THE DECOMPOSITION OF MATRICES

THE DECOMPOSITION OF MATRICES

By

WILFRED ALLEN WARD, B.SC.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Science

McMaster University

May 1970

MASTER OF SCIENCE (1970)
(Mathematics)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: The Decomposition of Matrices

AUTHOR: Wilfred Allen Ward, B.Sc. (McMaster University)

SUPERVISOR: Professor J. Csima

NUMBER OF PAGES: iv, 92

SCOPE AND CONTENTS:

This thesis deals with algorithms which, for a given square matrix A of order n, construct permutation matrices P and Q (if they exist) such that PAQ is a canonical form of A. The pertinent theory of fully indecomposable matrices is discussed and detailed description is given of the algorithm by Dulmage and Mendelsohn. The connection between irreducible and fully indecomposable matrices is also examined, and it is observed that Harary's algorithm for bringing a matrix to a normal form is interchangeable with the second part of the Dulmage and Mendelsohn algorithm. Efficient computer programs for the Dulmage and Mendelsohn algorithm are presented which are directly applicable to various numerical problems.

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Professor J. Csima of the Mathematics Department for introducing the author to the research problem reported in this thesis. His guidance, criticism and encouragement through all phases of the work has been deeply appreciated.

I also wish to thank Mrs. M. Hruboska who competently performed the typing necessary for a successful completion of this thesis.

Finally, I would like to express my gratitude for the patience and understanding of my wife, Pamela.

TABLE OF CONTENTS

1. FULLY INDECOMPOSABLE MATRICES

1.1 INTRODUCTION

This thesis deals with algorithms which, for a given square matrix $A=(a_{ij})$ of order n (henceforth denoted simply A), construct permutation matrices P and Q (if they exist) such that PAQ is a canonical form of A. The pertinent theory of fully indecomposable and irreducible matrices is presented in section 1.3. Since we are primarily concerned with the points of nonzero entries in A rather than the entries, we work with the pattern of a matrix (see section 1.2) and consider (0,1)-matrices only.

In Chapter 2, we discuss the term rank and coverance of a matrix. In this connection, the classical König-Hall Theorem plays an important role.

In section 3.3, we explain Egerváry's algorithm which in essence constitutes the first part of the Dulmage and Mendelsohn algorithm. The second part is presented in section 3.1. In section 3.2, Harary's algorithm is discussed which performs the same task as the second part of the Dulmage and Mendelsohn algorithm.

The remainder of the thesis is devoted to the presentation of computer programs of the Dulmage and Mendelsohn algorithm which should prove valuable in the realm of numerical analysis (e.g. Forsythe and Moler (1), pp. 14-15: A Research Problem).

1.2 PATTERN OF A MATRIX

We say that (i,j) is a nonzero point of A if $a_{ij} \neq 0$ where a_{ij} is the entry at point (i,j). The set of nonzero points of A is called the pattern of A. Hence a pattern is simply a subset of the set $J=\{(i,j) \mid i=1,2,\ldots,n; j=1,2,\ldots,n\}$ for some natural number n.

In connection with matrices, it is customary to use the word line to refer to either the word 'row' or the word 'column'. For the purpose of our discussion, it is convenient to define the i-th row as the set $\{(i,1),(i,2),\ldots,(i,n)\}$ and the j-th column as the set $\{(1,j),(2,j),\ldots,(n,j)\}$. This way a line is simply a pattern of a special kind.

Points (i_1,j_1), $(i_2,j_2),\ldots,$ (i_k,j_k) are independent if no two points appear on the same line. A diagonal of A consists of n independent points.

1.3 DEFINITIONS

A matrix whose entries consist solely of the integers 0 and 1 is called a (0,1)-matrix. A permutation matrix is a (0,1)-matrix which has exactly n 1's at independent points. Premultiplying A by a permutation matrix permutes the rows of A while postmultiplying A by a permutation matrix permutes the columns of A.

<u>Definition 1.3.1</u> If A is of order $n \geq 2$, then A is <u>fully indecomposable</u> if no permutation matrices P and Q exist such that

$$PAQ = \begin{pmatrix} B & O \\ C & D \end{pmatrix} \qquad (1)$$

where B and D are square submatrices and O is a zero submatrix. If A is of order 1, then A is fully indecomposable if its only entry is nonzero. If A is not fully indecomposable, then A is <u>partly decomposable</u>.

If A is partly decomposable and either of the diagonal submatrices B and D in (1) is partly decomposable, then there exist permutation matrices P_1 and Q_1 such that

$$P_1PAQQ_1 = \begin{pmatrix} B_{11} & O & O \\ B_{21} & B_{22} & O \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$$

Continuing this process, after a finite number of steps we obtain a canonical form

$$\begin{pmatrix} A_{11} & O & \cdot & \cdot & O \\ A_{21} & A_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & O \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix}$$

where each of the diagonal submatrices A_{pp}, $p=1,2,\ldots,m$ is either fully indecomposable or is a zero submatrix of order 1.

Definition 1.3.2 A is reducible if there exists a permutation matrix R such that

$$RAR^T = \begin{pmatrix} B & O \\ C & D \end{pmatrix} \qquad (2)$$

where B and D are square submatrices and O is a zero submatrix. If A is not reducible, then A is irreducible.

If A is reducible and either of the diagonal submatrices B and D in (2) is reducible, then there exists a permutation matrix R_1 such that

$$R_1 RAR^T R_1 T = \begin{pmatrix} B_{11} & O & O \\ B_{21} & B_{22} & O \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$$

Continuing this process, after a finite number of steps we obtain a normal form

$$\begin{pmatrix} A_{11} & O & \cdot & \cdot & O \\ A_{21} & A_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & O \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix}$$

where each of the diagonal submatrices A_{pp}, p=1,2,...,m is irreducible.

Remark 1.3.1 Every reducible matrix is partly decomposable and every fully indecomposable matrix is irreducible. The matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

is partly decomposable and irreducible.

2. THE TERM RANK AND COVERANCE OF A MATRIX

2.1 THE KÖNIG-HALL THEOREM

Let P be the pattern of A and let L be a set of lines. Then L is a k-cover of A if $|L|=k$ and $P \subseteq \bigcup_{h \in L} h$. The coverance of A is c if c is the minimum number of lines required to cover A. The term rank of A is r if r is the largest number such that P has a subset of r independent points.

Remark 2.1.1 The term rank and coverance of a matrix are invariant under permutation of rows and columns of a matrix.

From the definitions it follows immediately that the coverance of A is at least equal to the term rank of A. The following is a classical theorem of D. König (2).

Theorem 2.1.1 (D. König). The term rank of a matrix is equal to its coverance.

A corollary of this theorem is the well known theorem of P. Hall (3) on systems of distinct representatives. Let $F=(S_1,S_2,\ldots,S_n)$ be an n-tuple of (not necessarily distinct) subsets of an arbitrary finite set S, then the n-tuple $(a_1,a_2,\ldots,a_n)$ formed from distinct elements of S is called a system of distinct representatives (abbreviated SDR) for F.

Theorem 2.1.2 (P. Hall). F has an SDR if and only if the following condition holds: For each $k=1,2,\ldots,n$, any k of the sets $S_1, S_2, \ldots, S_n$ contain between them at least k distinct elements.

2.2 DULMAGE AND MENDELSOHN'S ALGORITHM

Since we are primarily concerned in this thesis with determining whether or not a matrix is fully indecomposable, the following theorem (4) reveals that we need only consider matrices with term rank n.

Theorem 2.2.1 (Frobenius-König). Every diagonal of A contains a zero entry if and only if A has an $s \times t$ zero submatrix with $s+t = n+1$.

Proof Assume A is of the form

$$\begin{pmatrix} B & O \\ C & D \end{pmatrix}$$

where O is an $s \times t$ zero submatrix. Suppose that A has a strictly positive diagonal. Then t points of the diagonal must lie in submatrix D. Hence $t = n-s$. But then $t<n-s+1$, a contradiction.

We prove the necessity by induction on n. If A is a zero matrix, there is nothing to prove. Assume $a_{ij} \neq 0$ for some point (i,j). Then each diagonal of A must contain a zero entry; and, by the induction hypothesis, A has a $u \times v$ zero submatrix with $u+v = (n-1)+1$. Hence there exist permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} B & O \\ C & D \end{pmatrix}$$

where O is a u x (n-u) zero submatrix. If all the entries of any diagonal of submatrix B are nonzero, then all the diagonals of submatrix D must contain a zero entry. It follows that all the diagonals of B or D must contain a zero entry. Without loss of generality, assume the former is the case. Then, by the induction hypothesis, B must contain a pxq zero submatrix with p+q = u+1. But then the first u rows of PAQ contain a $p \times (q+v)$ zero submatrix and $p + (q+v) = (p+q) + (n-u) = (u+1) + (n-u) = n+1$.

<u>Corollary 2.2.1</u> If A is fully indecomposable, the term rank of A is n.

<u>Remark 2.2.1</u> A square matrix has at least two n-covers: a set of n horizontal lines and a set of n vertical lines.

<u>Theorem 2.2.2</u> If the order of A is $n \geq 2$, then A is fully indecomposable if and only if A has precisely two n-covers.

<u>Proof</u> Assume A is fully indecomposable and has an n-cover consisting of r rows and n-r columns. Then there exist permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} B & O \\ C & D \end{pmatrix}$$

where B is a square submatrix of order n-r, D is a square submatrix of order r, and O is a zero submatrix. Hence A is partly decomposable, a contradiction.

Assume A has precisely two n-covers and is partly decomposable. Then there exist permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} B & O \\ C & D \end{pmatrix}$$

where B and D are square submatrices and O is a zero submatrix. Suppose B is of order n-r and D is of order r. Consequently the last r rows and first n-r columns of PAQ constitute an n-cover of PAQ other than a set of n horizontal lines and a set of n vertical lines, a contradiction.

<u>Remark 2.2.2</u> If the term rank of A is n and A is not fully indecomposable, then there exist permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} A_{11} & O & \cdot & \cdot & O \\ A_{21} & A_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & O \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix},$$

PAQ has a strictly positive main diagonal, and each of the diagonal submatrices A_{pp}, p=1,2,...,m, is fully indecomposable.

<u>Theorem 2.2.3</u> (5) A is fully indecomposable if and only

if there exist permutation matrices P and Q such that PAQ has a strictly positive main diagonal and is irreducible.

Proof Let A be fully indecomposable. Then, by Theorem 2.2.1, there exist permutation matrices P and Q such that PAQ has a strictly positive main diagonal. Since PAQ is fully indecomposable if and only if A is, PAQ is irreducible.

Conversely, suppose PAQ has a strictly positive main diagonal and is irreducible. Without loss of generality, we may assume $PAQ = A$. Suppose A is not fully indecomposable and let P_1 and Q_1 be permutation matrices such that

$$P_1AQ_1 = \begin{pmatrix} B & O \\ C & D \end{pmatrix}$$

where B and D are square submatrices and O is a zero submatrix. Suppose B is of order r and D is of order $n-r$. Then we may write $P_1AQ_1 = A^1Q^1$ where $A^1 = P_1AP_1^T$ is again a matrix with a strictly positive main diagonal and $Q^1 = P_1Q_1$ is a permutation matrix. But then it follows that Q_1 permutes the first r columns of A^1 among themselves and the last $n-r$ columns of A^1 among themselves. Hence A is reducible, a contradiction.

Herein is the Dulmage and Mendelsohn algorithm (6). Part one determines whether or not permutation matrices P and Q exist such that PAQ has a strictly positive main diagonal. If this is indeed the case, part two either concludes that A is fully indecomposable or decomposes PAQ to a normal form as given in Remark 2.2.2.

3. THE DECOMPOSITION OF MATRICES

3.1 THE DECOMPOSITION ALGORITHM

A partition of a set S is a family of nonvoid subsets $(S_1,S_2,\ldots,S_m)$ such that the union of all the subsets in the family is S and the intersection of any two distinct subsets in the family is empty.

Let $M = (M_1,M_2,\ldots,M_r)$ and $N = (N_1,N_2,\ldots,N_r)$ be two partitions of the set of first n natural numbers. Then to (M,N) there corresponds a function ρ_{MN} which maps the (0,1)-matrices of order n onto the (0,1)-matrices of order r according to the following rule: the (i^*,j^*)-entry of $A^*=\rho_{MN}(A)$ is 1 if and only if $\sum_{i\in M_{i^*}} \sum_{j\in N_{j^*}} a_{ij}\neq 0$. A^* is called the induced matrix.

A set of k independent nonzero points is called a k-transversal. We will refer to an n-transversal simply as a transversal.

A sequence of nonzero points $(i_1,j_1),(i_2,j_2),\ldots,(i_k,j_k)$ is a chain (of length k) if no two adjacent points are identical, every two consecutive points are on a line, but no three consecutive points are on a line. A chain is simple if no point appears twice. Let T denote a transversal. Then an alternating chain (with respect to T) is a chain with every other point in T. The chain $(i_1,j_1),(i_2,j_2),\ldots,$

(i_k, j_k) is a <u>cycle</u> if (i_{k-1}, j_{k-1}), (i_k, j_k), (i_1, j_1), (i_2, j_2) is a chain.

<u>Remark 3.1.1</u> The length of every cycle is even.

A nonzero point is <u>admissible</u> if it is a point of a transversal. Otherwise a nonzero point is <u>inadmissible</u>.

<u>Remark 3.1.2</u> Let C be an alternating cycle with respect to a transversal T. Let C_1 (C_2) denote the set of points of C which belong (do not belong) to T. Then the points of C_1 are admissible by definition. $(T-C_1) \cup C_2$ is also a transversal and hence the points in C_2 are admissible. Consequently every point in an alternating cycle is admissible.

<u>Remark 3.1.3</u> It follows from König's Theorem that a point (i,j) in A is inadmissible if and only if there is an n-cover of A such that (i,j) belongs to two lines of the n-cover.

<u>Remark 3.1.4</u> If the order of A is $n \geq 2$, a necessary condition for A to be fully indecomposable is that every line of A contains at least two nonzero points.

Repeating Remark 2.2.2, if the term rank of A is n and A is not fully indecomposable, then there exist permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} A_{11} & O & \cdot & \cdot & O \\ A_{21} & A_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & O \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix} \quad (3)$$

PAQ has a strictly positive main diagonal, and each of the diagonal submatrices A_{pp}, $p=1,2,\ldots,m$, is fully indecomposable. In addition, we have the following:

Remark 3.1.5 By Remark 3.1.3, every nonzero point in the diagonal submatrices in (3) is admissible while the remaining nonzero points are inadmissible.

Remark 3.1.6 If C is an alternating cycle in PAQ, then every point in C is in one of the diagonal submatrices.

Remark 3.1.7 All the nonzero points in one of the diagonal submatrices of order at least 2 form an alternating cycle.

The preceding remarks yield immediately the following theorem:

Theorem 3.1.1 If C is an alternating cycle $(i_1,i_1),(i_1,i_2),(i_2,i_2),(i_2,i_3),\ldots,(i_k,i_k),(i_k,i_1)$ in A and $D = \{i_1,i_2,\ldots,i_k\}$ then the restriction of A to DxD is fully indecomposable.

Let $S = \{1,2,\ldots,n\}$ and assume A has a strictly positive main diagonal. The decomposition algorithm begins by constructing a finite alternating chain $(i_1,i_1),(i_1,i_2),$ $(i_2,i_2),(i_2,i_3),\ldots,$ in A. If $i_{k+1} = i_r$ for some $r<k$, then $(i_r,i_r),(i_r,i_{r+1}),\ldots,(i_k,i_k),(i_k,i_{k+1})$ is a simple alternating cycle C_1 of length $2r_1$ $(r_1=k-r+1)$.

Let $B_1 = \{1,2,\ldots,n-r_1+1\}$ and let θ_1 be the surjection $\theta_1\colon S\to B_1$ such that $\theta_1(i) = 1$ if (i,i) is a point in C_1 and θ_1 preserves the order of magnitude of the integers in $S-\theta_1^{-1}(1)$. Then $D_1 = (\theta_1^{-1}(1),\theta_1^{-1}(2),\ldots,\theta_1^{-1}(n-r_1+1))$ is a partition of S and (D_1,D_1) induces a matrix A_1 from A. It is apparent that $(\theta_1(i_1),\theta_1(i_1))$, $(\theta_1(i_1),\theta_1(i_2)),\ldots,(\theta_1(i_r),\theta_1(i_r))$ is a simple alternating chain in A_1 (of course the last point of this chain is $(1,1)$ since $\theta_1(i_r)=1$).

Continue the chain from point $(1,1)$ in A_1 forming an induced matrix as above whenever a simple alternating cycle is found.

Eventually the alternating chain will terminate at some point (t,t) of A_k, the only nonzero point in row t of A_k. Let $\Phi = \theta_k\theta_{k-1}\cdots\theta_1$. Then the restriction of A to $\Phi^{-1}(t)\times\Phi^{-1}(t)$ is carried into point (t,t). By Theorem 3.1.1, the restriction of A to $\Phi^{-1}(t)\times\Phi^{-1}(t)$ is fully indecomposable. In addition, the restriction of A to $\Phi^{-1}(t)\times(S-\Phi^{-1}(t))$ is a zero submatrix. Hence we may construct a permutation matrix

R_1 such that

$$R_1 A R_1^T = \begin{pmatrix} A_{11} & 0 \\ C & D \end{pmatrix}$$

where A_{11} is fully indecomposable.

Repeating the above procedure, a new chain is constructed in the restriction of A_k to $(B_k - t) \times (B_k - t)$.

Ultimately m chains will be required to exhaust A. Hence, we may construct a permutation matrix $R = R_m R_{m-1} \ldots R_1$ such that

$$RAR^T = \begin{pmatrix} A_{11} & 0 & . & . & 0 \\ A_{21} & A_{22} & . & . & . \\ . & . & . & . & . \\ . & . & . & . & 0 \\ A_{m1} & . & . & . & A_{mm} \end{pmatrix},$$

RAR^T has a strictly positive main diagonal, each of the diagonal submatrices A_{pp}, p=1,2,...,m, is fully indecomposable, and A is in a normal form.

3.2 HARARY'S ALGORITHM (7)

A nonzero point (i_2, j_2) in A is reachable from a nonzero point (i_1, j_1) in A if either $(i_1, j_1) = (i_2, j_2)$ or there exists a chain in A starting with (i_1, j_1) and ending with (i_2, j_2).

Let $S = \{1,2,\ldots,n\}$ and assume $B = (b_{ij}^{(1)})$ is a (0,1)-matrix of order n with a strictly positive main diagonal. Let $T = \{(1,1), (2,2), \ldots, (n,n)\}$.

First compute matrix $B^{n-1} = (b_{ij}^{(n-1)})$ using boolean multiplication and addition. For $2 \le k \le n-1$, $b_{ij}^{(k)} = \sum_{r=1}^{n} b_{ir}^{(k-1)} b_{rj}^{(1)}$ is 1 if and only if either $b_{ij}^{(k-1)} = 1$ or there is a simple alternating chain in B (with respect to T) of length 2k-1 such that a nonzero point (i^*,j), $i^*\neq j$, in column j of B is reachable from a nonzero point (i,j^*), $j^*\neq i$, in row i of B.

Remark 3.2.1 If the (i,j)-entry of any power of B is 1, then the (i,j)-entry of any higher power of B is 1.

Remark 3.2.2 $B^{n-1} = B^p$ whenever $p\geq n-1$ since the longest simple alternating chain in B (with respect to T) as defined above consists of $2n-3$ points. B^{n-1} is called the reachability matrix.

We observe that only the sequence of matrices B^2, B^4, B^8, ... need be computed in order to find B^{n-1}.

Next each row of B^{n-1} is searched in turn until a row, say i, is found such that, for each nonzero point (i,j_1), $(2,j_2),\ldots,(i,j_k)$ in row i, $b_{ij_p}{}^{(n-1)} = b_{j_pi}{}^{(n-1)} = 1$ for $p=1,2,\ldots,k$. Let $S_1 = \{j_1,j_2,\ldots,j_k\}$. Consequently each nonzero point in the restriction of B to S_1xS_1 is a point of an alternating cycle; and, by Theorem 3.1.1, the restriction of B to S_1xS_1 is fully indecomposable. In addition, the restriction of B to $S_1x(S-S_1)$ is a zero submatrix. Hence we may construct a permutation matrix R_1 such that

$$R_1BR_1{}^T = \begin{pmatrix} A_{11} & O \\ C & D \end{pmatrix}$$

where A_{11} is fully indecomposable.

Delete from B^{n-1} all the rows and columns whose indices are in the set S_1 but preserve the original row and column indices of B^{n-1} in the resulting submatrix.

Repeat the above procedure on this submatrix.

Ultimately the partition $(S_1, S_2, \ldots, S_m)$ of S will be formed. Hence we may construct a permutation matrix $R = R_m R_{m-1} \ldots R_1$ such that

$$RBR^T = \begin{pmatrix} A_{11} & 0 & \cdot & \cdot & 0 \\ A_{21} & A_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix},$$

RBR^T has a strictly positive main diagonal, each of the diagonal submatrices A_{pp}, $p=1,2,\ldots,m$, is fully indecomposable, and B is in a normal form.

3.3 THE TRANSVERSAL ALGORITHM

If the term rank of A (denoted $\rho(A)$) is ρ, then A has a ρ-transversal. When A is large, the selection of ρ such nonzero points is not a trivial task. An efficient algorithm to do this has been provided by E. Egerváry (8). This algorithm is also known as the 'Hungarian Method' (see Kuhn (9)) and lends itself in a natural way to computer programming (10). If A has a transversal, then the algorithm constructs one. If A has no transversal, the algorithm can be used to detect this fact.

A k-transversal is a maximal transversal if it can not be extended to a (k+1)-transversal by the addition of a point.

Let T be a maximal transversal. A simple chain $(i_1,j_1),(i_2,j_2),\ldots,(i_{2q+1},j_{2q+1})$ is an <u>augmenting chain</u> (with respect to T) if $(i_{2r},j_{2r})\varepsilon T$ for $r=1,2,\ldots,q$ but no points from row i_1 or column j_{2q+1} belong to T.

<u>Theorem 3.3.1</u> (Egerváry) If T is a maximal transversal in A, then $\rho(A) = |T|$ if and only if there is no augmenting chain in A (with respect to T).

<u>Proof</u> Suppose $\rho(A) = |T| = k$ and there is an augmenting chain $(i_1,j_1),(i_2,j_2),\ldots,(i_{2q+1},j_{2q+1})$ in A (with respect to T). Then we can construct a (k+1)-transversal from T by deleting the points $(i_2,j_2),(i_4,j_4),\ldots,(i_{2q},j_{2q})$ and adding the points $(i_1,j_1),(i_3,j_3),\ldots,(i_{2q+1},j_{2q+1})$. Thus $\rho(A)>|T|$, a contradiction.

Suppose there is no augmenting chain in A (with respect to T). Clearly the coverance of A is at least equal to $|T|$. We shall now construct a minimum cover of A revealing the coverance of A is exactly equal to $|T|$. Select a row r_1 of A with a nonzero point such that T has no point in row r_1 and let T_1 denote the set of points in T which are reachable from some nonzero point in row r_1. Next select a different row r_2 in A with a nonzero point such that T has no point in row r_2 and let T_2 denote the set of points in $T-T_1$ which are reachable from some nonzero point in row r_2. Continuing in this manner, after a finite number of steps we will reach a row r_w of A with a nonzero point such that T has no point in row r_w but will be unable to find a row r_{w+1} with a nonzero point such that T has no point in row r_{w+1}. Let T_w denote the set of points in $T- \bigcup_{m=1}^{w-1} T_m$ which are reachable from some nonzero point in row r_w. Without loss of generality, we may assume $T = \{(i_1,j_1), (i_2,j_2),\ldots,(i_k,j_k)\}$ and $\bigcup_{m=1}^{w} T_m = \{(i_1,j_1), (i_2,j_2),\ldots,(i_p,j_p)\}$. Then columns $j_1,j_2,\ldots, j_p$ of A and rows $i_{p+1},i_{p+2},\ldots,i_k$ of A constitute a minimum cover of A. For, since T is maximal, any nonzero point (i,j) in a row r_v of A, $1\leq v\leq w$, is covered by a vertical line. In addition, a nonzero point (i,j) in A in the same row as a point in T but not in the same column as a point in T is covered by a horizontal line. Otherwise point (i,j) is reachable from a nonzero point in some row r_v of A, $1\leq v\leq w$, by an augmenting chain in A (with respect to T), a contradiction. It is evident from the construction that each nonzero point in the same row and column as a point in T is covered by at least one line. Thus the coverance of A is $|T|$; and, by König's Theorem, $\rho(A)=|T|$.

The algorithm begins by selecting nonzero independent points $(i,j_1),(2,j_2),\ldots$ from rows $1,2,\ldots$ of A respectively until a maximal transversal T is found.

If $|T| = n$, then there exists a permutation matrix P such that PA has a strictly positive main diagonal. Otherwise, assuming $\rho(A) = n$, we apply Egerváry's Theorem and continue as before.

3.4 Concluding Remarks

In the remainder of this thesis, computer programs of the foregoing algorithms are presented. For large matrices, a comparison of the tables in sections C and D of the Appendix appears to indicate that the second part of the Dulmage and Mendelsohn algorithm is more efficient than Harary's algorithm for decomposing a matrix to a normal form.

APPENDIX

A. PROGRAM DECOMP

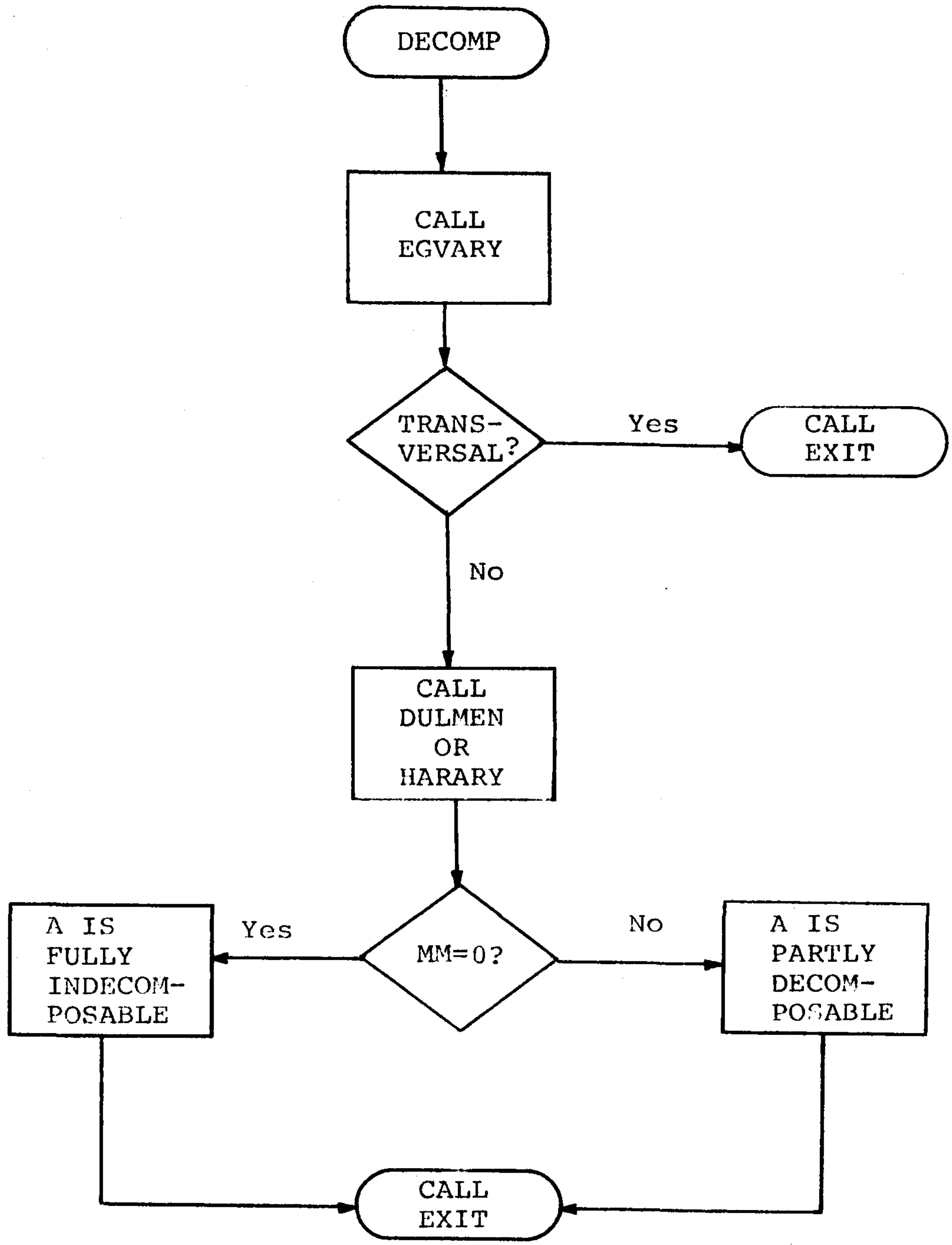
I. Purpose

Program DECOMP (written in FORTRAN) determines whether or not a matrix is fully indecomposable.

II. Method

The method has previously been presented in section 2.2 of the thesis. Program DECOMP reads and prints pertinent data pertaining to subroutine EGVARY and subroutine DULMEN or HARARY. DECOMP begins by calling subroutine EGVARY to determine whether or not A has a transversal. If a transversal is not found, DECOMP prints the term rank of A and a minimum cover of A. Otherwise A is returned with a nonzero main diagonal and either subroutine DULMEN or HARARY is called in order to decompose A to a normal form.

III. Flowchart, Listing and Typical Printouts



```
      PROGRAM DECOMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     PURPOSE
C        TO DETERMINE WHETHER OR NOT A MATRIX IS FULLY
C        INDECOMPOSABLE.
C
C     SUBROUTINES CALLED
C        EGVARY AND DULMEN
C
      INTEGER PER,HOR,VERT
      DIMENSION A(10,10),S(10,11),REPNG(10),REPBY(10),
     -          PER(10),SHORT(10),TAIL(10),HOR(10),VERT(10),
     -          IRR(10),WORK(10),HLDCH(10),ZERO1(11),
     -          CHAIN(11),HLDPER(11),START(11),FINISH(11),
     -          IND(10)
      EQUIVALENCE (REPNG,IRR),(REPBY,WORK),(SHORT,HLDCH),
     -            (TAIL,ZERO1),(HOR,CHAIN),(VERT,HLDPER)
      NN=10
C
C     NTIMES IS THE TOTAL NUMBER OF TIMES THE PROGRAM
C     IS TO BE EXECUTED.
C
      READ(5,1) NTIMES
      DO 350 LLL=1,NTIMES
C
```

```
C     READ AND PRINT MATRIX A.
C
      READ(5,1) N
      WRITE(6,2) N
      NP1=N+1
      DO 100 I=1,N
      READ(5,3) (A(I,J),J=1,N)
  100 WRITE(6,8) (A(I,J),J=1,N)
C
C     CALL SUBROUTINE EGVARY.
C
      CALL TOCKS
      CALL EGVARY(N,NN,LL,A,S,REPNG,REPRY,PER,SHORT,
     -            TAIL,IND,HOR,VERT)
      CALL TOCKP
      IF(LL.EQ.0) GO TO 310
C
C     NO TRANSVERSAL HAS BEEN FOUND. PRINT THE TERM RANK
C     OF A AND A MINIMUM COVER OF A.
C
      NMLL=N-LL
      WRITE(6,13) NMLL
      WRITE(6,15)
      K=0
      DO 200 I=1,N
```

```
      IF(VERT(I).EQ.0) GO TO 200
      K=K+1
      VERT(K)=I
  200 CONTINUE
      IF(K.EQ.0) GO TO 205
      WRITE(6,11) (VERT(I),I=1,K)
      K=0
  205 DO 210 I=1,N
      IF(HOR(I).EQ.0) GO TO 210
      K=K+1
      HOR(K)=I
  210 CONTINUE
      IF(K.EQ.0) GO TO 350
      WRITE(6,14) (HOR(I),I=1,K)
      GO TO 350
C
C     A TRANSVERSAL HAS BEEN FOUND. PRINT A WITH A NONZERO
C     MAIN DIAGONAL AND PRINT ARRAY PER.
C
  310 WRITE(6,4)
      DO 320 I=1,N
  320 WRITE(6,8)(A(I,J),J=1,N)
      WRITE(6,9)
      WRITE(6,10) (PER(I),I,I=1,N)
C
```

```
C     CALL SUBROUTINE DULMEN.
C
      CALL TOCKS
      CALL DULMEN(N,NN,KK,MM,A,S,PER,IRR,WORK,HLDCH,
     -            ZERO1,CHAIN,HLDPER,START,FINISH)
      CALL TOCKP
      IF(MM.EQ.1) GO TO 330
C
C     A IS FULLY INDECOMPOSABLE.
C
      WRITE(6,5)
      GO TO 350
C
C     A IS PARTLY DECOMPOSABLE. PRINT A IN A NORMAL FORM
C     AND PRINT ARRAYS PER AND IRR.
C
  330 WRITE(6,6)
      DO 339 I=1,N
  339 WRITE(6,8) (A(I,J),J=1,N)
      WRITE(6,9)
      WRITE(6,10) (PER(I),I,I=1,N)
      WRITE(6,9)
      WRITE(6,12) (IRR(I),I,I=1,KK)
  350 CONTINUE
      WRITE(6,7)
```

```
      CALL EXIT
    1 FORMAT(I5)
    2 FORMAT(1H1,////,10X,18HINITIAL MATRIX A.
     -                         17HTHE ORDER OF A IS,I3,/)
    3 FORMAT(8F10.1)
    4 FORMAT(1H0,10X,24HMATRIX A WITH A NONZERO
     -                    14HMAIN DIAGONAL.,/)
    5 FORMAT(1H0,10X,33HMATRIX A IS FULLY INDECOMPOSABLE.,/)
    6 FORMAT(1H0,10X,26HMATRIX A IN A NORMAL FORM.,/)
    7 FORMAT(1H0,10X,11HEND OF JOB.)
    8 FORMAT(1H ,10X,15F7.1)
    9 FORMAT(1H )
   10 FORMAT(1H ,10X,4(I3,5H=PER(,I3,1H),3X))
   11 FORMAT(1H0,15X,8HCOLUMNS ,10(I3,1H,))
   12 FORMAT(1H ,10X,4(I3,5H=IRR(,I3,1H),3X))
   13 FORMAT(1H0,10X,33HMATRIX A IS PARTLY DECOMPOSABLE. ,
     -                    21HTHE TERM RANK OF A IS,I3,/)
   14 FORMAT(1H0,15X,8H   ROWS ,10(I3,1H,))
   15 FORMAT(1H ,10X,21HA MINIMUM COVER OF A.)
       END
```

```
      PROGRAM DECOMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     PURPOSE
C        TO DETERMINE WHETHER OR NOT A MATRIX IS FULLY
C        INDECOMPOSABLE.
C
C     SUBROUTINES CALLED
C        EGVARY AND HARARY
C
      INTEGER PER,HOR,VERT
      DIMENSION A(10,10),S(10,11),B(10,10),REPNG(10),
     -          REPBY(10),PER(10),SHORT(10),TAIL(10),
     -          HOR(10),VERT(10),IRR(10),JHOLD(10),
     -          IWORK(10),IND(10)
      EQUIVALENCE (REPNG,IRR),(REPBY,JHOLD),(SHORT,IWORK)
      NN=10
C
C     NTIMES IS THE TOTAL NUMBER OF TIMES THE PROGRAM
C     IS TO BE EXECUTED.
C
      READ(5,1) NTIMES
      DO 350 LLL=1,NTIMES
C
C     READ AND PRINT MATRIX A.
C
```

```
      READ(5,1) N
      WRITE(6,2) N
      NP1=N+1
      DO 100 I=1,N
      READ(5,3) (A(I,J),J=1,N)
  100 WRITE(6,8) (A(I,J),J=1,N)
C
C     CALL SUBROUTINE EGVARY.
C
      CALL TOCKS
      CALL EGVARY(N,NN,LL,A,S,REPNG,REPRY,PER,SHORT,
     -            TAIL,IND,HOR,VERT)
      CALL TOCKP
      IF(LL.EQ.0) GO TO 310
C
C     NO TRANSVERSAL HAS BEEN FOUND. PRINT THE TERM RANK
C     OF A AND A MINIMUM COVER OF A.
C
      NMLL=N-LL
      WRITE(6,13) NMLL
      WRITE(6,15)
      K=0
      DO 200 I=1,N
      IF(VERT(I).EQ.0) GO TO 200
      K=K+1
```

```
      VERT(K)=I
  200 CONTINUE
      IF(K.EQ.0) GO TO 205
      WRITE(6,11) (VERT(I),I=1,K)
      K=0
  205 DO 210 I=1,N
      IF(HOR(I).EQ.0) GO TO 210
      K=K+1
      HOR(K)=I
  210 CONTINUE
      IF(K.EQ.0) GO TO 350
      WRITE(6,14) (HOR(I),I=1,K)
      GO TO 350
C
C     A TRANSVERSAL HAS BEEN FOUND. PRINT A WITH A NONZERO
C     MAIN DIAGONAL AND PRINT ARRAY PER.
C
  310 WRITE(6,4)
      DO 320 I=1,N
  320 WRITE(6,8)(A(I,J),J=1,N)
      WRITE(6,9)
      WRITE(6,10) (PER(I),I,I=1,N)
C
C     CALL SUBROUTINE HARARY.
C
```

```
      CALL TOCKS
      CALL HARARY(N,NN,KK,MM,A,S,B,PER,IRR,JHOLD,IWORK)
      CALL TOCKP
      IF(MM.EQ.1) GO TO 330
C
C     A IS FULLY INDECOMPOSABLE.
C
      WRITE(6,5)
      GO TO 350
C
C     A IS PARTLY DECOMPOSABLE. PRINT A IN A NORMAL FORM
C     AND PRINT ARRAYS PER AND IRR.
C
  330 WRITE(6,6)
      DO 339 I=1,N
  339 WRITE(6,8) (A(I,J),J=1,N)
      WRITE(6,9)
      WRITE(6,10) (PER(I),I,I=1,N)
      WRITE(6,9)
      WRITE(6,12) (IRR(I),I,I=1,KK)
  350 CONTINUE
      WRITE(6,7)
      CALL EXIT
    1 FORMAT(I5)
    2 FORMAT(1H1,////,10X,18HINITIAL MATRIX A.
```

```
     -                   17HTHE ORDER OF A IS,I3,/)
 3 FORMAT(8F10.1)
 4 FORMAT(1H0,10X,24HMATRIX A WITH A NONZERO
     -              14HMAIN DIAGONAL.,/)
 5 FORMAT(1H0,10X,33HMATRIX A IS FULLY INDECOMPOSABLE.,/)
 6 FORMAT(1H0,10X,26HMATRIX A IN A NORMAL FORM.,/)
 7 FORMAT(1H0,10X,11HEND OF JOB.)
 8 FORMAT(1H ,10X,15F7.1)
 9 FORMAT(1H )
10 FORMAT(1H ,10X,4(I3,5H=PER(,I3,1H),3X))
11 FORMAT(1H0,15X,8HCOLUMNS ,10(I3,1H,))
12 FORMAT(1H ,10X,4(I3,5H=IRR(,I3,1H),3X))
13 FORMAT(1H0,10X,33HMATRIX A IS PARTLY DECOMPOSABLE. ,
     -              21HTHE TERM RANK OF A IS,I3,/)
14 FORMAT(1H0,15X,8H   RCWS ,10(I3,1H,))
15 FORMAT(1H ,10X,21HA MINIMUM COVER OF A.)
   END
```

INITIAL MATRIX A. THE ORDER OF A IS 7

1.0	0.0	2.0	0.0	3.0	0.0	0.0
0.0	4.0	5.0	0.0	0.0	6.0	0.0
0.0	0.0	7.0	0.0	8.0	0.0	0.0
9.0	0.0	0.0	10.0	0.0	0.0	0.0
0.0	11.0	0.0	0.0	12.0	0.0	0.0
0.0	0.0	13.0	0.0	0.0	14.0	0.0
0.0	0.0	0.0	15.0	0.0	0.0	16.0

.006 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

1.0	0.0	2.0	0.0	3.0	0.0	0.0
0.0	4.0	5.0	0.0	0.0	6.0	0.0
0.0	0.0	7.0	0.0	8.0	0.0	0.0
9.0	0.0	0.0	10.0	0.0	0.0	0.0
0.0	11.0	0.0	0.0	12.0	0.0	0.0
0.0	0.0	13.0	0.0	0.0	14.0	0.0
0.0	0.0	0.0	15.0	0.0	0.0	16.0

1=PER(1)	2=PER(2)	3=PER(3)	4=PER(4)
5=PER(5)	6=PER(6)	7=PER(7)	

.042 SECONDS

MATRIX A IN A NORMAL FORM.

4.0	5.0	0.0	6.0	0.0	0.0	0.0
0.0	7.0	8.0	0.0	0.0	0.0	0.0
11.0	0.0	12.0	0.0	0.0	0.0	0.0
0.0	13.0	0.0	14.0	0.0	0.0	0.0
0.0	2.0	3.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	9.0	10.0	0.0
0.0	0.0	0.0	0.0	0.0	15.0	16.0

2=PER(1)	3=PER(2)	5=PER(3)	6=PER(4)
1=PER(5)	4=PER(6)	7=PER(7)	
4=IRR(1)	1=IRR(2)	1=IRR(3)	1=IRR(4)

```
INITIAL MATRIX A. THE ORDER OF A IS  5

     8.0      2.0      1.0      3.0      2.0
     1.0      0.0      0.0      0.0      3.0
     1.0      7.0      6.0      8.0      4.0
     2.0      6.0      3.0      2.0      1.0
     2.0      0.0      0.0      0.0      2.0

       .004 SECONDS


 MATRIX A WITH A NCNZERO MAIN DIAGONAL.

     2.0      0.0      0.0      0.0      2.0
     8.0      2.0      1.0      3.0      2.0
     1.0      7.0      6.0      8.0      4.0
     2.0      6.0      3.0      2.0      1.0
     1.0      0.0      0.0      0.0      3.0

   5=PER(   1)       1=PER(   2)       3=PER(   3)       4=PER(   4)
   2=PER(   5)

       .010 SECONDS


 MATRIX A IN A NORMAL FORM.

     2.0      2.0      0.0      0.0      0.0
     1.0      3.0      0.0      0.0      0.0
     8.0      2.0      2.0      1.0      3.0
     1.0      4.0      7.0      6.0      8.0
     2.0      1.0      6.0      3.0      2.0

   1=PER(   1)       5=PER(   2)       2=PER(   3)       3=PER(   4)
   4=PER(   5)

   2=IRR(   1)       3=IRR(   2)
```

INITIAL MATRIX A. THE ORDER OF A IS 8

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
32.0	33.0	34.0	35.0	36.0	37.0	38.0	39.0
24.0	25.0	26.0	27.0	28.0	29.0	30.0	31.0
20.0	21.0	20.0	21.0	22.0	23.0	0.0	0.0
14.0	15.0	16.0	17.0	18.0	19.0	0.0	0.0
5.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0
8.0	9.0	10.0	11.0	12.0	13.0	0.0	0.0

.012 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
5.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0
8.0	9.0	10.0	11.0	12.0	13.0	0.0	0.0
20.0	21.0	20.0	21.0	22.0	23.0	0.0	0.0
14.0	15.0	16.0	17.0	18.0	19.0	0.0	0.0
32.0	33.0	34.0	35.0	36.0	37.0	38.0	39.0
24.0	25.0	26.0	27.0	28.0	29.0	30.0	31.0

1=PER(1)	2=PER(2)	7=PER(3)	8=PER(4)
5=PER(5)	6=PER(6)	3=PER(7)	4=PER(8)

.014 SECONDS

MATRIX A IN A NORMAL FORM.

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
5.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0
8.0	9.0	10.0	11.0	12.0	13.0	0.0	0.0
20.0	21.0	20.0	21.0	22.0	23.0	0.0	0.0
14.0	15.0	16.0	17.0	18.0	19.0	0.0	0.0
32.0	33.0	34.0	35.0	36.0	37.0	38.0	39.0
24.0	25.0	26.0	27.0	28.0	29.0	30.0	31.0

1=PER(1)	2=PER(2)	3=PER(3)	4=PER(4)
5=PER(5)	6=PER(6)	7=PER(7)	8=PER(8)
2=IRR(1)	1=IRR(2)	3=IRR(3)	2=IRR(4)

INITIAL MATRIX A. THE ORDER OF A IS 9

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
0.0	5.0	6.0	0.0	0.0	0.0	7.0	0.0	0.0
0.0	8.0	9.0	0.0	10.0	11.0	12.0	13.0	0.0
0.0	36.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14.0	0.0	0.0	15.0	0.0	0.0	0.0	16.0	0.0
0.0	17.0	18.0	0.0	19.0	0.0	20.0	21.0	0.0
0.0	22.0	0.0	0.0	23.0	24.0	25.0	26.0	0.0
0.0	27.0	0.0	0.0	28.0	29.0	30.0	0.0	0.0
0.0	31.0	0.0	32.0	33.0	34.0	0.0	35.0	0.0

.010 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

14.0	0.0	0.0	15.0	0.0	0.0	0.0	16.0	0.0
0.0	36.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	5.0	6.0	0.0	0.0	0.0	7.0	0.0	0.0
0.0	31.0	0.0	32.0	33.0	34.0	0.0	35.0	0.0
0.0	8.0	9.0	0.0	10.0	11.0	12.0	13.0	0.0
0.0	27.0	0.0	0.0	28.0	29.0	30.0	0.0	0.0
0.0	17.0	18.0	0.0	19.0	0.0	20.0	21.0	0.0
0.0	22.0	0.0	0.0	23.0	24.0	25.0	26.0	0.0
1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0

5=PER(1) 4=PER(2) 2=PER(3) 9=PER(4)
3=PER(5) 8=PER(6) 6=PER(7) 7=PER(8)
1=PER(9)

.027 SECONDS

MATRIX A IN A NORMAL FORM.

36.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8.0	10.0	11.0	12.0	13.0	9.0	0.0	0.0	0.0
27.0	28.0	29.0	30.0	0.0	0.0	0.0	0.0	0.0
17.0	19.0	0.0	20.0	21.0	18.0	0.0	0.0	0.0
22.0	23.0	24.0	25.0	26.0	0.0	0.0	0.0	0.0
5.0	0.0	0.0	7.0	0.0	6.0	0.0	0.0	0.0
31.0	33.0	34.0	0.0	35.0	0.0	32.0	0.0	0.0
0.0	0.0	0.0	0.0	16.0	0.0	15.0	14.0	0.0
2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0

2=PER(1) 5=PER(2) 6=PER(3) 7=PER(4)
8=PER(5) 3=PER(6) 4=PER(7) 1=PER(8)
9=PER(9)

1=IRR(1) 5=IRR(2) 1=IRR(3) 1=IRR(4)
1=IRR(5)

INITIAL MATRIX A. THE ORDER OF A IS 5

1.0	0.0	0.0	2.0	0.0
3.0	0.0	0.0	4.0	0.0
5.0	0.0	0.0	6.0	0.0
7.0	0.0	0.0	0.0	8.0
0.0	9.0	10.0	0.0	0.0

.002 SECONDS

MATRIX A IS PARTLY DECOMPOSABLE. THE TERM RANK OF A IS 4

A MINIMUM COVER OF A.

COLUMNS 1, 4,

ROWS 4, 5,

INITIAL MATRIX A. THE ORDER OF A IS 5

1.0	0.0	2.0	3.0	0.0
0.0	4.0	5.0	6.0	0.0
7.0	0.0	0.0	0.0	0.0
8.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	9.0

.003 SECONDS

MATRIX A IS PARTLY DECOMPOSABLE. THE TERM RANK OF A IS 4

A MINIMUM COVER OF A.

COLUMNS 1,

ROWS 1, 2, 5,

INITIAL MATRIX A. THE ORDER OF A IS 8

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

.004 SECONDS

MATRIX A IS PARTLY DECOMPOSABLE. THE TERM RANK OF A IS 6

A MINIMUM COVER OF A.

COLUMNS 3,

ROWS 2, 3, 4, 5, 6,

INITIAL MATRIX A. THE ORDER OF A IS 8

1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0

.006 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0

1=PER(1)	2=PER(2)	3=PER(3)	4=PER(4)
5=PER(5)	6=PER(6)	7=PER(7)	8=PER(8)

.047 SECONDS

MATRIX A IS FULLY INDECOMPOSABLE.

INITIAL MATRIX A. THE ORDER OF A IS 9

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
0.0	5.0	6.0	0.0	0.0	0.0	7.0	0.0	0.0
0.0	8.0	9.0	0.0	10.0	11.0	12.0	13.0	0.0
14.0	0.0	0.0	15.0	0.0	0.0	0.0	16.0	0.0
0.0	17.0	18.0	0.0	19.0	0.0	20.0	21.0	0.0
0.0	22.0	0.0	0.0	23.0	24.0	25.0	26.0	0.0
0.0	27.0	0.0	0.0	28.0	29.0	30.0	0.0	0.0
0.0	31.0	0.0	32.0	33.0	34.0	0.0	35.0	0.0
0.0	36.0	0.0	37.0	0.0	0.0	0.0	0.0	38.0

.008 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
0.0	5.0	6.0	0.0	0.0	0.0	7.0	0.0	0.0
0.0	8.0	9.0	0.0	10.0	11.0	12.0	13.0	0.0
14.0	0.0	0.0	15.0	0.0	0.0	0.0	16.0	0.0
0.0	17.0	18.0	0.0	19.0	0.0	20.0	21.0	0.0
0.0	22.0	0.0	0.0	23.0	24.0	25.0	26.0	0.0
0.0	27.0	0.0	0.0	28.0	29.0	30.0	0.0	0.0
0.0	31.0	0.0	32.0	33.0	34.0	0.0	35.0	0.0
0.0	36.0	0.0	37.0	0.0	0.0	0.0	0.0	38.0

1=PER(1)   2=PER(2)   3=PER(3)   4=PER(4)
5=PER(5)   6=PER(6)   7=PER(7)   8=PER(8)
9=PER(9)

.017 SECONDS

MATRIX A IS FULLY INDECOMPOSABLE.

INITIAL MATRIX A. THE ORDER OF A IS 4

1.0	0.0	2.0	0.0
0.0	3.0	0.0	4.0
0.0	5.0	6.0	7.0
8.0	9.0	10.0	11.0

.003 SECONDS

MATRIX A WITH A NONZERO MAIN DIAGONAL.

1.0	0.0	2.0	0.0
0.0	3.0	0.0	4.0
0.0	5.0	6.0	7.0
8.0	9.0	10.0	11.0

1=PER(1) 2=PER(2) 3=PER(3) 4=PER(4)

.005 SECONDS

MATRIX A IS FULLY INDECOMPOSABLE.

B. Subroutine EGVARY

I. Purpose

Subroutine EGVARY (written in FORTRAN) determines whether or not a matrix has a transversal.

II. Method

The method has previously been presented in section 3.3 of the thesis.

III. Operation

(i) Description of variables

A- A is a matrix for which we wish a transversal. If a transversal is found, A is returned with a nonzero main diagonal.

HOR- HOR is an integer array. If a transversal is not found, then, if HOR(I)=1, I=1,2,...,N, row I of A is a horizontal line in a minimum cover of A.

I- I is a DO loop indexing variable or a variable subscript.

IHOLD- IHOLD is a temporary storage variable.

IL- IL is a temporary storage variable.

IND- In the search for an augmenting chain, point (I,S(I,IND(I)+1)) in A, $1 \leq I \leq N$, is the point under consideration.

J- J is a DO loop indexing variable or a variable subscript.

K- K is a variable subscript.

KK- KK is a temporary storage variable.

L- L is a DO loop indexing variable or a variable subscript.

L1- L1 is a variable subscript.

LL- At the end of the subroutine, if LL = 0, A has a transversal. N-LL is the term rank of A.

M- M is a variable subscript.

N- N is the order of matrix A.

NN- NN is the first dimension of doubly subscripted arrays A and S in the calling program.

NP1- NP1 = N+1

PER- PER is an integer array. If a transversal is found, row PER(I) of A, I=1,2,...,N, is permuted into row I.

REPBY- If REPBY(I) is not zero, I=1,2,...,N, then point (REPBY(I),I) in A is a point in the maximal transversal being constructed.

REPNG- If REPNG(I) is not zero, I=1,2,...,N, then point (I,REPNG(I)) in A is a point in the maximal transversal being constructed.

RHOLD- RHOLD is a temporary storage variable.

S- S is a representative matrix of A defined as follows: For I=1,2,...,N, if the nonzero points in row I of A are $(I,J_1),(I,J_2),\ldots,(I,J_K)$, then $S(I,1) = J_1, S(I,2) = J_2, \ldots, S(I,K) = J_K$, $S(I,K+1) = 0$ where the remainder of the points in row I of S are ignored.

SHORT- SHORT is an integer array. No point in the maximal transversal being constructed is in row SHORT(I), I=1,2,...,L1.

TAIL- TAIL is an integer array. In the search for an augmenting chain, TAIL(I), $1 \leq I \leq N$, was the previous row searched.

VERT- VERT is an integer array. If a transversal is not found, then, if VERT(I) = 1, I=1,2,...,N, column I of A is a vertical line in a minimum cover of A.

(ii) Entry points

EGVARY

(iii) Exit points

Normal return

(iv) Subprograms called

None

(v) Restrictions and limitations

None

IV. Storage Requirements

Subroutine EGVARY requires 236 words of computer memory on a CDC 6400. The following table shows additional storage required in the calling program for variable arrays.

ORDER OF MATRIX	CORE STORAGE (in decimal)
10	290
20	980
30	2070
40	3560
50	5450
60	7740

Table 1

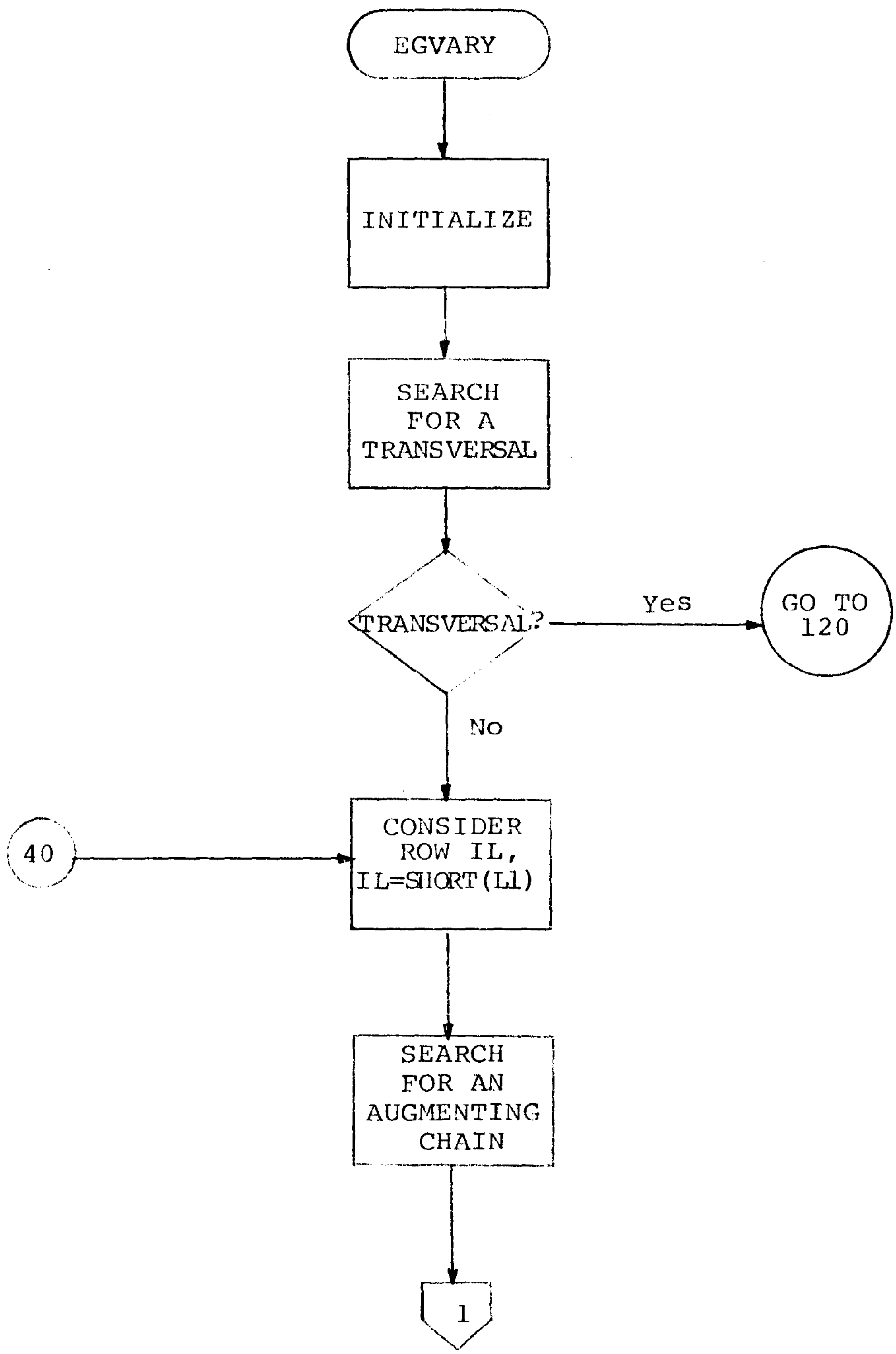
V. Timing

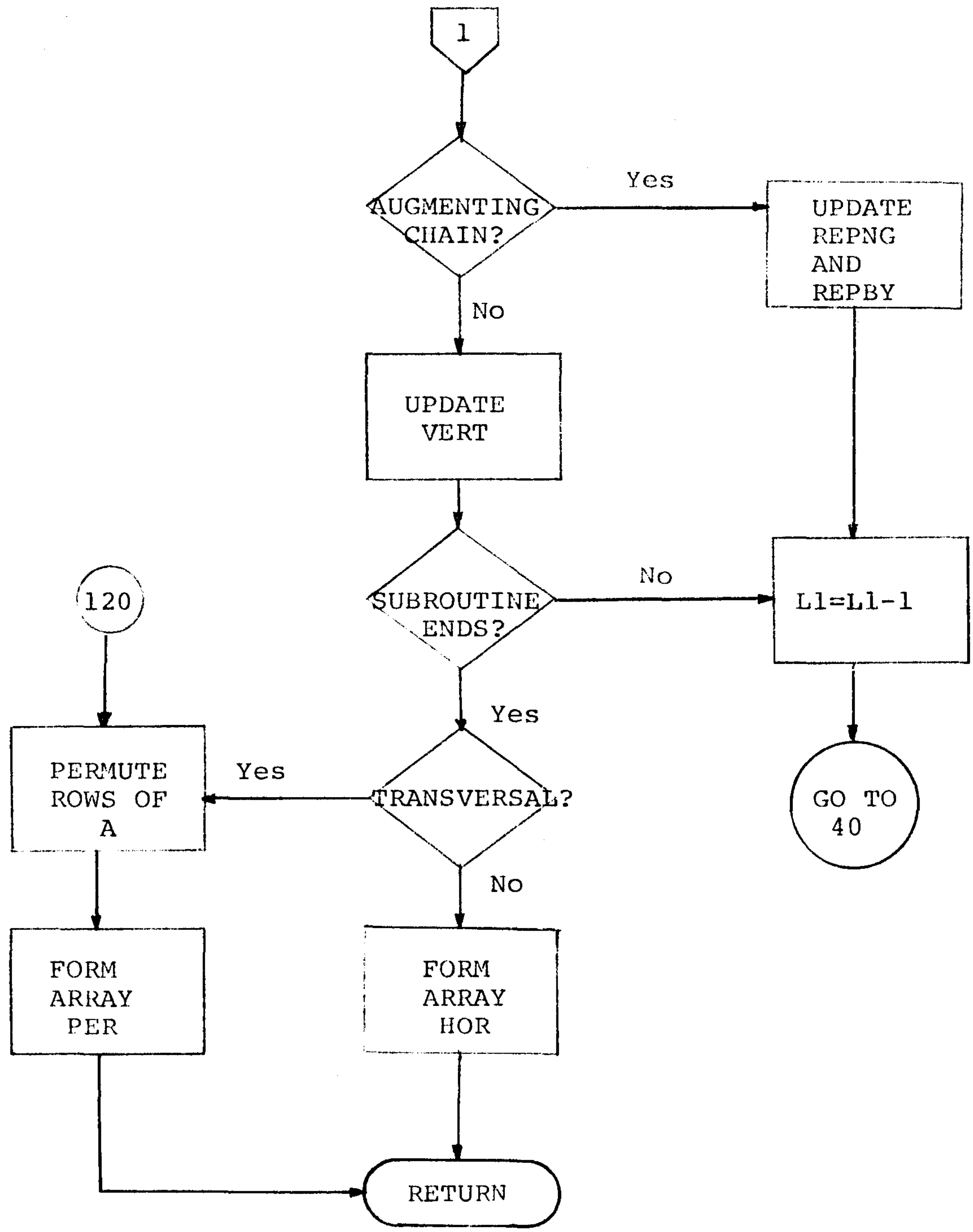
The identity matrix I yields the fastest computer time while matrix A = PL where L is a strictly positive lower triangular matrix and $P = (p_{ij})$ is a permutation matrix, $p_{n-i+1,i}=1$, $i=1,2,\ldots,n$, takes the longest computer time. Typical timings for these two types of matrices are shown in the following table.

ORDER OF MATRIX	TIME IN SECONDS	
	I	A
10	.005	.022
20	.021	.112
30	.027	.309
40	.044	.654
50	.121	1.187
60	.272	1.946

Table 2

VI. Flowchart and Listing




1
AUGMENTING CHAIN?
Yes
UPDATE REPNG AND REPBY
No
UPDATE VERT
SUBROUTINE ENDS?
No
L1=L1-1
Yes
GO TO 40
120
TRANSVERSAL?
Yes
PERMUTE ROWS OF A
No
FORM ARRAY PER
FORM ARRAY HOR
RETURN

```
      SUBROUTINE EGVARY(N,NN,LL,A,S,REPNG,REPBY,PER,SHORT,
     -                TAIL,IND,HOR,VERT)
C
C     PURPOSE
C        TO DETERMINE WHETHER OR NOT A MATRIX HAS A
C        TRANSVERSAL. IF SO, THE MATRIX IS RETURNED WITH A
C        NONZERO MAIN DIAGONAL.
C
C     DESCRIPTION OF PARAMETERS
C        INPUT-  N-      THE ORDER OF MATRIX A.
C                NN-     THE FIRST DIMENSION OF DOUBLY
C                        SUBSCRIPTED ARRAYS A AND S IN THE
C                        CALLING PROGRAM. THE SECOND
C                        DIMENSION OF ARRAY A IS AT LEAST
C                        N WHILE THE SECOND DIMENSION OF
C                        ARRAY S IS AT LEAST N+1.
C                S-      A DOUBLY SUBSCRIPTED WORK ARRAY.
C                REPNG-  A SINGLY SUBSCRIPTED WORK ARRAY
C                        OF DIMENSION AT LEAST N.
C                REPBY-  A SINGLY SUBSCRIPTED WORK ARRAY
C                        OF DIMENSION AT LEAST N.
C                SHORT-  A SINGLY SUBSCRIPTED WORK ARRAY
C                        OF DIMENSION AT LEAST N.
C                TAIL-   A SINGLY SUBSCRIPTED WORK ARRAY
C                        OF DIMENSION AT LEAST N.
```

```
C                 IND-    A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N.
C         IN/OUT- A-      A IS A MATRIX FOR WHICH WE WISH A
C                         TRANSVERSAL. IF A TRANSVERSAL IS
C                         FOUND, A IS RETURNED WITH A NONZERO
C                         MAIN DIAGONAL.
C         OUTPUT- PER-    PER IS AN INTEGER ARRAY. IF A
C                         TRANSVERSAL IS FOUND, ROW PER(I)
C                         OF A, I=1,2,...,N, IS PERMUTED
C                         INTO ROW I.
C                 VERT-   VERT IS AN INTEGER ARRAY. IF A
C                         TRANSVERSAL IS NOT FOUND, THEN, IF
C                         VERT(I)=1, I=1,2,...,N, COLUMN I
C                         OF A IS A VERTICAL LINE IN A
C                         MINIMUM COVER OF A.
C                 HOR-    HOR IS AN INTEGER ARRAY. IF A
C                         TRANSVERSAL IS NOT FOUND, THEN, IF
C                         HOR(I)=1, I=1,2,...,N, ROW I OF A
C                         IS A HORIZONTAL LINE IN A MINIMUM
C                         COVER OF A.
C                 LL-     IF LL=0, THEN A HAS A TRANSVERSAL.
C                         N-LL IS THE TERM RANK OF A.
C
      INTEGER REPNG,REPBY,SHORT,TAIL,PER,VERT,HOR
      DIMENSION A(NN,1),S(NN,1),REPNG(1),REPBY(1),SHORT(1),
```

```
     -             TAIL(1),PER(1),IND(1),HOR(1),VERT(1)
C
C     FORM REPRESENTATIVE MATRIX S.
C
      DO 20 I=1,N
      L=1
      DO 19 J=1,N
      IF(A(I,J).EQ.0.0) GO TO 19
      S(I,L)=J
      L=L+1
   19 CONTINUE
   20 S(I,L)=0
C
C     INITIALIZE.
C
      DO 21 I=1,N
      VERT(I)=0
      REPBY(I)=0
   21 REPNG(I)=0
C
C     K IS THE NEXT ENTRY IN ROW I. IF K=0, EITHER THERE
C     IS NO NONZERO POINT IN ROW I OR ELSE EACH NONZERO
C     POINT IN ROW I IS IN THE SAME COLUMN AS A POINT IN
C     THE MAXIMAL TRANSVERSAL BEING CONSTRUCTED.
C
```

```
      L1=0
      NP1=N+1
      DO 30 I=1,N
      DO 25 J=1,NP1
      K=S(I,J)
      IF(K.EQ.0) GO TO 24
      IF(REPBY(K).NE.0) GO TO 25
      REPBY(K)=I
      REPNG(I)=K
      GO TO 30
   24 L1=L1+1
C
C     NO POINT IN THE MAXIMAL TRANSVERSAL BEING CONSTRUCTED
C     IS IN ROW SHORT(I), I=1,2,...,L1.
C
      SHORT(L1)=I
      GO TO 30
   25 CONTINUE
   30 CONTINUE
C
C     IF L1=0, A TRANSVERSAL HAS BEEN FOUND. GO TO PERMUTE
C     THE ROWS OF A.
C
      LL=0
      IF(L1.EQ.0) GO TO 120
```

```
C
C     SEARCH FOR AUGMENTING CHAINS.
C
   40 DO 50 I=1,N
      IND(I)=0
   50 TAIL(I)=0
C
C     SEARCH FOR AN AUGMENTING CHAIN BEGINNING WITH A
C     NONZERO POINT IN ROW IL. POINT (I,S(I,IND(I)+1) IS THE
C     POINT UNDER CONSIDERATION. ROW TAIL(I) WAS THE
C     PREVIOUS ROW.
C
      IL=SHORT(L1)
      I=IL
   60 J=IND(I)
   70 J=J+1
      K=S(I,J)
C
C     IF K=0, EITHER EACH ENTRY IN ROW I IS ZERO OR
C     ELSE THERE IS NO AUGMENTING CHAIN THROUGH ROW I.
C
      IF(K.EQ.0) GO TO 80
C
C     IF VERT(K)=1, THEN POINT (I,K) CANNOT BE A POINT IN
C     THE AUGMENTING CHAIN BEING CONSTRUCTED.
```

```
C
      IF(VERT(K).EQ.1) GO TO 70
C
C     IF M=I, THEN POINT (I,K) IS A POINT IN THE MAXIMAL
C     TRANSVERSAL.
C
      M=REPBY(K)
      IF(M.EQ.I) GO TO 70
C
C     IF M=0, WE HAVE AN AUGMENTING CHAIN.
C
      IF(M.EQ.0) GO TO 100
C
C     IF TAIL(M) IS NOT ZERO, WE HAVE ALREADY TRAVELLED
C     THROUGH ROW M.
C
      IF(TAIL(M).NE.0) GO TO 70
C
C     IND AND TAIL ARE UPDATED.
C
      IND(I)=J
      TAIL(M)=I
C
C     WE HAVE ARRIVED FROM ROW I TO ROW M.
C
```

```
      I=M
      GO TO 60
C
C     IF I=IL, THERE IS NO AUGMENTING CHAIN FROM A NONZERO
C     POINT IN ROW I.
C
   80 IF(I.EQ.IL) GO TO 90
C
C     BACK UP FROM ROW I TO ROW TAIL(I).
C
      I=TAIL(I)
      GO TO 60
C
C     AT THE END OF THE SUBROUTINE, IF VERT(I)=1,
C     I=1,2,...,N, THEN COLUMN I IS A VERTICAL LINE IN A
C     MINIMUM COVER OF A.
C
   90 LL=LL+1
      DO 95 I=1,N
      K=TAIL(I)
      IF(K.EQ.0) GO TO 95
      L=REPNG(I)
      VERT(L)=1
   95 CONTINUE
      GO TO 110
```

```
C
C     THE MAXIMAL TRANSVERSAL IS AUGMENTED. REPNG AND REPBY
C     ARE AUGMENTED.
C
  100 KK=REPNG(I)
      REPNG(I)=K
      REPBY(K)=I
      IF(I.EQ.IL) GO TO 110
      I=TAIL(I)
      K=KK
      GO TO 100
C
C     L1 IS DECREMENTED. IF L1=0, SHORT IS EXHAUSTED.
C
  110 L1=L1-1
      IF(L1.NE.0) GO TO 40
C
C     IF LL=0, A TRANSVERSAL HAS BEEN FOUND.
C
      IF(LL.EQ.0) GO TO 120
C
C     IF HOR(I)=1, I=1,2,...,N, ROW I IS A HORIZONTAL LINE
C     IN A MINIMUM COVER OF A. NO TRANSVERSAL HAS BEEN
C     FOUND.
C
```

```
      DO 114 I=1,N
  114 HOR(I)=0
      DO 115 I=1,N
      K=REPBY(I)
      IF(K.EQ.0) GO TO 115
      IF(VERT(I).EQ.1) GO TO 115
      HOR(K)=1
  115 CONTINUE
      RETURN
C
C     INITIALIZE.
C
  120 DO 121 I=1,N
  121 PER(I)=I
C
C     PERMUTE THE ROWS OF A SUCH THAT A HAS A NONZERO MAIN
C     DIAGONAL.
C
      DO 190 L=1,N
      K=REPBY(L)
      IHOLD=PER(L)
      PER(L)=PER(K)
      PER(K)=IHOLD
      DO 168 J=1,N
      RHOLD=A(K,J)
```

```
    A(K,J)=A(L,J)
168 A(L,J)=RHOLD
170 REPNG(K)=REPNG(L)
    K=REPNG(L)
190 REPBY(K)=REPBY(L)
    RETURN
    END
```

C. Subroutine DULMEN

I. Purpose

Subroutine DULMEN (written in FORTRAN) decomposes a matrix to a normal form.

II. Method

The method has previously been presented in section 3.1 of the thesis.

III. Operation

(i) Description of variables

A- Initially A is a square matrix having a nonzero main diagonal. At the end of the subroutine, A is in a normal form.

C1- C1, $1 \leq C1 \leq NP1$, is augmented by 1 each time the alternating chain being constructed changes direction. C1 is an integer variable.

CEND- CEND = C1-1. CEND is an integer variable.

CHAIN- CHAIN(I), $1 \leq I \leq C1$, denotes a row of S in the construction of an alternating chain. When CHAIN(I) = CHAIN(C1), I<C1, a simple alternating cycle is found. CHAIN is an integer array.

FINISH-	FINISH is an integer variable array used for subscripting array HLDPER.
H1-	H1 is a variable integer subscript.
HH1-	HH1 is a variable integer subscript.
HLDCH-	HLDCH is an integer storage array.
HLDPER-	HLDPER is an integer array. At any stage of the subroutine, row I of S, I=NI,...,NL, represents rows HLDPER(START(I)) to HLDPER(FINISH(I)) inclusive of A.
I-	I is a DO loop indexing variable or a variable subscript.
IRR-	If A is not fully indecompsable, IRR(I), I=1,2,...,KK, is the coverance of the I-th fully indecomposable diagonal submatrix of A in a normal form.
J-	J is a DO loop indexing variable or a variable subscript.
K-	K is a variable subscript.
KK-	See the description of IRR.
L-	L is a variable subscript.
L1-	L1 is a DO loop indexing parameter.
L2-	L2 is a DO loop indexing parameter.
LL1-	LL1 is a DO loop indexing parameter.

LL2- LL2 is a DO loop indexing parameter.

M- M is a DO loop indexing variable.

M1- M1 is a DO loop indexing variable.

M2- M2 is a DO loop indexing variable.

M3- M3 is a DO loop indexing variable.

MM- At the end of the subroutine, if MM=0, A is fully indecomposable. If MM=1, A is partly decomposable.

MP1- MP1 is a variable subscript.

N- N is the order of matrix A.

NI- Initially NI=1. When a new alternating chain is started, NI is augmented by 1.

NI1- NI1=NI+1

NEWNL- NEWNL=NL-(CEND-I+1) where (CEND-I+1) is the length of a simple alternating cycle.

NL- Initially NL=N. After a simple alternating cycle has been found, NL is diminished by the length of the cycle.

NN- NN is the first dimension of doubly subscripted arrays A and S in the calling program.

NP1- NP1=N+1

P1- P1 is a variable integer subscript.

PER- PER is an integer array. If A is not fully indecomposable, row PER(I) of A, I=1,2,...,N, is permuted into row I while column PER(I) of A is permuted into column I.

R1- R1 is a variable integer subscript.

S- Initially S is a representative matrix of A. During the subroutine, S is the representative matrix of the induced matrix in question.

START- START is a variable integer array used for subscripting array HLDPER.

TEMP- TEMP is an integer storage variable.

TEMP1- TEMP1 is an integer storage variable.

TEMP2- TEMP2 is an integer storage variable.

TEMP3- TEMP3 is an integer storage variable.

WORK- **WORK** is an integer storage array.

ZERO1- ZERO1 is an integer work array.

(ii) Entry points

DULMEN

(iii) Exit points

Normal return

(iv) Subprograms called

None

(v) Restrictions and limitations

The only restriction (required by the algorithm) is that A must have a nonzero main diagonal.

IV. Storage Requirements

Subroutine DULMEN requires 534 words of computer memory on a CDC 6400. The following table shows additional storage required in the calling program for variable arrays.

ORDER OF MATRIX	CORE STORAGE (in decimal)
10	305
20	1005
30	2105
40	3605
50	5505
60	7805

Table 3

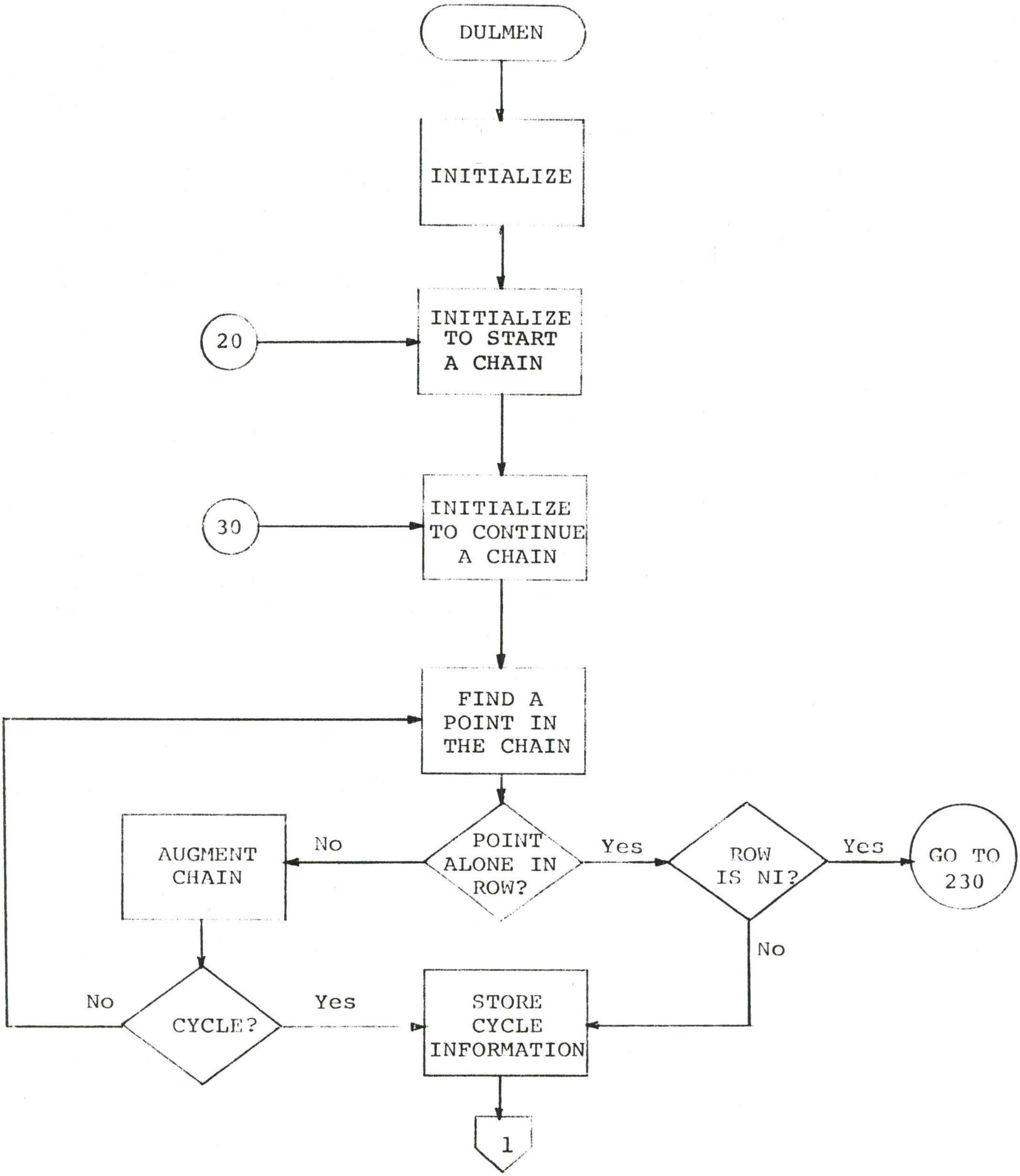
V. Timing

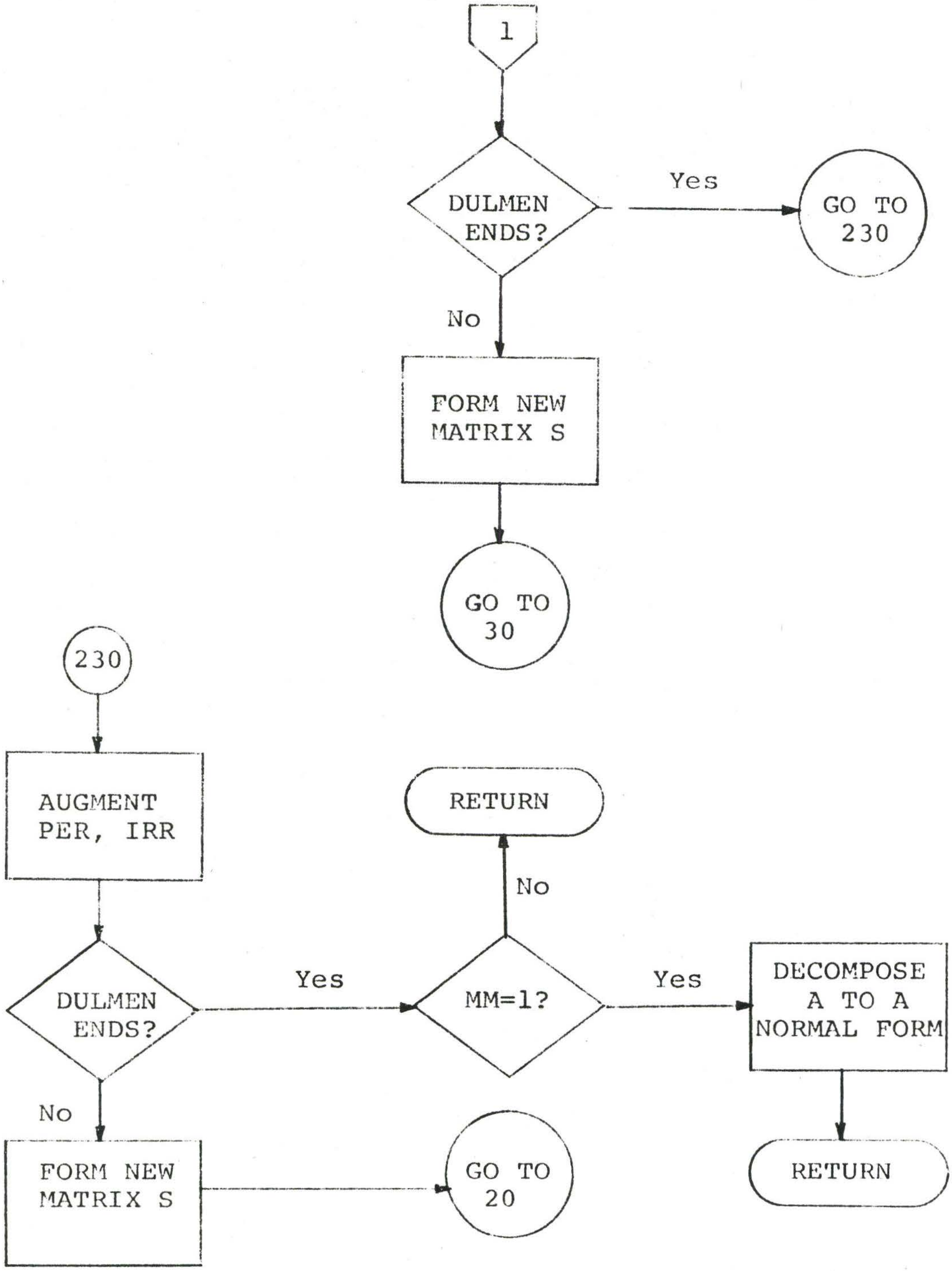
We define a step matrix A of order N as follows: $A(N,1)\neq 0$, the main diagonal and superdiagonal of A are strictly positive, the remaining entries of A are zero. A step matrix B yields the fastest computer time while a tridiagonal matrix D with a strictly positive band requires the longest computer time. Typical timings for these two types of matrices are shown in the following table.

ORDER OF MATRIX	TIME IN SECONDS	
	B	D
10	0.005	0.046
20	0.016	0.340
30	0.033	1.116
40	0.057	2.714
50	0.089	5.298
60	0.125	8.813

Table 4

VI. Flowchart and Listing




1
DULMEN ENDS?
Yes
GO TO 230
No
FORM NEW MATRIX S
GO TO 30
230
AUGMENT PER, IRR
DULMEN ENDS?
Yes
MM=1?
No
RETURN
Yes
DECOMPOSE A TO A NORMAL FORM
RETURN
No
FORM NEW MATRIX S
GO TO 20

```
      SUBROUTINE DULMEN(N,NN,KK,MM,A,S,PER,IRR,WORK,HLDCH,
     -                 ZERO1,CHAIN,HLDPER,START,FINISH)
C
C     PURPOSE
C        TO DECOMPOSE A MATRIX TO A NORMAL FORM.
C
C     DESCRIPTION OF PARAMETERS
C        INPUT-  N-       THE ORDER OF MATRIX A.
C                NN-      THE FIRST DIMENSION OF DOUBLY
C                         SUBSCRIPTED ARRAYS A AND S IN THE
C                         CALLING PROGRAM. THE SECOND
C                         DIMENSION OF ARRAY A IS AT LEAST
C                         N WHILE THE SECOND DIMENSION OF
C                         ARRAY S IS AT LEAST N+1.
C                S-       A DOUBLY SUBSCRIPTED WORK ARRAY.
C                WORK-    A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N.
C                HLDCH-   A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N.
C                ZERO1-   A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N+1.
C                CHAIN-   A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N+1.
C                HLDPER-  A SINGLY SUBSCRIPTED WORK ARRAY
C                         OF DIMENSION AT LEAST N+1.
```

```
C               START-  A SINGLY SUBSCRIPTED WORK ARRAY
C                       OF DIMENSION AT LEAST N+1.
C               FINISH- A SINGLY SUBSCRIPTED WORK ARRAY
C                       OF DIMENSION AT LEAST N+1.
C        IN/OUT- A-     INITIALLY A IS THE MATRIX WITH A
C                       NONZERO MAIN DIAGONAL TO BE
C                       DECOMPOSED. ON RETURN, A IS IN A
C                       NORMAL FORM.
C        OUTPUT- PER-   PER IS AN INTEGER ARRAY.
C                       IF A IS NOT FULLY INDECOMPOSABLE,
C                       ROW PER(I), I=1,2,...,N, OF A IS
C                       PERMUTED INTO ROW I AND COLUMN
C                       PER(I) OF A IS PERMUTED INTO
C                       COLUMN I.
C               IRR-    IF A IS NOT FULLY INDECOMPOSABLE,
C                       IRR(I), I=1,2,...KK,, IS THE
C                       COVERANCE OF THE I-TH FULLY
C                       INDECOMPOSABLE DIAGONAL SUBMATRIX
C                       OF A IN A NORMAL FORM.
C               KK-     SEE THE DESCRIPTION OF IRR.
C               MM-     IF MM=0, A IS FULLY INDECOMPOSABLE.
C                       IF MM=1, A IS PARTLY DECOMPOSABLE.
C
      INTEGER ZERO1,PER,HLDPER,CHAIN,HLDCH,START,FINISH,WORK,
     -      P1,H1,HH1,R1,C1,CEND,TEMP,TEMP1,TEMP2,TEMP3
```

```
      DIMENSION A(NN,1),S(NN,1),PER(1),IRR(1),WORK(1),
     -          HLDCH(1),ZERO1(1),CHAIN(1),HLDPER(1),
     -          START(1),FINISH(1)
C
C     INITIALIZE.
C
      NI=1
      NL=N
      P1=0
      KK=0
      MM=0
      NP1=N+1
      DO 10 I=1,NP1
      ZERO1(I)=1
      HLDPER(I)=I
      START(I)=I
   10 FINISH(I)=I
C
C     FORM REPRESENTATIVE MATRIX S FOR A.
C
      DO 18 I=1,N
      L=1
      DO 17 J=1,N
      IF(A(I,J).EQ.0.0) GO TO 17
      S(I,L)=J
```

```
      L=L+1
   17 CONTINUE
   18 S(I,L)=0
      I=1
C
C     START AT 20 TO BEGIN A CHAIN.
C
   20 NI1=NI+1
C
C     START AT 30 TO CONTINUE A CHAIN.
C
   30 C1=I
      CHAIN(I)=NI
      L=NI
      J=NI
C
C     IF S(L,J+1)=0, THEN THERE ARE NO ELEMENTS TO THE
C     RIGHT IN ROW L. OTHERWISE, S(L,J+1) BECOMES THE
C     NEXT ELEMENT IN THE CHAIN.
C
   40 K=J+1
      IF(S(L,K).NE.0.0) GO TO 60
C
C     IF J=NI, S(L,NI) IS ALONE IN ROW L.
C
```

```
      IF(J.NE.NI) GO TO 59
C
C     IF L=NI, S(NI,NI) IS ALONE IN ROW NI.
C
      IF(L.EQ.NI) GO TO 230
      I=C1
      CEND=C1
      GO TO 90
   59 K=NI
C
C     AN ELEMENT HAS BEEN FOUND FOR THE CHAIN.
C
   60 CEND=C1
      C1=CEND+1
      CHAIN(C1)=S(L,K)
      L=S(L,K)
C
C     BRANCH IF A CYCLE IS COMPLETED. OTHERWISE SEARCH
C     FOR A NEW MEMBER FOR THE CHAIN.
C
      DO 70 I=1,CEND
      IF(CHAIN(I).EQ.CHAIN(C1)) GO TO 90
   70 CONTINUE
C
C     BRANCHING ALWAYS OCCURS BY THE FOLLOWING DO LOOP.
```

```
C
      DO 80 J=NI,NL
      IF(S(L,J).EQ.FLOAT(L)) GO TO 40
   80 CONTINUE
C
C     A CYCLE HAS BEEN FOUND FOR THE CHAIN.
C
   90 NEWNL=NL-(CEND-I)
C
C     IF NEWNL=NI, GO TO PERMUTE THE ROWS AND COLUMNS OF A.
C
      IF(NEWNL.NE.NI) GO TO 100
C
C     IF MM=0, A IS FULLY INDECOMPOSABLE.
C
      IF(MM.EQ.0) RETURN
      L1=START(NI)
      L2=FINISH(NL)
      GO TO 235
C
C     INFORMATION STORED IN HLDPER IS LATER TRANSFERRED TO
C     PER WHICH CONTAINS ALL THE NECESSARY INFORMATION TO
C     PERMUTE THE ROWS AND COLUMNS OF A AT THE END OF THE
C     SUBROUTINE.
C
```

```
C     REINITIALIZE FOR NEW REPRESENTATIVE MATRIX.
C
  100 H1=0
      DO 110 M=I,CEND
      TEMP=CHAIN(M)
      ZERO1(TEMP)=0
      L1=START(TEMP)
      L2=FINISH(TEMP)
      DO 110 M1=L1,L2
      H1=H1+1
  110 HLDCH(H1)=HLDPER(M1)
      L1=START(NI)
      L2=FINISH(NL)
      DO 111 M=L1,L2
  111 WORK(M)=HLDPER(M)
      DO 112 M=1,H1
  112 HLDPER(M)=HLDCH(M)
      L1=1
      L2=H1
      R1=NI
      TEMP1=ZERO1(NI)
      TEMP2=START(NI)
      TEMP3=FINISH(NI)
      DO 120 M=NI,NL
      MP1=M+1
```

```
      IF(TEMP1.EQ.0) GO TO 115
      R1=R1+1
      HLDCH(M)=R1
      TEMP=TEMP1
      TEMP1=ZERO1(MP1)
      ZERO1(R1)=TEMP
      TEMP=TEMP2
      TEMP2=START(MP1)
      START(R1)=TEMP
      TEMP=TEMP3
      TEMP3=FINISH(MP1)
      FINISH(R1)=TEMP
      GO TO 120
  115 TEMP1=ZERO1(MP1)
      TEMP2=START(MP1)
      TEMP3=FINISH(MP1)
  120 CONTINUE
      ZERO1(NI)=1
      START(NI)=1
      FINISH(NI)=H1
      NL=NEWNL
      HH1=H1
      DO 126 M=NI1,NL
      L1=START(M)
      L2=FINISH(M)
```

```
      DO 125 M1=L1,L2
      HH1=HH1+1
  125 HLDPER(HH1)=WORK(M1)
      START(M)=HH1-(L2-L1)
  126 FINISH(M)=HH1
C
C     RETRIEVE THE FIRST PART OF THE CHAIN.
C
      IF(I.EQ.1) GO TO 128
      I1=I-1
      DO 127 M=1,I1
      TEMP=CHAIN(M)
  127 CHAIN(M)=HLDCH(TEMP)
C
C     FIRST ROW OF NEW REPRESENTATIVE MATRIX.
C
  128 R1=NI
      S(NI,NI)=NI
      DO 160 M=NI1,NL
      L1=START(M)
      L2=FINISH(M)
      DO 131 M1=1,H1
      TEMP1=HLDPER(M1)
      DO 131 M2=L1,L2
      TEMP2=HLDPER(M2)
```

```
      IF(A(TEMP1,TEMP2).NE.0.0) GO TO 150
  131 CONTINUE
      GO TO 160
  150 R1=R1+1
      S(NI,R1)=M
  160 CONTINUE
      S(NI,R1+1)=0
C
C     REMAINING ROWS OF NEW REPRESENTATIVE MATRIX.
C
      DO 220 M=NI1,NL
      R1=NI
      L1=START(M)
      L2=FINISH(M)
      DO 170 M1=1,H1
      TEMP2=HLDPER(M1)
      DO 170 M2=L1,L2
      TEMP1=HLDPER(M2)
      IF(A(TEMP1,TEMP2).NE.0.0) GO TO 190
  170 CONTINUE
      R1=R1-1
      GO TO 200
  190 S(M,NI)=NI
  200 DO 215 M1=NI1,NL
      DO 205 M2=L1,L2
```

```
      TEMP1=HLDPER(M2)
      LL1=START(M1)
      LL2=FINISH(M1)
      DO 205 M3=LL1,LL2
      TEMP2=HLDPER(M3)
      IF(A(TEMP1,TEMP2).NE.0.0) GO TO 210
  205 CONTINUE
      GO TO 215
  210 R1=R1+1
      S(M,R1)=M1
  215 CONTINUE
  220 S(M,R1+1)=0
C
C     CONTINUE THE CHAIN.
C
      GO TO 30
C
C     AUGMENT PER.
C
  230 L1=START(NI)
      L2=FINISH(NI)
  235 DO 238 M=L1,L2
      P1=P1+1
  238 PER(P1)=HLDPER(M)
C
```

```
C     AUGMENT IRR.
C
      KK=KK+1
      IRR(KK)=L2-L1+1
C
C     IF P1=N, GO TO PERMUTE THE ROWS AND COLUMNS OF A.
C
      IF(P1.EQ.N) GO TO 270
C
C     FORM A NEW REPRESENTATIVE MATRIX.
C
      DO 260 M=NI1,NL
      IF(S(M,NI).EQ.FLOAT(NI)) GO TO 260
      TEMP1=S(M,NI)
      DO 250 M1=NI1,NL
      TEMP2=S(M,M1)
      S(M,M1)=TEMP1
  250 TEMP1=TEMP2
      S(M,NL+1)=0
  260 CONTINUE
C
C     BEGIN A NEW CHAIN.
C
      MM=1
      NI=NI+1
```

```
      I=1
      GO TO 20
C
C     PERMUTE THE ROWS AND COLUMNS OF A.
C
  270 DO 280 M=1,N
      TEMP=PER(M)
      DO 280 M1=1,N
  280 S(M1,M)=A(M1,TEMP)
      DO 290 M=1,N
      TEMP=PER(M)
      DO 290 M1=1,N
  290 A(M,M1)=S(TEMP,M1)
      RETURN
      END
```

D. Subroutine HARARY

I. Purpose

Subroutine HARARY (written in FORTRAN) decomposes a matrix to a normal form.

II. Method

The method has previously been presented in section 3.2 of the thesis.

III. Operation

(i) Description of variables

A- Initially A is a square matrix having a nonzero main diagonal. At the end of the subroutine, A is in a normal form.

B- At the end of phase one of the subroutine, B is the reachability matrix of the (0,1)-matrix formed from A by replacing nonzero entries in A by 1.

I- I is a DO loop indexing variable.

INDEX- INDEX is the power of B at each stage of phase one of the algorithm. When INDEX is greater than or equal to NM1, phase one is complete and the reachability matrix B is found.

IRR- If A is not fully indecomposable, IRR(I), I=1,2,...,KK, is the coverance of the I-th fully indecomposable diagonal submatrix of A in a normal form.

IWORK- IWORK is a work array.

J- J is a DO loop indexing variable or a variable subscript.

JHOLD- JHOLD is a work array.

K- K is a DO loop indexing variable.

KK- See description of IRR.

L1- L1 is a DO loop indexing parameter.

L2- L2 is a variable subscript.

M- M is a DO loop indexing variable.

M1- M1 is a DO loop indexing variable.

MM- At the end of the subroutine, if MM=0, A is fully indecomposable. If MM=1, A is partly decomposable.

N- N is the order of A.

NM1- NM1 = N-1

NN- NN is the first dimension of doubly subscripted arrays A, S and B in the calling program.

PER- PER is an integer array. If A is not fully indecomposable, row PER(I) of A, I=1,2,...,N, is permuted into row I while column PER(I) of A is permuted into column I.

S- Initially S is the (0,1)-matrix formed from A by replacing nonzero entries in A by L. S is used to compute the reachability matrix B.

TEMP- TEMP is an integer storage variable.

(ii) Entry points

HARARY

(iii) Exit points

Normal return.

(iv) Subprograms called

None

(v) Restrictions and limitations

The only restriction (required by the algorithm) is that A must have a nonzero main diagonal.

IV. Storage Requirements

Subroutine HARARY requires 218 words of computer memory on a CDC 6400. The following table shows additional storage required in the calling program for variable arrays.

ORDER OF MATRIX	CORE STORAGE (in decimal)
10	340
20	1280
30	2820
40	4960
50	7700
60	11040

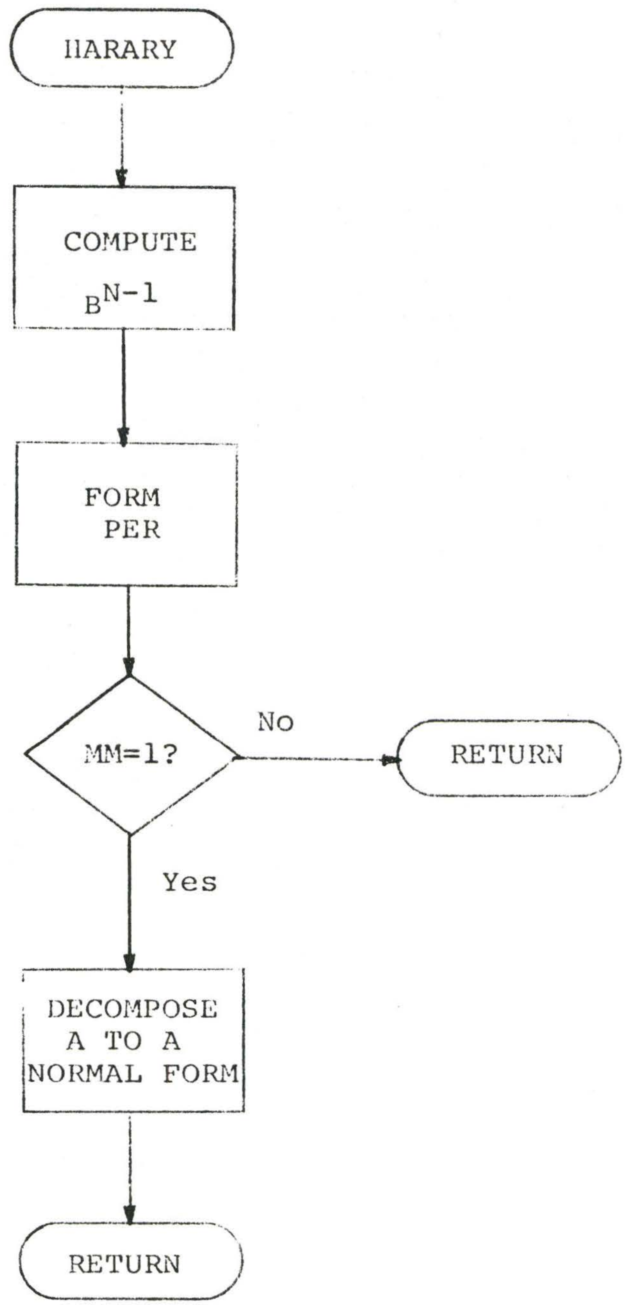
Table 5

V. Timing

Typical timings for the two types of matrices B and D (defined in section V of Appendix C) are shown in the following table.

ORDER OF MATRIX	TIME IN SECONDS	
	B	D
10	.132	.117
20	1.408	1.244
30	5.458	4.876
40	14.426	12.970
50	30.101	27.281
60	55.128	50.676

VI. Flowchart and Listing



```
      SUBROUTINE HARARY(N,NN,KK,MM,A,S,B,PER,IRR,JHOLD,
     -                  IWORK)
C
C     PURPOSE
C        TO DECOMPOSE A MATRIX TO A NORMAL FORM.
C
C     DESCRIPTION OF PARAMETERS
C        INPUT-  N-     THE ORDER OF MATRIX A.
C                NN-    THE FIRST DIMENSION OF DOUBLY
C                       SUBSCRIPTED ARRAYS A, S AND B IN
C                       THE CALLING PROGRAM. THE SECOND
C                       DIMENSION OF ARRAYS A, S AND B IS
C                       AT LEAST N.
C                S-     A DOUBLY SUBSCRIPTED WORK ARRAY.
C                B-     A DOUBLY SUBSCRIPTED WORK ARRAY.
C                JHOLD- A SINGLY SUBSCRIPTED WORK ARRAY OF
C                       DIMENSION AT LEAST N.
C                IWORK- A SINGLY SUBSCRIPTED WORK ARRAY OF
C                       DIMENSION AT LEAST N.
C        IN/OUT- A-     INITIALLY A IS THE MATRIX WITH A
C                       NONZERO MAIN DIAGONAL TO BE
C                       DECOMPOSED. ON RETURN, A IS IN A
C                       NORMAL FORM.
C        OUTPUT- PER-   PER IS AN INTEGER ARRAY.
C                       IF A IS NOT FULLY INDECOMPOSABLE,
```

```
C                          ROW PER(I), I=1,2,...,N, OF A IS
C                          PERMUTED INTO ROW I AND COLUMN
C                          PER(I) OF A IS PERMUTED INTO
C                          COLUMN I.
C                    IRR-  IF A IS NOT FULLY INDECOMPOSABLE,
C                          IRR(I), I=1,2,...,KK, IS THE
C                          COVERANCE OF THE I-TH FULLY
C                          INDECOMPOSABLE DIAGONAL SUBMATRIX
C                          OF A IN A NORMAL FORM.
C                    KK-   SEE THE DESCRIPTION OF IRR.
C                    MM-   IF MM=0, A IS FULLY INDECOMPOSABLE.
C                          IF MM=1, A IS PARTLY DECOMPOSABLE.
C
      INTEGER TEMP,PER
      DIMENSION A(NN,1),S(NN,1),B(NN,1),PER(1),IRR(1),
     -          JHOLD(1),IWORK(1)
C
C     INITIALIZE.
C
      IF(N.LE.2) STOP 44
      NM1=N-1
      INDEX=1
      L2=0
      KK=0
      MM=0
```

```
      DO 10 I=1,N
      DO 10 J=1,N
      S(I,J)=0
      IF(A(I,J).NE.0.0) S(I,J)=1
   10 CONTINUE
      DO 12 I=1,N
   12 IWORK(I)=0
C
C     PHASE 1. FORM MATRIX B**(N-1).
C
   15 DO 30 I=1,N
      DO 30 J=1,N
      DO 20 K=1,N
      B(I,J)=S(I,K)*S(K,J)
      IF(B(I,J).EG.1.0) GO TO 30
   20 CONTINUE
   30 CONTINUE
      INDEX=INDEX+INDEX
      IF(INDEX.GE.NM1) GO TO 50
      DO 40 I=1,N
      DO 40 J=1,N
   40 S(I,J)=B(I,J)
      GO TO 15
C
C     PHASE 2. FORM PER AND IRR.
```

```
C
   50 DO 70 I=1,N
      IF(IWORK(I).EQ.1) GO TO 70
      L1=0
      DO 60 J=1,N
      IF(B(I,J).EQ.0.0) GO TO 60
      IF(B(I,J).NE.B(J,I)) GO TO 70
      L1=L1+1
      JHOLD(L1)=J
   60 CONTINUE
      GO TO 75
   70 CONTINUE
   75 DO 80 K=1,L1
      L2=L2+1
   80 PER(L2)=JHOLD(K)
      KK=KK+1
      IRR(KK)=L1
      IF(L2.EQ.N) GO TO 110
      DO 100 K=1,L1
      J=JHOLD(K)
      IWORK(J)=1
      DO 100 I=1,N
  100 B(I,J)=0
      MM=1
      GO TO 50
```

```
C
C     PHASE 3. DECOMPOSE A TO A NORMAL FORM.
C
  110 IF(MM.EQ.0) RETURN
      DO 120 M=1,N
      TEMP=PER(M)
      DO 120 M1=1,N
  120 S(M1,M)=A(M1,TEMP)
      DO 130 M=1,N
      TEMP=PER(M)
      DO 130 M1=1,N
  130 A(M,M1)=S(TEMP,M1)
      RETURN
      END
```

BIBLIOGRAPHY

1. Forsythe, G. E. and Moler, C. B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1957, pp. 14-15.

2. König, D., Theorie der endlichen und unendlichen Graphen, Chelsea, New York, 1950.

3. Hall, P., On representatives of subsets, J. London Math. Soc., 10 (1935), pp. 26-30.

4. Marcus, M. and Minc, H., A Survey of Matrix Theory and Matrix Inequalities, Allyn and Bacon, Boston, 1964.

5. Brualdi, R. A., Parter, S. V. and Schneider, H., The diagonal equivalence of a nonnegative matrix to a stochastic matrix, Journal of Mathematical Analysis and Applications, 16 (1966), pp. 31-50.

6. Dulmage, A. L. and Mendelsohn, N. S., Two algorithms for bipartite graphs, J. Soc. Indust. Appl. Math., 11 (1963), pp. 183-194.

7. Harary, F., A graph theoretic method for the complete reduction of a matrix with a view toward finding its eigenvalues, _Journal of Mathematics and Physics_, 38 (1959), pp. 104-111.

8. Egerváry, E., Matrixok kombinatorikus tulajdonsagairol, _Mat. Fiz. Lapok_, 38 (1931), pp. 16-28.

9. Kuhn, H. W., The Hungarian Method for the assignment problem, _Naval Res. Logistics Quarterly_, 2 (1955), pp. 83-97.

10. Lions, J., Matrix reduction using the Hungarian Method for the generation of school timetables, _Communications of the ACM_, 9 (1966), pp. 349-354.