Chapter 5

Separators in planar graphs

5.1 Triangulation

We say a planar embedded graph is *triangulated* if each face's boundary has at most three edges.

Problem 5.1.1. *Provide a linear-time algorithm that, given a planar embedded graph G, adds a set of artificial edges to obtain a triangulated planar embedded graph. Show that the number of artificial edges is at most twice the number of original edges.*

5.2 Weights and balance

Let G be a planar embedded graph, and let α be a number between 0 and 1. An assignment of nonnegative weights to the faces, vertices, and edges of G is an α*-proper* assignment if no element is assigned more than α times the total weight. A subpartition of these elements is α*-balanced* if, for each part, the sum of the weights of the elements of that part is at most α times the total weight.

5.3 Fundamental-cycle separators

Let G be a plane graph. A simple cycle C of G defines a subpartition consisting of two parts, the strict interior and the strict exterior of the cycle. If the subpartition is α-balanced, we say that C is an α*-balanced cycle separator*. The subgraph induced by the (nonstrict) interior, i.e. including C, is one *piece* with respect to C, and the subgraph induced by the (nonstrict) exterior is the other piece.

First we give a result on fundamental-cycle separators that are balanced with respect to an assignment of weights only to *faces*.

Lemma 5.3.1 (Fundamental-cycle separator of faces)**.** *For any plane graph G, $\frac{1}{4}$-proper assignment of weights to faces, and spanning tree T of G such that the* *boundary of each face of G has at most three nontree edges, there is a nontree edge $\hat{e}$ such that the fundamental cycle of $\hat{e}$ with respect to T is a $\frac{3}{4}$-balanced cycle separator for G.*

Proof. Let T^* be the interdigitating tree, the spanning tree of G^* consisting of edges not in T. The vertices of T^* are faces of G, and are therefore assigned weights. The property of T ensures that T^* has degree at most three. Using Lemma 1.3.2, let $\hat{e}$ be an edge separator in T^* of vertex weight. By the Fundamental-Cut Lemma (Lemma 3.2.2), the fundamental cut of $\hat{e}$ in G with respect to T^* is a simple cut in G^* (each side of which has weight at most three-fourths of the total) so it is, by the Simple-Cycle/Simple-Cut Theorem (Theorem 4.6.2), a simple cycle in G that encloses between one-fourth and three-fourths of the weight. □

Assignments of weight to faces, vertices, and edges can also be handled:

Lemma 5.3.2. *There is a linear-time algorithm that, given a triangulated plane graph G, a spanning tree T of G, and a $\frac{1}{4}$-proper assignment of weights to faces, edges, and vertices, returns a nontree edge $\hat{e}$ such that the fundamental cycle of $\hat{e}$ with respect to T is a $\frac{3}{4}$-balanced cycle separator for G.*

Problem 5.3.3. *Prove Lemma 5.3.2 using Lemma 5.3.1*

We can give a better balance guarantee if we are separating just edge-weight.

Lemma 5.3.4 (Fundamental-cycle separator of edges)**.** *There is a linear-time algorithm that, given a triangulated plane graph G with a $\frac{1}{3}$-proper assignment of weights to edges, and a spanning tree T, returns a nontree edge $\hat{e}$ such that the fundamental cycle of $\hat{e}$ with respect to T is a $\frac{2}{3}$-balanced cycle separator for G.*

Problem 5.3.5. *Prove Lemma 5.3.4 by following the proof of Lemma 5.3.1 but using tree edge separators of vertex/edge weight (Lemma 1.3.3) instead of tree edge separators of vertex weight (Lemma 1.3.2).*

5.4 Breadth-first search

Let G be a connected, undirected graph, and let r be a vertex. For $i = 0, 1, 2, \ldots$, we say a vertex v of G has *level i with respect to r* and we define $\text{level}(v) = i$ if i is the minimum number of edges on an r-to-v path in G. (That is, the level of a vertex v is the distance of v from r where the edges are assigned unit length.) For $i = 0, 1, 2, \ldots$, let $L_i(G, r)$ denote the set of vertices having level i.

An edge uv is said to have level i (and we write $\text{level}(uv) = i$) if u has level i and v has level $i + 1$. (An edge whose endpoints have the same level is not assigned a level.)

Breadth-first search from r is a linear-time algorithm that finds

- the levels of vertices and edges, and

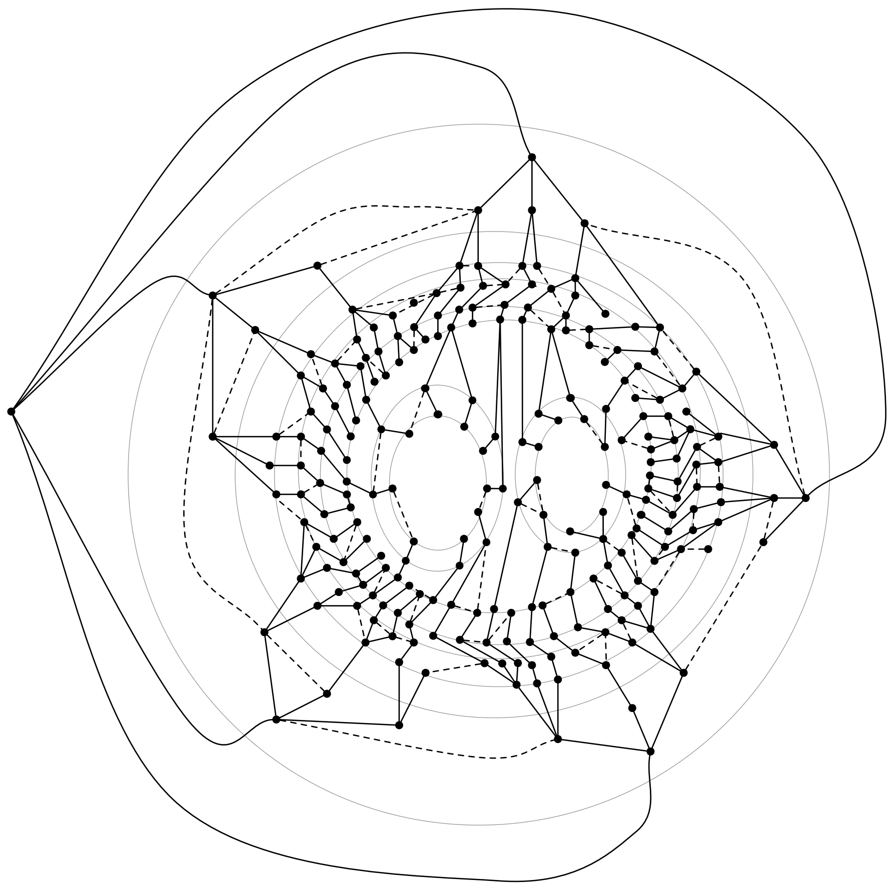

Figure 5.1: Shows the levels of breadth-first search.

- a spanning tree rooted at r such that, for each vertex v other than r, the parent of v has level one less than that of v (a *breadth-first-search tree*).

For brevity, we refer to the levels as *BFS levels* and we refer to the edges of the breadth-first-search tree as *BFS edges*.

5.5 $O(\sqrt{n})$-vertex separator

We use fundamental-cycle separators to prove a fundamental separator result for planar graphs. [3]

Theorem 5.5.1 (Planar-Separator Theorem with Edge-Weights)**.** *There is a linear-time algorithm that, for a plane graph G and $\frac{1}{3}$-proper assignment of weights to edges, returns subgraphs G_1, G_2 such that*

- $E(G_1), E(G_2)$ *is a partition of* $E(G)$*,*

- *The partition* $E(G_1), E(G_2)$ *is* $\frac{2}{3}$*-balanced, and*
- $|V(G_1) \cap V(G_2)| \leq 4\sqrt{|V(G)|}$

The subgraphs G_1, G_2 are called the *pieces*. The set $V(G_1) \cap V(G_2)$ of vertices common to the two subgraphs is called the *vertex separator*.

The Planar-Separator Theorem can be used with an assignment of weight to vertices instead of edges. In this case, in evaluating the resulting balance, we do not count the weight of the vertices in the vertex separator.

Theorem 5.5.2 (Planar-Separator Theorem with Vertex-Weights)**.** *There is a linear-time algorithm that, for a plane graph* G *and* $\frac{1}{3}$*-proper assignment of weights to vertices, returns subgraphs* G_1, G_2 *such that*

- $E(G_1), E(G_2)$ *is a partition of* $E(G)$*,*
- *The subpartition* $V(G_1) - V(G_2), V(G_2) - V(G_1)$ *of* $V(G)$ *is* $\frac{2}{3}$*-balanced, and*
- $|V(G_1) \cap V(G_2)| \leq 4\sqrt{|V(G)|}$

Problem 5.5.3. *Show that the Vertex-Weights version (Theorem 5.5.2) follows easily from the Edge-Weights version (Theorem 5.5.1).*

4

Now we give the proof of the Edge-Weights version. Let $w(\cdot)$ denote the edge-weight assignment. Assume for notational simplicity that the sum of edge-weights is 1. The basic idea is to find a balanced fundamental cycle separator, which might be too long, and shortcut it at small BFS levels.

The algorithm adds artificial zero-weight edges to G to triangulate it. Then, it performs a breadth-first search of G from r. Let T be the breadth-first-search tree. Let V_i denote the set of vertices at level i. Following Lemma 5.3.4, the algorithm finds a $\frac{2}{3}$-balanced fundamental-cycle separator C with respect to T. Let i_{min} and i_{max} denote the minimum and maximum level of a vertex of C, respectively.

For an integer i, let $w_{\leq i}$ denote $w(\{uv : u$ and v both have level $\leq i\}$, and let $w_{>i}$ denote $w(\{uv :$ at least one of u or v have level $> i\}$.

Let i_- be the greatest integer satisfying

- $i_{min} < i_- < i_{max}$
- $|V_{i_-}| \leq \sqrt{n}$
- $w_{\leq i_-} \leq 2/3$

If no such a level exists we set $i_- := i_{min}$. Let i_+ be the smallest integer satisfying

- $i_- < i_+ < i_{max}$

- $|V_{i_+}| \leq \sqrt{n}$
- $w_{>i_+} \leq 2/3$

If no such a level exists we set $i_+ := i_{max}$.

Since for any level i, the sets of edges corresponding to $w_{\leq i}$ and to $w_{>i}$ are disjoint, at least one of $w_{\leq i} \leq 2/3$, $w_{>i} \leq 2/3$ is true. It follows that each of the levels $i_- + 1, i_- + 2, \ldots, i_+ - 1$ has greater than $\sqrt{n}$ vertices. Hence, the total number of vertices among these levels is greater than $(i_+ - i_- - 1)\sqrt{n}$, so $i_+ - i_- - 1 < n/\sqrt{n} = \sqrt{n}$.

Let E_1 be the set of edges whose endpoints have level at most i_- if $i_- > i_{min}$, and the empty set if $i_- = i_{min}$. Let E_2 be the set of edges whose endpoints have level greater than i_+ if $i_+ < i_{max}$, and the empty set if $i_+ = i_{max}$. Let E_3 be the set of edges assigned to the interior of C, and that belong to neither E_1 nor to E_2, and let E_4 be the set of edges assigned to the exterior of C, and that belong to neither E_1 nor to E_2.

By choice of i_+ and i_-, for $j = 1, 2$ we have $w(E_j) \leq 2/3$. Since C is a balanced separator, for $j = 3, 4$ we have $w(E_j) \leq 2/3$.

For $j = 1, 2, 3$, or 4, if $w(E_j) \geq \frac{1}{3}$ then the algorithm sets $G_1 := E_j$ and $G_2 :=$ all other edges of G. Since $\frac{1}{3} \leq w(E_j) \leq \frac{2}{3}$, it follows that $w(E(G_1)) \leq \frac{2}{3}$ and $w(E(G_2)) \leq \frac{2}{3}$.

Assume therefore that $w(E_j) < \frac{1}{3}$ for $j = 1, 2, 3, 4$. Let j be the minimum integer such that $w(E_1) + \cdots + w(E_j) \geq \frac{1}{3}$. Then $w(E_1) + \cdots + w(E_{j-1}) < \frac{1}{3}$ and $w(E_j) < \frac{1}{3}$ so $w(E_1) + \cdots + w(E_j) < \frac{2}{3}$. The algorithm sets $G_1 := E_1 \cup \cdots \cup E_j$ and $G_2 :=$ all other edges of G. Then $w(E(G_1)) \leq \frac{2}{3}$ and $w(G_2) \leq \frac{2}{3}$.

5.5.1 Analysis

We have completed the description of the algorithm, and have ensured the balance property of the separator. It is easy to verify that E_1, E_2, E_3, and E_4 are disjoint, which implies that G_1 and G_2 are edge-disjoint.

The only vertices that are endpoints of edges in distinct sets E_i and E_j are the vertices at level i_-, the vertices at level i_+, and the vertices of the cycle C in the level range (i_-, i_+). There are at most $\sqrt{n}$ in each of the first two categories. We claim that the number in the third category is at most $2(i_+ - i_- - 1)$, which in turn is at most $2\sqrt{n}$. This is because the monotonicity property of levels of vertices on leafward paths in the BFS tree implies that each of the two leafward tree paths comprising C has at most one vertex at each level BFS level. This completes the proof of the theorem.

Problem 5.5.4. *Show how to modify the algorithm so that the size of the separator is at most $c\sqrt{n}$ for some constant $c < 4$.* Hint: *The size criterion used to decide whether a level i qualifies to be designated level i_- can depend on $|i_0 - i_-|$.*

5.6 Biconnectivity

A graph is *biconnected* if every pair of edges belong to some simple cycle.

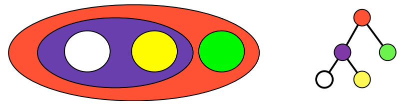

Figure 5.2: The diagram on the left shows a Venn diagram of some noncrossing sets, and the diagram on the right shows the corresponding rooted forest (a tree in this case).

Lemma 5.6.1. *If a planar embedded graph G is biconnected then so is the planar dual G^*.*

Lemma 5.6.2. *If G is a planar embedded biconnected graph then every face is a simple cycle.*

Problem 5.6.3. *Let G be a biconnected planar graph with m edges and face weights summing to 1 such that no face weighs more than $\frac{3}{4}$. Give an $O(m)$ algorithm that finds a balanced simple-cycle separator C in G. (Note there is no size requirement on C.)*

5.7 Noncrossing families of subsets

Two nonempty sets A and B *cross* if they are neither disjoint nor nested, i.e. if $A \cap B \neq \emptyset$ and $A \not\subseteq B$ and $B \not\subseteq A$.

A family $\mathcal{C}$ of nonempty sets is *noncrossing* (also called *laminar*) if no two sets in $\mathcal{C}$ cross.

As illustarted in Figure 5.2, under the subset relation, a noncrossing family $\mathcal{C}$ of sets forms a rooted forest $F_{\mathcal{C}}$. That is, each set $X \in \mathcal{C}$ is a node, and its ancestors are the sets in $\mathcal{C}$ that include X as a subset. To see that this is a forest, let X_1 and X_2 be two supersets of X in $\mathcal{C}$. Since X is nonempty, X_1 and X_2 intersect, so one must include the other. This shows that the supersets of X are totally ordered by inclusion. Hence if X has any proper supersets, it has a unique minimal proper superset, which we take to be the parent of X.

5.8 The connected-components tree

5.8.1 The face-vertex incidence graph

Let G be an embedded graph. A graph is a *face-vertex incidence graph* of G if its vertex set corresponds one-to-one with the union of the vertex set of G and the face set of G, and if, for a vertex v and a face f of G, the vertices corresponding to v and f are adjacent in $FV(G)$ if v occurs on the boundary of f.

Informally, we will refer to *the* face-vertex incidence graph of G since up to isomorphism there is just one.

5.8.2 Vertex and face labels

For a connected triangulated graph G, and a face f_∞, let $FV(G)$ be the face-vertex incidence graph of G. For a face f of G, define its level $\ell(f)$ to be one less than the minimum number of faces of G on a f_∞-to-f path in $FV(G)$. Similarly, for a vertex v of G, define its level $\ell(v)$ to be one less than the minimum number of vertices of G on a f_∞-to-v path in $FV(G)$. Thus, $\ell(f_\infty) = 0$, and all three vertices incident to f_∞ have level 0.

The definition of vertex- and face-labels implies the following lemma:

Lemma 5.8.1. *A vertex v of G with $\ell(v) = i$ is incident to at least one level-i face, at least one level-$(i+1)$ face, and no faces at other levels.*

5.8.3 The connected-components tree

For an integer $i \geq 0$, let F_i^+ denote the set of faces of G with level at least i. Let $\mathcal{K}_i^+$ denote the set of connected components of the subgraph of G^* induced by F_i^+. We refer to a connected component $K \in \mathcal{K}_i^+$ as a *level-i component*, and we define level$(K) = i$.

Lemma 5.8.2. *For any triangulated connected graph G and face f_∞, the components $\bigcup_i \mathcal{K}_i^+$ form a noncrossing family of subsets of $F(G)$.*

Proof. Let K_1 and K_2 be two components. Assume without loss of generality that level$(K_1) \leq$ level(K_2). If any face of K_2 belongs to K_1 then *every* face of K_2 belongs to K_1. □

It follows from Lemma 5.8.2 that the face sets of the components $\bigcup_i \mathcal{K}_i^+$ form a rooted forest with respect to the subset relation.

Lemma 5.8.3. *The parent of a level-i component is a level-$(i-1)$ component if $i > 0$.*

Proof. Let K be a level-i component where $i > 0$. Since the level-i components are disjoint, K is not contained in any other level-i component. Since K must contain some level-i face f of G, K is not contained in any level-j component where $j > i$. Let g be a level $i-1$ face that shares a vertex with f. Then the level-$i-1$ component K' containing g must contain K. Furthermore, for $j \leq i-1$, any level-j component that contains K must also contain K'. This shows that K' is the unique minimal proper superset of K among the components. □

Lemma 5.8.3 shows that a root of the forest of components must have level 0. Since we assume G is connected, there is only one level-0 component, namely the whole graph G^*, so it is the only root, and the forest is in fact a tree. We call it the *component tree*, and we denote it by $\mathcal{T}$

Lemma 5.8.4 (BFS-Component-Tree Construction)**.** *There is a linear-time algorithm to construct the component tree $\mathcal{T}$.*

Problem 5.8.5. *Prove the Component-Tree Construction Lemma.*

Lemma 5.8.6. *The cuts* $\{\delta_{G^*}(K) : K \in \mathcal{K}_i^+, i > 0\}$ *are edge-disjoint.*

Proof. Suppose K is a level-i component. Let fg be an edge of $\delta_{G^*}(K)$ where $g \in K$. Then the level of f is less than i (else f would belong to K). Since faces f and g share a vertex, their level differ by at most one, so level$(f) = i-1$. Therefore, for any component K' such that $fg \in \delta_{G^*}(K')$, we must have level$(K') = i$ and K' must contain g, hence $K' = K$. □

Lemma 5.8.7. *For any component* $K \in \mathcal{K}_i^+$, $\delta_{G^*}(K)$ *is a simple cut.*

Proof. Clearly K is connected in G^*. We need to show that $V(G^*) - K$ is also connected in G^*. We show that, every face $f \in V(G^*) - K$ other than f_∞ is connected, in G^*, to a face g whose level is less than that of f. An induction by level then shows that every vertex in $V(G^*) - K$ is connected to f_∞, and hence that $V(G^*) - K$ is connected.

Let $i > 0$ be the level of f. Consider the vertex v incident to f whose level is $i-1$ (such a vertex must exist by definition of levels with respect to $FV(G)$). By Lemma 5.8.1, v is incident to at least one level $i-1$ face and only to faces at levels $i-1$ or i. Let e be an edge of f incident to v. Let P be a path in G^* consisting of the duals of all edges in the orbit v in order, starting at e^* and ending at the first dual edge whose endpoint is a level-$(i-1)$ face. No vertex of P is in K, or f would be in K as well. Therefore, P shows the required property. □

The simple-cut/simple-cycle theorem (Theorem 4.6.2) implies that for any component K, the edges of $\delta_{G^*}(K)$ form a simple cycle in G whose enclosed set of faces is exactly K. We call such a cycle the *level cycle* associated with K and denote it $X(K)$. If K is a level-i component then we say that $X(K)$ is a level-i cycle.

The definition of face-levels with respect to the face-vertex incidence graph implies that level-cycles at different levels are vertex disjoint.

Lemma 5.8.8. *Let* K, K' *be components at distinct levels.* $X(K)$ *and* $X(K')$ *are vertex disjoint*

Proof. Suppose K is a level-i component, and let v be a vertex of $X(K)$. Thus v is incident to a level-i face and to a level-$(i-1)$ face. By Lemma 5.8.1, any face incident to v has levels i or $i-1$. Hence, if e is an edge of $X(K')$ incident to v, K' must be a level-i component as well. □

5.9 Cycle separators

Theorem 5.9.1 (Planar-Cycle-Separator Theorem)**.** *There is a linear-time algorithm that, for any simple undirected biconnected triangulated plane graph and any* $\frac{3}{4}$*-proper assignment of weights to faces, edges, and vertices, returns a* $\frac{3}{4}$*-balanced cycle separator* C *of size at most* $4\sqrt{n}$.

5

The algorithm is very similar to the vertex separator of Section 1.2.1. One difference is that it finds a fundamental cycle separator in a tree that is monotone with respect to the labeling induced by the face-vertex incidence graph, rather than a BFS labeling. Another crucial difference is that it shortcuts that fundamental cycle at small level-cycles, rather than at small BFS levels.

Assume for notational simplicity that the total weight is 1 and that weight is assigned only to faces. The algorithm first designates some face as f_∞ and computes the component tree $\mathcal{T}$.

5.9.1 The primal tree

The algorithm computes the tree described in the following lemma.

Lemma 5.9.2. *There exists a spanning tree T such that, for any vertex u with $\ell(u) > 0$, $\ell(parent_T(u)) = \ell(u) - 1$. T can be computed in linear time.*

Proof. Let T' be a BFS tree of $FV(G)$, rooted at f_∞, and directed towards the root. T is obtained from T' by contracting all the edges entering nodes of T' that correspond to vertices of G, deleting the three edges of T' entering the root and designating an arbitrary level-0 vertex to be the parent of the other two level-0 vertices.

T is a spanning tree of G since the nodes of T are in one-to-one correspondence with the vertices of G. Also, for nodes $u, v \in T$ such that $u = \text{parent}_T(v)$, the construction of T from T' guarantees that, in G, u and v are incident to the same face. Since G is triangulated, this implies that $uv \in E(G)$. □

5.9.2 Shortcutting a fundamental-cycle separator

Following Lemma 5.3.1, the algorithm finds a $\frac{3}{4}$-balanced fundamental-cycle separator C with respect to T. Let i_{min} and i_{max} denote the minimum and maximum level of a vertex of C, respectively.

We say that an edge uv *penetrates* a component K at u if $u \in X(K)$ and v is strictly enclosed by $X(K)$. A set of edges penetrates K if some edge in the set penetrates K.

Lemma 5.9.3. *An edge uv with $\ell(u) = i - 1, \ell(v) = i$ penetrates exactly one component K. The level of K is i.*

Proof. Since $\ell(u) = i - 1$, and $\ell(v) = i$, the two faces f_1 and f_2 to which uv belongs must have level i. Let K be the unique level-i component to which f_1 and f_2 belong. Since $\ell(u) = i - 1$, $u \in X(K)$. Since $\ell(v) = i$, v is strictly enclosed by $X(K)$.

We next show that K is the unique such component. Since the level-i components are disjoint, v is in no other level-i component. Since the level of u is $i-1$, u is in no level-j component for $j > i$. For any level-j component K' with $j < i$, The vertices of $X(K')$ have level $j - 1$, so $u \notin X(K')$. □

Lemma 5.9.4. *For every integer $i \in (i_{min}, i_{max})$, C penetrates exactly one level-i component K_i, and it does so at exactly two distinct vertices of $X(K_i)$.*

Proof. Let uw be the non-tree edge of C. Let v be the least common ancestor of u and w in T. The cycle C consists of the edge uw, of a v-to-u leafward paths P_1, and a v-to-w leafward path P_2. The level of v is $\ell(v) = i_{min}$. The level of u is $\ell(u) = i_{max}$, and the level of w is either i_{max} or $i_{max}-1$. By the monotonicity of labels along leafward paths of T, and by Lemma 5.9.3, each of these paths penetrates exactly one component at every level in (i_{min}, i_{max}). Let K_{max} be the level i_{max}-component penetrated by P_1. That is, u is strictly enclosed by $X(K_{max})$. Since the edge uw exists, w is also enclosed by $X(K_{max})$. It follows that P_1 and P_2 penetrate the same components. Since P_1 and P_2 are vertex disjoint (except at their start vertex v), C penetrates each of these components at exactly two distinct vertices. □

Let K_i denote the unique level-i component penetrated by C. For convenience, let X_i denote $X(K_i)$. Let $w(X_i)$ denote the weight of faces enclosed by X_i (that is, the weight of the component corresponding to X_i), and let $\bar{w}(X_i)$ denote the weight of faces not enclosed by X_i. Let i_- be the greatest integer satisfying

- $i_{min} < i_- < i_{max}$
- $|V(X_i)| \leq \sqrt{n}$
- $\bar{w}(X_i) \leq 3/4$

If no such a level exists we set $i_: = i_{min}$. Let i_+ be the smallest integer satisfying

- $i_- < i_+ < i_{max}$
- $|V(X_i)| \leq \sqrt{n}$
- $w(X_i) \leq 3/4$

If no such a level exists we set $i_+ := i_{max}$.

Since for any level i, the sets of faces enclosed and not enclosed by X_i are disjoint, at least one of $w(X_i) \leq 3/4$,$\bar{w}(X_i) \leq 3/4$ is true. It follows that each of the levels $i_-+1, i_-+2, \ldots, i_+-1$ has greater than $\sqrt{n}$ vertices. By Lemma 5.8.8 all these level-cycles are vertex disjoint, so the total number of vertices among these levels cycle is greater than $(i_+ - i_- - 1)\sqrt{n}$, so $i_+ - i_- - 1 < n/\sqrt{n} = \sqrt{n}$.

Let F_1 be the set of faces enclosed by X_{i_+} if $i_+ < i_{max}$, and the empty set if $i_+ = i_{max}$. Let F_4 be the set of faces of G not enclosed by X_{i_-} if $i_- > i_{min}$, and the empty set if $i_- = i_{min}$. Let F_2 be the set of faces enclosed by C, and that belong to neither F_1 nor to F_4, and let F_3 be the set of faces not enclosed by C, and that belong to neither F_1 nor to F_4.

By choice of i_+ and i_-, for $j = 3, 4$ we have $w(F_j) \leq 3/4$. Since C is a balanced separator, for $j = 1, 2$ we have $w(F_j) \leq 3/4$. Let j be such that $w(F_j) \geq \frac{1}{4}$. Let C^* be the boundary of F_j. That is, C^* is the set of edges that belong to exactly one face in F_j. The algorithm returns C^*.

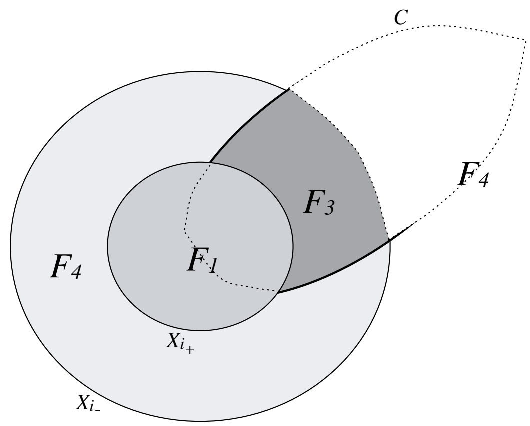

Figure 5.3: This figure shows the cycle separator C penetrating the vertex disjoint level-cycles X_{i_-} and X_{i_+}, each at two distinct nodes. The sets of faces F_j are also indicated. Their boundaries are simple cycles.

5.9.3 Balanced, short, and simple cycle

Lemma 5.9.5. *C^* is a simple cycle*

Proof. If $j = 1$ then C^* is the simple cycle X_{i_+}. If $j = 4$ then C^* is the simple cycle X_{i_-}. By lemma 5.9.4, C intersect each of $X(i_-)$ and $X(i_+)$ exactly twice. Thus, each of X_{i_-} and X_{i_+} can be decomposed into two paths, one enclosed by C and the other not enclosed by C. If $j = 2$ then C^* is the simple cycle formed by the edges of C enclosed by X_{i_-} but not by X_{i_+} and by the edges of X_{i_-} and X_{i_+} enclosed by C. If $j = 3$ then C^* is the simple cycle formed by the edges of C enclosed by X_{i_-} but not by X_{i_+} and by the edges of X_{i_-} and X_{i_+} not enclosed by C. See Figure 5.3 for an illustration. □

By choice of j, C^* is a $\frac{3}{4}$-balanced cycle separator. It remains to show that C^* consists of fewer than $4\sqrt{n}$ edges. By choice of i_- and i_+, $|V(X_{i_-})| \leq \sqrt{n}$. The monotonicity property of levels of vertices on leafward paths in T implies that each of the two leafward tree paths comprising C has at most one vertex at each level. Combining this with the fact that $i_+ - i_- \leq \sqrt{n}$, we get that the number of edges of C enclosed by X_{i_-} and not enclosed by X_{i_+} is $2\sqrt{n}$. Thus C^* has fewer than $4\sqrt{n}$ edges. This completes the proof of Theorem 5.9.1.

5.10 Division into regions

6

For constants c_1, c_2, an r-*division* of an m-edge graph G is a decomposition of G into at most $c_1 m/r$ edge-disjoint subgraphs $G_1, \ldots, G_k$, each consisting of at most r edges, such that $|\partial_G(G_i)| \leq c_2\sqrt{r}$. The subgraphs are called *regions*. For each region G_i, the vertices of $\partial_G(G_i)$ are called the *boundary vertices of region* G_i, and $\partial_G(G_i)$ is called the *boundary*.

We show in this section that an r-division can be found by using calls to a procedure for finding an $O(\sqrt{n})$-vertex separator.

Theorem 5.10.1. *There is an $O(n \log n)$ algorithm that, given a planar embedded graph G and a positive integer r, computes an r-division.*

5.10.1 Concave function

In our analysis, we use *concave functions*, which we define in terms of weighted averages. Any number $0 \leq \mu \leq 1$ defines a weighted average of a pair x, y of numbers: $\mu x + (1-\mu)y$. A continuous function $f(\cdot)$ defined on an interval is *concave* if, for every number $0 \leq \mu \leq 1$,

$$f(\mu x + (1-\mu)y) \geq \mu f(x) + (1-\mu) f(y) \tag{5.1}$$

That is, the weighted average of the images of x and y under f is at most the image under f of the weighted average of x and y.

Fact 5.10.2. *If the second derivative of a function is nonpositive over some interval then the function is concave over that interval.*

For example, a little calculus shows that $\log x$ and $\sqrt{x}$ are monotone nondecreasing and concave.

5.10.2 Phase One: Finding regions with on-average small boundaries

The r-division algorithm consists of two phases. In Phase One, the algorithm finds a decomposition of G into $O(|E(G)|/r)$ regions such that the average number of boundary vertices per region is $O(\sqrt{r})$. Phase One consists of processing the input graph using the procedure RECURSIVEDIVIDE:

```
def RecursiveDivide(H, r):
  if |E(G)| ≤ r then return {H}
  else
    assign equal weight to all edges of H
    find a vertex separator and let H_1, H_2 be the pieces
    return RecursiveDivide(H_1, r) ∪ RecursiveDivide(H_2, r)
```

Lemma 5.10.3. *After Phase One, the number of regions is at most $2|E(G)|/r$.*

The proof is left as an exercise.

For each vertex v, let $b(v)$ be one less than the number of regions containing v.

Lemma 5.10.4. *After* RecursiveDivide *is applied to an* m*-edge graph* G*,* $\sum_{v \in V(G)} b(v) \leq \frac{18m}{\sqrt{r}}$.

Proof. Let $B(m)$ be the maximum of $\sum_{v \in V(G)} b(v)$ over all m-edge graphs. The Planar-Separator Theorem applied to an m-edge graph G guarantees that the fraction of edges in each of piece lies in the interval $[\frac{1}{3}, \frac{2}{3}]$, and that the separator has size at most $4\sqrt{m}$. Therefore $B(\cdot)$ satisfies the recurrence

$$B(m) = 0 \text{ for } m \leq r$$
$$B(m) \leq 4\sqrt{m} + \max_{1/3 \leq \alpha \leq 2/3} B(\alpha m) + B((1-\alpha)m) \text{ for } m > r$$

For a constant c to be determined, we show by induction that $B(m) \leq \frac{18m}{\sqrt{r}} - c\sqrt{m}$ for $m \geq r/3$. Let us postpone the basis until c is determined. To show the induction step, suppose $m > r$. We show that, for any α in the range $[1/3, 2/3]$,

$$4\sqrt{m} + B(\alpha m) + B((1-\alpha)m) \leq \frac{18m}{\sqrt{r}} - c\sqrt{m}.$$

By the inductive hypothesis,

$$\begin{aligned} B(\alpha m) &\leq \frac{18\alpha m}{\sqrt{r}} - c\sqrt{\alpha m} \\ B((1-\alpha)m) &\leq \frac{18(1-\alpha)m}{\sqrt{r}} - c\sqrt{(1-\alpha)m} \end{aligned}$$

so

$$\begin{aligned} & 4\sqrt{m} + B(\alpha m) + B((1-\alpha)m) && (5.2) \\ \leq\ & 4\sqrt{m} + \frac{18\alpha m}{\sqrt{r}} - c\sqrt{\alpha m} + \frac{18(1-\alpha)m}{\sqrt{r}} - c\sqrt{(1-\alpha)m} \\ =\ & 4\sqrt{m} + \frac{18m}{\sqrt{r}} - c\sqrt{m}\left(\sqrt{\alpha} + \sqrt{1-\alpha}\right) && (5.3) \end{aligned}$$

In order to select a value of c for which the induction step can be completed, we must show that $\sqrt{\alpha} + \sqrt{1-\alpha} > 1$. Let $f(x) = \sqrt{x} + \sqrt{1-x}$. Taking the second derivative shows that $f(x)$ is a concave function for $0 < x < 1$. For any $\frac{1}{3} \leq \alpha \leq \frac{2}{3}$, we can write

$$\alpha = \mu\frac{1}{3} + (1-\mu)\frac{2}{3}$$

and solve for μ. Since the resulting value of μ is between 0 and 1, we can apply Inequality 5.1 to obtain

$$f(\alpha) \geq \mu f(1/3) + (1-\mu)f(2/3)$$

Since a weighted average of two numbers is at least the minimum,

$$\mu f(1/3) + (1-\mu)f(2/3) \geq \min\{f(1/3), f(2/3)\}$$

Since $f(1/3) = f(2/3) = 1.39...$, we have shown $\sqrt{\alpha} + \sqrt{1-\alpha} \geq 1.39....$ We set $c = 4/0.39$, for then $c\sqrt{m}(\sqrt{\alpha} + \sqrt{1-\alpha}) - 4\sqrt{m} \geq c\sqrt{m}$, which shows that the right-hand side of Inequality 5.3 is bounded by $\frac{18m}{\sqrt{r}} - c\sqrt{m}$, completing the induction step.

Finally, for the case $r/3 < m \leq r$, we have $B(m) = 0$, so we require only that $0 \leq \frac{18m}{\sqrt{r}} - c\sqrt{m}$. Setting $m = rx$, we require that $18\sqrt{r}x - c\sqrt{r}\sqrt{x} \geq 0$ for $x > 1/3$. The function $f(x) = c\sqrt{x} - 18x$ is monotone decreasing for $x \geq 1/3$. Since it is nonpositive for $x = 1/3$, it is therefore nonpositive for all $x \geq 1/3$. This proves the base case. □

5.10.3 Phase Two: Splitting small regions into small regions with small boundaries

The result of Phase One is a decomposition of the input graph G into at most $2|E(G)/r$ regions of size at most r. Lemma 5.10.4 shows that the regions have small boundaries *on average* but Phase Two is responsible for ensuring that each region has a small boundary. Phase Two resembles Phase One; the difference is that weight is assigned to boundary vertices instead of to all edges. Phase Two is as follows: for each region G resulting from Phase One, call SPLIT on G and B where B is the set of vertices of G that also appear in other regions.

def $\text{SPLIT}_r(G, B)$:
  if $|B| \leq 16\sqrt{r}$ then return $\{G\}$
  else
    assign equal weight to all vertices in B (others are assigned zero weight)
    find a vertex separator S and let G_1, G_2 be the pieces
    return $\text{SPLIT}_r(G_1, (B \cup S) \cap V(G_1)) \cup \text{SPLIT}_r(G_2, (B \cup S) \cap V(G_2))$

Lemma 5.10.5. *A call* $\text{SPLIT}_r(G, B)$ *creates at most* $\max\{0, \frac{|B|}{8\sqrt{r}} - 2\}$ *new regions.*

Proof. Let $R(k)$ be the maximum number of new regions created upon invoking $\text{SPLIT}_r(G, B)$ where $|B| = k$. We show by induction that $R(k) \leq \max\{0, \frac{k}{8\sqrt{r}} - 2\}$.

For the base case, if $|B| \leq 16\sqrt{r}$ then no additional regions are created. Assume therefore that $|B| > 16\sqrt{r}$. In this case there are two recursive calls, $\text{SPLIT}_r(G_1, B_1)$ and $\text{SPLIT}_r(G_2, B_2)$ where $B_i = (B \cup S) \cap V(G_i)$.

We write $|B \cap V(G_1)| = \alpha|B|$ and $|B \cap V(G_2)| = (1-\alpha)|B|$. The separator's size guarantee ensures that $|S| \leq 4\sqrt{r}$. Hence $|B_1| \leq \alpha|B| + 4\sqrt{r}$ and $|B_2| \leq (1-\alpha)|B| + 4\sqrt{r}$. The separator's balance condition ensures that $\frac{1}{3} \leq \alpha \leq \frac{2}{3}$. This combined with the fact that $|B| > 16\sqrt{r}$ implies that $|B_i| < |B|$ for $i = 1, 2$.

Hence, by the inductive hypothesis, the number of new regions created by the call $\text{SPLIT}_r(G_1, B_1)$ is at most $\max\{0, \frac{|B_1|}{8\sqrt{r}} - 2\}$, which is in turn at most

$\frac{\alpha|B|+4\sqrt{r}}{8\sqrt{r}} - 2$. Similarly, the number created by the call $\text{Split}_r(G_2, B_2)$ is at most $\frac{(1-\alpha)|B|+4\sqrt{r}}{8\sqrt{r}} - 2$. Since the call $\text{Split}_r(G, B)$ directly created one new region, the total number of new regions created is at most

$$1 + \frac{\alpha|B| + 4\sqrt{r}}{8\sqrt{r}} - 2 + \frac{(1-\alpha)|B| + 4\sqrt{r}}{8\sqrt{r}} - 2$$

which is at most $\frac{|B|}{8\sqrt{r}} - 2$, proving the induction step. □

Suppose Phase One and Phase Two are executed on a graph with m edges. After Phase One, the sum over all regions R of the number of vertices in R that also appear in other regions is at most $\sum_v 2b(v)$. By Lemma 5.10.4, this sum is at most $\frac{36m}{\sqrt{r}}$. By Lemma 5.10.4, the number of new regions introduced by Phase Two is therefore at most $5\frac{m}{r}$. By Lemma 5.10.3, the number of regions resulting from Phase One is at most $2m/r$. Hence the total number of regions after Phase One and Phase Two is at most $7m/r$. Since Phase Two ensures that each region has at most $16\sqrt{r}$ boundary vertices, the resulting set of regions form an r-division.

Now we consider the running time. Traditional analysis of Phase One, using the fact that finding a separator in an m-edge graph, shows that it runs in $O(m \log m)$ time. The analysis of Lemma 5.10.5 shows that calling $\text{Split}_r(G, B)$ results in at most $\max\{0, \frac{|B|}{8\sqrt{r}} - 1\}$ calls to the vertex-separator algorithm; as in the analysis of new regions, by Lemma 5.10.4 the total number of such calls for all of Phase Two is at most $\frac{m}{r}$. Each call is on a graph consisting of at most r edges, so the total time for Phase Two is $O(m)$. We have proved Theorem 5.10.1.

5.11 Recursive divisions

Let $\bar{r} = (r_0, r_1, \ldots)$ be an increasing sequence of positive integers. We say the *height* of a graph G with respect to $\bar{r}$ is the smallest integer i such that the graph has at most r_i arcs. For a fixed sequence $\bar{r}$, we denote the height of G by $\text{height}(G)$.

A *recursive $\bar{r}$-division* of a nonempty graph G is a rooted tree whose vertices are subgraphs of G, defined inductively as follows. The root of the tree is the graph G. If G has one arc, the root has no children. Otherwise, the children of the root are the regions $G_1, \ldots, G_k$ forming an $r_{\text{height}(G)-1}$-division of G. Moreover, each child is the root of a recursive $\bar{r}$-division. The parent of a region R is denoted by $\text{parent}(R)$. The region consisting of a single arc uv is denoted $R(uv)$. Such a region is said to be *atomic*.

5.12 History

Ungar [**?**] gave the first separator theorem for planar graph. His separator had $O(\sqrt{n} \log^{3/2} n)$ vertices. Lipton and Tarjan [Lipton and Tarjan, 1979] proved the first $O(\sqrt{n})$ separator theorem for planar graphs. Constant-factor Improvements were found by Djidjev [Djidjev, 1981] and Gazit [Gazit, 1986]. Goodrich [Goodrich, 1995] gave a linear-time algorithm to find a recursive decomposition using planar separators. Miller [Miller, 1986] proved the first cycle-separator theorem for planar graphs. A constant-factor improvement was found by Djidjev and Venkatesan [Djidjev and Venkatesan, 1997]. The notion of an r-division is due to Frederickson [Frederickson, 1987], who gave the algorithm for finding one.