Vertex Separators and low tree-width k-coloring

Lectures 11 and 12, January 12 and 19, 2012

1 A theorem about vertex separators

Given a graph $G(V,E)$ and a set of vertices $S \subset V$, an *S-flap* is the set of vertices in a connected component of the graph induced on $V \setminus S$. A set S is a *vertex separator* if no S-flap has more than $n/2$ vertices.

Lipton and Tarjan showed that every planar graph has a separator of size $O(\sqrt{n})$. This was later generalized by Gilbert, Hutchinson and Tarjan to any graph embeddable on a surface of bounded genus. This was further generalized by Alon, Seymour and Thomas to any family of graphs that excludes some fixed (arbitrary) subgraph H as a minor. Their proof (like all previous proofs) is constructive – it provides a polynomial time algorithm that finds the desired separator.

Theorem 1 *There a polynomial time algorithm that given a parameter h and an n vertex graph $G(V,E)$ either outputs a K_h minor, or outputs a vertex separator of size at most $h\sqrt{hn}$.*

Theorem 1 does not completely subsume the previous results on separators, because there are some differences regarding the leading constant in the $O(\sqrt{n})$ size bound, and differences in the running time of the algorithms (the algorithm of Lipton and Tarjan runs in linear time). Moreover, in some cases one would like the separators to have certain structure (e.g., be a cycle in a planar graph), and Theorem 1 does not guarantee this. We remark that it is an open question whether the size bound in Theorem 1 can be improved (possibly to $h\sqrt{n}$).

2 Proof of the vector separator theorem

Let X be a proposed vertex separator. Let Y be the largest connected component that remains after removing X. We shall only consider vertex separators X satisfying the following *key properties*:

1. $|X| \leq h\sqrt{hn}$. More specifically, $X \subset \cap_{i=1}^{k} C_i$ for C_i as defined shortly.
2. For some $k \leq h$, there are k disjoint sets of vertices $C_1, \dots C_k$ in $V \setminus Y$ such that:
   (a) For every $1 \leq i \leq k$, the subgraph induced on C_i is connected.
   (b) For every $1 \leq i \leq k$, $0 < |X \cap C_i| \leq \sqrt{hn}$.

(c) If $k \geq 2$, then for every $1 \leq i < j \leq k$ there is an edge between some vertex of C_i and some vertex of C_j.

We present an iterative algorithm that progresses over a sequence of such X candidates, until either one is found for which the corresponding Y satisfies $|Y| \leq n/2$, or $k = h$, in which case the respective $\{C_i\}$ form an K_h minor. Our measure of progress in the construction will be $P^t = 3|Y^t| + |X^t| + k^t$, where superscript t is the iteration number. We show that if both $k^t < h$ and $|Y^t| > n/2$ the necessarily $P^{t+1} < P^t$. As P^t is integer and nonnegative, it must be that eventually either $k = h$ or $|Y| \leq n/2$.

Initially, X^1 can be taken as an arbitrary vertex $v \in V$, one can take $k = 1$ and $C_1 = \{v\}$. All key properties above hold. We consider now the inductive step, in which X^t, Y^t, k^t and $C_1^t, \ldots C_{k^t}^t$ are given, with $|Y^t| > n/2$ and $k^t < h$.

Suppose that for some i, C_i^t has no neighbor in Y^t. In this case, let $X^{t+1} = X^t \setminus C_i^t$, $Y^{t+1} = Y^t$, $k^{t+1} = k^t - 1$, and the C_j^{t+1} are the same as C_j^t except that C_i^t is removed. Observe that indeed X^{t+1} separates $Y^{t+1} = Y^t$ from the rest of the graph, because X^t did, and $(N(Y^{t+1}) \cap X^t) = (N(Y^{t+1}) \cap X^{t+1})$ (where for $T \subset V$, $N(T)$ denotes the set of neighbors of T in G). Necessarily $P^{t+1} < P^t$, because $k^{t+1} < k^t$ whereas $|X^{t+1}| \leq |X^t|$ and $|Y^{t+1}| \leq |Y^t|$.

Hence it remains to deal with the case that for every i, C_i^t has a neighbor in Y^t. For every $1 \leq i \leq k^t$, let $A_i = (N(C_i^t) \cap Y^t)$. All A_i are nonempty, and they may intersect. We shall now use the following lemma.

Lemma 2 *Let $G(V, E)$ be a connected graph on n vertices and let for $1 \leq i \leq k$ let A_i be subsets of V. Then for every r one of the following two alternatives must hold:*

1. *There is a subgraph C which is connected and intersects all A_i, with $|C| \leq r$.*
2. *There is a set S of vertices such that no S-flap intersects all the A_i, and moreover, $|S| \leq kn/r$.*

Moreover, there is a polynomial time algorithm that finds either C or S as above.

Proof: For intuition, consider first the special case where $|A_i| = 1$ for every i, but one does not simply want to pick $S = A_i$ in alternative 2. Then we could check whether $d(A_i, A_{i+1}) \leq r/k$ for every $i < k$. If so, concatenation of the respective paths gives C (the first alternative). If not, then for some i, r/k levels of BFS from A_i do not suffice in order to reach A_{i+1}. One of these levels is of size at most nk/r, and it separates A_i from A_{i+1}. This gives S (the second alternative). This argument does not prove the lemma when the A_i are sets, because paths between sets cannot be concatenated (the end vertex of the path $A_{i-1} - A_i$ might not coincide with the starting vertex of $A_i - A_{i+1}$). Hence we modify the above argument.

One may assume that $k \geq 3$, because for $k \leq 2$ the above proof works. Construct a graph $G'(V', E')$ with $k - 1$ blocks. Each block is a copy of G, where vertex v^i in block i denotes the ith copy of $v \in V$. For $2 \leq i \leq k - 1$, for every vertex v^i with $v \in A_i$, include the edge (v^i, v^{i-1}).

For a vertex $u \in V'$, let $d(u)$ be the length of the shortest path in G' from u to vertices of A_1 in the first block. There are two alternatives.

1. There is some vertex u of A_k in the last block with $d(u) \leq r$. The path of length r certifying this value of d necessarily goes through all A_i (it starts at A_1, ends at A_k, and crosses to layer i through A_i, for $2 \leq i \leq k-1$). This is the first alternative in the lemma.

2. For every vertex u of A_k in the last block, $d(u) > r$. By comparing with the average, there is some value $j \leq r$ such that the set of vertices with $d(u) = j$ contains at most $(k-1)n/r$ vertices. This vertex set S' separate A_1 of the first block from A_k of the last block in G'. Consequently, the vertices of V represented in S' form S in the second alternative in the lemma. (If there were an S-flap with $a_1 \in A_1$, $a_2 \in A_2$, ..., $a_k \in A_k$, then a tour visiting these vertices could also be followed in $V' \setminus S'$.)

□

We now apply Lemma 2 on Y^t and the associated A_i from above, and with $r = \sqrt{hn}$. There are two cases to consider.

The first alternative holds. In this case, take $X^{t+1} = X^t \cup C$ (for C as in the first alternative in the lemma), $k^{t+1} = k^t + 1$, $C_i^{t+1} = C_i^t$ for $i \leq k^t$ and $C_{k^t+1}^{t+1} = C$, and Y^{t+1} to be the largest connected component of $Y^t \setminus C$ (if any of them is larger than $n/2$ – otherwise we are done). Observe that $|Y^{t+1}| \leq |Y^t| - |C|$ and hence indeed $P^{t+1} < P^t$.

Only the second alternative holds (in particular implying that $k^t \geq 2$). Intuitively, we would like X^{t+1} to be $X^t \cup S$ (for S as in the second alternative in Lemma 2), but this would cause a problem with key property 2. Hence we will do something slightly different.

Let Y' be the largest S-flap in Y^t. We may assume that $|Y'| > n/2$, as otherwise X^{t+1} as above is indeed a separator of size at most $(k^t+1)\sqrt{hn} \leq h\sqrt{hn}$. By Lemma 2, for some $i \leq k^t$, A_i is disjoint from Y'. This A_i allows us to extend the corresponding C_i^t into Y^t while maintaining C_i^t connected. We extend C_i^t into Y^t as much as possible, but without invading Y'. This will be the new C_i^{t+1}. As Y^t was connected, necessarily C_i^{t+1} extends into S. Now let $X^{t+1} = (X^t \setminus C_i^t) \cup (S \cap C_i^{t+1})$. Observe that this X^{t+1} satisfies key properties 1 and 2 above, with $k^{t+1} = k^t$. Let us now analyze the corresponding Y^{t+1}. As Y' is still connected and $|Y'| > n/2$ we have that Y^{t+1} is the connected component containing Y' in $G \setminus X^{t+1}$. The crucial observation is that this connected component cannot contain any vertex from C_i^{t+1}, because any path from C_{t+1} to Y' must go through S, and its intersection with S must be in X^{t+1}. Consequently, Y^{t+1} is entirely contained in Y^t (it cannot invade X^t because the only vertices of X^t not in X^{t+1} are in C_i^{t+1}). It follows that

$$P^{t+1} = 3|Y^{t+1}| + |X^{t+1}| + k^{t+1} \leq 3(|Y^t| - |S \cap C_i^{t+1}|) + |X^t| + |S \cap C_i^{t+1}| + k^t < P^t$$

as desired.

3 Extensions and applications

Theorem 1 has several useful corollaries.

Corollary 3 *Let $G(V,E)$ be an arbitrary graph with no K_h minor, and let $W \subset V$. Then one can find in polynomial time a set S of at most $h\sqrt{hn}$ vertices such that every S-flap contains at most $|W|/2$ vertices from W.*

Proof: The proof given for Theorem 1 can easily be adapted to this setting. □

Corollary 4 *Every graph with no K_h as a minor has treewidth $O(h\sqrt{hn})$. Moreover, a tree decomposition with this treewidth can be found in polynomial time.*

Proof: We have seen last week an algorithm that given a graph of treewidth p constructs a tree decomposition of treewidth $8p$. Using Corollary 3, that algorithm can be modified to give a tree decomposition of treewidth $8h\sqrt{hn}$ in our case, and do so in polynomial time. (The leading constant of 8 can be reduced with extra care.) □

It is known that computing maximum independent sets in planar graphs is NP-hard. As an easy consequence of Corollary 4, maximum weighted independent set can be solved in time $2^{O(\sqrt{n})}$ in planar graphs, and more generally, in graphs with excluded minors. In general graphs, no algorithm with running time $2^{o(n)}$ is known.

Rather than spend superpolynomial time and solve an optimization problem exactly, one is sometimes interested in polynomial time algorithms that solve problems approximately. The following corollary is useful for this purpose.

Corollary 5 *In every n-vertex graph with no K_h-minor and for every k, one can find in polynomial time a set S of vertices with $|S| \leq O(hn\sqrt{h/k})$ such that no S-flap contains more than k vertices.*

Proof: For simplicity (and with no loss of generality, up to a choice of constants in the O notation), assume that n and k are both powers of 2. We shall prove a bound of $|S| \leq 4hn\sqrt{h/k}$. By Theorem 1, the statement holds for $k = n/2$. We shall now assume the Corollary for $2k$, and prove for k. Hence we already have S' of size at most $4hn\sqrt{h/2k}$ for which no S'-flap has size more than $2k$. Consider now only those S'-flaps of size at least k. There are at most n/k such S'-flaps. For each of them use Theorem 1 to find a vertex separator of size at most $h\sqrt{hk}$. The combination of S' and all these new separators is the desired S. Its size is at most $4hn\sqrt{hn/2k} + h\sqrt{hk}\frac{n}{k} = (\frac{4}{\sqrt{2}} + 1)hn\sqrt{h/k} \leq 4hn\sqrt{h/k}$, as desired. □

Observe that every n-vertex planar graph has an independent set of size at least $n/4$ by the 4-color theorem. More generally, the average degree of any graph with no K_h minor is known to be bounded as a function of h, and hence when fixing h (and treating it as a constant) every n vertex graph with no K_h minor has an independent set of size $\Omega(n)$. Applying Corollary 5 with $k = \log n$ and discarding the vertices of S, at most $|S|$ vertices of the maximum independent set are lost. On the small components that remain, a maximum independent set in each of them can be found by exhaustive search in time proportional to $2^k \leq n$. The union of these independent sets is by itself an independent set, of size equal to the maximum independent set minus $O(n/\sqrt{\log n})$, which is a low order term.

The above argument can be strengthened by taking $k = (\log n)^2$, and using also Corollary 4 to find maximum independent sets in components of size $(\log n)^2$. Hence we have the following corollary.

Corollary 6 *For every fixed h there is a polynomial time algorithm that given any graph G on n vertices with no K_h minor finds an independent set of size $(1 - O(1/\log n))\alpha(G)$, where $\alpha(G)$ is the size of the maximum independent set in G.*

4 Low treewidth k-coloring

Observe that Corollary 6 was stated only for maximum cardinality independent set, and not for maximum weight independent set (in graphs in which vertices have nonnegative weight). This is because for the separator vertices all that we argued was that their number is small, rather than that their weight is small. We now develop a different algorithmic paradigm that is in many cases easier to apply than the vertex separator framework. In particular, it gives an $1 - O(1/\log n)$ approximation to maximum weight independent set in planar graphs.

The algorithmic paradigm is based on the following Theorem of DeVos, Ding, Oporowski, Sanders, Reed, Seymour and Vertigan.

Theorem 7 *For every graph H and every k, there is an integer p such that the vertex set of every graph $G(V,E)$ that does not contain H as a minor can be partitioned into k sets $V_1, \ldots, V_k$ such that for every $1 \le i \le k$, the graph induced on $V \setminus V_i$ has treewidth at most p. Moreover, such a partition can be found in polynomial time.*

The proof of Theorem 7 uses structural properties of graphs with excluded minors, and is beyond the scope of the course. Instead, we shall prove the theorem in the interesting special case that G is planar. This (in slightly different formulation) is a result of Brenda Baker that predated (and motivated) Theorem 7.

Theorem 8 *For every k, the vertex set of every planar graph $G(V,E)$ can be partitioned into k sets $V_1, \ldots, V_k$ such that for every $1 \le i \le k$, the graph induced on $V \setminus V_i$ has treewidth at most $3(k-1)$. Moreover, such a partition and the associated tree decompositions can be found in polynomial time.*

Before we prove Theorem 8 we provide some definitions. A planar graph is *outerplanar* if it can be embedded in the plane with all vertices on the outer face. It is k-outerplanar if it can be embedded in the plane in a way that becomes $(k-1)$-outerplanar after all vertices of the outer face are removed (together with their edges). Given an embedding of a planar graph in the plane, a vertex is said to be in layer i if an iterative procedure that in each step removes the vertices of the current outer face would remove it in step i.

Throughout the proofs we assume that a particular embedding of the planar graph is given, and use the terms outerplanar, k-outerplanar and layer as referring to this particular embedding.

Lemma 9 *In a planar graph, for every $1 \le i < j - 1$, there cannot be an edge between a vertex v in layer i and a vertex u in layer j.*

Proof: If v is at layer i and has an edge to u, then after the removal of layer i, vertex u is visible from the outer face (along the corridor opened by the removal of edge (v,u)), and hence the layer of u is at most $i + 1 < j$. $\square$

We now prove of Theorem 8:

Proof: Consider an arbitrary planar embedding of G. For $1 \le i \le k$, let V_i be all the vertices in layers of the form $kN + i$ for integer N. Removing V_i, the graph decomposes into connected components (by Lemma 9), where each component is $(k-1)$-outer-planar. The proof of Theorem 8 now follows from Lemma 12 below. □

Before proving Lemma 12 we shall need two propositions.

Proposition 10 *Every k-outerplanar graph is a minor of a k-outerplanar graph of degree at most 3.*

Proof: Consider an arbitrary vertex v of degree $d > 3$ at layer i and number its edges clockwise (with respect to the layout of the planner embedding). Let us first single out two of the edges incident to v via the following procedure. Remove all vertices that belong to layers less than i together with their edges, except for the edges that connect to v (such edges remain dangling with one endpoint in v). Since v is in layer i, it is now visible from the outside, and so are at least two of its edges, say e_1 and e_2. These are the edges that we single out.

Now let us return to the original graph. Replace v by two vertices v_1 and v_2, where v_1 keeps $d-2$ edges of v including e_1, v_2 keeps the remaining two edges including e_1, and an edge (v_1, v_2) is added. Hence the degrees of v_1 and v_2 are smaller than the degree of v. The original graph is a minor of the new graph. The new graph is planar because neighbors of v_1 are consecutive neighbors of v. The graph remains k-outerplanar because e_1 and e_2 are visible once all vertices up to layer $i-1$ and their edges are removed (except for possibly e_1 and e_2), and hence so are v_1 and v_2. Hence they both are in layer i.

Continuing as above, all vertices can be made of degree at most 3. □

Given a spanning forest F of a graph $G(V, E)$, the *load* of a vertex v is the number of different edges $e \in (E \setminus F)$ whose addition to F closes a cycle through v. The load of an edge $e' \in F$ is one plus the number of different edges $e \in (E \setminus F)$ whose addition to F closes a cycle through e'. The load of F is the maximum load of any of its vertices and edges.

Proposition 11 *If G has a spanning forest F of maximum degree 3 and load ℓ than it has treewidth at most ℓ.*

Proof: Consider the natural tree (in fact, forest, if G is not connected) decomposition T of F whose bags are the vertices of G and the edges of F (a vertex bag in connected to the bags of its incident edges). Consider a non-forest edge (u, v). This requires to have a bag containing both u and v. Follow the path between u and v in F and add u to the respective bags, if not already there. Doing so for all non-forest edges, each vertex bag increased by at most the load of its respective vertex, and each edge bag increased by at most one less than its load. □

We now state and prove Lemma 12.

Lemma 12 *Every k-outerplanar graph has treewidth at most $3k$.*

Proof: By Proposition 10 and the fact that treewidth does not increase by taking minors, it suffices to prove the lemma for k-outerplanar graphs of degree at most 3. By Proposition 11, it suffices to show that every k-outerplanar graph of degree at most 3 has a spanning forest of load at most $3k$. We prove this by induction.

Base case. A 1-outerplanar graph is simply an outerplanar graph. Consider an arbitrary outerplanar embedding. Remove all edges visible from the outer face. No cycle remains. (This is one place where we use the restriction that the maximum degree is 3.) Hence we have a forest. Put back some of the removed edges until a maximal forest is obtained. Thereafter, every non-forest edge closes a cycle. It is important to note that this cycle must be a face in the outerplanar embedding – no edge of G not in F can form an inner cord of this cycle, because all edges not in F are on the outer face. As every edge is only on two faces, the load per forest edge is at most 2. As every vertex is on at most three faces (this is another place where we use the restriction that the maximum degree is 3), its load is at most 3.

Inductive step. Assume the claim for $(k-1)$-outerplanar graphs. Consider now an arbitrary k-outerplanar graph $G(V, E)$ of degree at most 3 and an arbitrary outerplanar embedding for it. Remove the set O of edges of the outer face to obtain a graph G'. The vertices of layer 1 in G have degree at most 1 in G'. Hence G' is $(k-1)$-outerplanar, and has a spanning forest F' of load at most $3(k-1)$. Let R' denote the set of edges not in F' and not in O. Put back removed outer face edges until a maximal forest F is obtained. Let R be the edges of O not in F. Edges of R can contribute at most 2 to the load of any edge of F and at most 3 to the load of any vertex. (For the graph containing only the edges of F and R every face contains an outer edge. Hence every edge of R closes a cycle which is a face, similar to the base case.) The load incurred by edges or R' was accounted for already in F'. □

As an application of Theorem 8, we can prove:

Theorem 13 *For every k there is at algorithm that runs in time $n^{O(1)}2^{O(k)}$ and approximates maximum weight independent set (MWIS) in planar graphs within a ratio of $1-1/k$.*

Proof: Apply Theorem 8. One of the sets V_i contains at most a $1/k$ fraction of the weight of the MWIS. Try all possible values of i. Remove the corresponding set V_i and solve MWIS on the remaining graph, using dynamic programming on graphs of treewidth $O(k)$. □