A Simple Algorithm for Computing a Cycle Separator

Sariel Har-Peled* Amir Nayyeri†

February 5, 2022

Abstract

We present a linear time algorithm for computing a cycle separator in a planar graph that is (arguably) simpler than previously known algorithms. Our algorithm builds on, and is somewhat similar to, previous algorithms for computing separators. In particular, the algorithm described by Klein and Mozes [KM17] is quite similar to ours. The main new ingredient is a specific layered decomposition of the planar graph constructed differently from previous BFS-based layerings.

1. Introduction

The planar separator theorem is a fundamental result in the study of planar graphs that has been used in many divide and conquer algorithms. The theorem guarantees for planar graphs the existence of $O(\sqrt{n})$ vertices whose removal breaks the graph into “small” pieces, connected components of size at most $\alpha n$ for a constant $\alpha$. For triangulated planar graphs, a stronger result is known – the separator is a simple cycle of length $O(\sqrt{n})$ whose inside and outside (in the planar embedding) each contains at most $\alpha n$ vertices.

The separator theorem was first proved by Ungar [Ung51] with a slightly weaker upper bound of $O(\sqrt{n} \log n)$. Lipton and Tarjan [LT79] showed how to compute, in linear time, a separator of size $O(\sqrt{n})$. Later, Miller [Mil86] described a linear time algorithm for computing a cycle separator.

In this paper, we describe a simple algorithm for computing a cycle separator. We believe the simplicity of our algorithms is comparable to that of the original algorithm of Lipton and Tarjan [LT79].

Existential proofs. Alon et al. [AST94] described an existential proof of the cycle separator theorem using a maximality condition. Miller et al. [MTTV97] showed how to compute a planar separator in a planar graph if its circle packing realization is given (this proof was later simplified by Har-Peled [Har13]). In particular, the planar separator theorem is an easy consequence of the work of Paul Koebe [Koe36] (see [Har13] for details). A nice property of the proof of Miller et al. [MTTV97], is that it immediately implies the cycle separator theorem. Unfortunately, there is no finite algorithm for computing the circle packing realization of a planar graph – all known algorithms are iterative convergence algorithms. That is, the proof of Miller et al. is an existential proof.
Constructive proofs. As mentioned above, Miller [Mil86] gave a linear time algorithm for computing the cycle separator. A somewhat different algorithm is also provided in the work of Klein et al. [KMS13], which computes the whole hierarchy of such separators in linear time. Fox-Epstein et al. [FMPS16] also provides an algorithm for computing a cycle separator in linear time.

This paper. A simple cycle is a $\alpha$-separator if its inside and outside each contains at most $\lceil \alpha f \rceil$ faces, where $f$ is the number of faces of the graph. We present a linear time algorithm for computing a cycle $2/3$-separator – see Theorem 3.6. The algorithm is somewhat similar in spirit to the work of Fox-Epstein et al. [FMPS16]. A closer algorithm to ours is described by Klein and Mozes [KM17, Section 5.9]. The new algorithm is (arguably) slightly simpler than these previous versions.

The rest of the paper is composed of two section. In Section 2 we define some required basic concepts, and in Section 3 we describe the new algorithm.

2. Preliminaries

Let $G$ be a triangulated planar graph embedded in the plane, with vertex set $V$, edge set $E$, and face set $F$, and let $G^* = (V^*, E^*, F^*)$ be the dual of $G$. A vertex $x \in V$ corresponds to a face $x^* \in F^*$, an edge $xy \in E$ to an edge $(xy)^* \in E^*$, and a face $xyz \in F$ to a vertex $(xyz)^* \in V^*$. Because of the last correspondence, and since $G$ is triangulated, $G^*$ is 3-regular: all its vertices have degree three. For any spanning tree $T = (E_T, V_T)$ of $G$, the duals of the edges $E \setminus E_T$ form a spanning tree of the dual graph $G^*$.

For any simple cycle $C$ in the embedding of $G$, the inside (resp., outside) of $C$, denoted by $\text{in}(C)$ (resp., $\text{out}(C)$), is the bounded (resp., unbounded) region of $\mathbb{R}^2 \setminus C$. Each vertex of $V$ is inside, outside or on $C$. A face is inside (resp. outside) $C$ if its interior is a subset of $\text{in}(C)$ (resp. $\text{out}(C)$). It follows that each face of $G$ is either inside or outside $C$. If a face $\theta$ is inside $C$, then $C$ contains $\theta$.

Definition 2.1. For a cycle $C$, and an $1/2 \leq \alpha < 1$, $C$ is an $\alpha$-cycle separator of a graph $G$, if the number of faces inside (resp. outside) $C$ is at most $\lceil \alpha \cdot f \rceil$, where $f = |F|$ is the number of faces of $G$.

For two cycles $C_1$ and $C_2$ of $G$, $C_1$ is inside $C_2$, denoted by $C_1 \leq C_2$, if $\text{in}(C_1) \subseteq \text{in}(C_2)$. For $C_1 \leq C_2$, a face is between $C_1$ and $C_2$, if it is inside $C_2$ and outside $C_1$.

Let $\gamma$ be a simple path or cycle in $G$. The length of $\gamma$, denoted by $|\gamma|$, is the number of edges of $\gamma$. If $\gamma$ is a path, and $x, y$ are vertices on $\gamma$, $\gamma[x, y]$ denotes the subpath of $\gamma$ between $x$ and $y$. For two internally disjoint paths $\gamma_1$ and $\gamma_2$, if the last vertex of $\gamma_1$ and the first vertex of $\gamma_2$ are identical, $\gamma_1 \circ \gamma_2$ denotes the path or cycle obtained by their concatenation.

3. The cycle separator theorem

Let $G = (V, E, F)$ be a triangulated planar graph embedded on the plane, and let $n = |V|$, and $f = |F|$. In this section, we describe the linear time algorithm for computing a cycle separator of $G$.

Our construction is composed of three phases. First, we find a possibly long cycle separator $S$, by finding a spanning tree $T$ of $G$, and a balanced edge separator $(uv)^*$ in its dual tree. The unique cycle in $T \cup \{uv\}$ is guaranteed to be a (possibly long) cycle separator (Section 3.1). This part of the construction is similar to Lemma 2 of Lipton and Tarjan [LT79], and we include the details for completeness. Next, we build a nested sequence of cycles $C_1 \preceq C_2 \preceq \ldots \preceq C_k$ (Section 3.2). The specific construction of these cycles, which is guided by $S$, is the main new ingredient in the new algorithm. Finally, we consider the
collection of cycles $C_1, \ldots, C_k$ and $S$, and construct a few short cycles, such that one them is guaranteed to be a balanced separator (Section 3.3).

3.1. A possibly long cycle separator

We start by computing a balanced separator that, unfortunately, can be too long. For a BFS tree $T$, we denote by $\pi(T, u)$ the unique shortest path in $T$ between the root of $T$ and $u$.

Lemma 3.1 ([LT79]). Given a triangulated planar graph $G$, one can compute, in linear time, a BFS tree $T$ rooted at a vertex $r$, and an edge $uv \in E(G)$, such that:

(A) the (shortest) paths $p_u = \pi(T, u)$ and $p_v = \pi(T, v)$ are edge disjoint,
(B) the cycle $S = p_u \cup p_v \cup uv$ is a $2/3$-separator for $G$.

Proof: Our proof is a slight modification of the one provided by Lipton and Tarjan [LT79], and we include it for the sake of completeness. Let $r' \in V$ be any vertex, and let $T = (V, E_T)$ be a BFS tree rooted at $r'$. Also, let $D = E \setminus E_T$, and note that the dual set of edges $D^*$ is a spanning tree of the dual $G^*$. Since $G$ is a triangulation, $D^*$ has maximum degree at most three. Thus, it contains an edge $(uv)^*$ whose removal leaves two connected components, $D^*_{in}$ and $D^*_{out}$, each with at most $\lceil (2/3)f \rceil$ (dual) vertices, see Lemma A.1, where $f = |F|$ is the number of faces of $G$. Let $D^*_{out}$ be the connected component that contains the dual of the outer face, and let $D^*_{in}$ be the other one.

Let $uv$ be the original edge that is dual of $(uv)^*$, and $S$ the unique cycle in $T \cup \{uv\}$. The sets of faces inside and outside $S$, correspond to the vertex sets of $D^*_{in}$ and $D^*_{out}$, respectively. Thus, $S$ is a $2/3$-cycle separator.

Now, let $r$ be the lowest common ancestor of $u$ and $v$ in $T$. The cycle $S$ is composed of $p_u = T[r, u]$, $p_v = T[r, v]$ and the edge $uv$. Since $T$ is a BFS tree, and $r$ is an ancestor of $u$ and $v$, the paths $p_u$ and $p_v$ are shortest paths in $G$.

To get a BFS tree rooted at $r$, one simply recompute the BFS tree starting from $r$, where we include the edges of $p_u$ and $p_v$ in the newly computed BFS tree $T$.

For the rest of the algorithm, let $S$, $r$, $uv$, $p_u$ and $p_v$ be as specified by Lemma 3.1. We emphasize that the graph is unweighted, $p_u$ and $p_v$ are shortest paths, and $u$ and $v$ are neighbors.
3.2. A nested sequence of short cycles

Let \( r \) be the root node of the BFS tree \( T \) computed by Lemma 3.1. For \( x \in V(G) \), let \( \ell(x) \) be the distance in \( T \) of \( x \) from the root \( r \). The level of a (triangular) face \( \eta = xyz \) of \( G \) is \( \ell(\eta) = \max(\ell(x), \ell(y), \ell(z)) \). In particular, a face \( \eta = uvz \in F(G) \) is \( i \)-close to \( r \) if \( \ell(\eta) \leq i \). The union of all \( i \)-close faces, form a region \( P_{\leq i} \) in the plane\(^1\). This region is simple, but it is not necessarily simply connected.

Let \( h = \max(\ell(u), \ell(v)) \), and let \( \psi \in \{u, v\} \) be the vertex realizing \( h \). We assume, for the sake of simplicity of exposition, that \( \psi \) is one of the vertices of the outer face \(^2\).

For \( i < h \), let \( \xi_i \) be the outer connected component of \( \partial P_{\leq i} \). This is a closed curve in the plane, with \( \psi \) being outside it (as long as \( i < h \)), and let \( C_i \) be the corresponding cycle of edges in \( G \) that corresponds to \( \xi_i \). The resulting set of cycles is \( C_0, \ldots, C_{h-1} \) (i.e., a cycle \( C_i \) is empty if \( i \geq h \)).

**Lemma 3.2.** We have the following:

(A) For any \( i < h \), the vertices of \( C_i \) are all at distance \( i \) from \( r \) in \( T \).

(B) For any \( i < h \), the cycle \( C_i \) is simple.

(C) For any \( i < j < h \), the cycles \( C_i \) and \( C_j \) are vertex disjoint.

(D) For \( i < h \), the cycle \( C_i \) intersects the cycle \( S \).

**Proof:**

(A) Consider a vertex \( x \) in \( G \) with \( \ell(x) < i \). As \( T \) is a BFS tree, we have that all the neighbors \( y \) of \( x \) in \( G \), have \( \ell(y) \leq \ell(x) + 1 \leq i \). Namely, all the triangles adjacent to \( x \) are \( i \)-close, and the vertex \( x \) is internal to the region \( P_{\leq i} \), which implies that it can not appear in \( C_i \).

(B) Since \( \xi_i \) is the (closure) of the outer boundary of a connected set, the corresponding cycle of edges \( C_i \) is a cycle in the graph. The bad case here is that a vertex \( x \) is repeated in \( C_i \) more than once. But then, \( x \) is a cut vertex for \( P_{\leq i} \) – removing it disconnects \( P_{\leq i} \) – see Figure 3.3. Now, \( \ell(x) < i \) as the BFS from \( r \) must have passed through \( x \) from one side of \( P_{\leq i} \) to the other side. Arguing as in (A), implies that \( x \) is internal to \( P_{\leq i} \), which is a contradiction.

(C) is readily implied by (A).

---

\(^1\)Here, conceptually, we consider the embedding of the edges of \( G \) to be explicitly known, so that \( P_{\leq i} \) is well defined. The algorithm does not need this explicit description.

\(^2\)This can be ensured by applying inversion to the given embedding of \( G \) – but it is not necessary for our algorithm.
(D) Indeed, $C_i$ must intersect the shortest path $p_\psi$ from $r$ to $\psi$, and as this path is part of $S$, the claim follows.

Computing the cycles $C_i$, for all $i$, can be done in linear time (without the explicit embedding of the edges of $G$). To this end, compute for all the (triangular) faces of $G$ their level. Next, mark all the edges between faces of level $i$ and $i+1$ as boundary edges forming $\partial P_{\leq i}$ — this yields a collection of cycles. To identify the right cycle, consider the shortest $p_\psi$ path between $r$ and $\psi$. The cycle with a vertex that belongs to $p_\psi$ is the desired cycle $C_i$. Clearly, this can be done in linear time overall for all these cycles.

**Lemma 3.3.** Let $\Delta > 0$ be an arbitrary parameter. If $h = \ell(\psi) > \Delta$, then there exist an integer $i_0 \in [\Delta]$, such that $|C_{i_0}| > 0$ and $\sum_{j \geq 0} |C_{i_0+j\Delta}| \leq n/\Delta$, where $|C_k|$ denotes the number of vertices of $C_k$.

**Proof:** Setting $g(i) = \sum_{j \geq 0} |C_{i+j\Delta}|$. By Lemma 3.2 (D), $g(i) > 0$, for $i = 0, \ldots, \Delta - 1$. We have

$$\sum_{i=0}^{\Delta-1} g(i) \leq \sum_{i=0}^{\Delta-1} \sum_{j \geq 0} |C_{i+j\Delta}| = \sum_{k \geq 0} |C_k| \leq |V(G)| \leq n,$$

as the cycles $C_0, C_1, \ldots, C_{\Delta-1}$ are disjoint. As such, there must be an index $i = i_0$ of the first summation that does not exceed the average.

### 3.3. Constructing the cycle separator

#### 3.3.1. The algorithm

Let $\Delta = \Theta(\sqrt{n})$ be a parameter to be specified shortly. Let $S$ be a $2/3$-cycle separator, and $r, u, v, p_u$, and $p_v$ as specified by Lemma 3.1. If $|S| \leq 2\Delta$ then this is the desired a short cycle separator. So, assume that $h \geq |S|/2 > \Delta$.

For $j \geq 0$, let $\alpha_j = i_0 + (j - 1)\Delta$ be the index of the $j$th cycle in the small “ladder” of Lemma 3.3. Since $h > \Delta$ and by Lemma 3.2 (D), the cycles $C_{i_0} = C_{\alpha_0}$ of the ladder intersects $S$. In particular, let $D_j = C_{\alpha_j}$, for $j = 1, \ldots, k-1$, be the $j$th nested cycles of this light ladder that intersects $S$. Specifically, let $k$ the minimum value such that $\alpha_k \geq h$. Let $D_0$ be the trivial cycle formed by the root vertex $r$. Similarly, let $D_k$ be the trivial cycle formed only by the vertex $\psi$, such that its interior contains the whole graph.

For $j = 0, \ldots, k$, let $f_j$ be the number of faces in the interior of $D_j$. If for some $j$, we have that $|f/3| \leq f_j \leq \lceil(2/3)f\rceil$, then $D_j$ is the desired separator, as its length is at most $n/\Delta$ by Lemma 3.2, where $f$ is the number of faces of $G$.

Otherwise, there must be an index $i$, such that $f_i < f/3$, and $f_{i+1} > (2/3)f$. Assume, for the sake of simplicity of exposition that $0 < i < k - 1$ (the cases that $i = 0$ or $i = k - 1$ are degenerate and can be handled in a similar fashion to what follows).

Consider the “heavy” ring $R$ bounded by the two of the nested cycles $D_{i+1}$ and $D_i$, see Figure 3.4.

**Observation 3.4.** By Lemma 3.2, the cycles $D_i$ and $D_{i+1}$ each intersects $S$ in two vertices exactly. And $D_i$ is nested inside $D_{i+1}$.
Let $I_i$ and $O_i$ the portions of $D_i$ inside and outside $S$, respectively (define $I_{i+1}$ and $O_{i+1}$ similarly). Let $p_i$ and $q_i$ (resp., $p_{i+1}$ and $q_{i+1}$) be the endpoints of $I_i$ (resp., $I_{i+1}$), such that $p_i$ is adjacent to $p_{i+1}$ along $S$.

We can now partition $R$ into two cycles $R_1$ and $R_2$. The region $R_1$ is bounded by the cycle formed by $J_1 = S[q_i, q_{i+1}] \circ I_{i+1} \circ S[p_{i+1}, p_i] \circ I_i$. The region $R_2$ is bounded by the cycle formed by $J_2 = S[q_i, q_{i+1}] \circ O_{i+1} \circ S[p_{i+1}, p_i] \circ O_i$, see Figure 3.5.

We have that $|J_1| \leq |D_i| + |D_{i+1}| + 2\Delta \leq n/\Delta + 2\Delta$, by Lemma 3.3. In particular, if $f(R_1) \geq f/3$, then $J_1$ is the desired cycle separator, since $f(R_1) \leq f(S) \leq [(2/3)f]$.

Similarly, if $f(R_2) \geq f/3$, then $J_2$ is the desired cycle separator, since $f(R_2) \leq f(S) \leq [(2/3)f]$.

Otherwise, the algorithm returns the cycle $K$ formed by $O_i \circ S[q_i, q_{i+1}] \circ I_{i+1} \circ S[p_{i+1}, p_i]$ as the desired separator.

3.3.2. Analysis

**Lemma 3.5.** Assume that $f(R_1) < f/3$ and $f(R_2) < f/3$. Consider the region $Z$, formed by the union of the interior of $D_i$, together with the interior of $R_1$. Its boundary, is the cycle $K$ formed by $O_i \circ S[q_i, q_{i+1}] \circ I_{i+1} \circ S[p_{i+1}, p_i]$, see Figure 3.6. The cycle $K$ is a $2/3$-cycle separator with $n/\Delta + 2\Delta$ edges.

**Proof:** We have the following: (i) $f_i < f/3$, (ii) $f_i + f(R_1) + f(R_2) = f_{i+1} > (2/3)f$, (iii) $f(R_1) < f/3$, and (iv) $f(R_2) < f/3$. Assume that $f_i + f(R_1) < f/3$. But then $f_{i+1} = f_i + f(R_1) + f(R_2) < (2/3)f$, which is impossible. The region $Z$ bounded by $K$ contains $f_i + f(R_1)$ faces, and we have $f/3 < f_i + f(R_1) < (2/3)f$, which implies the separator property.

As for the length of $K$, observe that $|K| \leq |D_i| + |D_{i+1}| + |S[p_i, p_{i+1}]| + |S[q_i, q_{i+1}]| \leq n/\Delta + 2\Delta$, by Lemma 3.3.
Theorem 3.6. Given an embedded triangulated planar graph \( G \) with \( n \) vertices and \( f \) faces, one can compute, in linear time, a simple cycle \( K \) that is a \( 2/3 \)-separator of \( G \). The cycle \( K \) has at most \( O(1) + \sqrt{8n} \) edges.

This cycle \( K \) also \( 2/3 \)-separates the vertices of \( G \) — namely, there are at most \( (2/3)n \) vertices of \( G \) on each side of it.

Proof: The construction is described above. As for the length of \( K \), set \( \Delta = \lceil \sqrt{n/2} \rceil \). By Lemma 3.5, we have \(|K| \leq 2\Delta + n/\Delta \leq O(1) + \sqrt{2n} + \sqrt{2n} \leq O(1) + \sqrt{8n} \). (The separator cycle is even shorter if one of the other cases described above happens.)

As for the running time, observe that the algorithm runs BFS on the graph several times, identify the edges that form the relevant cycles. Count the number of faces inside these cycles, and finally counts the number of edges in \( R_1 \) and \( R_2 \). Clearly, all this work (with a careful implementation) can be done in linear time.

The second claim follows from a standard argument, see Lemma 3.7 (C) below for details.

3.4. From faces separation to vertices separation

Lemma 3.7. (A) A simple planar graph \( G \) with \( n \) vertices has at most \( 3n - 6 \) edges and at most \( 2n - 4 \) faces. A triangulation has exactly \( 3n - 6 \) edges and \( 2n - 4 \) faces.

(B) Let \( G \) be a triangulated planar graph and let \( C \) be a simple cycle in it. Then, there are exactly \( (f(C) - |C|)/2 + 1 \) vertices in the interior of \( C \), where \( f(C) \) denotes the number of faces of \( G \) in the interior of \( C \).

(C) A simple cycle \( C \) in a triangulated graph \( G \) that has at most \( \lceil (2/3)f \rceil \) faces in its interior, contains at most \( (2/3)n \) vertices in its interior, where \( n \) and \( f \) are the number of vertices and faces of \( G \), respectively.

Proof: (A) is an immediate consequence of Euler’s formula.

(B) Let \( n' \) be the number of vertices of \( G \) in or on \( C \) — delete the portion of \( G \) outside \( C \), and add a vertex \( v \) to \( G \) outside \( C \), and connect it to all the vertices of \( C \). The resulting graph is a triangulation with \( n' + 1 \) vertices, and \( 2(n' + 1) - 4 = 2n' - 2 \) triangles, by part (A). This counts \( |C| \) triangles that were created by the addition of \( v \). As such, \( f(C) = 2n' - 2 - |C| \implies n' = f(C)/2 + 1 + |C|/2 \). The number of vertices inside \( C \) is \( n' - |C| = (f(C) - |C|)/2 + 1 \).

(C) Part (B) implies that number of vertices inside the region formed by the cycle \( C \) is

\[
\frac{(f(C) - |C|)/2 + 1}{2} \leq \frac{[(2/3)f] - |C|}{2} + 1 = \frac{[(2/3)(2n - 4)] - |C|}{2} + 1
\]

\[
\leq \frac{(2/3)(2n - 4) + 1 - |C|}{2} + 1 \leq \frac{2}{3}n,
\]

as claimed.

References

[AST94] N. Alon, P. Seymour, and R. Thomas. Planar separators. SIAM J. Discrete Math., 2(7):184–193, 1994.

[FMPS16] E. Fox-Epstein, S. Mozes, P. M. Phothilimthana, and C. Sommer. Short and simple cycle separators in planar graphs. ACM Journal of Experimental Algorithmics, 21(1):2.2:1–2.2:24, 2016.
A. Balanced edge separator in a low-degree tree

The following lemma is well known, and we provide a proof for the sake of completeness.

Lemma A.1. Let \( T \) be a tree with \( n \) vertices, with maximum degree \( d \geq 2 \). Then, there exists an edge whose removal break \( T \) into two trees, each with at most \( \lceil (1 - 1/d)n \rceil \) vertices. This edge can be computed in linear time.

Proof: Let \( v_1 \) be an arbitrary vertex of \( T \), and root \( T \) at \( v_1 \). For a vertex \( v \) of \( T \) let \( n(v) \) denote the number of nodes in its subtree – this quantity can be precomputed, in linear time, for all the vertices in the tree using BFS.

In the \( i \)th step, \( v_{i+1} \) be the child of \( v_i \) with maximum number of vertices in its subtree. If \( n(v_{i+1}) \leq \lceil (1 - 1/d)n \rceil \), then the algorithm outputs the edge \( xy \) as the desired edge separator, where \( x = v_i \) and \( y = v_{i+1} \). Otherwise, the algorithm continues the walk down to \( v_{i+1} \). Since the tree is finite, the algorithm stops and output an edge.

Assume, for the sake of contradiction, that \( n(y) < n/d \). But then, \( x \) has at most \( d(x) - 1 \leq d - 1 \) children (in the rooted tree), each one of them has at most \( n(y) \) nodes (since \( y \) was the “heaviest” child). As such, we have \( n(x) \leq 1 + (d - 1)n(y) < 1 + (d - 1)n/d \leq \lceil (1 - 1/d)n \rceil \) if \( d \) does not divides \( n \). If \( d \) divides \( n \) then \( n(x) \leq 1 + (d - 1)n(y) \leq 1 + (d - 1)(n/d - 1) = ((d - 1)/d)n + 2 - d \leq \lceil (1 - 1/d)n \rceil \).

Namely, the algorithm would have stopped at \( x \), and not continue to \( y \), a contradiction.

As such, \( n/d \leq n(y) \leq \lceil (1 - 1/d)n \rceil \). But this implies that \( xy \) is the desired edge separator. \( \blacksquare \)