The Complexity of Optimal Elimination Trees

Alex Pothen

CS-88-16 April 1988

Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

The Complexity of Optimal Elimination Trees

Alex Pothen[1]
Computer Science Department
The Pennsylvania State University
University Park, PA 16802

April 1988

Abstract

We prove that finding a shortest elimination tree of a graph is an NP-complete problem. This justifies the Jess and Kees method for computing a good parallel ordering in sparse Cholesky factorization, which first computes a chordal filled graph (with small fill), and then finds a shortest elimination tree of the chordal graph. Finding the longest elimination tree with bounded fill is also NP-complete.

[1]Research supported by NSF grant CCR-8701723, NSF Equipment grant CCR-8705110, and by Oak Ridge National Laboratory and The University of Tennessee, Knoxville, during their Numerical Linear Algebra Year.

1 Introduction

In surveying the role played by the elimination tree in sparse matrix factorization, Liu [8] writes:

> Another important research direction is the characterization of the best elimination tree for a given computational environment, be it on a parallel architecture, a vector machine, or a virtual memory system. Current recommendations on desirable structures are mostly based on intuition and experience. A vigorous approach to provide theoretical justification seems to be an interesting and important area.

We examine the problem of computing a good elimination tree in the context of a medium grained parallel algorithm. In the elimination tree model of sparse Cholesky factorization, a shortest elimination tree is appropriate in a parallel environment, since the height of the tree is a measure of the inherently sequential part of the computation.

Jess and Kees [4] suggested that a good parallel ordering be computed in two steps. Initially an ordering that reduces the fill is computed; the resulting filled graph is a chordal graph. In the second step, a perfect elimination re-ordering that preserves the graph but reduces the height of the tree is computed. Liu [6] showed that the Jess and Kees method computes a shortest elimination tree of the chordal graph over all perfect elimination orders of the graph.

We show that the problem of computing a shortest elimination tree of a general graph is NP-complete. The problem of computing a longest elimination tree with bounded fill is also NP-complete. The latter problem is of interest when the factorization is computed out of core.

There are two possible approaches to coping with the intractability of the shortest tree problem: One is to adopt the two step Jess and Kees method, and compute the shortest tree of the chordal filled graph over all of its perfect elimination orders. Thus we provide a justification for the Jess and Kees approach. Liu and Mirzaian [7] and this author [9] have designed $O(n + e)$ time algorithms to implement the Jess and Kees method.

The second approach is to compute a vertex ordering of the general graph that tries to reduce fill and the elimination tree height simultaneously. The results in this paper suggest that this method will have to settle for an approximation to the shortest tree of the general graph. More work is needed to determine if this approach can be implemented efficiently, and if so, how the heights of the computed trees compare with those in the first approach.

The rest of this paper is organized as follows. In the next section we review background material concerning the elimination graph model, chain graphs, chordal graphs, and elimination trees. Section 3 contains a proof that the shortest elimination tree problem is NP-complete. In the last Section, we show that the longest elimination tree problem is also NP-complete, if the fill is bounded. Without a bound on the fill, this problem is trivial.

2 Background

The elimination graph model. The computation of the Cholesky factors of a sparse positive definite matrix A can be modeled on its adjacency graph $G(A)$. We assume the reader is familiar with this *elimination graph model*; a good description may be found in Chapter 5 of George and Liu [2]. Briefly, each column in A corresponds to a vertex in $G(A)$, and each nonzero a_{ij} to an edge between vertices i and j. By convention, the loop (i, i) corresponding to a_{ii} is omitted from the set of edges. The computation of a column of the Cholesky factor L corresponds to the elimination of a vertex—marking it as deleted, and adding the edges necessary to make all of its unmarked neighbors a clique. The added edges are called *fill edges*. The order in which the vertices are eliminated defines an *elimination order*, which is a mapping $\alpha : V \to \{1, 2, \ldots, n\}$, where $|V| = n$. An elimination order on the graph corresponds to a symmetric permutation of the rows and columns of A.

The *filled graph* G is the graph obtained by adding the fill edges generated by an elimination order to the adjacency graph of A. It is well known that G is a chordal graph—i.e., every cycle of length greater than 3 has a *chord*, an edge joining two non-consecutive vertices on the cycle. A *perfect elimination order* is an elimination order that incurs no fill. A graph can have a perfect elimination order iff it is chordal. If the vertices of a graph $G(A)$ are ordered by an elimination order α, and G is the filled graph corresponding to α, then α is a perfect elimination order on G. Chapter 4 of Golumbic [3] contains a detailed study of chordal graphs.

A *simplicial vertex* in a chordal graph is a vertex whose neighbors form a clique. Thus a simplicial vertex can be eliminated without incurring fill. A chordal graph is either a clique, or it has at least two non-adjacent simplicial vertices. Further, if G is chordal and v is a simplicial vertex, the induced subgraph $G - v$ is chordal.

In sequential computation, an elimination order that reduces the fill edges added is desirable, since storage and factorization time are proportional to the number of nonzeros in L. Yannakakis [11] showed that minimizing fill is NP-complete. (Garey and Johnson [1] contains a good exposition of the theory of NP-completeness.) Since we use some techniques from this proof in the sequel, it is appropriate to review them here.

Let $\Gamma(v)$ denote the adjacency set of v. A *bipartite graph* $B = (P, Q, E)$ has the vertex set $P \cup Q$ and each edge in E has one endpoint in P and the other in Q. A bipartite graph is a *chain graph* if the adjacency sets of vertices in P form a chain; i.e., vertices in P can be ordered such that

$$\Gamma(v_1) \supseteq \Gamma(v_2) \ldots \supseteq \Gamma(v_p).$$

Yannakakis [12], (pp. 318) proves that then vertices in Q also form a chain, and hence the definition is unambiguous.

Given a bipartite graph $B = (P, Q, E)$, we can construct a biclique $C = (P, Q, E \cup F)$, where F contains all the edges necessary to make each of P and Q cliques. Yannakakis [11] proved

Lemma 1 *A bipartite graph B is a chain graph iff the biclique C is chordal.*

The Lemma shows that if the addition of a set of edges to B makes it a chain graph $\overline{B}$, adding the same set of edges to C makes it a chordal graph $\overline{C}$. We will call $\overline{C}$ a *chordal completion* of the biclique C.

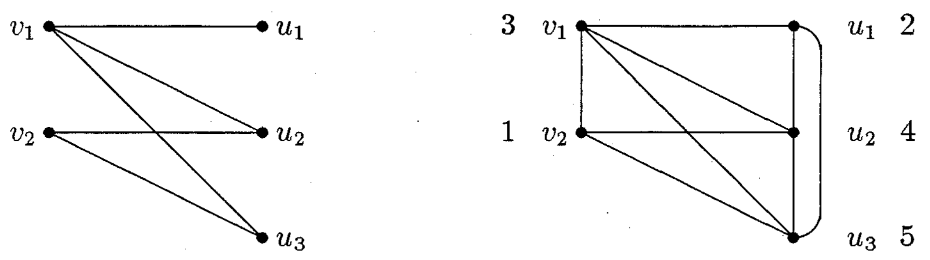
We can find two simplicial vertices of $\overline{C}$ from an ordering of the vertices of a chain graph $\overline{B}$. Let the P-vertices of $\overline{B}$ be ordered as above, and let $\Gamma(v)$ refer to the neighbors of v in the chain graph. Then v_p is simplicial in $\overline{C}$, since $P \cup \Gamma(v_p)$ is a clique in $\overline{C}$. For, all other P-vertices are adjacent to $\Gamma(v_p)$ by the ordering, and each of the sets P, $\Gamma(v_p)$, is a clique. Similarly, if the Q-vertices in $\overline{B}$ are ordered as

$$\Gamma(u_1) \subseteq \Gamma(u_2) \ldots \subseteq \Gamma(u_q),$$

then u_1 is simplicial in $\overline{C}$.

The elimination tree. Assume that the vertices of a chordal graph are renumbered (from 1 to n) according to a perfect elimination order. Let $hadj(v)$ be the set of vertices adjacent to v that are numbered higher than v.

For each vertex v, define $parent(v)$ to be the lowest numbered vertex in $hadj(v)$. If G is connected, and we define $E_T = \{(v, parent(v)) : v \in V\}$, then the subgraph $T = (V, E_T)$ is a rooted tree called the *elimination tree*. The root is the vertex numbered n in the elimination order. The height of a vertex v in the tree is the length of a longest path from a leaf to v. The height of the tree is the height of the root.



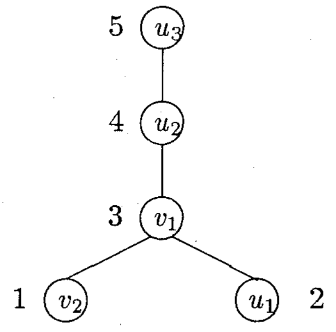


Figure 1: A chain graph, its biclique, and an elimination tree.

Liu [8] surveys the role that the elimination tree plays in sparse factorizations; Zmijewski and Gilbert [13] provide an especially lucid exposition of several properties of the elimination tree.

Schreiber [10] proved the following

Lemma 2 *Let (v, w) be an edge in a chordal graph G whose vertices are numbered in a perfect elimination order. If $v < w$, there exists a monotone path from v to w in the elimination tree, and for all vertices x on this path, (x, w) is an edge in the graph.*

This lemma has an important consequence: Let $T[x]$ and $T[y]$ be two subtrees of the elimination tree rooted at vertices x and y respectively. If $T[x]$ and $T[y]$ are vertex disjoint, then these sets are mutually independent in the chordal graph G. In particular, all the leaves of the elimination tree are pairwise independent.

Let C be a biclique with vertex sets P and Q. Then since the set P is a clique in C, it remains a clique in a chordal completion $\overline{C}$. The result above then implies that the elimination tree contains a monotone path from the lowest numbered vertex in P to the highest numbered vertex in P. Thus all vertices in P belong to a path from a leaf to the root in the tree. (Such a path may include some vertices from Q as well.) A similar result holds for vertices in Q. Thus an elimination tree of $\overline{C}$ can have at most two leaves, one belonging to P, and the other to Q. Figure 1 shows a chain graph, a biclique obtained from it (which is chordal), a perfect elimination order of the chordal graph, and the corresponding elimination tree.

For ease of notation, if $G = (V, E)$ is a graph, and R is a subset of its vertices, then $G \backslash R$ is the induced subgraph on the vertices $V \backslash R$. If $R = \{v\}$, then we write the subgraph as $G - v$. In a bipartite graph $G = (P, Q, E)$, if a vertex $v \in P$, we call it a P-vertex, and similarly for Q. We denote by n the number of vertices in a graph G.

3 The shortest tree problem

In this section we prove that computing a shortest elimination tree of a graph is NP-complete. We will need some preliminary results.

The *Mutual Independent Set* problem is the following: Given a bipartite graph $G = (P, Q, E)$, are there sets $V_1 \subseteq P$, $V_2 \subseteq Q$, with $|V_1| = |V_2| = k$, such that V_1 and V_2 are mutually independent? That is, no edge joins a vertex in V_1 to a vertex in V_2?

Lemma 3 *The Mutual Independent Set problem for bipartite graphs is NP-complete.*

Proof: The reduction is from the Balanced complete bipartite subgraph problem, (Garey and Johnson [1], pp.196, [GT24]). In a bipartite graph $G = (P, Q, E)$, this is the problem of finding disjoint sets $V_1 \subseteq P$ and $V_2 \subseteq Q$, with $|V_1| = |V_2| = k$, such that the subgraph induced by V_1 and V_2 is a complete bipartite graph.

Given an instance of the Balanced complete bipartite subgraph problem, construct the *bipartite complement* $\overline{B} = (P, Q, \overline{E})$ with $e = (i, j) \in \overline{E}$ iff $i \in P$, $j \in Q$, and $(i, j) \notin E$. Then B has a complete bipartite subgraph induced by the sets V_1 and V_2 iff these sets are mutually independent in $\overline{B}$. Since the former problem is NP-complete, the result follows. ■

Let $B = (P, Q, E)$ be a bipartite graph with corresponding biclique C. Let $\overline{B}$ be a chain graph that can be obtained from B by the addition of edges, and let $\overline{C}$ be the chordal graph that is obtained by adding the same edges to C.

Lemma 4 *B has mutually independent sets of size k iff there exists a chordal completion $\overline{C}$ with an elimination tree of height $n - k - 1$.*

Proof: If B has mutual independent sets of size k, let these sets be $\overline{P} \subseteq P$, and $\overline{Q} \subseteq Q$. Construct a chain graph $\overline{B}$ by ordering the P-vertices such that vertices in $\overline{P}$ are ordered last:

$$\Gamma(v_1) \supseteq \Gamma(v_2) \ldots \supseteq \Gamma(v_{p-k}) \supseteq \Gamma(v_{p-k+1}) \ldots \supseteq \Gamma(v_p),$$

where the last k vertices belong to $\overline{P}$. Since vertices in $\overline{P}$ are ordered last, and this set is independent of $\overline{Q}$, it is only necessary to add edges that join $P \setminus \overline{P}$ to Q, and $\overline{P}$ to $Q \setminus \overline{Q}$. Thus $\overline{P}$ and $\overline{Q}$ continue to be mutually independent in the chain graph. Since $\overline{B}$ is a chain graph, vertices in Q can be ordered such that

$$\Gamma(u_1) \subseteq \Gamma(u_2) \ldots \subseteq \Gamma(u_k) \subseteq \Gamma(u_{k+1}) \ldots \subseteq \Gamma(u_q),$$

with the first k vertices belonging to $\overline{Q}$.

In the chordal graph $\overline{C}$, vertices v_p and u_1 are simplicial and independent, and can be eliminated together. Next, vertices v_{p-1} and u_2 can be eliminated, and so on, till v_{p-k+1} and u_k are eliminated together. The resulting elimination tree has two leaves, v_p and u_q. Further, the subtrees rooted at v_{p-k+1} and u_k are vertex disjoint. Hence a longest path from a leaf to a root excludes at least k vertices, and the height of the tree is at most $(n - k) - 1$.

Conversely, suppose a chordal completion $\overline{C}$ of the biclique C has an elimination tree with height at most $n-k-1$. Since the height of the tree is less than $n-1$, the tree has two leaves, and one leaf belongs to P, the other to Q. Since P is a clique, all P-vertices are contained in a path from the P-leaf to the root. At least k Q-vertices do not lie on this path, since its length is at most $n-k$. Similarly, at least k P-vertices do not lie on the path from the Q-leaf to the root. Thus these two paths can have at most $n-2k$ vertices in common.

Thus the first k vertices on the path from the P-leaf to the root (counting from the leaf) belong to P. Call this set $\overline{P}$. Similarly the first k vertices on the path from the Q-leaf to the root belong to Q, and we call this set $\overline{Q}$. Since $\overline{P}$ and $\overline{Q}$ are vertex disjoint, by Lemma 2, these sets are mutually independent. ■

Theorem 1 *The shortest elimination tree problem for a general graph is NP-complete.*

Proof: Follows immediately from Lemma 3 and Lemma 4. ■

4 Other measures of optimality

Elimination trees optimal with respect to other measures are appropriate in computing environments different from a parallel architecture. We consider the problem of computing a long elimination tree, which is appropriate when the factorization is computed out of core [8].

We study the problem of computing a longest elimination tree when the size of the fill edges is bounded. Such a bound is necessary, for otherwise a tree of height $n-1$ could be computed for any connected graph, by the following strategy. We choose the first vertex in an elimination order arbitrarily, and then at each step, choose a vertex which is adjacent in the filled graph to the last eliminated vertex.

We show that the problem of computing a longest elimination tree by adding at most k edges to a graph is NP-complete. We require the following lemma.

Lemma 5 *Let $B = (P, Q, E)$ be a chain graph with corresponding biclique C. Then C has an elimination tree of height $n-1$.*

Proof: Let the P-vertices in B be ordered as follows:

$$\Gamma(v_1) \supseteq \Gamma(v_2) \ldots \supseteq \Gamma(v_p).$$

Then v_p is simplicial in C, and can be eliminated first in a perfect elimination order. In the graph $C - v_p$, v_{p-1} is simplicial, and can be eliminated next, and continuing this way we can choose $v_p, \ldots, v_1$ to be the first p vertices in a perfect elimination order of C.

Since Q is a clique in C, we can eliminate the vertices in Q in any order to complete this perfect elimination order. Further, since B is connected, $\Gamma(v_1) = Q$. Since each of P, Q is a clique in C, the elimination tree is a path of P-vertices followed by a path of Q-vertices, and has height $n - 1$. ∎

Theorem 2 *It is NP-complete to find a chordal graph with a longest elimination tree by adding at most k edges to a graph G.*

Proof: We restrict G to be a biclique. Every chordal completion of G has an elimination tree of height $n - 1$, by Lemma 5. It is NP-complete to find if we can obtain a chordal graph by adding at most k edges to a biclique (Yannakakis [11]). ∎

In contrast to this result, the complexity of finding a longest elimination tree of a chordal graph over all perfect elimination orders, is unknown. Also of interest in sparse Cholesky factorization is the computation of an elimination tree that leads to the largest average supernode size. Such trees are of interest when the factorization is computed on a vector machine. (For a definition of a supernode, see [9].) The complexity of this problem is also unknown. Johnson's column [5] tabulates the known (and unknown) complexity results for ten famous graph problems when restricted to several classes of graphs, including chordal graphs.

Acknowledgements. A part of this work was done while the author was visiting Oak Ridge National Laboratory and the University of Tennessee, during their Special Year in Numerical linear algebra. I thank my colleagues at both institutions for their hospitality and encouragement. I also thank my colleague Georg Schnitger for several helpful conversations.

References

[1] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[2] Alan George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Englewood Cliffs, N.J., 1981.

[3] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[4] J. A. G. Jess and H. G. M. Kees, A data structure for parallel *LU* decomposition, *IEEE Trans. Comput.*, C-31: 231–239 (1982).

[5] David S. Johnson, The NP-Completeness column: an ongoing guide, *J. Algorithms*, 6: 434–451 (1985).

[6] Joseph W. H. Liu, Reordering sparse matrices for parallel elimination, Tech. Report CS-87-01, Computer Science, York University, 1987.

[7] Joseph W. H. Liu and Andranik Mirzaian, A linear reordering algorithm for parallel pivoting of chordal graphs, Tech. Report CS-87-02, Computer Science, York University, 1987.

[8] Joseph W. H. Liu, The role of elimination trees in sparse factorization, Tech. Report CS-87-12, York University, 1987.

[9] Alex Pothen, Simplicial cliques, shortest elimination trees, and supernodes in sparse Cholesky factorization, Tech. Report CS-88-13, Computer Science, Penn State, April 1988.

[10] Robert Schreiber, A new implementation of sparse Gaussian elimination, *ACM Trans. Math. Software*, 8: 266–283 (1982).

[11] Mihalis Yannakakis, Computing the minimum fill-in is NP-complete, *SIAM J. Alg. Disc. Meth.*, 2: 77–79 (1981).

[12] Mihalis Yannakakis, Node-deletion problems on bipartite graphs, *SIAM J. Comput.*, 10: 310–327 (1981).

[13] Earl Zmijewski and John R. Gilbert, A parallel algorithm for sparse symbolic Cholesky factorization on a multiprocessor, to appear in *Parallel Computing*, (1987).