Or il y avait des graines terribles sur la planète du petit prince . . . c'étaient les graines de baobabs. Le sol de la planète en était infesté. Or un baobab, si l'on s'y prend trop tard, on ne peut jamais plus s'en débarrasser. Il encombre toute la planète. Il la perfore de ses racines. Et si la planète est trop petite, et si les baobabs sont trop nombreux, ils la font éclater.

[Now there were some terrible seeds on the planet that was the home of the little prince; and these were the seeds of the baobab. The soil of that planet was infested with them. A baobab is something you will never, never be able to get rid of if you attend to it too late. It spreads over the entire planet. It bores clear through it with its roots. And if the planet is too small, and the baobabs are too many, they split it in pieces.]

— Antoine de Saint-Exupéry (translated by Katherine Woods)
Le Petit Prince [The Little Prince] (1943)

11 Treewidth

In this lecture, I will introduce a graph parameter called *treewidth*, that generalizes the property of having small separators. Intuitively, a graph has small treewidth if it can be *recursively* decomposed into small subgraphs that have small overlap, or even more intuitively, if the graph resembles a 'fat tree'. Many problems that are NP-hard for general graphs can be solved in polynomial time for graphs with small treewidth. Graphs embedded on surfaces of small genus do not necessarily have small treewidth, but they can be covered by a small number of subgraphs, each with small treewidth. This covering can be used to develop efficient algorithms (either exact or approximate) for a huge number of problems on surface graphs that are NP-hard for general graphs. As an example of this technique, I'll describe a polynomial-time approximation scheme for the maximum independent set problem.

11.1 Definitions

The concept of treewidth was discovered independently by several different researchers and given several different names. The actual term 'treewidth' and its definition in terms of tree decompositions were introduced by Robertson and Seymour [15]. Almost simultaneously, Arnborg and Proskurowski independently began the systematic study of *partial k-trees* [3, 2, 1]. Both groups were apparently unaware of Bertelè and Brioschi's earlier equivalent definition of the *dimension*[1] of a graph [6], or Halin's related study of *S-functions* [13]. Other equivalent definitions are surveyed by Bodlaender [8, 9].

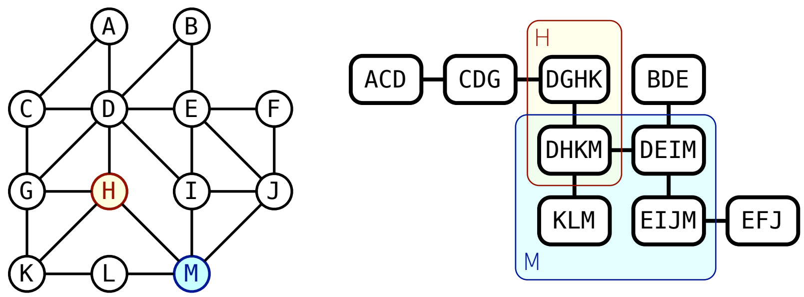
A ***tree decomposition*** (T, X) of a graph $G = (V, E)$ consists of a tree $T = (I, F)$ and a function $X : I \to 2^V$ satisfying three constraints. To avoid confusion, I will always refer to *vertices* of G, but *nodes* of T. We say that a vertex v is ***associated with*** a node i, or vice versa, whenever $v \in X(i)$.

- Every vertex in G is associated with at least one node in T. More formally, we have $\bigcup_{i \in N} X(i) = V$.
- For every edge uv in G, at least one node in T is associated with both u and v.
- The nodes in T associated with any vertex of G define a subtree (connected subgraph) of T. Equivalently, $X(i) \cap X(k) \subseteq X(j)$ for every node j on the path from node i to node k in T.

The ***width*** of a tree decomposition is $\max_i |X(i)| - 1$. Finally, the ***treewidth*** of G is the minimum width of any tree decomposition of G. A single graph can have several tree decompositions of minimum width.

Conversely, let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a set of subtrees of a tree T. The ***intersection graph*** $\boldsymbol{I(\mathcal{S})}$ is the graph with vertices $\{1, 2, \ldots, n\}$, where two vertices i and j are joined by an edge if and only if S_i and S_j share a node in T. The ***depth*** of $\mathcal{S}$ is the maximum number of subtrees that share any node in T,

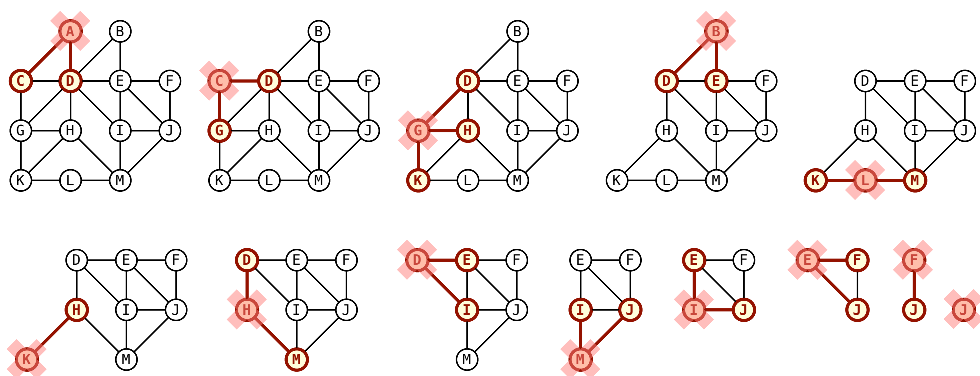
[1] Erdős, Harary, and Tutte defined the dimension of a graph to be the minimum dimension of a Euclidean space that allows an embedding where every edge is a line segment of length 1 [12]. This is *not* the same as treewidth.



A graph with treewidth 3 and an optimal tree decomposition.
The subtrees of T corresponding to vertices H and M are highlighted.

minus 1. The treewidth of G is the minimum depth of a subtree system $\mathcal{S}$ whose intersection graph $I(\mathcal{S})$ contains G as a subgraph.

A graph G is called a ***k*-tree** if and only if either G is the complete graph with k vertices, or G has a vertex v with degree $k-1$ such that $G \setminus v$ is a k-tree. A ***partial k*-tree** is any subgraph of a k-tree. In particular, a connected graph G is a 1-tree if and only if G is a tree. Unwinding the recursion gives us an ***elimination order*** $v_1, v_2, \ldots, v_n$ for the vertices of any partial k-tree, in which each node v_i has at most $k-1$ neighbors with larger indices.



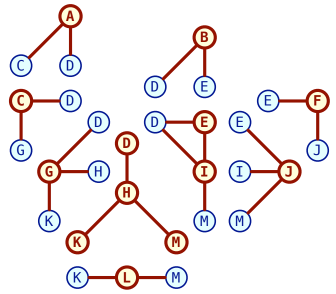
An elimination order for the example graph.

We can easily derive a tree decomposition of width k for any partial k-tree from an elimination order. Specifically, the decomposition tree T has n nodes $\{1, 2, \ldots, n\}$, and for any node i, the subset $X(i)$ contains vertex v_i and its (at most $k-1$) larger-indexed neighbors. The converse is true as well; every tree decomposition of width k defines an elimination order proving that the decomposed graph is a partial k-tree. (I'll leave the proof as an exercise.)

Lemma 11.1 (Wimer [17], Scheffler [16]). *A graph has treewidth k if and only if it is a partial k-tree.*

Finally, a tree decomposition (T, X) can be interpreted as a recursive separator hierarchy as follows. Choose any node of T to to be the root. For each node i of T, let $D(i)$ denote the union of the subsets $X(j)$ for all descendants of j of i (including i itself), and let $A(i)$ denote the union of the subsets $X(j)$ for all proper ancestors of j of i. Each node i is associated with a set of ***boundary*** vertices $B_i = D(i) \cap A(i)$ and a subgraph G_i, which is the induced subgraph $G[D(i)]$ minus edges between boundary nodes. Each subset $X(i)$ is a separator of the corresponding subgraph G_i.

We can relate the treewidth of a graph to the existence of small *balanced* separators as follows. We say that an n-vertex graph G is ***s*-separable**, for some non-decreasing function $s\colon \mathbb{N} \to \mathbb{N}$, if either $n = 1$ or G has a 2/3-separator S of size $s(n)$, such that the components of $G \setminus S$ are also s-separable.



A separator hierarchy for the example graph, rooted at DHKM.
Lighter (blue) nodes are boundary nodes inherited from ancestors in the hierarchy.

Lemma 11.2. *Any graph of treewidth k is $O(k)$-separable. Conversely, any s-separable n-vertex graph has treewidth $O(s(n)\log n)$, or treewidth $O(s(n))$ if $s(n) = \Omega(n^c)$ for some constant $c > 0$.*

Proof (sketch): Let G be a graph with treewidth k, and let (T,X) be a tree decomposition of width k. Without loss of generality, every node in T has degree at most 3. Thus, there is a node i that separates T into subtrees, each with at most $2/3$ of the nodes of T. The corresponding subset $X(i)$ is a $2/3$-separator for G of size at most $k+1$, and the components of $T \setminus X(i)$ have treewidth at most k.

On the other hand, suppose $G = (V,E)$ is s-separable, and $|V| = n > 1$. I claim that G has a ***path*** decomposition of the appropriate width. Let S be a $2/3$ separator of size $s(n)$, such that both components of $G \setminus S$ are also s-separable. Let (T_1, X_1) and (T_2, X_2) be path decompositions of the components of $G \setminus S$. Let T be a path obtained by concatenating paths T_1 and T_2. For any node i in T_1, let $X(i) = X_1(i) \cup S$, and for any node j in T_2, let $X(i) = X_2(i) \cup S$. Then (T,X) is a path decomposition whose width $k(n)$ obeys the recurrence $k(n) \le k(2n/3) + s(n)$. □

Corollary 11.3. *Every planar graph has treewidth $O(\sqrt{n})$. Every graph that has a cellular embedding on a surface of genus $g > 0$ has treewidth $O(\sqrt{gn})$.*

11.2 Dynamic Programming: Maximum Independent Set

An ***independent set*** in a graph G is a subset of the vertices, no two of which are connected by an edge in G. Compute the largest independent set in a graph is one of Karp's classical NP-hard problems. Moreover, even approximating the largest independent set to within a factor of $n^{1-\varepsilon}$, for any $\varepsilon > 0$, is NP-hard [14, 18]. However, for any graph of constant (or even logarithmic) treewidth, we can compute the largest independent set in polynomial time.

Theorem 11.4 (Arnborg and Proskurowski [4]). *The size of the largest independent set in a graph of treewidth k can be computed in $O(4^k k^2 n)$ time.*

Proof: Let G be a graph of treewidth k, and let (T,X) be a tree decomposition of width k. Without loss of generality, the tree T has at most n nodes. Let r denote the arbitrarily chosen root of T. For each node i, let G_i and B_i denote the corresponding subgraph and boundary vertices (as defined above).

For any node i of T and any subset $A \subseteq B_i$, let $\boldsymbol{MIS(i,A)}$ denote the size of the largest independent set I in G_i such that $I \cap B_i = A$. This function obeys the recurrence

$$MIS(i,A) = |A| + \max \left\{ \sum_{j \text{ child of } i} MIS(j, B_j \cap A') - \left|B_j \cap A\right| \;\middle|\; \begin{array}{c} J \text{ is an independent set in } G_j \\ \text{and } A \subseteq J \subseteq X(i) \end{array} \right\},$$

with the base case $MIS(i,A) = |A|$ whenever $G_i \setminus B_i = \varnothing$ (in particular, when i is a leaf of T). The largest independent set in G has size $MIS(G,\varnothing)$. We can easily evaluate the recurrence using dynamic programming.

For each node i, there are at most 2^{k+1} subsets A to consider. For each subproblem (i,A), and each child j of i, there are at most 2^{k+1} subsets J to consider; for each subset J, we can easily determine in $O(k^2)$ time whether J is an independent set in G_j. Thus, the total time to evaluate $MIS(i,A)$, not counting recursion, is $O(2^k k^2 \cdot \deg(i))$, where $\deg(i)$ is the number of children of node i. Thus, the total running time of the dynamic programming algorithm is $\sum_i O(4^k k^2 \cdot \deg(i)) = O(4^k k^2 n)$. □

Similar strategies apply to a truly *enormous* number of other NP-hard optimization problems [1]; some of these are surveyed by Arnborg and Proskurowski [4] and by Bodlaender [7].

11.3 PTAS for Planar Graphs

Lemma 11.5. *A planar graph with diameter D has treewidth at most $3D-2$.*

Proof (Eppstein [10]): Without loss of generality, assume that G is a triangulation; adding edges does not increase the diameter of the graph. Let $(T,\varnothing,C)$ be a tree-cotree decomposition where T is a BFS tree rooted at an arbitrary node r. For each face f of T, let $X(f)$ be the setof vertices on the shortest paths from the vertices of f to r. Then (C^*,X) is a tree decomposition; each subset $X(f)$ has size at most $3D-1$, so the width of the decomposition is at most $3D-2$. □

Let G be an *arbitrary* graph. Choose an arbitrary root vertex r, and for any vertex r, let $\boldsymbol{d(v)}$ denote the ***depth*** of v in a breadth-first search tree rooted at r. For any nonnegative integers $i \le j$, let $G[i,j]$ be the subgraph of G induced by vertices v such that $i \le d(v) \le j$, and let $\boldsymbol{G\langle i,j]} := G[0,j] \,/\, G[0,i-1]$. Observe that $G[i,j]$ is a subgraph of $G\langle i,j]$, and that $G[i,j] \setminus G\langle i,j]$ consists of a single vertex: the contraction of $G[0,i-1]$.

Lemma 11.6. *For any graph G and any nonnegative integers $i \le j$, the graph $G\langle i,j]$ has diameter at most $2j-2i+1$.*

Finally, for any integers $i \ge 0$ and $k \ge 2$ such that $i < k$, let $\boldsymbol{G_{i,k}}$ denote the subgraph of G induced by vertices whose depth mod k is *not* equal to i.

Corollary 11.7 (Baker [5]). *For any planar graph G and any integers $i \ge 0$ and $k \ge 2$ such that $i < k$, the subgraph $G_{i,k}$ has treewidth $O(k)$.*

Proof: $G_{i,k}$ is the disjoint union of all graphs of the form $G[ak+i+1,(a+1)k+i-1]$. For every integer a, the graph $G\langle ak+i+1,(a+1)k+i-1]$ has diameter at most $2k-3$, and therefore has treewidth at most $3(2k-3)-2 = 6k-11$. Thus, each subgraph $G[ak+i+1,(a+1)k+i-1]$ also has treewidth at most $6k-11$. □

Theorem 11.8 (Baker [5]). *For any integer $k > 1$, we can compute a $(1-1/k)$-approximation of the largest independent set in any n-node planar graph in $2^{O(k)}n$ time.*

Proof: Let G be a planar graph, and let S be the largest independent set in G. For each integer $0 \le i < k$, let $S_i = \{v \in S \mid d(v) \bmod k = i\}$. For any i, our dynamic programming algorithm can compute the largest independent set S_i' in $G_{i,k}$ in $2^{O(k)}n$, because $G_{i,k}$ has treewidth $O(k)$. The set $S \setminus S_i$ is an independent set in $G_{i,k}$, so $|S'| \ge |S \setminus S_i|$. Finally, the pigeonhole principle implies that $|S \setminus S_i| \ge (1-1/k)|S|$ for some i. □

Equivalently, we can compute a $(1-\varepsilon)$-approximation in time $2^{O(1/\varepsilon)}n$. In particular, we can compute a $(1-1/\log n)$-approximation in polynomial time.

11.4 PTAS for Surface Graphs

Lemma 11.9 (Eppstein [11]). *A graph with diameter D and genus g has treewidth $O(gD)$.*

Proof: Let G be a cellular graph on a surface of genus g. Let (T, L, C) denote a tree-cotree decomposition of G where T is a breadth-first-search tree. Let R be the reduced cut graph obtained by repeatedly removing vertices of degree 1 from the cur graph $T \cup L$. We proved earlier in the semester that R is the union of $O(g)$ shortest paths. It follows that R contains $O(gD)$ nodes.

We can embed the graph G / R in the plane by first embedding $G \setminus R$ so that all neighbors of R are on the outer face, and then connecting the neighbors of R to a new super-vertex. Thus, G / R is planar. Contracting an edge can only reduce the diameter of a graph, so G / R has diameter at most D. Thus, Baker's Lemma (Corollary 10.7) implies that G / R has a tree decomposition (T', X) of width at most $3D - 1$.

We can extend (T', X) to a tree decomposition (T', X') of G by setting $X'(i) = X(i) \cup V(R)$. The resulting decomposition has width $O(gD)$. □

Corollary 11.10. *For any genus-g graph G and any integers $i \geq 0$ and $k \geq 2$ with $i < k$, the subgraph $G_{i,k}$ has treewidth $O(gk)$.*

Theorem 11.11 (Eppstein [11]). *For any integer $k > 1$, we can compute a $(1 - 1/k)$-approximation of the largest independent set in any n-node graph of genus g in $2^{O(gk)}n$ time.*

Equivalently, we can compute a $(1-\varepsilon)$-approximation in time $2^{O(g/\varepsilon)}n$. In particular, we can compute a $(1 - g/\lg n)$-approximation in polynomial time.

References

[1] S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability: A survey. *BIT* 25:2–23, 1985.

[2] S. Arnborg and A. Proskurowski. Characterization and recognition of partial k-trees. *Congr. Numer.* 47:69–75, 1985. Proc. 16th S.E. Int. Conf. Combinatorics, Graph Theory, and Computing.

[3] S. Arnborg and A. Proskurowski. Characterization and recognition of partial 3-trees. *SIAM J. Alg. Disc. Meth.* 7(2):305–314, 1986.

[4] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Appl. Math.* 23:11–24, 1989.

[5] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.* 41(1):153–180, 1994.

[6] U. Bertelè and F. Brioschi. *Nonserial Dynamic Programming*. Academic Press, New York, 1972.

[7] H. L. Bodlaender. Dynamic programming on graphs of bounded treewidth. *Proc. 15th Int. Colloq. Automata Lang. Prog.*, 105–119, 1988. Lecture Notes Comput. Sci. 317, Springer-Verlag.

[8] H. L. Bodlaender. Some classes of graphs with bounded treewidth. *Bull. EATCS* 36:116–126, 1988. Technical Report RUU-CS-76-22, Dept. Comput. Sci., Univ. Utrecht, 1998.

[9] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.* 209:1–45, 1998.

[10] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.* 3(3):1–27, 1999.

[11] D. Eppstein. Diameter and treewidth in minor-closed families. *Algorithmica* 27:275–291, 2000.

[12] P. Erdős, F. Harary, and W. T. Tutte. On the dimension of a graph. *Mathematika* 12:118–122, 1965.

[13] R. Halin. S-functions for graphs. *J. Geom.* 8(1–2):171–186, 1976.

[14] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica* 182(1):105–142, 1999.

[15] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms* 7(3):309–322, 1986.

[16] P. Scheffler. Linear-time algorithms for NP-complete problems restricted to partial k-trees. Tech. Rep. R-Math-87/03, Akademie der Wissenschaften der DDR, Karl-Weierstrass-Institut für Mathematik, 1987.

[17] T. V. Wimer. *Linear algorithms on k-terminal graphs*. Ph.D. thesis, Clemson University, 1987.

[18] D. Zuckerman. Linear degree extractors and the inapproximability of MAX CLIQUE and CHROMATIC NUMBER. *Theory Comput.* 3:103–128, 2007.