

Consiglio Nazionale delle Ricerche

**ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE**

PISA

The Average Parallel Complexity of some
Sparse Problems.
B.Codenotti, G.Resta.

Nota Interna B4-02
Gennaio 1990

The average parallel complexity of some sparse problems

B. Codenotti⁽¹⁾ and G. Resta⁽²⁾

(1) Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche,
via S. Maria 46, 56100 Pisa, Italy.

(2) Dipartimento di Informatica, Corso Italia 40, 56100 Pisa, Italy.

1. Introduction.

In this paper, we give a first look at the average parallel complexity of some problems concerning random graphs and matrices. More precisely, we consider the solution of large sparse positive definite linear systems, which arise in a variety of contexts, e.g. scientific computation.

The direct solution of such systems is typically performed by Cholesky factorization [6][4], $A=LL^T$, in four stages:

- 1) ordering; 2) symbolic factorization; 3) numerical factorization;
- 4) solution of triangular systems.

Stages 1) and 2) are of combinatorial type. Ordering consists of permuting the rows and the columns of A in order to decrease the 'fill-in', i.e. the number of additional nonzeros. The choice of a good permutation for A is critical: without any ordering, a sparse problem can evolve in a dense one in few elimination steps. Finding an ordering that minimizes the fill of the factor L is an NP complete problem [8], so heuristic strategies have been proposed. *Minimum degree algorithm* (MD from now to forth)[5] and its variants are the most popular among these techniques.

The symbolic factorization consists of determining if a given entry of L is nonzero or not. After the symbolic factorization, the nonzero pattern of L is obtained. Starting from it, the notion of *elimination tree* [4] is introduced to formalize the precedence relations existing between different elimination steps.

In this paper, we import the notions

- of 'elimination tree', from numerical analysis, and
- of 'random graph', from graph theory,

and we analyze the average parallel cost of solving a sparse linear system by Cholesky factorization, through stages 1)-4). More precisely we show that

i)
$$\sum_{i=0}^{n-1} \left(1 - \sum_{j=0}^i \binom{n}{j} p^j q^{n-j} \right)^n$$
, is the average minimum number of

nonzeros in the row chosen by MD, where n is the size of the matrix, p is the probability that an element of the matrix is nonzero, and $q=1-p$;

ii)
$$\frac{1}{e^{1+p/4}} \sum_{k=1}^{n-1} e^{qk} = \alpha n$$
, $\alpha \in (e^{-1}, 1)$, is the average number of leaves in

the elimination tree, where n, p, q are defined as above;

iii) we give also formulas to estimate the average fill-in.

Furthermore, we conjecture that the average depth of an elimination tree is $\theta(n)$.

Our investigation uses probabilistic techniques involving matrices with uniform random nonzero pattern, to study, on the average, how the coefficient matrix evolves during the factorization. Note that we deal only with the size of the matrix and with the probability of each element to be nonzero.

As mentioned above, we estimate the probabilistic shape of L and the average fill-in in the matrix L . Moreover we predict the average number of leaves, in the elimination trees in order to give the maximum number of rows that can be eliminated at the first step of the factorization. Finally we find an experimental evidence that this number is related to the overall number of steps needed to perform the parallel factorization.

Previous works in the field has been done in [3][7], where some probabilistic estimates were proposed for fill-in during gaussian elimination. More recently [2] a graph-theoretic approach has been used for the (worst-case) parallel solution of sparse systems.

2. Preliminaries

We assume the reader to be familiar with Cholesky factorization for positive definite symmetric matrices. We use the following notation:

$C \equiv (c_{ij})$ denotes a matrix C whose (i,j) -th entry is c_{ij} ; $C^{(k)} \equiv (c_{ij}^{(k)})$

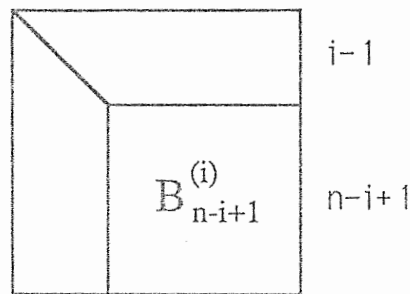
denotes the transformed matrix $C^{(k)}$, at the k -th elimination step, with entries $c_{ij}^{(k)}$.

Minimum Degree Algorithm.

It works with the nonzero structure of the matrix A (i.e. with a boolean matrix B with $b_{ij}=1 \Leftrightarrow a_{ij} \neq 0$, and $b_{ij}=0 \Leftrightarrow a_{ij}=0$), as follows:

1) Step $i=1, \dots, n$: we look for the row, say r , of the $(n-i+1) \times (n-i+1)$ matrix

$B_{n-i+1}^{(i)}$ (see below) with minimum number of nonzero elements.



2) Row r (and column r) is then put in "pivot" position and elimination takes place.

Elimination.

We consider the elimination involving the row r .

Let $b_{ri} = \begin{cases} 1 & \text{with probability } p_r \\ 0 & \text{with probability } 1-p_r \end{cases} \quad i=1, \dots, n, i \neq r,$

and

$b_{jh}^{(k)} = \begin{cases} 1 & \text{with probability } p_m \\ 0 & \text{with probability } 1-p_m \end{cases}$,

where $b_{jh}^{(k)}$ is the (j,h) -th element of $B^{(k)}$. Then the probability of an element of the submatrix to be nonzero, after the elimination, can be approximated by the formula

$$E(p_r, p_m) = p_m + (1-p_m)p_r^2 \quad (1).$$

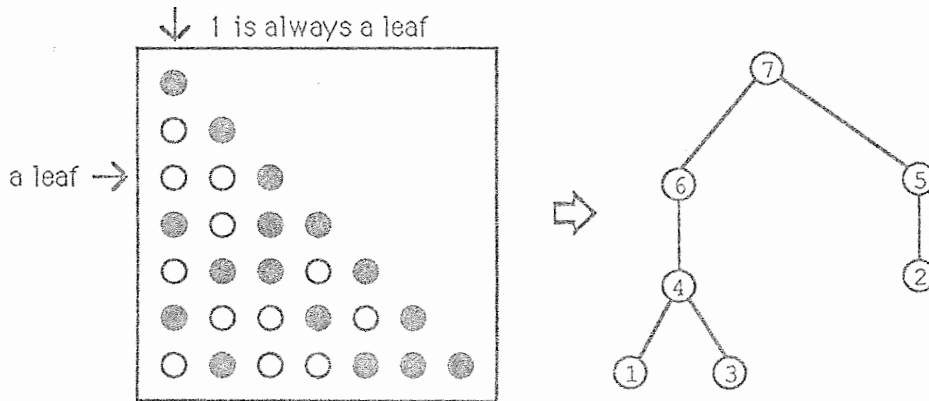
This follows from the fact that, if $b_{ij}^{(k-1)}=1$ or ($b_{ir}^{(k-1)}=1$ and $b_{jr}^{(k-1)}=1$), then $b_{jh}^{(k)}=1$. ■

Elimination Tree.

Let $L=(l_{ij})$ be an $n \times n$ irreducible triangular matrix. For each column j of L define

$$PARENT[j]=\begin{cases} \min \{i|i>j \text{ and } l_{ij} \neq 0\} & \exists k \text{ s.t. } l_{kj} \neq 0, k \neq j, \\ j & \text{otherwise.} \end{cases}$$

The elimination tree is a tree with n nodes, labelled from 1 to n , and with arcs $(j, PARENT[j])$, $j=1, \dots, n$, $j \neq PARENT[j]$. One can readily see that the arcs of the elimination tree correspond to dependences among the elimination steps, i.e. to sequential constraints.



There are at least two important measures of parallelism beside elimination trees, i.e. their depth and width (number of leaves).

- Depth: gives the number of parallel steps, where each step is an elimination process, provided that enough processors are available.
- Width : gives an upper bound to the number of processors sufficient to perform the elimination with maximum parallelism. In other words, it gives the maximum number of eliminations which can take place simultaneously.

Random matrices.

Let $\mathcal{P}\{x\}$ denote the probability of the event x .

To investigate some aspect of ordering and factoring we use a special type of random matrix [1]. In the following, an $n \times n$ matrix

$B=(b_{ij})$ with elements in $\{0,1\}$, is called random matrix of size n and probability p , if

$$\mathcal{P}\{b_{ij}=1\}=\begin{cases} p & \text{for } i \neq j \\ 1 & \text{for } i=j \end{cases} \quad 0 < p < 1.$$

Note that B is unsymmetric; this simplifies our calculus and is a reasonable assumption, being the size of n large enough [1].

3. Theoretical results.

Average number of leaves in the elimination tree.

Assume that no row and column interchanges have to be done, and that the matrix does not suffer from fill-in. In this case we have the following preliminary result.

Lemma 1.

Let p_i the probability of the node i to be a leaf, then

$$\begin{cases} p_1=1 \\ p_k = \prod_{i=0}^{k-2} (1-pq^i), \quad k=2, \dots, n, \end{cases}$$

where p is the probability of an (off-diagonal) element of the matrix to be 1, and $q=1-p$.

Sketch of the proof.

The n nodes of the elimination tree correspond to the rows of L . The first row is always a leaf. The second row is a leaf with probability q (i.e. the probability for which $l_{21}=0$); for the i -th row, one can consider the upper-left $i \times i$ submatrix of L , and combine the probability of being nonzero for l_{ir} , $r=1, \dots, i-1$, with the probability of being zero for l_{jr} , $j=1, \dots, i-1$, for any r . Then the thesis follows, since a node is a leaf if and only if the corresponding row is null (except that for the diagonal element) or each nonzero in the row has at least one (off-diagonal) nonzero in the same column and in one of the previous rows. ■

Proposition 2.

Let L be an $n \times n$ lower triangular matrix, with $\mathcal{P}\{l_{ij} \neq 0\} = p$, $i > j$, $0 < p < 1$. Then the average number of leaves in the elimination tree derived from L is

$$1 + \sum_{k=2}^n \prod_{i=0}^{k-2} (1-pq^i) \quad (2),$$

where $q=1-p$. ■

We assume that the probabilities of two nodes i and j , to be leaves, are independent. This is asymptotically true for large sparse matrices; the formula (2) gives also good approximations for small matrices.

Corollary 3.

Let n_1 be the average number of leaves. Under the hypotheses of Proposition 2 we have, for n large enough,

$$n_1 = 1 + \sum_{k=2}^n \prod_{i=0}^{k-2} (1-pq^i) \approx \frac{1}{e^{1+p/4}} \sum_{k=1}^{n-1} e^{q^k} = \alpha n, \quad \alpha \in (e^{-1}, 1). \quad \blacksquare$$

Corollary 4.

Under the hypotheses of Lemma 1 we have, for n large enough, and assuming that $p=c/n$, $c=O(1)$,

$$\lim_{n \rightarrow \infty} p_n = e^{e^{-c}-1} = \frac{e^c}{e}. \quad \blacksquare$$

From corollary 4 we can further approximate the average number of leaves in the elimination tree, as $n_1 = n\beta$, $\beta = e^{e^{-c}-1}$.

Minimum Degree.

We give two formulas that approximate the number of nonzeros in the matrices $B^{(k)}$, after each step of factorization.

A preliminary result is the following

Proposition 5.

Let $A=(a_{ij})$ be an $n \times n$ matrix, with $\mathcal{P}\{a_{ij} \neq 0\}=p$, $0 < p < 1$. Then the average number of nonzeros in the row chosen (at the first step) by MD is

$$M(n,p) = \sum_{i=0}^{n-1} \left(1 - \sum_{j=0}^i \binom{n}{j} p^j q^{n-j} \right)^n \quad (3)$$

where $q=1-p$.

Sketch of the proof.

The probability $P\{\min=m\}$ that the minimum number of nonzeros in a row of A is, say, m , can be evaluated using the equality $P\{\min=m\}=P\{\min \geq m\}-P\{\min \geq m+1\}$. Since $\min \geq k$ if and only if each row of the matrix has at least k nonzeros, then the thesis follows by summing up all the terms $h P\{\min=k\}$ for $h=0, \dots, n$. ■

When we choose a row with $M(n,p)$ nonzeros, the probability of an element of the submatrix $B^{(k)}$ to be nonzero is

$$R(n,p) = \frac{n(n-1)p - 2M(n,p)}{(n-1)(n-2)} \quad (4)$$

Starting from (1), (3) and (4), one can obtain the formula:

$$\begin{cases} p_0 = p \\ p_{i+1} = E(M(n-i+1, p_i), R(n-i+1, p_i)) \\ n_i = (n-i)p_i \end{cases} \quad i=1, \dots, n-1. \quad (5)$$

This formula gives an approximation, from below, to the number of nonzeros in each row (and column).

The error introduced by (5) depends on

- i) the uniform (unsymmetric) distribution assumed for the pattern of A ;
- ii) the underestimate of the value of the minimum chosen by MD.

The largest error we make is due to the properties of the MD; in fact, when we apply MD, the number of nonzeros in the row chosen, can not be less than the number of nonzeros in the row selected by the previous application, minus one. In the following we obtain another approximation, based on the last argument.

Let us consider an $m \times n$ random matrix. We define :

$P_j(m, n, p)$, as the probability that the minimum number of nonzeros in a row is j ;

$P_{j,k}(n, p)$, as the probability that the minimum number of nonzeros at the k -th step is j ;

$M_k(n, p)$, the expected minimum number of nonzeros at the k -th step;

$E_j(n, p)$ is like E , with the additional constraint that the number of nonzeros in the row eliminated is j ;

$R_j(n, p)$ is like R , with the additional constraint that the row eliminated contains j nonzeros;

The following equalities hold:

$$R_j(n,p) = \frac{n \cdot (n-1) \cdot p - 2 \cdot j}{(n-1)(n-2)},$$

$$E_j(n,p) = p + (1-p)(j/n)^2,$$

$$P_j(m,n,p) = \left(1 - \sum_{i=0}^{j-1} \binom{n}{i} p^i (1-p)^{n-i} \right)^m - \left(1 - \sum_{i=0}^j \binom{n}{i} p^i (1-p)^{n-i} \right)^m.$$

Now, let us suppose that the MD after k steps is j , with probability $P_{j,k}(n,p)$; hence, each row contains at least j nonzeros. Using R_j and E_j we first simulate the elimination of the selected row and then we calculate the probability that the next MD is $h \geq j$, as the probability that the MD of a matrix of size $(n-k) \times (n-k-j-1)$ is $h-j$, i.e. $P_{h-j}(n-k+1, n-k-j, p')$, where p' is appropriately chosen.

In this way, we have calculated the probability that the MD is h , provided that the previous one was j . Summing up for all $j \leq h$, we obtain a formula for $P_{h,k+1}(n,p)$.

More formally, we have:

$$M_k(m,p) = \sum_{j=0}^{m-k} j \cdot P_{j,k}(m,p), \quad k=1, \dots, m-1,$$

$$\begin{cases} P_{j,1}(m,p) = P_j(m, m-1, p, j), \\ P_{j,k+1}(m,p) = \sum_{i=0}^j P_{i,k}(m,p) \cdot P_{j-i}(m-k, m-k-i-1, r_k(m,p,i)), \quad k=1, \dots, m-1, \end{cases}$$

$$r_k(m,p,i) = \frac{(m-k+1) \cdot E_i(m-k+1, R_i(m-k+1, p_{k-1}(m,p))) - i}{m-k-i-1}, \quad k=1, \dots, m-1,$$

$$\begin{cases} p_0(m,p) = p, \\ p_k(m,p) = E(M_k(m,p), R'_k(m,p)), \quad k=1, \dots, m-1, \end{cases}$$

$$R'_k(m,p) = \frac{(m-k+1)(m-k)p_{k-1}(m,p) - 2M_k(m,p)}{(m-k)(m-k-1)}, \quad k=1, \dots, m-1.$$

These formulas allow to estimate the probability of an element of $B^{(k)}$ to be nonzero, i.e. the average fill-in. Note that one can combine these formulas with (2) and obtain an evaluation of the number of leaves in the elimination tree after MD.

4. Experimental results.

We have tested the behaviour of elimination trees. Experiments have been performed on matrices of size from 50x50 to 600x600, with random (uniform) distribution of nonzeros, and $p=c/n$, $c=1,2,4,8$.

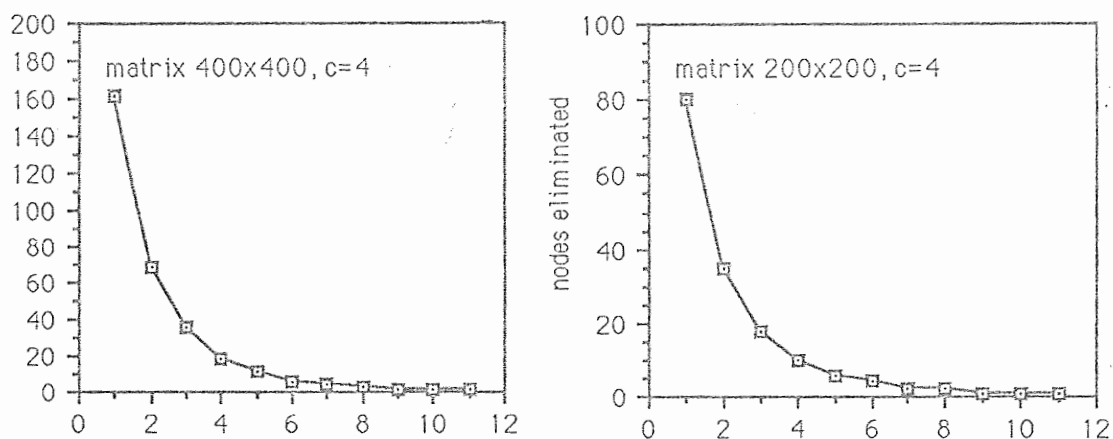


Fig.1. Number of nodes eliminated versus the number of steps.

In fig. 1 we have plotted the average number of nodes eliminated, i.e. the number of leaves of the elimination tree, versus the elimination steps, with the condition that, at each step, we eliminate as many nodes as possible. It turns out that the average number of nodes eliminated approaches 1 exponentially.

It is possible to introduce a formula that expresses the average numbers n_f of leaves eliminated at the first step, i.e.

$$n_f \approx \frac{n}{2} - \frac{n \log_2 c}{10}, \quad 2 < c < 50.$$

5. Conclusions.

The main results of the paper are probabilistic evaluations of the structure of elimination tree and of the MD method.

A number of computational consequences follows, related to the parallel average-case complexity of sparse linear systems solvers.

It remains an open question whether or not stages 1)-4) of section 1 could be performed in polylogarithmic time. About this problem we conjecture that the average depth of an elimination tree is $\theta(n)$, i.e all these methods require at least linear time.

6. References.

- [1] B. Bollobás, *Random Graphs*, Academic Press Inc. (London) LTD, 1985.
- [2] E. Dahlhaus, and M. Karpinski, *An Efficient Parallel Algorithm for the Minimal Elimination Ordering (MEO) of an Arbitrary Graph*, Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, pp. 454-459.
- [3] I.S. Duff, *On the Number of Nonzeros Added when Gaussian Elimination is Performed on Sparse Random Matrices*, Mathematics of Computation 28 (1974), pp. 219-230.
- [4] A. George, and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1981.
- [5] A. George, and J. Liu, *The Evolution of Minimum Degree Ordering Algorithm*, SIAM Review 31 (1989), pp. 1-19.
- [6] G.H. Golub, and C.F. Van Loan, *Matrix Computations*, J. Hopkins University Press, Baltimore, MD, 1983.
- [7] H.Y. Hsieh, and M.S. Ghausi, *A Probabilistic Approach to Optimal Pivoting and Prediction of Fill-In for Random Sparse Matrices*, IEEE Trans. on Circuit Theory Ct-19 (1972), pp. 329-336.
- [8] M. Yannakakis, *Computing the Minimum Fill-In is NP-complete*, SIAM J. Alg. Disc. Meth. 2 (1981), pp. 77-79.