Well-Connected Separators for Planar Graphs

Michelangelo Grigni* Andrzej P. Woloszyn

September 11, 2000

Abstract

Given an n-vertex weighted planar graph G, a separator is a subset S of vertices such that each component of $G - S$ has at most two-thirds of the original weight. We give an algorithm finding a separator while balancing these two parameters of $G[S]$: the number of components, and the cost of a minimum spanning forest. In particular, given uniform edge costs and a positive integer $k \leq \sqrt{n}$, we find a separator S such that $|S| = O(n/k)$ and $G[S]$ has $O(k)$ components.

1 Introduction

Given an n-vertex weighted planar graph G, the separator theorem of Lipton and Tarjan [12] gives us a subset S of $O(\sqrt{n})$ vertices such that each component of $G-S$ has at most $2/3$ the total weight. This result immediately found many divide-and-conquer applications [13]. However, the result says nothing about the structure of the induced graph $G[S]$; it could be a set of discrete points. Recent approximation schemes [4, 10] for the TSP (traveling salesman problem, and similar problems) in planar graphs require finding a separator S controlling the following quantities:

1. the number of connected components of $G[S]$; and
2. the cost of a minimum spanning forest in $G[S]$, as a fraction of the total cost of all edges in G.

We present an algorithm finding a separator with a multiplicative tradeoff between these two quantities.

In fact our separator S is the set of vertices along a Jordan curve C in an embedding of G. Curve C is a *V-cycle*: that is, each arc of C between consecutive vertices is either an edge of G or a *face edge*, contained in the interior of some face. The interior and exterior of C each have weight at most

*Contact author. Email: mic@mathcs.emory.edu. Address: Emory U. Math/C.S. Dept, 1784 North Decatur Rd., Atlanta GA 30322, USA.

2/3 the total, therefore each component of $G - S$ has weight at most 2/3 the total. We say that such a curve C is a separator. Furthermore, we bound the number of components of $G[S]$ by the number of face edges on C. The following is our main result.

Theorem 1.1 *Let G denote an embedded, 2-connected planar graph on n vertices. Suppose G has nonnegative weights on its vertices and faces, and nonnegative costs on its edges. Suppose W is the total weight of the graph, M is the total cost, and k is a positive integer. Then in $O(n \log n)$ time we can find a V-cycle C such that:*

- *The interior and exterior of C each has weight at most 2/3 W.*
- *C uses $O(k)$ face edges.*
- *C uses ordinary edges of total cost $O(M/k)$.*

Remarks. If a vertex or edge is used by the V-cycle or if a face is crossed by a face-edge of the V-cycle, then it counts towards neither the interior nor the exterior of the V-cycle. In particular, if any vertex or face has weight at least $W/3$, then we may take as our V-cycle a nearly trivial loop through that element (and one other). So in the following, we may assume that no single weight exceeds $W/3$.

Our result extends a similar claim for unweighted planar graphs [10]; actually [10] contains a gap which we correct in Section 4. That result and our result both depend on Miller's simple cycle separator [14] for triangulated planar graphs. A result similar to ours is sketched in [4], using a very different argument based on the method of Lipton and Tarjan.

Figure 1 provides an example of a V-cycle separator of a grid graph that uses four face-edges and two ordinary edges.

Preliminaries

We use standard graph theoretic terminology; for example see [6]. When G is a graph and W is a subset of its vertex set $V(G)$, $G[W]$ denotes the subgraph induced by W.

A *cycle* is a walk in a graph that returns to its starting vertex. A *path* is a walk with distinct vertices, and a *simple cycle* is a cycle with distinct vertices (excepting the first and last).

A *planar graph* can be drawn in the plane so that its edges intersect only at their endpoints. Such a drawing of a planar graph G is called a *planar embedding* of G, or a *plane graph.*

A planar embedding of a simple cycle can be viewed as a continuous non-self-intersecting curve whose origin and terminus coincide. Such a curve is called a *Jordan curve.* By the Jordan Curve Theorem (see [11]), a Jordan curve J

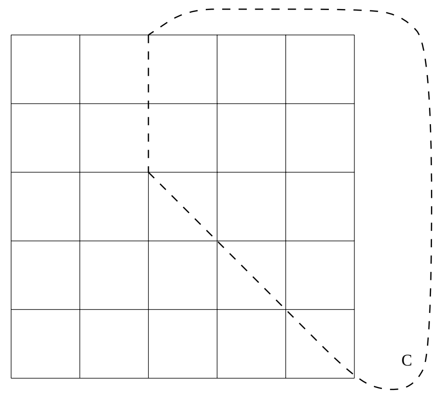


Figure 1: A V-cycle separator in a grid graph.

divides the rest of the plane into two disjoint open sets, the *interior* and *exterior*, denoted by $\operatorname{int} J$ and $\operatorname{ext} J$ respectively. Any curve joining $\operatorname{int} J$ and $\operatorname{ext} J$ meets J in at least one point.

A plane graph G partitions the plane into a number of connected open regions, the *faces* of G. A face f is said to be *incident* with the vertices and edges in its boundary. If two distinct faces f_1 and f_2 are incident with an edge e, we say that *e separates f_1 and f_2*.

Given a plane graph G, we denote its *dual* plane graph by G^*. Each face f of G corresponds to a dual vertex f^* of G^*. Each e edge of G corresponds to a dual edge e^* of G^*, connecting the faces incident to e.

A *spanning forest* of graph G is a forest subgraph whose connected components are exactly those of G. When G is connected, it is a spanning tree.

With each vertex v and face f of G, we may associate a non-negative real *weight*, denoted by $w(v)$ and $w(f)$, respectively. With each edge e we may associatd a non-negative real *cost*, denoted by $c(e)$. A planar graph G, together with these costs and weights on its elements, is called a *weighted graph*.

2 The Cycle Separator

For the remainder of this paper assume that G is a weighted planar graph on n vertices with total weight on its vertices and faces equal to W, and total cost on its edges equal to M.

2.1 Miller's Cycle Decomposition

Theorem 1.1 is based on the result of Miller [14], who considered the case $k = 0$ (a simple cycle separator). Miller labeled the faces and vertices of G in the following way: an arbitrarily chosen root face f_0 gets label 0, all the vertices adjacent to it get label 1, all unlabeled faces adjacent to a vertex with label 1 get label 2, and so on. In other words, this is the distance labeling from a breadth first search of the vertex-face incidences in G.

Vertices around a face of label $2n$ can only have label $2n-1$ or $2n+1$. Consequently, the endpoints of an edge e have either the same label or they differ by two. The same is true of the two face labels adjacent to e. Local arguments [14] show that the set of edges whose adjacent faces have different labels forms a vertex-disjoint collection of simple cycles. All vertices lying on such a cycle C have the same label, and so we define that to be the label of C. Further we observe that cycles with the same label are non-nesting, and that a vertex v on a label $2n+1$ cycle shares a face f with some vertex w on a $2n-1$ cycle (See Fig. 2). We will refer to this fact as the *visibility property* of the decomposition, and we call f and w the (not necessarily unique) *parent* and *grandparent* of v, respectively.

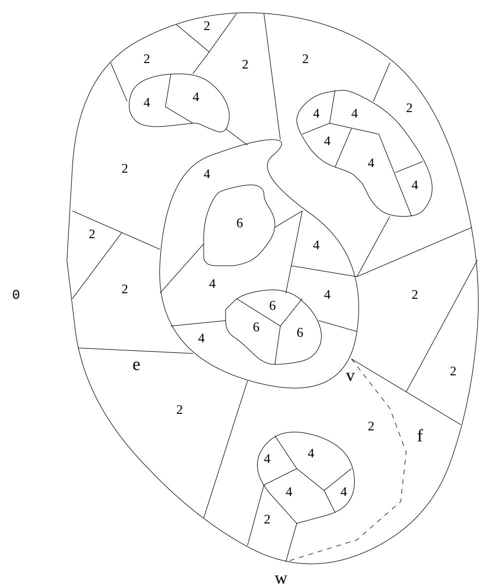


Figure 2: A Cycle Decomposition with face labels.

We define the *partition tree* $\mathcal{T}$ as follows: $V(\mathcal{T})$ is the collection of cycles in the above decomposition, and two cycles are adjacent if they share a face and differ in label. The properties of the decomposition insure that $\mathcal{T}$ is a tree.

2.2 The Cycle Tree

Let $\mathcal{T}$ be the partition tree of G rooted at an arbitrary face f_0, as described above. Each cycle separates the plane into two disjoint regions; we will refer to the region containing f_0 as the *interior* of the cycle; we call the other region the *exterior* of the cycle. The *weight* of the interior (exterior) is the total weight of all vertices and faces lying in the interior (exterior). We construct the *main sequence of cycles* on $\mathcal{T}$ as the path from the root cycle to a leaf cycle, such that from each cycle we go to its child whose exterior has the largest weight (break ties arbitrarily).

We will call the perimeter of a cycle *heavy* if the total cost of its edges exceeds M/k; otherwise we call the perimeter *light*. We will call the exterior (interior) of a cycle *heavy* if its weight exceeds $W/2$; otherwise the exterior (interior) will be called *light*. Let B denote the first cycle on the main sequence of cycles with a light exterior. We proceed along the main sequence of cycles and denote by C the first cycle after B that has light perimeter. The case when no such C can be found, that is, the leaf on the main sequence has exterior weight larger than $W/2$ or heavy perimeter, will be handled separately in Section 4. For now, we assume that the exterior of C is light and that its perimeter is light. We recall here that no face or vertex has weight exceeding $W/3$.

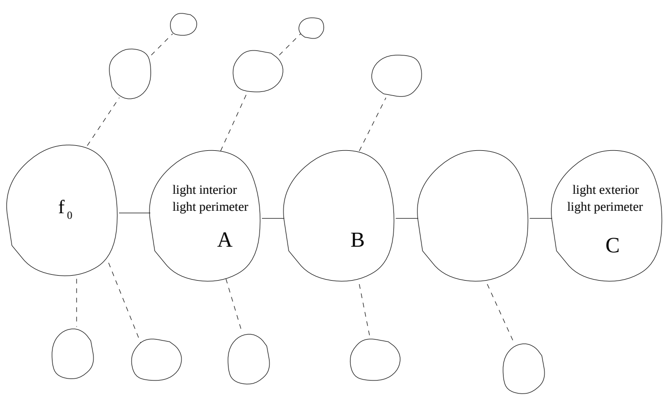


Figure 3: Main Sequence of cycles.

From B, we also walk back towards the root and denote by A the first cycle found with light perimeter. If no such cycle is found, then A denotes the root cycle around f_0.

2.3 Pruning the Graph

We replace the interior of A by an empty face of the same weight, and we replace $\mathcal{T}$ by its subtree rooted at A. Similarly we replace the exterior of C by an empty face of the same weight, and we remove the descendants of C in $\mathcal{T}$ (making C a leaf). Note that the modified $\mathcal{T}$ is the cycle tree of the modified G. We call this procedure *capping* the interior or exterior of a given cycle.

Consider now the subcollection of cycles that are at the same distance in $\mathcal{T}$ from the segment A-C. We examine the total edge cost of the perimeter of all cycles in a given subcollection, level by level, starting with the direct descendants of A-C. When this cost drops below M/k or when all the cycles in the decomposition have been examined, we will stop this procedure and cap the exteriors of all cycles on that level. We refer to this surgery as *pruning*.

Note that this search may not terminate with a light perimeter subcollection of cycles, namely when all of G has been exhausted. Should that be the case, for each capped leaf of the pruned $\mathcal{T}'$ we attach a trivial cycle of perimeter cost 0. Whichever the case, the pigeonhole principles guarantees that the number of levels in the pruned $\mathcal{T}'$ does not exceed k.

Thus the pruned tree $\mathcal{T}'$ has the following properties:

1. Each leaf cycle of $\mathcal{T}'$ has light exterior and light perimeter.

2. The root A of $\mathcal{T}'$ has light interior and is either the cycle around f_0 or it has light perimeter.

3. The diameter of the tree is at most k.

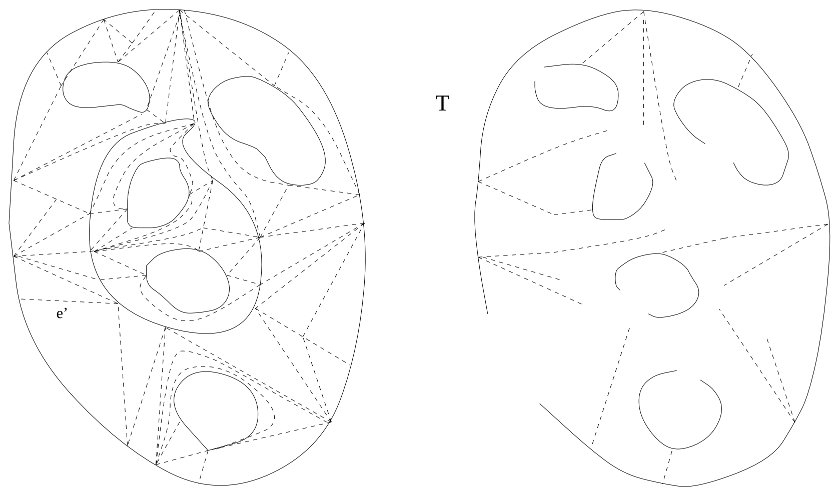
Note that the pruned tree $\mathcal{T}'$ is the cycle tree of the pruned graph G'. In the following we focus on $\mathcal{T}'$ and G' rather than $\mathcal{T}$ and G.

3 Triangulation

We now proceed with triangulation of G', the uncapped part of G. Because of the visibility property, each face in the uncapped part of the graph with label $2n$ is adjacent with at least one level-$(2n-1)$-vertex w. Therefore, we can triangulate each face by adding face-edges from the parent w to all vertices around that face.

Furthermore, notice that each non-cycle ordinary edge u_1u_2 in G' separates two faces that are of the same label l (otherwise u_1u_2 would lie on some cycle of $\mathcal{T}'$). To simplify the cost analysis in Section 3.1 we remove each non-cycle edge u_1u_2 from the triangulated graph. The thus created new face will still have label l and vertices u_1 and u_2 on its boundary. We then insert a face-edge that spans the new face between vertices u_1 and u_2 and replaces the removed ordinary edge u_1u_2 (compare ordinary edge e in Fig. 2 with corresponding face-edge e' in Fig. 4(a)).

Next, we remove all face-edges adjacent to the capped leaf cycles except for an arbitrarily chosen one and construct the breadth first search tree T on the modified G' starting with an arbitrary vertex of the root face.



(a) (b)

Figure 4: The triangulated G' (a), and a corresponding BFS tree T (b).

Notice that the only true edges present in T are those belonging to A or to the capped leaf cycles of $\mathcal{T}'$ (see Fig. 4(b)).

3.1 Construction of the Cycle Separator

We are now ready to employ Lipton and Tarjan's algorithm and search for the cycle separator.

Lipton and Tarjan [12] have shown that given nested cycles C_0 and C_1, where C_1 lies in the exterior of C_0, and $w(extC_0) \leq 2W/3$, $w(intC_1) \leq 2W/3$, there exists a cycle C_2 such that C_0 lies in $intC_2$, C_1 lies in $extC_2$, and both the weight of the interior and exterior of C_2 does not exceed $2W/3$ (see Fig. 5).

We will give a more precise characterization of C_2 in Lemma 3.2.

3.1.1 Balance argument

Definition 3.1 *Given a planar graph G with weights on its vertices and faces, we say that the weighting is* proper *if:*

- *All weights are non-negative.*
- *The total weight is 1.*

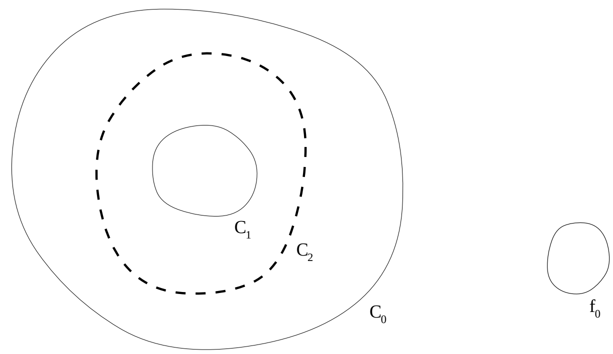


Figure 5: LT nested cycles lemma.

- *Every face has weight at most 1/3.*

A *2/3-separating cycle* is a simple cycle C in a graph G such that its interior and exterior (excluding C itself) both have weight at most 2/3.

Lemma 3.2 (LT modified) *Given a triangulated planar graph G, a proper weighting, and given a spanning tree T in G, then there is a 2/3-separating cycle in G of one of the following two forms:*

1. *The cycle C_e induced in T by a non-tree edge e.*
2. *A cycle induced in T by two adjacent non-tree edges e_1 and e_2, such that e_1 and e_2 lie on a common triangle Δ with a third non-tree edge e_3.*

Remark 1 *The second type is almost the same as the cycle C_{e_3}, except that we replace e_3 by the two adjacent edges e_1 and e_2. Cycles of the second type are necessary, as we may see by considering K_4 with equal weights on its four faces.*

Proof: Consider a non-tree edge e of G, which induces a cycle C_e in the tree T. If any such C_e is a 2/3-separating cycle, then we are done. So suppose instead that for every non-tree edge e, the cycle C_e has more than 2/3 weight in either its interior or exterior. In this way we may orient the dual edge of e to point towards the heavier side.

These oriented duals of non-tree edges form a tree T' in the dual G' of G (a dual cycle would imply that the original spanning tree T is not connected). Because each face weighs at most 1/3, all the leaf edges of T' are oriented into T' (that is, away from the leaf). Also since G was triangulated, T' has degree at most three (see Fig. 6).

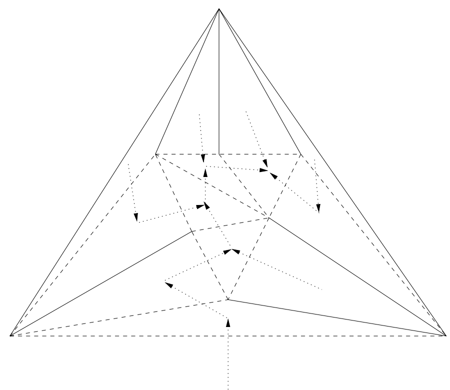


Figure 6: Oriented Dual Tree on T.

Thus T' contains one of the two configurations depicted in Fig. 7;

Figure 7: LT search algorithm: basic dual configurations.

corresponding to these are primal configurations in T described in Fig. 8.

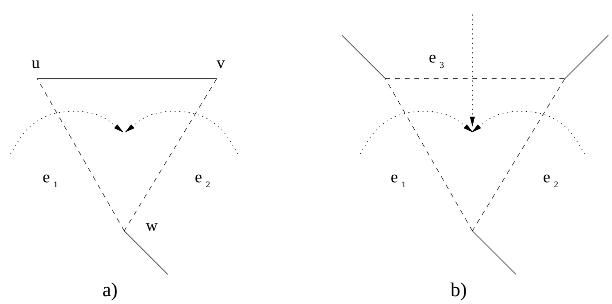


Figure 8: LT search algorithm: basic primal configurations.

In each case, let S_i denote the subset of the graph on the "heavy side" (interior or exterior) of C_{e_i}, and U_i the subset on the "light side" of C_{e_i}. Note that

$$w(U_i) < \frac{1}{3}. \tag{1}$$

In case (a), let u and v denote the distinct endpoints of e_1 and e_2, and let w denote their common endpoint. Also, let P_1 be the unique path between u and w in T, and let P_2 be the unique path between v and w. Now, if v lies on P_1, then $P_1 = uv \cup P_2$. If v does not lie on P_1, then $P_2 = uv \cup P_1$. In either case, the symmetric difference between C_{e_1} and C_{e_2} is $uv \cup e_1 \cup e_2$, which implies that S_1 and S_2 only share the triangle face Δ. Thus $w(S_1) + w(S_2) \leq w(G) + w(\Delta) \leq 4/3$, but this implies that one of them is at most $2/3$.

Case (b) is only slightly more complex. Consider the cycle C'_{e_3} which we get by replacing e_3 with e_1e_2 in C_{e_3}. Compared to the cut induced by C_{e_3}, the (formerly) heavy side has been reduced by $w(\Delta) + w(e_1) + w(e_2)$, and the (formerly) light side has been increased by $w(\Delta) + w(e_3)$. Call the (formerly) heavy side S'_3 and the (formerly) light side U'_3.

Suppose now that neither one of the one-edge induced cycles nor any of the two-edge induced cycles is balanced. For the two-edge induced cycles, two mutually exclusive possibilities arise: either at least two out of three formerly light sides became "heavy" (that is, $w(U'_i) > 2/3$) or at least two out of three formerly "heavy" sides remained "heavy" ($w(S'_i) > 2/3$). However, if $w(U'_i) > 2/3$ holds for any two sides U'_i, then $w(\Delta) > 1/3$, Δ being the only element of the graph that appears twice in the union $U'_1 \cup U'_2$. Now, since at most one U'_i is "heavy", at least two S'_i must still be "heavy", say S'_1 and S'_2. However, the intersection of S'_1 and S'_2 is exactly U_3, and consequently $w(U_3) > 1/3$, which contradicts 1. □

3.1.2 Cost analysis

With the exception of the root and leaf cycles of $\mathcal{T}'$, all edges in T are face-edges across the faces of G' between vertices of different labels. The induced cycle of Lemma 3.2 in T will involve at most two leaf cycles, twice the diameter of T of face-edges and possibly the root cycle A (see Fig. 9).

Moreover, if $A{=}f_0$ a path that encircles it can be replaced by a face-edge across its face (if $A \neq f_0$, then its perimeter is necessarily light). Hence, a V-cycle in G' will involve $O(k)$ face-edges and cost at most $2M/k$.

4 The Heavy Leaf Case

In the previous section we showed an algorithm searching for a V-cycle separator in a graph whose cycle decomposition tree does not contain a "heavy" (with weight of the exterior exceeding $W/2$) leaf, or a leaf with heavy perimeter. We will deal with the heavy leaf case by reducing it to the previous case. We insert an artificial weightless triangle Δ inside a face F of the heavy leaf, which

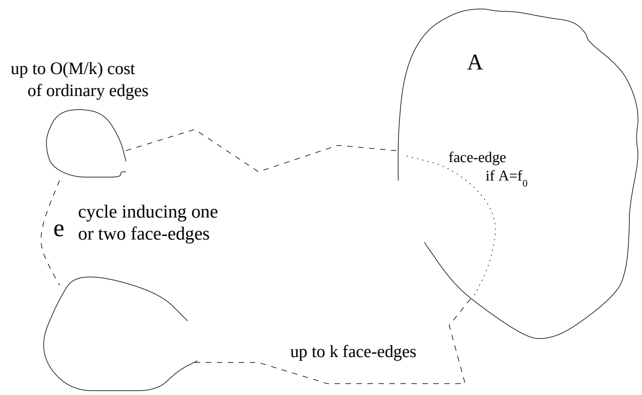


Figure 9: LT cycle in capped and pruned G'.

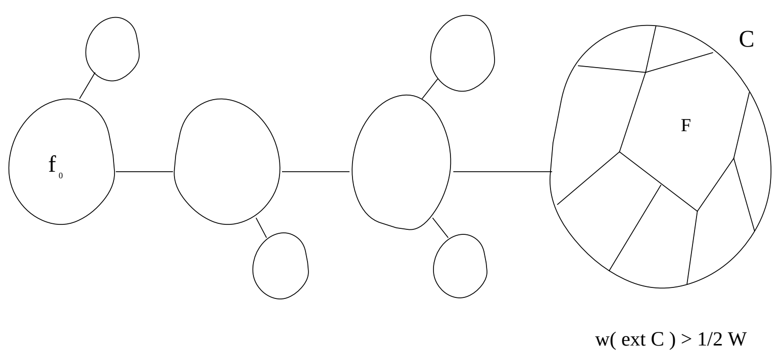


Figure 10: Heavy Leaf.

will then become the leaf in $\mathcal{T}$, and proceed with the cycle separator finding algorithm on this extended graph. Finally, we will contract the separator found in the augmented graph to recover the separator of the original graph G.

The insertion of Δ will take place after the triangulation of the heavy leaf; the triangulation of the remainder of the face F will proceed in a way depicted in Fig. 11: two edges from each of the grandparent vertices of Δ will join two of the vertices of Δ. The weight of the $v_1v_2v_3$-triangle surrounding Δ will be distributed arbitrarily so as to leave Δ weightless.

Let C be the V-cycle separator in the augmented graph. Notice that C may miss the vertices of Δ altogether, in which case it is also a V-cycle separator in G. Suppose it does not. Then it must enter the exterior (the part of the plane that does not contain f_0) of the $v_1v_2v_3$-cycle through, say, v_1 and leave through, say, v_2. We will replace the part of C in the exterior of the $v_1v_2v_3$-cycle with a single v_1v_2 face-edge through F to obtain a V-cycle $\hat{C}$ in G. This contraction

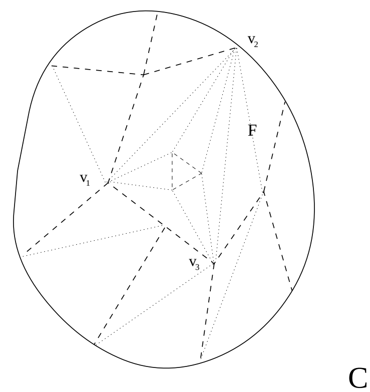


Figure 11: Heavy Leaf - insertion of the triangle.

will increase the length of C by at most the length of one face-edge; it will not increase the weight of either the interior or the exterior of C. It may in fact decrease it by the weight of some elements in the exterior of the $v_1v_2v_3$-cycle. This implies that $\hat{C}$ is a V-cycle separator of G having the properties claimed in Theorem 1.1.

5 Concluding Remarks

As mentioned before, our result is motivated by the problem of finding an efficient approximation scheme for the traveling salesman problem in weighted planar graphs [4]. A similar problem is the Steiner version: given a set of terminals $S \subseteq V$, find a minimum-cost circuit visiting all those terminals. It turns out that this problem is a common generalization of both the planar-graph TSP and the Euclidean TSP (even with obstacles, which remains unsolved). In order to obtain a PTAS for this problem, it would be sufficient to give a polynomial time algorithm to obtain a $(1+\epsilon)$-spanner of a Steiner type for weighted planar graphs, namely:

Conjecture 1 *There exists a function $f(\cdot)$ such that: given $\epsilon > 0$, a planar graph G with edge costs, and a subset S of vertices, there exists an edge-induced subgraph G' which $(1+\epsilon)$-approximates all internode distances in S, and furthermore G' has total edge cost at most $f(\epsilon)$ times the minimum Steiner tree cost for S.*

Such a result would let us approximate (in planar graphs) the metrical problems considered by Arora [3] for Euclidean space.

A recent result generalizes Theorem 1.1 to graph families with a fixed forbidden minor [9], albeit with worse constants and efficiency. This implies a quasi-polynomial time approximation scheme for metric TSP (and some similar problems) in such graphs.

References

[1] N. Alon, P. Seymour, R. Thomas, "A separator theorem for graphs with an excluded minor and its applications", in *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pp. 293-299, 1990.

[2] I. Althofer, G. Das, D. Dobkin, D. Joseph, L. Soares, "On sparse spanners of weighted graphs", *Journal of Discrete and Computational Computational Geometry*, Vol. 9, no. 1, pp. 81-100, 1993.

[3] S. Arora, "Polynomial-time approximation schemes for Euclidean TSP and other geometric problems", in *Proceedings of 37th IEEE Symposium on Foundations of Computer Science*, pp. 2-12, 1996.

[4] S. Arora, M. Grigni, D. Karger, P. Klein and A. Woloszyn, "A Polynomial-Time Approximation Scheme for Weighted Planar Graph TSP", in *Proc. ACM-SIAM 9th Annual Symp. on Discrete Algorithms*, pp. 33-41, 1998.

[5] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem", in J.F. Traub, editor, *Symposium on new directions and recent results in algorithms and complexity*, p. 441. Academic Press, NY, 1976.

[6] R. Diestel. *Graph theory.* Springer-Verlag, New York, 1997.

[7] Greg N. Frederickson, "A distributed shortest path algorithm for a planar network", *Journal of Information and Computation*, Vol. 86 no. 2, pp. 140-159, 1990.

[8] Greg N. Frederickson, "Space efficient message routing in c-decomposable networks", *SIAM Journal on Computing*, Vol. 19, no. 1, pp. 164-181, 1990.

[9] M. Grigni, "Approximate TSP in graphs with forbidden minors", in *Proc. Automata, Languages and Programming: 27th International Colloquium*, pp. 869–877, 2000. Vol. 510 of *Lecture Notes in Computer Science*, Springer-Verlag.

[10] M. Grigni, E. Koutsoupias and C.H. Papadimitriou, "An Approximation Scheme for Planar Graph TSP", in *Proc. IEEE Symposium on Foundations of Computer Science*, pp. 640-645, 1995.

[11] J. R. Munkres, *Topology: a First Course*, Prentice Hall, Englewood Cliffs, NJ, 1975.

[12] R.J. Lipton and R.E. Tarjan, "A separator theorem for planar graphs", *SIAM Journal on Applied Mathematics*, Vol. 36, pp. 177-189, 1979.

[13] R.J. Lipton and R.E. Tarjan, "Applications of a planar separator theorem", *SIAM Journal on Computing*, Vol. 9, pp. 615-627, 1980.

[14] Gary L. Miller, "Finding small simple cycle separators for 2-connected planar graphs", *Journal of Computer and System Sciences*, Vol. 32, pp. 265-279, 1986.

[15] S. Sahni and T. Gonzales, "P-Complete Problems and Approximate Solutions", in *Proc. IEEE 15th Annual Symposium on Switching and Automata Theory*, pp. 28-32, 1974.

[16] W-K. Shih, S. Wu, Y. Kuo, "Unifying maximum and minimum cut of a planar graph", *IEEE Journal of the Institute of Electrical and Electronics Engineers – Transactions on Computers*, Vol. 39, no. 5, pp. 694-697, 1990.