Handout on vertex separators and low tree-width k-partition

January 12 and 19, 2012

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 generalized by Alon, Seymour and Thomas to any family of graphs that excludes some fixed (arbitrary) subgraph H as a minor.

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 $h\sqrt{hn}$.*

Corollary 2 *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 in class for Theorem 1 easily extends to this setting. □

Corollary 3 *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 an algorithm that given a graph of treewidth p constructs a tree decomposition of treewidth $8p$. Using Corollary 2, 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 reader is advised to verify this claim.) □

We remark that we have seen in previous lectures that graphs of treewidth p have separators of size at most $p + 1$. Corollary 3 is an approximate reverse implication.

The following corollary is useful in designing polynomial time approximation schemes (PTAS).

Corollary 4 *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.*

Here is one such PTAS.

Corollary 5 *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.*

A related algorithmic paradigm is based on the following theorem of DeVos, Ding, Oporowski, Sanders, Reed, Seymour and Vertigan.

Theorem 6 *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 6 uses structural properties of graphs with excluded minors, and is beyond the scope of the course. Instead, we shall prove a theorem (due to Baker) in the interesting special case that G is planar.

Theorem 7 *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 can be found in polynomial time.*

As an application of Theorem 7, we can prove:

Theorem 8 *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 $1 - 1/k$.*

Homework:

1. Lipton and Tarjan showed that every planar graph has a separator of size $2\sqrt{2n}$ (not proved in class). The leading constant was subsequently improved. Use Theorem 7 to prove that every planar graph has a separator of size at most $2\sqrt{3n} + 1$.

2. Max cut is the problem of partitioning the vertex set of a graph into two sets in a way that maximizes the number of edges between the sets. For given H, design a PTAS for max cut in graphs with no H-minor. Namely, given a graph G that does not contain H as a minor and a parameter $\epsilon > 0$, your algorithm needs to produce a cut of size at least $(1-\epsilon)$ times the optimal, and do so in time $O(n^{O(1)})$, where the O notation may hide constants that depend on H and on ϵ.

   Remark. In planar graphs, max cut can be solved exactly in polynomial time, via a completely different approach.