Integer LU-Factorizations

Jean H. Bevis and Frank J. Hall
Department of Mathematics and Computer Science
Georgia State University
Atlanta, Georgia 30303

Submitted by Wayne Barrett

ABSTRACT

Let A be an m-by-n integer matrix and $r = \text{rank}(A)$. A necessary and sufficient condition is given for A to have an integer LU-factorization, and a modification of Gaussian elimination is given for finding such factorizations when the first r leading principal minors are nonzero. An algorithm is given for ordering the vertices of a tree with a loop at its root so that its adjacency matrix has an integer LU-factorization.

INTRODUCTION

In the first section we consider factorizations $A = LDU$ of an integer matrix A into integer matrices L, D, and U where L is lower triangular, D is diagonal, and U is upper triangular. By restricting the diagonal entries of L and U we define special types of such factorizations. Conditions are given for the existence of these factorizations, and a modification of Gaussian elimination is presented for finding the special types of integer LU-factorizations of a rectangular matrix when they exist. In the second section we give a method for ordering the vertices of a tree with a loop at one vertex so that its adjacency matrix will have an integer LU-factorization.

I. INTEGER FACTORIZATIONS

Let A be an m-by-n matrix. If there are integer matrices L, U, and D where L is m-by-t lower triangular, U is t-by-n upper triangular, and D is t-by-t diagonal such that $A = LU$ or $A = LDU$, then we say that A has an

655 Avenue of the Americas, New York, NY 10010 0024-3795/91/$3.50

integer LU-factorization. Any common integer factor of a column of L or a row of U may be moved to the corresponding position in D. When this process is completed, the greatest common divisor of the entries in each column of L and each row of U will be 1, and we say that the factorization has been normalized. We say that $A = LDU$ is a left (right) unit LU-factorization if the diagonal entries of L (U) are all 1's. Finally, $A = LDU$ is called a unit LU-factorization if the diagonal entries of both L and U are 1's.

In finding an LU-factorization of an m-by-n matrix A by Gaussian elimination, L is usually taken to be square with a unit diagonal. With no row or column interchanges A appears at the beginning of step k as

$$A = \begin{bmatrix} L'_k & 0 \\ X_k & I_{m-k} \end{bmatrix} \begin{bmatrix} U'_k & Y_k \\ 0 & S_k \end{bmatrix}, \qquad k = 0, 1, 2, \dots,$$

where L'_k is k-by-k lower triangular, U'_k is k-by-k upper triangular, and I_{m-k} is the $(m-k)$-by-$(m-k)$ identity matrix. Let s_k denote the $(1,1)$ entry of S_k. When $s_k \neq 0$ it can be used as the pivot in the next stage of the elimination. If $s_k \neq 0$ for $k = 0, 1, 2, \dots, r-1$, where $r = \operatorname{rank}(A)$, then S_r is a zero matrix or empty, and the elimination proceeds to

$$A = \begin{bmatrix} L'_r & 0 \\ X_r & I_{m-r} \end{bmatrix} \begin{bmatrix} U'_r & Y_r \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} L'_r \\ X_r \end{bmatrix} \begin{bmatrix} U'_r & Y_r \end{bmatrix},$$

which gives an LU-factorization where L is m-by-m or m-by-r. If we partition such a matrix as

$$A = \begin{bmatrix} A_k & B_k \\ C_k & G_k \end{bmatrix}$$

where A_k is k-by-k, $1 \leqslant k \leqslant r$, then $S_k = G_k - C_k A_k^{-1} B_k$ and is called the kth Schur complement of A. Since A_0 is empty, we take $S_0 = G_0 = A$ and $\det(A_0) = 1$. Also $s_k = \det(A_{k+1})/\det(A_k)$.

Integer factorizations $A = LDU$ are possible where one or more of the diagonal entries of L, D, or U are zero, as in

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix},$$

where $\operatorname{rank}(A) = 2$. This matrix has no LU-factorization where the diagonal entries of either L or U are nonzero [5, Theorem 2]. We do not know, in general, how to modify Gaussian elimination to obtain an integer LU-factorization of this type of matrix. Rather our modification deals with divisibility conditions which allow calculations to be done over the integers.

We modify Gaussian elimination by requiring that the pivot s_k be factored, if possible, as $s_k = p_k q_k$ where p_k divides the entries in the first row of S_k, q_k divides the entries in the first column of S_k, and 0 is not allowed as a divisor of 0. If there is no such factorization of s_k or $s_k = 0$, and if S_k is not upper triangular, then the elimination terminates without finding an LU-factorization of A. Next p_k is factored from the first row of S_k and put on the diagonal of L, leaving q_k as the new pivot. Since q_k divides the first column of S_k, integral multiples of the first row can be subtracted from the other rows of S_k to perform the elimination. This procedure is made explicit in the algorithm below. The proof of Theorem 1 below shows that at each step of the algorithm $A = L_k U_k$ where each L_k is an m-by-m lower triangular matrix. If sufficient elimination steps are completed so that some U_k is upper triangular, then an integer LU-factorization is obtained. When there is more than one factorization of $s_k = p_k q_k$ satisfying the above requirements, the proof of Theorem 1 also shows that we may use any one of them, if we do not necessarily want a unit factorization.

Algorithm 1 (Integer Gaussian elimination). Given an integer m-by-n rank-r matrix A, let

$$S_0 = A, \qquad L_0 = \begin{bmatrix} L_0' & 0 \\ X_0 & I_m \end{bmatrix}, \quad \text{and} \quad U_0 = \begin{bmatrix} U_0' & Y_0 \\ 0 & S_0 \end{bmatrix},$$

where L_0', X_0, U_0', and Y_0 are empty matrices.

Repeat the following if possible for $k = 0, 1, \dots, \min(r-1, m-2)$: If $s_k \neq 0$, let $q_k = \gcd(\mathrm{Col}_1(S_k))$ be the greatest common divisor of the entries in the first column of S_k, and take the sign of q_k to be the same as that of s_k, the $(1,1)$ entry of S_k. Set $p_k = s_k / q_k$ so that $s_k = p_k q_k$. If p_k is not a divisor of each entry in row 1 of S_k stop, otherwise S_k has the form

$$S_k = \begin{bmatrix} p_k q_k & p_k \beta_k^T \\ \alpha_k q_k & W_k \end{bmatrix}$$

for integer matrices α_k, β_k, and W_k. Partition X_k and Y_k as

$$X_k = \begin{bmatrix} \varepsilon_k^T \\ X_k' \end{bmatrix} \quad \text{and} \quad Y_k = \begin{bmatrix} \delta_k & Y_k' \end{bmatrix},$$

where ε_k^T is the first row of X_k and δ_k is the first column of Y_k. Finally set

$$L'_{k+1} = \begin{bmatrix} L'_k & 0 \\ \varepsilon_k^T & p_k \end{bmatrix}, \qquad X_{k+1} = \begin{bmatrix} X'_k & \alpha_k \end{bmatrix},$$

$$L_{k+1} = \begin{bmatrix} L'_{k+1} & 0 \\ X_{k+1} & I_{m-k-1} \end{bmatrix}, \qquad S_{k+1} = W_k - \alpha_k \beta_k^T,$$

$$Y_{k+1} = \begin{bmatrix} Y'_k \\ \beta_k^T \end{bmatrix}, \qquad U'_{k+1} = \begin{bmatrix} U'_k & \delta_k \\ 0 & q_k \end{bmatrix}, \quad \text{and} \quad U_{k+1} = \begin{bmatrix} U'_{k+1} & Y_{k+1} \\ 0 & S_{k+1} \end{bmatrix}.$$

If $r = m$ (the repeated step above does not determine q_{r-1} and p_{r-1}), set $q_{r-1} = s_{r-1}$ and $p_{r-1} = 1$.

End of algorithm.

THEOREM 1. *Let A be an m-by-n rank-r integer matrix, and suppose that the first r leading principal minors are nonzero. Let $s_k = p_k q_k$, $k = 0, 1, 2, \dots, r-1$, be the factorization of the pivots determined by Algorithm 1, so that $q_k = \gcd(\mathrm{Col}_1(S_k))$ and $p_k = s_k / q_k$. Then A has an integer LU-factorization $A = LDU$, where D is r-by-r, if and only if p_k divides every entry in the first row of S_k for $k = 0, 1, \dots, r-1$. Furthermore,*

(i) *A has a unit LU-factorization $A = LDU$, where D is r-by-r, if and only if s_k divides every entry in the first row and column of S_k for $k = 0, 1, \dots, r-1$;*

(ii) *A has a left (right) unit LU-factorization $A = LDU$, where D is r-by-r, if and only if s_k divides every entry in the first column (row) of S_k for $k = 0, 1, \dots, r-1$.*

Proof. To obtain the sufficiency of the conditions on the s_k we note at the beginning of step k that A has the form

$$A = L_k U_k = \begin{bmatrix} L'_k & 0 \\ X_k & I_{m-k} \end{bmatrix} \begin{bmatrix} U'_k & Y_k \\ 0 & S_k \end{bmatrix} \tag{1}$$

where L'_k and U'_k are k-by-k lower and upper triangular nonsingular integer matrices, respectively. For $k = 0$ such a factorization is given by $A = I_m S_0$. If A has an integer factorization of form (1) and $s_k = p_k q_k$, where $q_k =$

$\gcd(\mathrm{Col}_1(S_k))$ and p_k divides $\mathrm{Row}_1(S_k)$, then using the notation of Algorithm 1,

$$A=\begin{bmatrix} L'_k & 0 \\ X_k & I_{m-k} \end{bmatrix}\begin{bmatrix} U'_k & \multicolumn{2}{c}{Y_k} \\ 0 & \begin{matrix} p_k q_k \\ q_k\alpha_k \end{matrix} & \begin{matrix} p_k\beta_k^T \\ W_k \end{matrix} \end{bmatrix}$$

$$=\begin{bmatrix} L'_k & 0 \\ X_k & I_{m-k} \end{bmatrix}\begin{bmatrix} I_k & \multicolumn{2}{c}{0} \\ 0 & \begin{matrix} p_k \\ \alpha_k \end{matrix} & \begin{matrix} 0 \\ I_{m-k-1} \end{matrix} \end{bmatrix}\begin{bmatrix} U'_k & \multicolumn{2}{c}{Y_k} \\ 0 & \begin{matrix} q_k \\ 0 \end{matrix} & \begin{matrix} \beta_k^T \\ W_k-\alpha_k\beta_k^T \end{matrix} \end{bmatrix}$$

$$=\begin{bmatrix} L'_k & \multicolumn{2}{c}{0} \\ X_k & \begin{matrix} p_k \\ \alpha_k \end{matrix} & \begin{matrix} 0 \\ I_{m-k-1} \end{matrix} \end{bmatrix}\begin{bmatrix} U'_k & \multicolumn{2}{c}{Y_k} \\ 0 & \begin{matrix} q_k \\ 0 \end{matrix} & \begin{matrix} \beta_k^T \\ S_{k+1} \end{matrix} \end{bmatrix}$$

$$=\begin{bmatrix} L'_k & 0 & \\ \varepsilon_k^T & p_k & 0 \\ X'_k & \alpha_k & I_{m-k-1} \end{bmatrix}\begin{bmatrix} U'_k & \delta_k & Y'_k \\ 0 & q_k & \beta_k^T \\ \multicolumn{2}{c}{0} & S_{k+1} \end{bmatrix}=L_{k+1}U_{k+1}.$$

Thus if p_k divides $\mathrm{Row}_1(S_k)$ for $k=0,1,\dots,r-1$, we may proceed by induction to obtain a factorization of the form (1) with $k=r=\mathrm{rank}(A)$. Now S_r cannot have a nonzero row; otherwise $\mathrm{rank}(A)>r$. Hence we obtain the integer LU-factorization $A=LV$ with

$$L=\begin{bmatrix} L'_r \\ X_r \end{bmatrix},\qquad V=\begin{bmatrix} U'_r & Y_r \end{bmatrix},$$

p_k's on the diagonal of L, and q_k's on the diagonal of V.

Note that so far we have only used the condition that $s_k=p_kq_k$ where p_k divides $\mathrm{Row}_1(S_k)$ and q_k divides $\mathrm{Col}_1(S_k)$. Thus it is possible to use this more general factorization of s_k in Algorithm 1 instead of requiring that $q_k=\gcd(\mathrm{Col}_1(S_k))$. We may now factor V as $V=DU$, where D is an r-by-r diagonal matrix whose diagonal entries d_k are the greatest common divisors of the corresponding rows of V, to obtain the integer factorization $A=LDU$. We take the sign of d_k to be the same as the sign of s_k. Now if s_k divides $\mathrm{Col}_1(S_k)$ [s_k divides $\mathrm{Row}_1(S_k)$], $k=0,1,\dots,r-1$, then $q_k=s_k$ and $p_k=1$ [$d_k=q_k$] and we obtain a left [right] unit LU-factorization. If s_k divides both

$\operatorname{Row}_1(S_k)$ and $\operatorname{Col}_1(S_k)$, $k = 0, 1, \dots, r-1$, then $p_k = 1$ and q_k may be factored from row $k+1$ of V to obtain a unit LU-factorization.

Conversely, let $A = MDV$ be an integer LU-factorization where D is r-by-r. Partition M, D, and V as

$$M = \begin{bmatrix} M_1 & 0 \\ M_2 & M_3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \quad V = \begin{bmatrix} V_1 & V_2 \\ 0 & V_3 \end{bmatrix},$$

where M_1, D_1, and V_1 are k-by-k. Then

$$A = \begin{bmatrix} M_1 D_1 V_1 & M_1 D_1 V_2 \\ M_2 D_1 V_1 & M_2 D_1 V_2 + M_3 D_2 V_3 \end{bmatrix}.$$

Also, at the beginning of step k of the algorithm,

$$A = \begin{bmatrix} L_k' & 0 \\ X_k & I_{m-k} \end{bmatrix} \begin{bmatrix} U_k' & Y_k \\ 0 & S_k \end{bmatrix} = \begin{bmatrix} L_k' U_k' & L_k' Y_k \\ X_k U_k' & X_k Y_k + S_k \end{bmatrix}.$$

Equating the corresponding parts of these two partitions of A, we obtain $M_l D_l V_l = L_k' U_k'$, and over the rational numbers

$$X_k Y_k = X_k U_k' (L_k' U_k')^{-1} L_k' Y_k = M_2 D_1 V_1 (M_1 D_1 V_1)^{-1} M_1 D_1 V_2 = M_2 D_1 V_2.$$

Thus $S_k = M_3 D_2 V_3$ and $s_k = pdq$, where p, d, and q are the $(1,1)$ entries of M_3, D_2, and V_3 respectively. Also, the first row of S_k is $\operatorname{Row}_1(S_k) = \operatorname{Row}_1(M_3 D_2) V_3 = pd \operatorname{Row}_1(V_3)$, and the first column of S_k is $\operatorname{Col}_1(S_k) = M_3 \operatorname{Col}_1(D_2 V_3) = dq \operatorname{Col}_1(M_3)$. Thus q_k is a multiple of dq, p_k is a divisor of p, and thus p_k is a divisor of $\operatorname{Row}_1(S_k)$. This analysis holds for any $k = 0, 1, \dots, r-1$, so the condition on the p_k's is necessary for A to have an integer LU-factorization LDU where D is r-by-r. Similarly, if $A = MDV$ is a unit LU-factorization, then $s_k = d$, which divides both the first row and first column of S_k. If this is a left [right] unit LU-factorization, then $s_k = dq$ [$s_k = pd$], which divides the first column [row] of S_k. ∎

Corollary 1. *An integer rank-r matrix with r nonzero leading principal minors has a unit LU-factorization if and only if it has both a left and a right unit LU-factorization.*

A modification of the converse portion of the proof of Theorem 1 shows that:

COROLLARY 2 (Uniqueness). *If $A = LDU = L'D'U'$ are two integer LU-factorizations of the integer rank-r matrix A where L, L', D, D', U, and U' have nonzero diagonal entries, D and D' are r-by-r, the greatest common divisors of the entries of each column of L and L' are* 1, *and the greatest common divisors of the entries of each row of U and U' are* 1, *then $L = L'$, $D = D'$, $U = U'$.*

Let A_k denote the k-by-k leading principal submatrix of a rank-r matrix A. If A has an integer LU-factorization $A = LDU$ where D is r-by-r, and the first r leading principal minors are nonzero, then $\det(A_k)$ divides $\det(A_{k+1})$ for $k = 1,2,\ldots,r-1$. The converse is true for 2-by-2 matrices, but not true in general. For example, if

$$A = \begin{bmatrix} 6 & -2 & 1 \\ 3 & 1 & 1 \\ 1 & -1 & 2 \end{bmatrix},$$

then the leading principal minors are 6, 12, and 24, but A has no integer LU-factorization. However if rows 1 and 3 are interchanged, then the resulting matrix has a left unit factorization. We also note that

$$\begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$$

is an example of a unimodular matrix which does not have an integer LU-factorization no matter how the rows and columns are arranged. However, if an n-by-n unimodular matrix A does have an integer factorization $A = LDU$ where D is n-by-n, then this must be a unit factorization, since $\pm 1 = \det(A) = \det(L)\det(D)\det(U)$.

Let $A = LDU$ be an integer LU-factorization of A, where L, D, and U may have zeros on the diagonal. Partition L into its columns $L_{(k)}$, partition U into its rows $U^{(k)}$, and let $D = \operatorname{diag}[d_1, d_2, \ldots, d_t]$. Then $A = LDU = \sum_{k=1}^{t} d_k L_{(k)} U^{(k)}$, which, once the zero terms are omitted, is a linear combination of rank-1 matrices. Furthermore, the first $k-1$ rows and columns of the kth term in this sum, $d_k L_{(k)} U^{(k)}$, contain only zeros. DeCaen and others [3] have made effective use of such decompositions in the study of binary factorizations, that is, factorizations $A = BC$ into 0, 1 matrices. Such decompositions are useful in studying matrices associated with combinatorial objects. Theorem 2 below captures the spirit of such decompositions and is useful in studying the adjacency matrices of graphs. The proof of Lemma 1 is straightforward and thus omitted.

DEFINITION. A sequence of matrices $B_1, B_2, \dots, B_t$ of the same size is said to be *shrinking* if the first $k-1$ rows and columns of B_k contain only zero entries, $k = 1, 2, \dots, t$. If in addition the (k,k) entry of each B_k is nonzero, we say that the sequence is *descending*.

LEMMA 1. *Any rank-1 integer m-by-n matrix B has an integer LU-factorization $B = LU$ where L is a m-by-1 matrix and U is a 1-by-n matrix.*

THEOREM 2. *An integer matrix has an integer LU-factorization if and only if it is a shrinking sum of rank-1 integer matrices. It has an integer LU-factorization where the diagonal entries of L and U are nonzero if and only if it is a descending sum of rank-1 integer matrices.*

Proof. If A has an integer LU-factorization $A = LU = \Sigma_{k=1}^{t} L_{(k)} U^{(k)}$, then once zero terms are omitted the remaining terms may be renumbered as $B_1, B_2, \dots$ so that $B_i = L_{(k)} U^{(k)}$ with $i \leqslant k$. Now B_i contains only zeros in the first $k-1$ rows and columns and thus in the first $i-1$ rows and columns. A is thus a shrinking sum of rank-1 integer matrices, and if the diagonal entries of L and U are nonzero, this sum is descending. Conversely, if $A = \Sigma_{k=1}^{t} B_k$ is a shrinking sum of rank-1 integer matrices B_k, then by Lemma 1 there are integer column matrices L_k and row matrices U_k such that

$$A = \sum_{k=1}^{t} L_k U_k = \begin{bmatrix} L_1 & \cdots & L_t \end{bmatrix} \begin{bmatrix} U_1 \\ \vdots \\ U_t \end{bmatrix} = LU.$$

Since $\Sigma_k B_k$ is a shrinking sum, the first $k-1$ entries of L_k and U_k can be taken to be 0's, so that L is lower triangular and U is upper triangular. If the sum is descending, then the diagonal entries of L and U are nonzero.

II. ADJACENCY MATRICES OF TREES

Matrices associated with combinatorial objects such as graphs, networks, and block designs would seem to be important application areas for integer LU-factorizations. In a later paper we will report on some results concerning unit LU-factorizations of incidence matrices for graphs and block designs. For now consider the adjacency matrix $A = \text{adj}(G)$ of a graph $G = (V, E)$ where the vertices have been ordered, $V = \{v_1, v_2, \dots, v_n\}$, and $A = (a_{ij})$ with $a_{ij} = 1$ if $(v_i, v_j) \in E$ is an edge of the graph, otherwise $a_{ij} = 0$. Ordinarily A would not have an integer LU-factorization, but if the vertices have been properly ordered such a factorization maybe possible. We prefer to apply the same permutation to both the rows and columns of A, since they are both indexed by the vertices of G. For many graphs the diagonal of A consists entirely of 0's, and there is no way to start the factorization process for any ordering of the vertices [5, Corollary 1]. The addition of isolated vertices and loops [an edge of the form (v_i, v_i)] are two methods of dealing with this problem. For example, if G consists of an isolated vertex v_1 and the complete graph on $v_2, \dots, v_n$, then its adjacency matrix may be factored as

$$A = \begin{bmatrix} 0 & 0 \\ e & I_{n-1} \end{bmatrix} \begin{bmatrix} 0 & e^T \\ 0 & -I_{n-1} \end{bmatrix},$$

where I_{n-1} is the $(n-1)$-by-$(n-1)$ identity matrix and $e = [1 \ 1 \ \cdots \ 1]^T$ is a column of 1's. If G consists of a complete graph with a loop at each vertex, then its adjacency matrix may be factored as $A = ee^T$.

The problem of ordering the vertices of an arbitrary graph so that its adjacency matrix has an integer LU-factorization seems to be nontrivial. If the $(1,1)$ entry of the adjacency matrix is nonzero, then the graph must have a loop at the first vertex. For a tree with a loop added at one vertex, we have found that it is possible to order the vertices so that the adjacency matrix has an integer LU-factorization. The algorithm below gives such an ordering. Depending on the structure of the tree, there may be several orderings of the vertices which are consistent with this algorithm, and there may be orderings which yield integer LU-factorizations of the adjacency matrix which are not consistent with the algorithm. The orderings provided by this algorithm are *specialized* depth-first-search orderings that ensure the pivots are ± 1 until the factorization is complete. Thus, ordinary Gaussian elimination can be used to find an LU-factorization over the integers.

A result of F. Harary [4] is useful in determining when 0 occurs in the position of a pivot. This result states that the determinant of the adjacency matrix of a graph may be expanded as the sum of the determinants of the adjacency matrices of its linear subgraphs. A linear subgraph is a spanning subgraph whose components are edges or cycles (a loop is considered to be 1-cycle). Furthermore, the determinant of the adjacency matrix of a linear subgraph is $(-1)^e 2^c$, where e is the number of components with an even number of vertices and c is the number of components with more than two vertices.

LEMMA 2. *If G is a tree with n vertices which has a loop added at one vertex and $B = \operatorname{adj}(G)$ is the adjacency matrix of G, then* $\det(B) = 0$ *if G has no linear subgraph; otherwise,* $\det(B) = (-1)^e$, *where $e = n/2$ if n is even or $e = (n-1)/2$ if n is odd.*

Proof. G has at most one linear subgraph and no cycles with more than two vertices. Thus the sum given in [4,(8)] for $\det(B)$ is 0 if G has no linear subgraph, or $(-1)^e$ is G has a linear subgraph. In the latter case the linear subgraph consists of $e = n/2$ disjoint edges when n is even, or the loop and $e = (n-1)/2$ disjoint edges when n is odd.

We also need a modification of F. Harary's result for adjacency matrices with a row and a column deleted. For simplicity, let $1, 2, \dots, n$ be the vertices of a graph G, and let B be the $(n-1)$-by-$(n-1)$ matrix obtained by deleting the jth row and ith column $(i \neq j)$ of the adjacency matrix of G. A modification of F. Harary's proof shows that $\det(B)$ is the sum of the determinants of adjacency matrices with row j and column i deleted of subgraphs H which contain a path p from vertex i to vertex j such that $H - p$ is a linear subgraph of $G - p$. Furthermore, if H is such a subgraph of G and p contains m vertices of G, then the determinant of the adjacency matrix of H with row j and column i deleted is $(-1)^{m+|i-j|-1}$ times the determinant of the $(n-m)$-by-$(n-m)$ adjacency matrix of $H - p$. Now if G is a tree with a loop at one vertex called the root, then G has at most one such subgraph H, and $H - p$ contains no cycles with more than two vertices. Thus we obtain:

LEMMA 3. *If G is a tree with a loop at one vertex called the root, and B is obtained from the adjacency matrix of G by deleting the jth row and ith column, $i \neq j$, where neither i nor j corresponds to the root, then* $\det(B) = \pm 1$ *if G has a subgraph H which contains a path p from the ith vertex to the jth vertex such that $H - p$ is a linear subgraph of $G - p$; otherwise* $\det(B) = 0$.

Let G be a tree with a loop at one vertex, and let $r = \operatorname{rank}(\operatorname{adj}(G))$. If the first r leading principal minors are to be nonzero, then, according to Lemma 2, we must order the vertices of G as $V = \{1, 2, \dots, n\}$ so that for each $k \leqslant r$, the subgraph G_k spanned by vertices $\{1, 2, \dots, k\}$ has a linear subgraph. By Lemma 4 below, this is obtained by the ordering of Algorithm 2. It is well known, and follows directly from the result of F. Harary [4] noted above, that the rank of the adjacency matrix of a tree (without a loop) is the number of vertices in the largest subtree which has a linear subgraph. This is also true of a tree with a loop at one vertex, since the value for r produced by Algorithm 2 is the rank of adj(G) by Theorem 3, and there is a subtree G_r with r vertices which has a linear subgraph by Lemma 4. There is no larger subtree with a linear subgraph, since by Lemma 2 the principal minor of adj(G) corresponding to its vertices would be nonzero, but by Theorem 3 there is no larger nonzero principal minor.

By Lemma 2, the adjacency matrix of a tree (with or without a single loop) with an even number of vertices is nonsingular if and only if the tree has a linear subgraph. In this case the linear subgraph consists of a set of disjoint edges which span the tree and is called a perfect matching. For each leaf (a vertex of degree 1) v of the tree, such a matching must contain the edge from v to its only neighbor, which is called the parent of v when v is not the root. When two leaves have the same parent, the required edges are not disjoint and there is no perfect matching. For an arbitrary vertex this situation is described in a result of Caro and Schonheim [2, Theorem 2] as "A tree has a perfect matching if and only if at each vertex there is exactly one branch with an odd number of vertices." (A branch at a vertex v is a connected component of the tree with vertex v removed.) A perfect matching of a tree pairs each vertex v with a vertex in the odd branch of v.

A general overview of Algorithm 2 follows. Here the parity of branches refers to their parity in a largest subtree which has a linear subgraph, in particular for the subgraph which becomes known as G_r. Information about the parity of branches in G_r is obtained in step 0 of Algorithm 2 by the "even"/"odd" labeling process. However, this is not apparent until the proof of Theorem 3 is completed. Step 1 of the algorithm causes the root to be numbered vertex 1 and calls a procedure which numbers the vertices of the subgraph which becomes G_r. When the root is labeled even, it is paired with a child in an odd branch by the procedure call in step 1a. When the root is labeled odd, all of its children belong to even branches, so the root is covered in the linear subgraph by the loop, and its descendants $2,\dots,r$ are ordered by the procedure calls in step 1b. Finally the remaining vertices are ordered by step 3.

Next consider the tree of Figure 1(a). If OrdDesc(u) is called, and k has the value $h-1$ at that time, then u is numbered as vertex h. If, as in Figure 1(b), c_1 is numbered as vertex $h+1$ and c_2 as vertex $h+2$ by the call OrdDesc(c_1) in step 3, then (u,c_1) is an edge of linear subgraph of G_{h+1} but is not an edge of any linear subgraph of G_j for $j>h+1$. However, if, contrary to the procedure in Figure 1(c), v is numbered as vertex $h+1$, c_5 as vertex $h+2$, and c_6 as vertex $h+3$, then (u,v) is an edge of any linear subgraph of G_{h+1} and G_j for any $j \geqslant h+3$ (v heads an odd branch at u). In particular, for the ordering of Figure 1(c), G_{h+4} has no linear subgraph.

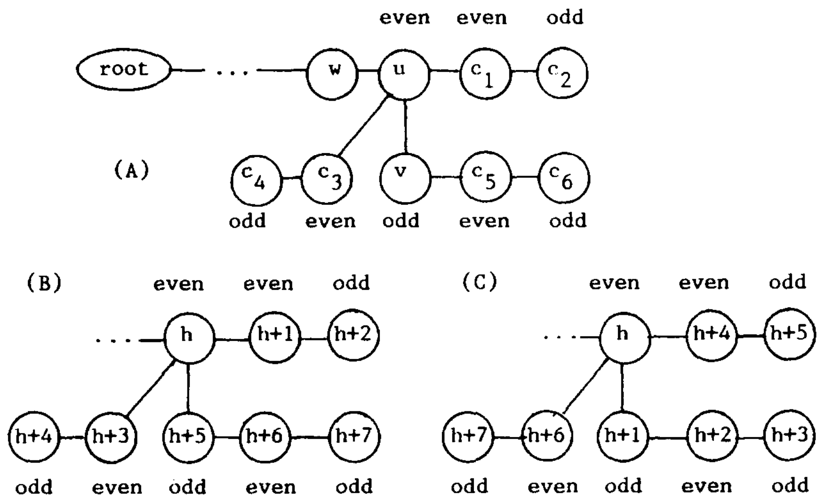


FIG. 1.

Thus, c_1, c_2 and similarly c_3, c_4 must precede v in the ordering. Hence, OrdDesc(u) identifies vertex v in step 2a as a vertex to be numbered only after descendants of u by other children have been numbered by procedure calls in step 3. The calls in step 3 increase the value of k by an appropriate amount. Vertex v is then numbered ($h+5$ in this example) in step 4, and the descendants of v are numbered by procedure calls in step 6. Note also that the orderings of the descendants of u in both Figure 1(b) and 1(c) are depth-first-search orderings, but as noted above, the ordering of Figure 1(c) cannot be obtained from Algorithm 2. In fact, for the depth-first-search ordering of Figure 1(c), the leading $(h+4)$-by-$(h+4)$ principal minor of the adjacency matrix is zero, and by [5, Corollary 1] the adjacency matrix has no LU-factorization with a unit diagonal.

Step 5 of the procedure is unnecessary for its operation, but is used in the proof of Theorem 3. In the proof of Lemma 4 we show that the marked edges (together with the loop when r is odd) form a linear subgraph of G_r. Also, if p is the path in G_k from vertex 1 to vertex k, where $1 \leqslant k < r$, then a linear subgraph of G_k consists of the loop when k is odd, the marked edges in $G - p$, and edges of p which may or may not be marked. See, for example, the graphs of Figure 2.

The level of a vertex v, Level(v), in a tree is defined as the number of edges in the simple path from v to the root. Our algorithm for ordering the vertices of a tree is then:

Algorithm 2 (A vertex ordering). Let t be the root of the tree (the vertex with a loop) with n vertices, and let K be the maximum level of the vertices.

0. For $k = K, K-1, \dots, 0$ label each level-k vertex v as odd if v is a leaf or all of its children are labeled even; otherwise label v as even.

1a. If t is labeled even, then set $k = 0$ and call the recursive procedure OrdDesc(t);

1b. otherwise, let t be vertex 1, set $k = 1$, and for each child c of t call OrdDesc(c).

2. Let r be the value of k when the procedure ends.

3. Order any remaining vertices as $r+1, \dots, n$ in any fashion.

4. End of algorithm.

Procedure OrdDesc(u):

1. Increase k by 1, and let u be the kth vertex.

2a. If possible, choose an odd child v of u;

2b. otherwise, choose an even child v of u.

3. For each child c of u such that c is not a leaf and $c \neq v$, call OrdDesc(c).

4. Increase k by 1, and let v be the kth vertex.

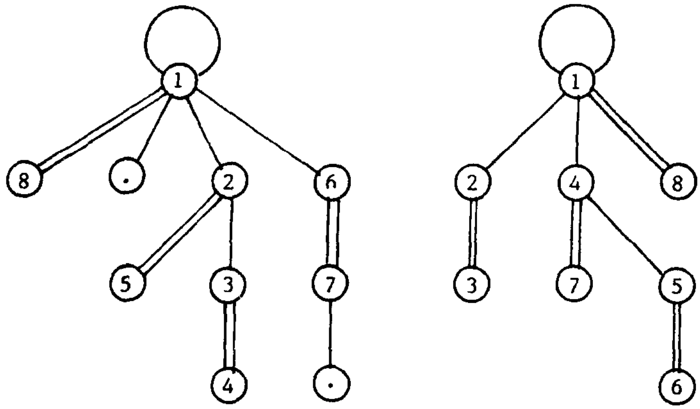
5. [Mark the edge (u, v); call u its top and v its bottom.]

6. For each child c of v such that c is not a leaf, call OrdDesc(c).

7. End of procedure.

In Figure 2, we show two graphs with the vertices ordered by step 1 of the algorithm with the marked edges shown as double lines. In Figure 2(a) the vertices without numbers would be numbered 9 and 10 in step 3 of the algorithm. The numbering in Figure 2(a) was obtained by selecting the vertex which became 8 as v in step 2 of OrdDesc(t). In step 3 of OrdDesc(t) the child which became 2 was treated before the one that became vertex 6. For the graph in Figure 2(b) with vertex 8 removed a linear subgraph must contain the loop. With vertex 8 included, a linear subgraph of the graph in Figure 2(b) must contain the edge $(1, 8)$, but in order to obtain the result of Lemma 4 below, vertex 8 cannot be numbered until all descendants of vertex 1 by other children have already been considered.

The complexity of Algorithm 2 is linear in the number of vertices. To see this note that step 0 can be replaced by a depth-first search that produces the odd/even labeling and reconstructs the adjacency lists of the vertices with



(a) (b)

FIG. 2.

odd children preceding even children. Once this is done, each vertex is considered only once in the remainder of the algorithm.

THEOREM 3. *If G is a tree with n vertices and a loop at one vertex t called the root, and the vertices of G have been ordered by Algorithm 2, then the adjacency matrix A of G has a unit factorization $A = LDL^T$ where all entries of L are* 0, 1, *or* -1 *and* $D = \mathrm{diag}(1, -1, 1, -1, \dots)$ *is r-by-r with the value of r supplied by Algorithm* 2, *which is thus the rank of A.*

Three lemmas are presented before completing the proof of Theorem 3. In these lemmas we assume that the vertices of G have been numbered by Algorithm 2 and that the value of r is determined by this algorithm. Recall that G_k denotes the subgraph of G spanned by vertices numbered $1, 2, \dots, k$; also let $G_{r,i}$ denote the subgraph spanned by vertices $1, 2, \dots, r, i$, let $G_{r,i,j}$ denote the subgraph spanned by vertices $1, 2, \dots, r, i, j$, and for an edge (u, v) of G_r let $D_r(u, v)$ denote the subgraph of G_r spanned by vertices u, v, and the descendants of v in G_r.

LEMMA 4. *For $k \leqslant r$, G_k has a linear subgraph.*

Proof. There is a path $p = u_0, u_1, \ldots, u_m$ in G_k where $u_0 = t$, u_m is the vertex numbered k, and for $i = 1, 2, \ldots, m$

(1) u_i is numbered in step 4 of OrdDesc(u_{i-1}) or

(2) u_i is numbered in step 1 of OrdDesc(u_i), which was called in step 3 of OrdDesc(u_{i-1}), step 6 of OrdDesc(u_{i-2}) (with u_{i-1} being numbered in step 4 of this call), or step 1b of the algorithm when t is even and $i = 1$.

When u_m is numbered, the calls to these procedures have not yet ended, while other calls to OrdDesc may have already been completed. Let E_k be the set of edges $(u_0, u_0), (u_1, u_2), (u_3, u_4), \ldots, (u_{m-1}, u_m)$ when m is even or $(u_0, u_1), (u_2, u_3), \ldots, (u_{m-1}, u_m)$ when m is odd, together with the edges marked in previously completed calls to OrdDesc. Now E_k is a linear subgraph of G_k, since: (i) the edges specified from p are disjoint and span vertices in p; (ii) marked edges are disjoint and span vertices in previously completed calls to OrdDesc; (iii) marked edges of previously completed calls to OrdDesc are not incident with p. ■

LEMMA 5. *If (u, v) is a marked edge with top u and bottom v, then v has an even number of proper descendants in G_r, and thus*

(a) *any linear subgraph of $D_r(u, v)$ or $G_{r,i}$, where i is not a descendant of v, must contain the edge (u, v), and*

(b) *if H is a subgraph of $G_{r,i,j}$ containing a path p from i to j such that $H - p$ is a linear subgraph of $G_{r,i,j} - p$, and u is on p, then u and v are both on p.*

Proof. The proper descendants of v in G_r are numbered in step 6 of OrdDesc(u), where each call OrdDesc(c) numbers an even number of vertices. The edges of a linear subgraph of $D_r(u, v)$ thus pair the proper descendants of v in $D_r(u, v)$ among themselves, and thus pair v with u. Now, if i is not a descendant of v, then the descendants of v in $G_{r,i}$ are the same as the descendants of v in $D_r(u, v)$, so any linear subgraph of $G_{r,i}$ must also contain (u, v). For part (b), if p passes through u but not v, then the descendants of v in $G_{r,i,j} - p$ are the same as the descendants of v in $D_r(u, v)$, and we reach a contradiction. ■

The hypothesis of the next lemma describes a situation we would like to avoid because of [2, Theorem 2]. The conclusion shows why the situation was not avoided and is illustrated in Figure 3. Since the even vertex v_0 must have an odd child, it is also shown in the figure.

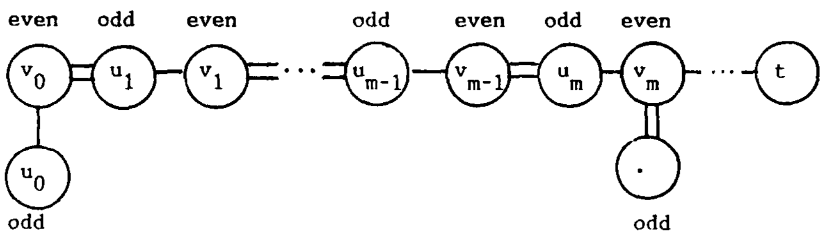


FIG. 3.

LEMMA 6. *If* (u_1, v_0) *is a marked edge whose bottom* v_0 *is even, then there is a subpath* $p = v_0, u_1, v_1, \dots, u_{m-1}, v_{m-1}, u_m, v_m$ *of the simple path* q *from* v_0 *to the root* t, *where* v_m *is the top of a marked edge which is not* (v_m, u_m), *and for* $j = 1, 2, \dots, m$, u_j *is odd*, v_j *is even, and* u_j *is the top of the marked edge* (u_j, v_{j-1}).

Proof. We proceed by induction on the length of the path q. Now v_0 was selected in step 2b of OrdDesc(u_1); thus all children of u_1 are even and u_1 is odd. Note that $u_1 \neq t$, since OrdDesc(t) is not called when t is odd, and hence u_1 has some parent v_1 which is even. The base case when $t = v_1$ is thus $q = v_0, u_1, t$, which is a path with the required properties, since t would be even, and OrdDesc(t) would be called but (t, u_1) is not marked. Now if $v_1 \neq t$, we have either (1) v_1 is the top of a marked edge and u_1 is not selected in step 2 of OrdDesc(v_1), so that $p = v_0, u_1, v_1$ is the desired path, or (2) v_1 is the bottom of some marked edge (u_2, v_1). In this case the path from the even vertex v_1 to t is shorter than the path from v_0 to t. By induction there is a path $v_1, u_2, \dots, u_m, v_m$ satisfying the above conditions, and $p = v_0, u_1, v_1, u_2, \dots, u_m, v_m$ is the desired path. ■

Proof of Theorem 3. Suppose that Algorithm 1 is applied to the adjacency matrix A. In the notation of Algorithm 1, we will show below that: (i) the (1,1) entry of S_k is $(-1)^k$ for $k = 0, 1, \dots, r-1$, so that we may choose $p_k = 1$, with $q_k = (-1)^k$, and (ii) the entries of α_k and β_k are 0, 1, or -1. From (i) it follows that Algorithm 1 will complete steps $k = 0, 1, \dots, r-1$, yielding

$$A = L_r U_r = \begin{bmatrix} L_r' & 0 \\ X_r & I_{n-r} \end{bmatrix} \begin{bmatrix} U_r' & Y_r \\ 0 & S_r \end{bmatrix},$$

where the diagonal entries of L'_r are 1's and the diagonal entries of U'_r are $1, -1, 1, -1, \ldots$. From (ii) it follows that all other entries of L'_r, X_r, U'_r, and Y_r are 0, 1, or -1. Finally we show below that (iii) $S_r = 0$. Then with D as the r-by-r diagonal matrix $D = \mathrm{diag}(1, -1, 1, -1, \ldots)$,

$$L = \begin{bmatrix} L'_r \\ X_r \end{bmatrix}, \quad \text{and} \quad U = D\begin{bmatrix} U'_r & Y_r \end{bmatrix},$$

a unit LU-factorization is given by $A = LDU$, and since $A = A^T$, Corollary 2 implies $U = L^T$.

For (i) we proceed inductively on k. Thus suppose that we have factored A as

$$A = L_k U_k = \begin{bmatrix} L'_k & 0 \\ X_k & I_{n-k} \end{bmatrix} \begin{bmatrix} U'_k & Y_k \\ 0 & S_k \end{bmatrix},$$

where L_k has a unit diagonal. For $k = 0$ such a factorization is given by $A = IA$. If A_{k+1} is the adjacency matrix of G_{k+1} and $k < r$, then

$$A_{k+1} = \begin{bmatrix} L'_k & 0 \\ \varepsilon_k^T & 1 \end{bmatrix} \begin{bmatrix} U'_k & \delta_k \\ 0 & s \end{bmatrix},$$

where s is the $(1,1)$ entry of S_k. Hence $\det(A_{k+1}) = s\det(U'_k)$. By Lemmas 2 and 4, $\det(A_{k+1}) = (-1)^e$, where $e = k/2$ if k is even or $e = (k+1)/2$ if k is odd. Now U'_k is k-by-k and by the inductive hypothesis has a diagonal of the form $1, -1, 1, -1, \ldots$. Hence $s = \det(A_{k+1})/\det(U'_k) = (-1)^k$, which completes the proof of (i).

Now let s be the (i', j') entry of S_k, $0 \leqslant k \leqslant r$, and set $i = i' + k$ and $j = j' + k$. Let $G_{k,i}$ be the subgraph spanned by vertices $1, 2, \ldots, k, i$ and let $G_{k,i,j}$ be the subgraph spanned by vertices $1, 2, \ldots, k, i, j$. Form the matrix B by deleting all of the last $n - k$ rows and columns of A except for row i and column j. Then

$$B = \begin{bmatrix} L'_k & 0 \\ \alpha^T & 1 \end{bmatrix} \begin{bmatrix} U'_k & \beta \\ 0 & s \end{bmatrix},$$

where α^T is row i' of X_k, β is column j' of Y_k, and $\det(B) = s\det(U'_k)$. For $i = j$, B is the adjacency matrix of $G_{k,i}$ and $\det(B)$ is given by Lemma 2. For $i \neq j$ and $k \leqslant r$, the modification of F. Harary's result in Lemma 3 gives

$\det(B)$ as ± 1 or 0 depending on whether or not $G_{k,i,j}$ contains a subgraph H consisting of a path p from i to j together with a linear subgraph $H-p$ of $G_{k,i,j}-p$. In either case $s=\det(B)/\det(U_k')=\pm 1$ or 0. We use the notation of this paragraph in the remainder of the proof.

When $i=k+1$ $(j=k+1)$ and $i\neq j$, s is an entry of $p_k\beta_k$ $(\alpha_k q_k)$. Hence the entries in α_k and β_k are 0, 1, or -1, which is part (ii) above.

For part (iii) take $k=r$, so that s is an entry of S_r. We show by contradiction that $G_{r,i}$ does not have a linear subgraph H, and $G_{r,i,j}$ does not have a subgraph H consisting of a path p from i to j and a collection of disjoint edges spanning $G_{r,i,j}-p$. Since $i,j>r$, these vertices were not numbered by the procedure OrdDesc, but by step 3 of Algorithm 2. All such vertices are leaves of the tree. Otherwise there would be an unnumbered vertex closest to the root which should have been numbered as a result of a call made in step 3 or step 6 of OrdDesc when its parent was numbered. Now the parents of i and j are not leaves and are numbered by OrdDesc. We consider four cases:

(a) $i=j$ has a parent which is the top of some marked edge,

(b) $i\neq j$, and one of i or j has a parent which is the top of some marked edge,

(c) $i=j$ has a parent which is the bottom of some marked edge, and

(d) $i\neq j$, and both have parents which are the bottoms of some marked edges.

If the parent of i or j, say i, is the top u of some marked edge (u,v), then vertex i is a possible selection in step 2a of OrdDesc(u). For $i=j$ there is no linear subgraph of $G_{r,i}$, since it would have to contain both (u,i) and [by Lemma 5(a)] (u,v). For $i\neq j$ the path $p=i,u,\dots,j$ joining i and j in H, according to Lemma 5(b), must contain the edge (u,v). Thus j is a descendant of v. Also, v is odd, since it was selected in step 2a instead of i, and thus v is not the parent of j. By Lemma 5b the parent of j is not the top of a marked edge, and thus is the bottom y_0 of some marked edge (x_1,y_0). Now y_0 is even, so we may obtain the path $p_y=y_0,x_1,y_1,\dots,x_m,y_m$ from Lemma 6 and set $p_j'=x_0,p_y$, where $x_0=j$. Let c be the child of y_m for which (y_m,c) is marked. Now (u,v) is not on this path, since it is a marked edge whose top is even. Hence y_m is a descendant of v, and p_j' is a subpath of p. This contradicts Lemma 5b, since p passes through the top y_m of a marked edge (y_m,c) without containing c. Thus in cases (a) and (b) the required subgraphs H do not exist.

For case (c) let the parent of $i=j$ be v_0, which is the bottom of the marked edge (u_1,v_0). Let $p_v=v_0,u_1,v_1,\dots,u_m,v_m$ be the path given by Lemma 6, and set $p_i'=u_0,p_v$, where $u_0=i$. We show by induction that $(v_k,u_k)\in H$, $k=0,1,\dots,m$, for any linear subgraph H of $G_{r,i}$. Since $u_0=i$ is a leaf, $(v_0,u_0)\in H$. If $(v_{k-1},u_{k-1})\in H$, $k\leqslant m$, then $(u_{k-1},v_k)\notin H$, since the edges of H are disjoint; and for any other child c of u_k in G_r, $(u_k, c) \notin H$, since c is the top of a marked edge (c, g) and by Lemma 5(a) $(c, g) \in H$. Thus $(v_k, u_k) \in H$. Now $(v_m, u_m) \in H$, but by Lemma 6 there is a marked edge (v_m, c) where $c \neq u_m$, and by Lemma 5(a) $(v_m, c) \in H$. This contradiction disposes of case (c).

For case (d) let v_0 and y_0 be the parents of i and j respectively, and let $p_v = v_0, \ldots, u_m, v_m$ and $p_y = y_0, \ldots, x_{m'}, y_{m'}$ be the paths provided by Lemma 6. Set $p_i' = u_0, p_v$ and $p_j' = x_0, p_y$, where $u_0 = i$ and $x_0 = j$. Recall that we are to show by contradiction that there is no subgraph H of $G_{r,i,j}$ that contains a path p from i to j such that $H - p$ is a linear subgraph of $G_{r,i,j} - p$. Let p be the path from i to j in such a subgraph H. Now p goes from i towards the root t until it reaches the lowest mutual ancestor z of i and j, and then goes away from t to j. By Lemma 5(b) applied to the marked edges at v_m and $y_{m'}$, z cannot be above v_m or $y_{m'}$, so the paths p_i' and p_j' intersect at z. Now z cannot be odd, for then it would be the top of two different marked edges (u_k, v_k) and $(x_{k'}, y_{k'})$. If $z = v_m = y_{m'}$, then by Lemma 6 there is a child c of z such that (z, c) is marked. Now $c \neq u_m$, $c \neq x_{m'}$, since (z, u_m) and $(z, x_{m'})$ are unmarked. Thus z but not c is on the path p, contradicting Lemma 5(b). If $z = v_k$ for some $k < m$ or $z = y_k$ for some $k < m'$, then by symmetry we need only consider the case $z = v_k$. Now $(v_k, u_{k+1}) \notin H$, since $z = v_k$ is the closest p gets to the root and z is on p. Since u_{k+1} is the top of a marked edge, all children $c \neq v_k$ of u_{k+1} in G_r are also tops of marked edges, and by Lemma 5(a) $(u_{k+1}, c) \notin H$. Hence $(v_{k+1}, u_{k+1}) \in H$. We proceed inductively as in case (c) above to obtain $(v_m, u_m) \in H$ and the same contradiction as in (c), to conclude case (d). ■

The authors wish to thank the referee for helpful suggestions.

REFERENCES

1 William N. Anderson, Maximum matching and the rank of a matrix, *SIAM J. Appl. Math.* 28:114–123 (1975).

2 Y. Caro and J. Schonheim, Generalized 1-factorization of trees, *Discrete Math.* 33:319–321 (1981).

3 D. De Caen and D. A. Gregory, Factorizations of symmetric designs, *J. Combin. Theory Ser. A* 49:323–337 (1988).

4 Frank Harary, The determinant of the adjacency matrix of a graph, *SIAM Rev.* 4:202–210 (1962).

5 C. M. Lau and T. Markham, *LU* factorizations, *Czechoslovak Math. J.* 29:546–550 (1979).

Received 20 September 1989; final manuscript accepted 14 July 1990