CSC 565: Graph Theory **North Carolina State University**
Fall 2019 **Computer Science**

Lecture 27: Nov 25, 2019

Lecturer: Don Sheehy <drsheehy@ncsu.edu> *Scribe: K. Gardner, G. Miller*

1 Recap

Last week we discussed how to solve systems of linear equations of the form

$$Mx = b$$

in the special case where M is positive definite (P.D.). While this can be done using the Cholesky decomposition the performance depends on the structure of the matrix. In the case of sparse matrices, for example, we would want to keep the matrix sparse. We do not want to introduce more nonzero terms into the matrix This requires an ordering on the vertices which minimizes the number of contractions.

2 Reordering Sparse Matrices

The ordering is given by a permutation matrix P known as the *pivot order*. We then solve

$$PMP^\top = Pb$$

which is still symmetric and P.D. and return $x = P^\top y$. This reordering can sometimes dramatically improve the performance.

3 Finding a Good Ordering

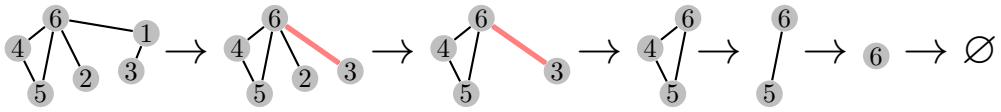

Figure 1: An elimination ordering of vertices and the resulting graphs with fill edges in red.

For some ordering $v_1, \ldots, v_n$ on the vertices of a graph G let G_i be the graph after vertices 1-i are eliminated. Figure 1 depicts an elimination ordering on a graph with 6 vertices and the resulting graphs $G = G_0, G_1, \ldots, G_6 = \emptyset$. The *fill graph* is the graph $G^+ = (\{1, \ldots, n\}, E^+)$ with the same vertex set and all of the *fill edges* $E^+ = \bigcup_{i=1}^{n} E_{G_i}$ resulting from eliminations. For a given ordering we would like to minimize the number of fill edges.

We will build an *elimination tree* rooted on the last vertex eliminated v_n. The parent of every other node v_i is defined to be $\min\{j > i \mid (i,j) \in E^+\}$. Note that the initial graph is a subgraph of the fill graph.

Throughout we assume an edge written (b, c) implies $b < c$. We would like a mapping $m : E^+ \to E$ that tells us which edge in the initial graph to "charge" for each fill edge in E^+ so that

$$|E^+| = \sum_{e \in E} |m^{-1}(e)|.$$

We define $m(b, c) = (a, c)$ where a is the minimal descendant of b in the elimination tree such that $(a, c) \in E$.

Lemma 1. For every $(b, c) \in E^+$ there exists $(a, c) \in E$ such that a is a descendant of b

Proof. If b is minimal then set $a = b$ and we are done. So we may assume w.l.o.g. that $(b, c) \notin E$.

Suppose, for the sake of contradiction, that there exists some $(b, c) \in E^+$ with no descendant a of b such that $(a, c) \in E$. So (b, c) was added by a reduction after removing a vertex x that was adjacent to both b and c. Let $x = max\{v < b \mid (x, b), (x, c) \in E^+\}$ and let y be the parent of x in the elimination tree. That is, y is the smallest larger value connected to a vertex in the fill graph.

Note that $y \neq c$ as c is connected to b. If $y \neq b$ then $x < y \leq b$, but y is connected to b and c in the fill graph, contradicting our choice of x. There exists a descendant a of x such that $(a, c) \in E$ in the same way, therefore (a, c) is a descendant of b. □

4 Nested Dissection

In nested dissection a separator C is used between subgraphs A and B to order eliminations. That is, a vertex set V is broken as the disjoint union $V = A \sqcup B \sqcup C$ and the separator order is $\text{order}(V) = (\text{order}(A), \text{order}(B), C)$. If we build an elimination tree from this ordering we will have a tree of separators connected into separators of subgraphs. We can group vertices in an elimination tree where the separator tree is given by grouping the vertices by separator.

In the case of a grid with n vertices has $2\sqrt{2}$ vertices in the separator, decreasing geometrically as we move down the separator tree for a total height of $4\sqrt{n}$. Each fill edge in the elimination tree is along a path between the vertices of the corresponding edge that gets "charged" in the initial graph. The length of this path is at most the height of the tree. That is, an edge in the initial graph cannot be charged more than the height of the tree.

In the top level separator there are $2\sqrt{n}$ vertices and height at most $4\sqrt{n}$. So every vertex in the separator has degree at most 4 in E, and each descendant pays the height of the tree for a total of $2\sqrt{n} \cdot 4 \cdot 4\sqrt{n} = 32n$ fill edges in the top-level separator, reducing geometrically as we move down the tree. We can write the following recurrence relation

$$f(n) \leq 32n + 4f(n/4)$$

which gives $O(n \log n)$ fill edges.

5 Perfect Elimination Graphs

A *perfect elimination graph* is a graph $G = (V, E)$ such that there exists an ordering of V such that no new fill edges are introduced in the elimination tree. That is, for each vertex removed there exist edges between all of its neighbors in E. Examples of perfect elimination graphs include trees and triangulated polygons.

Definition 1. A *chordal graph* is a graph G such that every induced cycle of G is K_3.

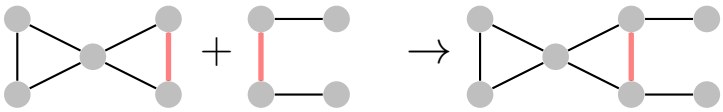

Figure 2: A 2-clique sum of two graphs.

We can glue graphs along shared k-cliques using the *k-clique sum*, depicted in Figure 2. Note that the clique sum of chordal graphs is chordal. Moreover, every chordal graph can be constructed in this way by iteratively gluing chordal subgraphs. Further note that $A_i \cap G_{i-1} \subseteq A_j$ for some $j < i$. In other words, every clique in G_i is contained in A_j for some $j \leq i$.

We can think of the gluing of cliques as edges in a tree with cliques as vertices. For an ordered set of cliques $A_1, \ldots, A_r$ and $G_i = G_{i-1} + A_i$ we define a tree with vertex set $V = \{A_1, \ldots, A_r\}$ rooted at A_1 such that the parent of all other nodes in the tree is the clique $\text{parent}(A_i) = A_j$ such that $A_i \cap G_{i-1} \subseteq A_j$ and $j < i$. A leaf of this tree is a clique that has just been added on. For example, A_r will be a leaf in this tree. This clique must have at least one vertex that isn't in any of the other cliques. If this vertex is removed all its neighbors are in the corresponding clique. Therefore, all fill edges that would be added when this vertex is removed are in this clique, so we have a perfect elimination order.

Corollary 1. *Clique sums of cliques have perfect elimination orders.*

If we have a corded graph and a minimal separator then there must be a path between any two vertices in the separator. Otherwise, there is a smaller separator. We therefore have a cycle of length at least four in a chordal graph, which implies that the minimum separator must be a clique.

Lemma 2. Chorded graphs are clique sums of cliques.

Proof. If the graph is a clique we are done. Otherwise take any minimal separator C which separates subgraphs A and B. C is a clique and, because subgraphs of chordal graphs are chordal, A and B are chordal. Therefore $G[A \cup C]$ and $G[B \cup C]$ are clique sums of cliques by induction. □

Theorem 1. *G is a perfect elimination graph if and only if it is chordal.*

Proof. ($\Rightarrow$) Suppose there is a perfect elimination ordering and an induced cycle that is not a triangle. If we have a perfect elimination ordering one of these vertices must be removed first. This vertex has two neighbors in the cycle that do not share an edge, otherwise the cycle would be a triangle. Therefore, a fill edge must be added when the vertex is removed, a contradiction, so we may conclude that the induced cycle is a triangle.

($\Leftarrow$) By Lemma 2 G is a sum of cliques, which has a perfect elimination order by Corollary 1. □

Definition 2. A graph G has *treewidth* $\leq k$ if G is a subgraph of a clique sum of cliques of size $k + 1$.