of a "real" tree start from a point. At first the frontier of the search will be a simple cycle. But in general the frontier will not remain simple and it will decompose into a collection of simple cycles which may share isolated vertices. Now, each simple cycle will correspond to an unsearched subtree. We mimic this search on an embedded planar graph.

Suppose G is an embedded planar graph with simple faces. We start our search from a face F. Thus, initially the frontier of the search is a simple cycle, the boundary of F. So, suppose that C is a simple cycle of G.

We next define a natural breath first search into the exterior of C. Let EF be the faces of G in the exterior of C which share a vertex or an edge with C, $R(e) \equiv e$. Consider the sum $C' = C + \sum F$ for $F \in EF$. We call C a *leaf* if C' is empty, i.e., all faces exterior to C are in EF. Otherwise, we show that C' can be written as an edge disjoint sum of simple cycles such that their exteriors are also disjoint. Let $\text{ext}(C')^*$ be the subgraph in the geometric dual of G induced by the faces in $\text{ext}(C')$. Further, let L be the faces in a connected component of $\text{ext}(C')^*$. Consider D the boundary of the union of the faces of $L, D = \sum F$ for $F \in L$. It follows that $R(D)$, the reflection of all the darts of D, is contained in C'. They cannot belong to faces in $\text{ext}(C')$ since L is connected component of $\text{ext}(C')^*$. We need only show that D is a simple cycle.

LEMMA 3. *The subgraph D as described above is a simple cycle.*

Proof. As noted in the preliminaries the boundary of a region consist of a collection of simple cycles. We will show that if the boundary of L contains more than one cycle then the underling surface that G is embedded on contains a "handle." By way of a contradiction suppose that D consists of 2 or more simple cycles. Further, let e and e' be two darts of D on distinct simple cycles, say C_1 *and* C_2. Since the regions defined by L and C' are connected there are two vertex disjoint paths, one in the interior of L and other in the interior of C' which only share a point on e and a point on e'. These two paths form a cycle T on the surface that crosses over C_1 in a fundamental way. Thus T and C_1 form an embedded graph of genus 1, the underling surface contains a "handle." This is a contradiction. Thus we may conclude that D is simple. ▮

Using Lemma 3 the unsearched region decomposes into a collection of connected regions each with a boundary consisting of a simple cycle. We shall call C' the *next level out* from C and each $R(D)$ a *branch* of C.

Note that this search is just regular breath first search on the face incidence graph. The *face incidence graph* consists of the faces of G as vertices and two faces are incident if they share an edge or a vertex as elements of G, see Fig. 1.

4. Finding a Subgraph of Small Diameter

The proof of Theorem 1 consists of two phases. In the first phase, outlined in this section, we find a subgraph H which has $O(\sqrt{d\cdot n})$ diameter and $O(\sqrt{d\cdot n})$ face size. While in the second phase, we will find a separator contained in H. The planar embedding of H will be the one induced by the embedding of G, the original graph. The weight on a vertex or an edge of H will equal the weight assigned in G. A face F of H will have weight equal to the sum of the weights of faces, vertices, and edges of G which are embedded in F, i.e., $\#\,(\mathrm{int}(F))$ in G. This weight will be called the *induced* weight on F. A face is a *leaf* with respect to a spanning tree if it contains only one nontree edge on its boundary. We give the main theorem of this section.

THEOREM 4. *If G is a 2-connected embedded planar graph with weights which sums to* 1, *no face weight* $>\frac{2}{3}$ *and the maximum face size is d, then there exists a 2-connected subgraph H with spanning tree T satisfying*:

1. *The diameter dia of T plus the maximum size h of any non-leaf face of H is at most* $2\sqrt{2\lfloor d/2\rfloor n}$, *i.e.*, $\mathrm{dia}+h\leqslant 2\sqrt{2\lfloor d\backslash 2\rfloor n}$.
2. *The maximum induced weight on any face of H is* $\leqslant\frac{2}{3}$.

Proof. Note that G is 2-connected if and only if every face of G is simple. Let G satisfy the hypothesis of the theorem and F be some face of G. Further, let $\#$ be an assignment of weights also satisfying the hypothesis.

We start by constructing a breath-first search of the levels from F as defined in Section 3. Namely, we construct the next level out from F and decompose it into branches. For each branch we again construct its branches. This gives us a tree of branches with root F. Note that by starting from the leaves of this tree we can compute the induced weight on the interior and exterior of each branch in linear time. We discuss the parallel implementation in Section 6. We next pick a path of branches $F=C_1,\dots,C_k$ called a *trunk* by starting at F and picking at each stage a branch on the next level whose exterior is of maximum size. That is, if C_i is on the trunk and $B_1,\dots,B_t$ are the branches of C_i then we pick B_j such that $\#\,(\mathrm{ext}(B_j))\geqslant\#\,(\mathrm{ext}(B_k))$, for $1\leqslant k\leqslant t$.

Let C be the first branch on the trunk such that $\#(\text{int}(C)) \geq \frac{1}{3}$, the interior of C is the side containing F.

Of the branches of G consider those branches which are ancestors of C, including C. Let n_1 be the number of vertices lying on these branches. Further, let $d' = \lfloor d/2 \rfloor$. We next exhibit a small size ancestor of C. The *size* of a cycle B is the number of darts it contains which we denote by $|B|$.

LEMMA 5. *There exists an integer $\alpha_1 \geq 0$ such that the α_1th ancestor of C, say B, satisfies $2d'\alpha_1 + |B| \leq 2\sqrt{d' \cdot n_1}$.*

Proof. We show that B must exist by contradiction. If no such B exists then the ith ancestor of C must have size $> b_i = 2\sqrt{d' \cdot n_1} - 2d' \cdot i$, for $0 \leq i \leq \sqrt{n_1/d'}$. Since the B_i's are vertex disjoint, the sum of the b_i's must be $\leq n_1$. That is, the sum

$$\sum_{i=0}^{\lfloor\sqrt{n_1/d'}\rfloor} \lceil 2\sqrt{d' \cdot n_1}\rceil - 2d' \cdot i$$

must be $\leq n_1$. The sum can be rewritten as follows:

$$\begin{aligned}
&(\lfloor\sqrt{n_1/d'}\rfloor + 1)\lceil 2\sqrt{d' \cdot n_1}\rceil - 2d' \sum_{i=1}^{\lfloor\sqrt{n_1/d'}\rfloor} i \\
&\quad = (\lfloor\sqrt{n_1/d'}\rfloor + 1)(\lceil 2\sqrt{d' \cdot n_1}\rceil - d'\lfloor\sqrt{n_1/d'}\rfloor) \\
&\quad > (\sqrt{n_1/d'})(2\sqrt{d' \cdot n_1} - d'\sqrt{n_1/d'}) = n_1.
\end{aligned}$$

Thus the B_i's contain more than n_1 vertices, a contradiction. ∎

Let $C = B_0, \dots, B_{\alpha_1} = B$ be the ancestors of C up to B. Consider the subgraph H' obtained from G by (1) deleting the exterior of any branch of B_1 thru B_{α_1} which is distinct from $B_1, \dots, B_{\alpha_1 - 1}$ and (2) deleting the interior of B. Note that we have deleted the exterior of C. Since the exterior of the branches deleted are not on the trunk their weight must be $\leq \frac{1}{2}$. Thus this subgraph H' already satisfies the second condition of the theorem. We next enlarge the graph so that it also satisfies the first condition. Figure 3 gives a simple possible topological surface corresponding to H' in the case when $\alpha_1 = 4$. Here, B_1 has two branches C and A_1, both have been removed. B_2 has three branches B_1, A_2, and A_3. Again, A_2 and A_3 have been removed. The subgraph H' has induced face weights $\leq \frac{2}{3}$. As we shall see, it also has small diameter. But, the face sizes may be too large.

Simultaneously for each face of H' except B construct the next level out until the maximum number of levels constructed α_2 and the maximum branch size f satisfies $d' \cdot \alpha_2 + f \leq 2\sqrt{d' \cdot n_2}$, where n_2 is the number of vertices of G interior to some face of H except B. By similar arguments as used above this procedure will terminate. The subgraph H will be G minus the exteriors of these branches. We call the portion of G which is added onto the face of H' a *cap*. This H' plus its caps has small face size. We next construct the spanning tree T.

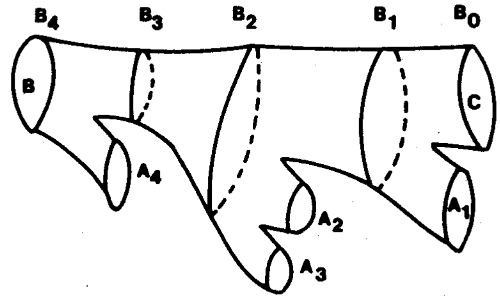

FIG. 3. A topological picture of a possible subgraph H'.

The spanning tree T will consist of all the edges in B minus one edge plus the edges from a breath first search spanning tree of H starting from B. Notice, we have chosen T such that B is a leaf. The distance between levels is not too large since if D is a simple cycle and x is a vertex on the next level out from D then x can be at most a distance d' from D since they must share a face of size $\leqslant d$. Thus, a breath first search from any point on B in H' will generate paths of length at most $(d' \cdot \alpha_1 + |B|/2)$. By similar argument, any point in a cap is at most a distance of $d' \cdot \alpha_2$ away from H'. Thus, H has a spanning tree of diameter $2d'(\alpha_1 + \alpha_2) + |B|$. We need only show that T and H satisfy Condition 1) of Theorem 4. Since the maximum non-leaf face size is f, we need only estimate $2d'\alpha_1 + |B| + 2d'\alpha_2 + f$ which is bounded by $2\sqrt{d' \cdot n_1} + 2\sqrt{d' \cdot n_2} = 2\sqrt{d'}(\sqrt{n_1} + \sqrt{n_2})$. Since n_1 and n_2 are the sizes of disjoint subsets of vertices in G the sum $\sqrt{n_1} + \sqrt{n_2}$ is bounded by $\sqrt{2}\sqrt{n}$. Thus, the diameter of T plus the maximum face size (except B) is at most $2\sqrt{2 \cdot d' \cdot n}$. ∎

5. Finding the Separator in a Graph of Small Diameter

By the last section we can find a subgraph of radius $O(\sqrt{d \cdot n})$. Here we find a small simple cycle which is a separator. The main theorem of this section is:

THEOREM 6. *If G is a 2-connected weighted and embedded planar graph with no face weight $> \frac{2}{3}$ and T is a spanning tree of G then there exists a weight separator of size at most the diameter of T plus the maximum non-leaf face size.*

Proof. The proof will consist of a sequence of successive simple cycle approximations that will converge to a cycle that is a weighted separator. We say a dart e is a *nontree dart* if the edge containing e is not a tree edge. For each non-tree dart e let C_e be the induced simple cycle from the spanning tree T.

If C_e is not a weighted separator then, without loss of generality, we may assume that the weight of the interior of $C_e > \frac{2}{3}$. Let F be the face in G containing e, F will be on the interior of C_e. Further, let $e_1, \ldots, e_k$ be the non-tree darts on F distinct from e as they appear on F starting at e. Note that $k \geqslant 1$. For, if $k = 0$ then F would be the interior of C_e since F is simple. This contradicts the facts: $\frac{2}{3} > \#(F) = \#(\text{int}(C_e)) > \frac{2}{3}$. Thus, F is a non-leaf face. We now partition $\text{int}(C_e)$.

For $i = 1$ to k let C_i be the cycle induced by $R(e_i)$, i.e., $\text{int}(C_i)$ is contained in $\text{int}(C_e)$. Thus the regions $\text{int}(C_1), \ldots, \text{int}(C_k)$, $\text{int}(F)$ form a partition of $\text{int}(C_e)$ up to vertices and edges on their boundaries. We first reduce the problem to the case when

$$\#(\text{ext}(F)),\ \#(\text{ext}(C_1)), \ldots, \#(\text{ext}(C_k)) > \tfrac{2}{3}: \tag{1}$$

1. If $\#(\text{int}(C_i)) > \frac{2}{3}$ for some $1 \leqslant i \leqslant k$ then set e to $R(e_i)$. Using this new e recompute e_1 thru e_k and repeat.

2. If $\#(\text{ext}(F)) \leqslant \frac{2}{3}$ then F is a weighted simple cycle separator of size $\leqslant d$. Return F and quit.

3. If $\#(\text{ext}(C_i)) \leqslant \frac{2}{3}$ for some $k < i < k$ then C_i is a weighted simple cycle separator of size $\leqslant$(diameter of T) + 1. Return C_i and quit.

After step 3 if we have not found a simple cycle separator satisfying the theorem then $F, C_1, \ldots, C_k$ must satisfy (1). For the rest of the proof we shall assume condition (1) and we will use it to construct a separator whose interior will be a region containing F plus some of the C_i's. As we combine F with the interiors of some of the C_i's we must insure that the boundary is simple. For this end, we introduce a partial order on the C_i's.

Let x and y be the end points of the edge e. Since F is a simple cycle, if we remove e from F we obtain a simple path from x to y on F. Recall that the boundary of F is the darts and vertices in counterclockwise order. Let $x = x_1, \ldots, x_t = y$ be the vertices on the path in the order they appear. Given any cycle C_i it will contain a vertex α_i of minimum index and a vertex r_i of maximum index in $\{x_1, \ldots, x_t\}$. We shall call these vertices the *left-most* and *right-most* vertices of C_i, respectively. Thus, associated with C_i is an interval or path of darts on F from the vertex of index α_i to the vertex of index r_i. We say C_i *dominates* C_j if the interval for C_i contains the interval for C_j. We say C_i is *left* of C_j if the darts on the interval of C_i come before those of C_j. Since the embedding is planar these intervals must nest, i.e., either one contains the other or they overlay by at most a single vertex. The C's can be viewed as vertices of a graph by adding a directed edge from C_i to C_j if C_i dominates C_j and there is no k such that C_j dominates C_k and C_k dominates C_j. In fact, we get a forest. In a natural way we let C_e be a common root. Thus dominates relation plus the left and right relation gives us a rooted ordered tree on $C_e = C_0$, $C_1, \ldots, C_k$.

We associate with each region $\text{int}(C_i)$ the union of all regions dominated by it, i.e., $\bar{C}_i = \sum \{C_j \mid C_i \text{ dominates } C_j \text{ for } i \neq 0\}$ and $\bar{C}_0 = (\sum_{i=1}^{k} C_i)$. Similar to the fact that trees have a separator consisting of a single vertex we get the following lemma:

LEMMA 7. *Either* (a) *there exists an $i \neq 0$ such that $F + \bar{C}_i$ is a weighted separator or* (b) *there exists an i such that $\#(\text{int}(F + \bar{C}_i)) > \frac{2}{3}$ and for all j, such that $\bar{C}_j$ is a child of $\bar{C}_i$, $\#(\text{ext}(F + \bar{C}_j)) > \frac{2}{3}$.*

First, suppose that condition (a) in Lemma 7 is true. Clearly, $F+\bar{C}_i$ is a weighted separator. We must show that $F+\bar{C}_i$ is a simple cycle and determine its weights. Note that $\bar{C}_i$ forms a simple cycle which intersects F on some interval of F. Thus, $F+\bar{C}_i$ will consist of an interval of F plus an interval of $\bar{C}_i$ which are disjoint except at their end points. Since this interval of $\bar{C}_i$ is contained as a simple path in T the size of $F+\bar{C}$ is at most diameter of T plus maximum over non-leaf face sizes. Thus it is a small simple cycle weighted separator. For the remainder of the proof of Theorem 5 we may assume that $\bar{C}_i$ satisfies condition (b) of Lemma 7.

Let $L_1,\ldots, L_s$ be the children of C_i which are left of C_i ordered from right to left. Similarly, let $R_1,\ldots, R_t$ be the children of C_i which are right of C_i ordered from left to right. We shall successively add $\bar{L}_j$ to $F+\sum_{k=1}^{j-1}\bar{L}_k$ while $\#\text{ext}(F+\sum_{k=1}^{j-1}\bar{L}_k)>\frac{2}{3}$. If the weight of the left children is not large enough to form a separator we will successively add the right children to the sum. Suppose that A is the union of some of the regions and we want to add region B. Since $\text{int}(A+B)$ will also contain the boundary between A and B both $\#(\text{int}(A))$ and $\#(\text{int}(B))$ may be $<\frac{2}{3}$ while $\#(\text{int}(A+B))>\frac{2}{3}$. By using the fact that $\#(\text{ext}(A))>\frac{2}{3}$ we show that $\#(\text{int}(A+B))<\frac{2}{3}$.

LEMMA 8. *If G is an embedded graph, A and B are two face disjoint regions such that $\#(\text{ext}(A))>\frac{2}{3}$, and $\#(\text{int}(B))<\frac{1}{3}$ then $\#(\text{int}(A+B))<\frac{2}{3}$.*

Proof. Let A and B satisfy the hypothesis. The exterior of A decomposes into the exterior of $A+B$, the interior of B, and that part of the boundary of B not in the boundary of A. Setting b equal to the last term above we can express this as $\#(\text{ext}(A))=\#(\text{ext}(A+B))+\#(\text{int}(B)+\#(b)$. By the hypothesis $\#(\text{ext}(A))>\frac{2}{3}$ and $\#(\text{int}(B))<\frac{1}{3}$. Therefore, $\#(\text{ext}(A+B))+\#(b)>\frac{1}{3}$. But both $\text{ext}(A+B)$ and b are disjoint and not in the $\text{int}(A+B)$. Thus, $\#(\text{int}(A+B))<\frac{2}{3}$. ∎

Using the last lemma we can simply pick $D_1,\ldots, D_j$ for some j such that $F'=F+D_1+\cdots+D_j$ is a separator. We must show that F' is simple and of small size. We state without proof the following simple lemma:

LEMMA 9. *If $D_1,\ldots, D_j$ are consecutive and all left (right) of e_i then $F+D_1+,\ldots,+D_j$ is simple and consists of an interval from F plus a simple path in T, the spanning tree.*

Thus, the new region will consist of F plus consecutive elements from the left of e_i and consecutive elements from the right of e_i. Its boundary will consist of two paths from the tree plus 2 paths from F. Thus, the size of this region is at most twice the diameter of T plus the maximum over the non-leaf face sizes, $2\,\text{dia}+S$. Actually these two paths in the tree can be joined to form one simple path in T. This gives the desired result that the size $\leqslant \text{dia}+S$.

6. Sequential and Parallel Implementation of the Separator Theorem

So far we have only implicitly given an algorithm to find a simply cycle separator for a planar graph. This algorithm naturally decomposes into 9 steps. We first enumerate these steps and then discuss both sequential and parallel implementations of each of these steps:

A List of Computations Used to Find a Separator

1. Pick a face F and compute the branches derived from F.
2. Determine the weight of each branch.
3. Find the subgraph H and spanning tree T satisfying Theorem 4.
4. Determine the weight on the interior and exterir of each induced cycle of T.
5. Check if any face or induced cycle is a weighted simple cycle separator. If not there must exist an edge e and a face F satisfying condition (1). Find it.
6. Compute the dominance tree associated with F and C and compute the weight of each $\bar{C}_i$.
7. Find a $\bar{C}_i$ satisfying either condition (a) or (b) of Lemma 6. If condition (a) holds then we are done.
8. If condition (b) of Lemma 6 holds then compute the accumulative weights of the left and right children of $\bar{C}_i$.
9. Pick an appropriate sum of left and right children of C_i. Which forms the separator.

It is assumed that the graph has all ready been embedded on the sphere or plane. If not there are several linear time sequential algorithms which find the embedding [10, 2], as well as, polylog parallel time parallel algorithms [11, 17]. In either case we may assume the graph is embedded where the embedding is presented by a permutation ϕ of the darts such that ϕ cyclically permutes the darts radiating from each vertex. The permutation may be stored as a array by first assigning a number to each dart, viewing ϕ as a permutation of these numbers, and storing $\phi(i)$ in location i. The permutation $\phi^* = \phi R$ which traverses the boundary of the faces in counterclockwise order can be computed by evaluating the product in linear sequential time or constant parallel time.

We start by analyzing step 1 which seems to be the most costly step of the algorithm especially in the parallel case. In the sequential case we implement a breath-first search from the face F as described in the preliminaries. It is not hard to see that this can be implemented only maintaining three permutations ϕ, R, and B, where initially B is F the boundary of the face F. Eventually either $B(e) = e$ if e is not in some successive branch of F or $B(e)$ is the next dart in the branch containing e.

To implement step 1 in parallel we construct the face incidence graph storing it as an incidences matrix. Using an $O(\log n)$ parallel time matrix multiplication algorithm where the two scalar operations are sum and minimum, we compute the distance of each face from the face F in $O(\log^2 n)$ time. A dart belonging to some branch of F will have a face of distance i on the left side and a face of distance $i+1$ on the right. To determine the image of B applied to a dart e with head x which belongs to some branch of F we must determine the first dart leaving x scanning clockwise from e at x, see Fig. 4. This can be done by standard doubling-up tricks.

To implement step 2 we use the fact that the branches of F form a tree rooted at F (Lemma 3). Thus, we compute the weight of the exterior of each branch starting at the leaves. The sequential case is straight forward. We discuss in more detail the parallel case.

We first present an $O(\log n)$ time $O(n^2)$ processor algorithm and then show how to modify it to reduce the processor count to $O(n)$. Let D be a simple cycle and $G^* - D^*$ be the geometric dual of G minus those edges dual to D. We saw in Lemma 3 that the faces in the exterior of D form a connected component in $G^* - D^*$. We can find this component in $O(\log n)$ time using $O(n)$ processors [18]. Thus in $O(\log n)$ time and $O(n)$ processors we can compute $\#\mathrm{ext}(D)$. To compute the weight of the exterior of n simple cycles will add a factor of n to the number of processors. Since the cycles in our case form a tree we show how to reduce the problem to the problem of evaluating all subexpressions of an arithmetic expression.

The weight $\#(\mathrm{ext}(D))$ of some branch D with children $D_i, \ldots, D_k$ will equal: $\sum_{i=1}^{k} \#(\mathrm{ext}(D_i)) + \sum_{i=1}^{k} \#(D_i) +$ (the weight of the faces, edges and vertices between D and $D_i, \ldots, D_k$). To compute the last term τ in the sum suppose that D is between faces at level $j-1$ and j. Note that the faces in the sum τ are those faces at level j which share a vertex or an edge with D. They form a connected component in the geometric dual. While the edges and vertices in the sum are those elements which are only common to the faces in this component. Notice that the τ terms are formed from disjoint sets of vertices, edges, and faces. By assigning a processor to each vertex, edge, and face one can in $O(\log n)$ time determine to which branch they belong using a connected components algorithm [18]. In $O(\log n)$ parallel

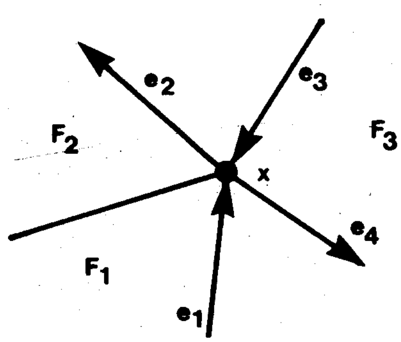

FIG. 4. The boundary of the region given by faces F_1, F_2, and F_3 will send e_1 to e_2 and e_3 to e_4.

additions we can compute the last two terms in the sum. If W_D is the last two terms in the sum then $\#(\text{ext}(D)) = W_D + \sum_{i=1}^{k} \#(\text{ext}(D_i))$. We view the evaluation of $\#(\text{ext}(D))$ as simply the evaluation of an expression of size n. Brent [3] has shown that expression can be evaluated in $O(\log n)$ time in parallel for a fixed expression. Here we need to dynamically evaluate expressions in polylog parallel time. A simple $O(\log n)$ time P-RAM algorithm using $O(n)$ processors is given in [17].

Step 3 is fairly straight forward in both the sequential and parallel case.

Step 4 is very similar to step 2. By the following lemma we view computing the weights on all induced cycles as an expression evaluation problem.

LEMMA 10. *If G is a 2-connected embedded planar graph, G^* its geometric dual, T a spanning tree of G, and H the subgraph of G^* consisting of those edge in G^* whose dual is not in T then H is a tree.*

Thus step 4 for face weights can be viewed as determining the induced weights on the edges of H. As in step 2, we start at the leaves. Thus, we can compute the weight of the faces interior to each induced cycle. We must also determine the contribution from the vertices and the edges. It will suffice to determine the smallest induced cycle containing a given edge. It is not hard to see this problem reduces to the least common accessors problem for rooted trees. This last problem can be preformed in $O(\log n)$ time using $O(n/\log n)$ processors for planar graphs [17].

Steps 5–9 are fairly straightforward. Note that step 6 uses ideas similar to steps 2 and 4.

7. CONCLUSIONS

In this paper we have concentrated on worst case separators. That is, we give an algorithm which finds a relatively small separator when the smallest separator is relative large. It is open whether there is a polynomial time algorithm which finds the optimal separator for planar graphs. It is easy to show that there is always an optimal separator which consists of non-nesting simple cycles if the graph is triangulated. We say a simple cycle C is a *separator of ratio k* if $\text{size}(C)/\min\{\#(\text{int}(C)), \#(\text{ext}(C))\} = k$.

Questions. Is finding an optimal ratio separator for planar graphs polynomial time computable?

It seems that most divide-and-conquer algorithms only need separators with small ratios. Thus optimal ratio separators may be a more natural or applicable then "$\frac{1}{3}$–$\frac{2}{3}$" separators.

ACKNOWLEDGMENTS

I thank Richard Anderson, Joan Hutchinson, Tom Leighton, and Vijaya Ramachandran for most helpful comments and discussions.

REFERENCES

1. A. AHO, J. HOPCROFT, AND J. ULLMAN, "*The Design and Analysis of Computer Algorithms*," Addison–Wesley, Reading, Mass., 1974.
2. K. BOOTH AND G. LUEKER, Testing for the consecutive ones property, interval graphs, and graph planarity using *PQ*-tree algorithms, *J. Comput. System. Sci.* **13** (1976), 335–379.
3. R. P. BRENT, The parallel evaluation of general arithmetic expressions, *J.* **21**, No. 2 *Assoc. Comput. Mach.* (1974), 201–208.
4. H. N. DJIDJEV, A separator theorem, *C. R. Acad. Bulgare Sci.* **34**, No. 5 (1981), 643–645.
5. H. N. DJIDJEV, On the problem of partitioning planar graphs, *SIAM J. Alg. Discrete meth.* **3**, No. 2 (1982), 229–240.
6. H. N. DJIDJEV, A linear algorithm for partitioning graphs, *C. R. Acad. Bulgare Sci.* **35**, No. 8 (1982), 1053–1056.
7. S. EVEN, "*Graph Algorithms*," Comput. Sci. Press, Potomac, Md., 1979.
8. H. GAZIT, An improved algorithm for separating a planar graph, manuscript.
9. J. R. GILBERT, J. P. HUTCHINSON, AND R. E. TARJAN, "*A Separation Theorem for Graphs of Bounded Genus*," Technical Report TR82-506, Cornell University, July 1982.
10. J. HOPCROFT AND R. TARJAN, Efficient planarity testing, *J. Assoc. Comput. Mach.* **21**, No. 4 (1974), 549–568.
11. J. JA'JA', AND J. SIMON, Parallel algorithms in graph theory: Planarity testing, *SIAM J. Comput.* **11**, No. 2 (1982), 314–328.
12. D. B. JOHNSON AND S. M. VENKATESAN, Partition of planar flow networks, *in* "24th Annu. Sympos. Found. of Comput. Sci.," pp. 259–263, IEEE, New York, 1983.
13. C. LEISERSON, Area-efficient graph layouts (for VLSI), *in* "21st Annu. Sympos Found. of Comput. Sci.," pp. 270–281, IEEE, New York, 1980.
14. A. LEMPLE, S. EVEN, AND I. CEDERBAUM, An algorithm for planarity testing of graphs, *in* "Theory of Graphs, International Symposium, Rome, July 1966," pp. 215–232, Gordon & Breach, New York, 1967.
15. R. J. LIPTON, D. J. ROSE, AND R. E. TARJAN, Generalized nested dissection, *SIAM J. Numer. Anal.* **16** (1979), 346–358.
16. R. J. LIPTON AND R. E. TARJAN, A separator theorem for planar graphs, *SIAM J. Appl. Math.* **36**, (1979), 177–189.
17. G. L. MILLER AND J. H. REIF, Parallel tree contraction and its applications, *in* "26th Symposium on Foundations of Computer Science," pp. 478–489, IEEE, Portland, Ore., 1985.
18. Y. SHILOACH AND U. VISKIN, An $O(\log n)$ parallel connectivity algorithm, *J. Algorithms* **3** (1982), 57–67.
19. R. E. TARJAN AND U. VISHKIN, Finding biconnected components and computing tree functions in logarithmic parallel time, *in* "25th Annu. Sympos. Found. of Comput. Sci.," pp. 12–22, IEEE, New York, 1984.
20. L. G. VALIANT, Universality consideration in VLSI circuits, *IEEE Trans. Comput.* **30**, No. 2 (1981), 135–140.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium