Planar Separators and the Euclidean Norm

Hillel Gazit[*]
Department of Computer Science
Duke University

Gary L. Miller[†]
School of Computer Science
Carnegie Mellon University &
Dept of Computer Science
University of Southern California

Abstract

In this paper we show that every 2-connected embedded planar graph with faces of sizes $d_1, \ldots, d_f$ has a simple cycle separator of size $1.58\sqrt{d_1^2 + \cdots + d_f^2}$ and we give an almost linear time algorithm for finding these separators, $O(n\alpha(n,n))$. We show that the new upper bound expressed as a function of $|G| = \sqrt{d_1^2 + \cdots + d_f^2}$ is no larger, up to a constant factor than previous bounds that where expressed in terms of $\sqrt{d \cdot v}$ where d is the maximum face size and v is the number of vertices and is much smaller for many graphs. The algorithms developed are simpler than earlier algorithms in that they work directly with the planar graph and its dual. They need not construct or work with the face-incidence graph as in [Mil86, GM87, GM].

1 Introduction

Planar graphs have played an important role in both sequential as well as parallel algorithm design. They arise in may areas of computation including: numerical analysis, animation, and VLSI. One of the important properties possessed by planar graphs, but not true for general graphs, is that they have small separators. Historically, a separator, in a graph $G = (V, E)$, is a subset $C \subset V$ such that (1) the remaining vertices can be partitioned into two sets: A and B, (2) $|A|, |B| \leq 2/3|V|$, and (3) there are no edges between vertices in A and vertices in B. Lipton and Tarjan were the first to show that paper graphs have $O(\sqrt{v})$ separators, [LT79]. Improvements in the constant have been made by Djidjev and the first author, [Dji81, Gaz86]. There are two important additional properties we require of separators. First, we shall assign a weighting function # to the vertices, edges, and faces of the embedded planar graph and require that the separator will "separate" the weighted graph. Second, we shall require that the separator be a cycle or collection of cycles. Intuitively, this means that we separate the planar graph "drawn" on the plane by cutting along the edges of the graph. Since all the separators we construct in this paper are in fact simple cycles, we will restrict our attention to the simple cycle case. These two restrictions have been addressed by the second author in [Mil86]. We will assume that the reader has some knowledge of this paper. The following definition is from [Mil86].

[*]This work was supported in part by grant number N00014-88-K-0623
[†]This work was supported in part by National Science Foundation grant DCR-8713489.

Definition 1.1 *Let G be an embedded planar graph and $\#$ an assignment of nonnegative 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-simple-cycle separator** *if both the weight of the interior of C and the weight of the exterior is $\leq 2/3$.*

Let G be an embedded planar graph. The size of a face of G is the number of edges, equivalently the number of faces, on its boundary counting multiplicity. The **Euclid norm** of G equals

$$|G|_2 = \sqrt{d_1^2 + \ldots + d_f^2} \quad \text{where } d_i \text{ is the size of the } i\text{th face of } G.$$

We can now state the main theorem of this paper.

Theorem 1.2 *A planar embedded graph G has a weighted-simple-cycle separator of size $\leq 1.58|G|_2$. The separator is computable in $O(n\alpha(n,n))$ sequential time where $\alpha(n,n)$ is the inverse of Ackerman's function.*

We first observe, up to a constant factor that Theorem 1.2 is stronger than the best previous bound of $2\sqrt{d \cdot v}$ where d is the maximum face size and v is the number of vertices, [Mil86].

Lemma 1.3 *The Euclidean norm $|G|_2 < \sqrt{6d \cdot v}$ if G has no faces of size ≤ 2.*

Proof: It will suffice to show $(|G|_2)^2 < 6d \cdot v$. We first observe that $\sum_{i=1}^{f} d_i < 6v$. By Euler's formula $e - f < v$. Substituting in the facts that $\sum_{i=1}^{f} d_i = 2e$ and $\sum_{i=1}^{f} d_i \geq 3f$ into the Euler equality from above we get:

$$\sum_{i=1}^{f} d_i = 3\sum_{i=1}^{f} d_i - 2\sum_{i=1}^{f} d_i \leq 6e - 6f < 6v.$$

To finish the proof consider the following equalities:

$$(|G|_2)^2 = \sum_{i=1}^{f} d_i^2 \leq d\sum_{i=1}^{f} d_i < 6d \cdot v.$$

□

2 A New Way to Level a Planar Graph

In our previous papers we performed a BFS numbering in the face-incidence graph. We performed the numbering in this graph so that frontier of the search at any stage will always be a collection of cycles. In the previous papers we assumed that all the faces were basically the same size. Therefore, we added to the new level all the faces adjacent to the frontier at the same time. Here we do not assume that all the faces are the same size. Therefore, we add faces adjacent to the frontier in a more judicious manner. The main new idea in this section is to perform the BFS numbering in G itself, and to introduce artificial edges called level-edges, to be defined later, such that the frontier is again a set of simple cycles. We can not use these edges in the final cycle separator, but they will be important for bookkeeping.

A **BFS numbering** of G from a subset of vertices $S \subset V$ is an assignment of a number to each vertex of G equal to its distance to the nearest vertex in S.

Let L be BFS numbering of G from a single source s. Let v be some vertex at level $i > 0$. The neighboring vertices of v will be at levels $i-1$, i or $i+1$. We introduce two level-edges at v for each consecutive, with respect to the cyclic ordering derived from the embedding of G in the plane, block of level $i+1$ vertices adjacent to v. If some edge e common to v is also common to another vertex at level i, then mark e level i. Otherwise, let F be the face common to v such that one edge at v is common to a level $i+1$ vertex, and the other edge is common to one with level number $i-1$. In this case, a level i edge will be added to the face F and the other attachment of this level-edge is obtained by following the boundary of F, using vertices of BFS number $< i$, around until a vertex of level i is found, see figure 1. Weight on the level-edge (x, y) in face F is the distance between x and y by following the boundary of F in the shorter of the two possible ways.

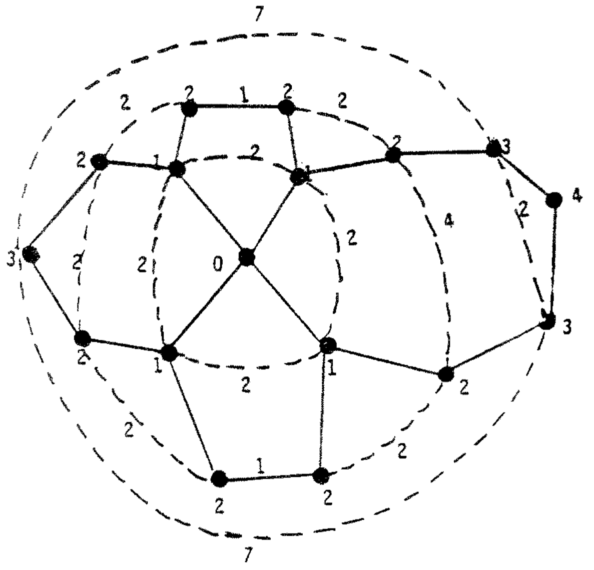

Figure 1: A BFS Number and Its Level-Edges with Weights. The solid edges are the graph edges and the dashed are the level-edges.

In prior separator papers the size of the separator was determined partly in terms of the number of vertices. In this paper it will be solely in terms of the sum of the weights on the level-edges. Let $W(G)$ equal the sum of the weights on the level-edges with respect to some BFS of G. We next show that $W(G)$ is bounded by $(|G|_2)^2$.

Lemma 2.1 *The sum of the level weights* $W(G) \leq \frac{1}{8}(|G|_2)^2$.

Proof: We show the inequality in the lemma holds for each face, and thus for the sum of the faces as a whole. Consider the following combinatorial problem: Let C be a simple cycle of size n drawn in the plane and A a set of noncrossing, vertex disjoint chords of C, i.e., C plus A forms an outer planar graph of degree at most 3. Let the weight of each chord (x, y) in A by equal to the distance between x and y in C. We will show that the sum of the weights of the chords $W(A) \leq n^2/8$. It should be clear that if we prove the last inequality then the lemma follows.

Suppose that $W(A)$ is a maximum over all such sets of chords. We first observe that no interior face of the corresponding outerplanar graph contains more than two chords or two cycle edges. If the chords of A form faces with more chords or cycle edges, we can rearrange the chords of A to strictly increase the value of $W(A)$.

Except for the middle chord, if it exists, each chord has a unique shortest path in C. Thus, we can partition the chords into two sets, two chords belong to the same set if their shortest paths intersect. Thinking of the chords as drawn vertically, we therefore partition the chords in the the left, middle, and right chords. There are several cases depending on whether the left, middle, and right chords have even or odd length and whether or not the middle chord exists. We will only handle the case when the chords all have even length and the middle chord exists. The other cases are similar. Let $2k+2$ be the length of the middle chord. Thus $n = 2(2k+2) = 4(k+1)$. In this case weight of the left chord equals those from the right and therefore we get the following equalities:

$$W(A) = (2k+2) + 2\sum_{i=1}^{k} 2i = (2k+2) + 2(k+1)k = 2(k+1)^2 = n^2/8.$$

□

As in [Mil86] the level edges are formally directed edges or arcs. We decompose the level-edges into a collection of simple cycles called **level-cycles**. Since the edges in the level-cycle need not be edges in G, we will "pull" the level-edges back to paths in G. The **pull-back** of a level-cycle C is defined as follows: Let $e = (x, y)$ be level-edge in C which belongs to a face F. The vertices x and y decompose the boundary of F into two paths P_1 and P_2 from x to y. We replace e by the shorter of P_1 or P_2. If they are equal, we pick the one consisting of smaller BFS numbers, see Figure 2. Observe that the number of edges on the pull-back of level-cycle is at most the weight of the level-cycle. Thus the length of a level-cycle is greater than or equal to its pull-back.

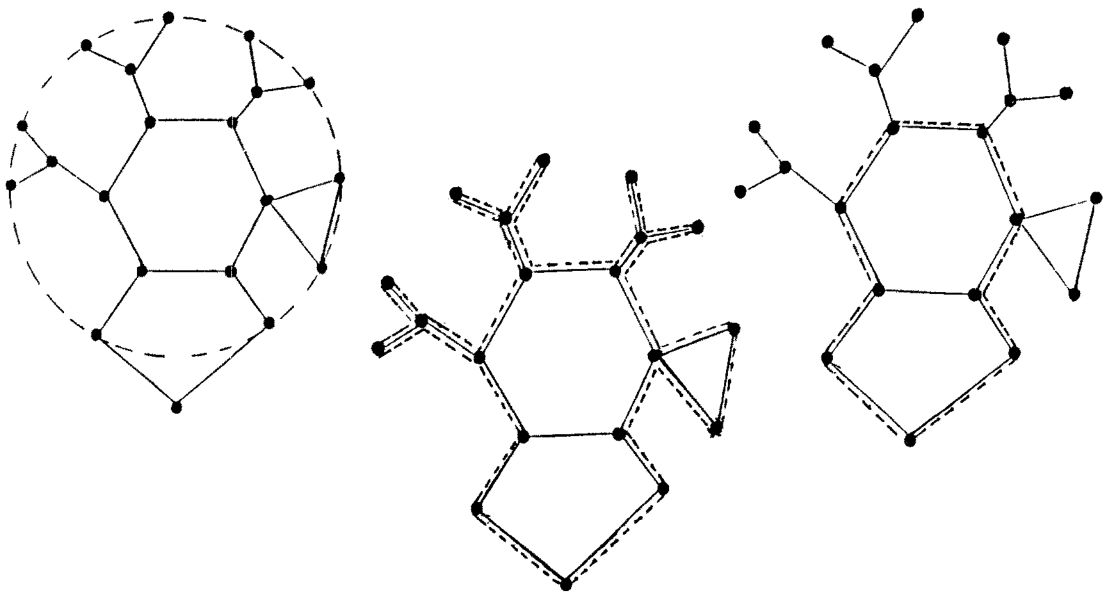

Figure 2: A Level-Cycle, Its Pull-back, and Its Retrack. The first is a graph and a level-cycle, the second is its pull-back, and the third is its retrack-cycle.

In the remainder of the paper we will assume the BFS is performed from the boundary vertices of some face F_0 of G. In this case we get the notion of a retract. The **retract** of a level-cycle C is the subgraph P' contained in the pull-back P of C defined as follows: Let H be all the faces in G which are reachable from F_0 in G^* without using edges that cross edges in P. The boundary, see [Mil86], of H is the retract of C. In the special case when H contains all the faces of G we return some fixed vertex in P such as the vertex with largest level number. If there is more than one we pick the most centered one (if there are two just pick one). In general P' is not a simple cycle but we can decomposed it into simple cycles, as we did to get the level-cycles.

The retrack-cycles form tree structure as did their level-cycles. In section 5 we show how to determine this structure, the length of each cycle, and the weight on the exterior and interior of each cycle. We will assume that we have computed this information.

Let F_0 be the face of G from which we perform the BFS of G and $R_1, \ldots, R_k$ be the corresponding set of retract-cycles. The rest of the algorithms will only work with this tree of retrack-cycles. The **retract-weight** of G with respect to F_0 is:

$$|G|_R = |R_1| + \cdots + |R_k| \quad \text{where } |R_i| \text{ is the weighted length of its level-cycle.}$$

Since the retack-cycles form a tree rooted at F_0, we get an ancestor/descendent relation. If R and R' are retrack-cycles where R' is a descendent of R, then the distance from R to R' is the maximum distance from any point in R' to the closest point in R. The distance between retrack-cycles may be quite large even though they are close in the descendent relation. We can bound this distance as follows:

Lemma 2.2 *If R' is a kth direct descendent of R then the distance from R to R' is at most $k + |R|/2 + |R'|/2$.*

Proof: Observe that the distance in G from a level-cycle C and its retract can be at most half the weighted length of C. Let C and C' be the level-cycles of R and R', respectively. Therefore the distance from R' to C' is at most $|R'|/2$, the distance from C' to C is at most k, and the distance from C to R is at most $|R|/2$. Thus the distance from R' to R is at most $|R'|/2 + k + |R|/2$. □

3 Phase I

As in [Mil86], the algorithm shall have two phases. In the first phase, developed in this section, we find a subgraph H of G with diameter and maximum face size $|G|_2$. In phase 2, as described in [Mil86], we find an $O(|G|_2)$ separator in H.

In this section we show that

Theorem 3.1 *If G is a 2-connected embedded planar graph with weights which sums to 1, no face weight $> 2/3$, 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 $|G|_2$, i.e., $dia + h \leq 1.58|G|_2$.

2. The maximum induced weight on any face of H is $\leq 2/3$.

The form of the algorithm follows quite closely that of Section 4 from [Mil86]. We will use the notation and idea from this section. Rather than use the branch cycles, defined in [Mil86], we will use the retract-cycles. As in [Mil86], the **trunk** $R_1, \ldots, R_t$ is a sequence of retract-cycles where R_1 is the root cycle, R_{i+1} is the direct descendence of R_i, where R_{i+1} has the largest exterior weight amongst their siblings. Let $C = R_\tau$ be the first cycle in the trunk with interior weight is at least $1/3$. Set $n_1 = |R_1| + \cdots + |R_\tau|$ and n_2 equal to the sum of the weights of the remaining retract-cycles. By the same arguments used in the proof of Lemma 5 in [Mil86] we get:

Lemma 3.2 *There exists an integer $\alpha_1 \geq 0$ such that some αth ancestor of C, say B, satisfies $\alpha_1 + \frac{3}{4}|B| \leq \sqrt{\frac{3}{2}}\sqrt{n_1}$.*

Proof: The proof is very similar to that of Lemma 5 in [Mil86]. We give the details for completeness.

Suppose the lemma is false and B_i is the ith ancestor of C. In this case $\frac{3}{4}|B_i| > \sqrt{\frac{3}{2}}\sqrt{n_1} - i$ for $0 \leq i \leq \lfloor\sqrt{2n_1}\rfloor$. Now the sum of the weights of the B_i satisfy the following:

$$\sum_{i=0}^{\lfloor\sqrt{\frac{3}{2}}\sqrt{n_1}\rfloor} \lceil\sqrt{\frac{3}{2}}\sqrt{n_1}\rceil - i = \sum_{i=1}^{\lceil\sqrt{\frac{3}{2}}\sqrt{n_1}\rceil} i \geq (\sqrt{\frac{3}{2}}\sqrt{n_1} + 1)(\sqrt{\frac{3}{2}}\sqrt{n_1})/2 > \frac{3}{4}n_1$$

But this contradicts the fact that the sum can be at most $\frac{3}{4}n_1$. □

Let H' be the subgraph consisting of vertices and edges "between" B and C. We next cap the large faces of H' by simultaneously adding on retract-cycles. We stop after α_2 levels where α_2 is given by the following lemma:

Lemma 3.3 *There exists an integer $\alpha_2 \geq 0$ such that after adding all α_2 direct descents of the retract-cycles in H' the maximum retract-cycle size β, which is a leaf in the construction, satisfies $2\alpha_2 + \frac{1}{2}\beta \leq \sqrt{2}\sqrt{n_2}$.*

Proof: The proof of this lemma is the same as that of Lemma 3.2.

The subgraph H will consist of a subtree of retract-cycles rooted at B plus all vertices and edges in G "between" these cycles. The height of this tree of retract-cycles will be $\alpha_1 + \alpha_2$. Our spanning tree of H will consist of all edges in B but one plus a BFS spanning tree from B into H. By Lemma 2.2 the total distance between the retract-cycles will be at most $\alpha_1 + \alpha_2 + \beta$. Thus the diameter of the spanning tree will be at most

$$\frac{3}{2}|B| + 2\alpha_1 + 2\alpha_2 + \frac{1}{2}\beta \leq 2\sqrt{\frac{3}{2}}\sqrt{n_1} + \sqrt{2}\sqrt{n_2} \leq \sqrt{6W(G)} \leq \sqrt{\frac{3}{4}}|G|_2.$$

We next must bound the maximum nonleaf face size of H. The maximum face is either a face from G or a new face in one of the caps. In the latter case, its size can be at most β from Lemma 3.3. In the prior case it will be at most d, the largest size face in G. If we start the BFS from the largest face, then the largest face from G in H can be at most the second largest face of G, which can be at most $|G|_2/\sqrt{2}$. We claim that $\beta \leq |G|_2/\sqrt{2}$. By Lemma 3.3 we know that $\beta \leq \sqrt{2n_2} \leq \sqrt{2W(G)} \leq \frac{1}{4}|G|_2$. Thus the diameter of the spanning tree plus the size of the maximum nonleaf face size is at most $\sqrt{\frac{3}{4}}|G|_2 + \sqrt{\frac{1}{2}}|G|_2. \leq (\sqrt{\frac{3}{4}} + \sqrt{\frac{1}{2}})|G|_2. < 1.58$.

The above discussion proves Theorem 3.1.

4 Phase II

In this section we simply observe that we can apply Theorem 5 from [Mil86]. We state the Theorem here for completeness.

Theorem 4.1 *If G is a 2-connected weighted and embedded planar graph with no face weight $> 2/3$ 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.*

Combining Theorems 3.1 and 4.1 we get a proof of Theorem 1.2.

5 Algorithms for Finding the Separator

In this section we discuss implementation details for finding the separator in Theorem 1.2. The sequential algorithm we presented uses Union-Find, and thus is not linear, but may require time $O(nG(n))$ time, see [AHU74]. The main difficulty is finding the tree of retract-cycles and the associated weights. The rest of the construct is straight forward and follows, for the most part, from the algorithms in [Mil86]. First we begin by determining when a face is interior to the ith level and also interior to the ith retract. We start by constructing the geometric dual of G and determining the size of each face. Next, we add the level edge to G in a BFS manner. We determine as we go the weight of each level edge. Observe that when the weight on a level reaches half the face size we use the face size in determining the rest of the edge weights. At this point the face, say F, is interior to the level. We mark the face as captured. To determine if it is interior to the retract, we check if any of the faces adjacent to F in G^* are in the retract. If we find such a face we mark F interior and check if F is adjacent to any faces which are marked as captured but not interior. We mark each such face interior and inspect in a DFS manner their adjacent faces. It follows that we traverse each edge and its dual at most a constant number of times, and determine for each face the level at which its is interior to the retract cycles, its index.

We next compute the tree determined by the retract-cycles. Observe that we cannot even explicitly write out these cycles because the sum of their lengths may be as large as n^2. Thus, we must find this tree without explicitly computing the cycles. By Lemma 3 in [Mil86] each retract-cycle at level i corresponds to the boundary of a connected component in the subgraph G_i^* of G^* induced by the faces of index $> i$. We need the following three pieces of information:

- A rooted tree corresponding to the tree of retract-cycles.
- The length of each retract-cycle.
- The weight exterior to each retract-cycles.

We compute all the information in one pass using one Union-Find. Suppose the face indexes run from 0 to k. We start by doing a Union for each edge between two faces of index k, technically we are working in the dual G^*. Using Finds we determine a representative for each connected component consisting of faces with index k. For a component, we determine its weight and boundary length. Thus, we have computed the three pieces of information for the leaf retact-cycles of index k. Suppose we have computed this information for index $k' + 1$. We compute the information for index k as follows.

1. Perform a Union for each edge between two faces one of index k' and the other of index $\geq k'$.
2. Perform a Find for each face of index k' and each representative from level $k' + 1$, obtaining new representatives for level k'.
3. The weight of each component at level k' will be the sum of the weights of the component from level $k' + 1$ in it, plus the weight contributed to it from its faces of index k' and their boundaries.
4. The new boundary will be the sum of its children's boundary sizes from level $k' + 1$, plus one for each new edge on the boundary of an index k' face, minus one for each old edge.

With this tree, all the remaining algorithms implicit in Phase I can be done in linear time. The algorithms for Phase II are in [Mil86].

6 Applications

6.1 Edge Separators

By considering the dual graph to G, we can construct edge separators.

Definition 6.1 *If $G = (V.E)$ is a graph with nonnegative weights on the vertices and edges which sums to one then a subset $E' \subset E$ of edges is a* **weighted-edge separator** *if E' partitions G into two disjoint subgraphs such that the weight of each is at most $2/3$.*

Theorem 6.2 below improves the previous best upper bound of $\sqrt{k \cdot v}$ obtained by [Mil86, DDSV]. In [Mil86] the weaker bound was proved for 2-connected planar graphs and was extended to all planar graphs in [DDSV]. By arguments similar to those used in Lemma 1.3, we see that this new bound is better up to constant factor.

Theorem 6.2 *A planar graph G with vertices of degree $k_1, \ldots, k_v$ has an edge separator of size $1.58\sqrt{k_1^2 + \cdots + k_v^2}$.*

Proof: Suppose G is an embedded planar for which we would like a separator. It is well known that G is 2-connected if and only if its dual G^* is 2-connected, see [Eve79]. Thus, if G is 2-connected, we simply find a simple cycle separator in G^* where the weight on a vertex is zero, the weight on an edge is the weight assigned to its dual, and the weight on a face is the weight assigned to its dual, a vertex in G. It follows that the simple cycle separator of G^* corresponds to an edge separator in G. If G is not 2-connected, we compute the tree T of 2-connected components of G. The vertices of T are either 2-connected components of G or cut points. We weight each vertex of T as in the proof of Theorem 2 in [Mil86]. If T has a separating vertex v, which is also a cut vertex of G, it follows that the edges of v form an edge separator of G of size at most $|G|_2$. If on the other hand, the only separating vertex of T corresponds to a proper 2-connected component C then we proceed as follows: Consider the weight graph C where the weight of each cut vertex x belonging to C includes the weight of the subtree of T which was attached to x. The graph C is 2-connected, no vertex weight greater than $2/3$, and any weighted-edge separator of C is one for G. Thus, we have reduced the general case to the 2-connected case which we handle as above. □

6.2 The Finite Element Graph from Numerical Analysis

Planar separators can be used for direct as well as indirect methods for solving certain linear systems. Possibly the most famous of these methods is nested and parallel nested dissection, [LRT79, PR85].

Definition 6.3 *The Finite Element graph $\hat{G} = (V, E')$ of a planar embedded graph $G = (V, E)$ has as its edge set*

$$\{(v, w) | \quad v \text{ and } w \text{ share a face of } G.\}$$

By Theorem 1.2 a finite element graph $\hat{G} = (V, E)$ has a separator of size $O(|E|)$. Thus if the nonzero entries of an n by n matrix A form a finite element graph then the linear system $Ax = b$ can be solved in time $O(w(A)^{1.5})$ where $w(A)$ is the number of nonzero entries in A.

7 Open Questions

It is open whether or not the $O(\sqrt{n}\log n)$ time using $\sqrt{n}/\log n$ processors parallel algorithms in [GM] can also be used to find a Euclidean separator. It would be very interesting to know if there is a processor-efficient NC algorithm which finds Euclidean separators, [GM87].

References

[AHU74] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[DDSV] K. Diks, H. N. Djidjev, O. Sykora, and I. Vrto. Edge separators for planar graphs and their applications. In Borlin, editor, *Proc. of 13th Mathematical Foundation of Computer Science*, pages 280–290, Carlsbad. Springer Verlage. LNCS 324.

[Dji81] H. N. Djidjev. A separator theorem. *Compt. End Acad. Bulg. Sci.*, 34(5):643–645, 1981.

[Eve79] S. Even. *Graph Algorithms.* Computer Science Press, Potomac, Maryland, 1979.

[Gaz86] Hillel Gazit. An improved algorithm for separating a planar graph. manuscript, 1986.

[GM] Hillel Gazit and Gary L. Miller. An $O(\sqrt{n}\log n)$ optimal parallel algorithm for a separator for planar graphs. manuscript.

[GM87] Hillel Gazit and Gary L. Miller. A parallel algorithm for finding a separator in planar graphs. In *28th Annual Symposium on Foundations of Computer Science*, pages 238–248, Los Angeles, October 1987. IEEE.

[LRT79] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. on Numerical Analysis*, 16:346–358, 1979.

[LT79] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. of Appl. Math.*, 36:177–189, April 1979.

[Mil86] Gary L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *Journal of Computer and System Sciences*, 32(3):265–279, June 1986. invited publication.

[PR85] Victor Pan and John Reif. Efficient parallel solution of linear systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 143–152, Providence,RI, May 1985. ACM.