THE BORDERED TRIANGULAR MATRIX AND MINIMUM ESSENTIAL SETS OF A DIGRAPH

by

by

L. K. Cheung and E. S. Kuh

Memorandum No. ERL-M375

5 February 1973

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

The Bordered Triangular Matrix and Minimum Essential Sets of a Digraph*

L.K. Cheung and E.S. Kuh

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory,
University of California, Berkeley, California 94720

Abstract

This paper deals with a partitioning strategy of sparse matrices. In particular, the problem of transforming a non-singular matrix by symmetric permutation to an optimal bordered Triangular Form is solved. It is shown that the problem is equivalent to the determination of a minimum essential set of a directed graph.

An efficient algorithm is given for finding minimum essential sets of a digraph. The method depends on, as a preliminary step, graph simplication using local information at a vertex. A circuit-generation technique based on vertex elimination is then introduced. The algorithm is illustrated with a complete example.

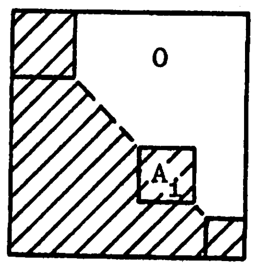
* Research sponsored by the National Science Foundation Grant GK10656X2 and Joint Services Electronics Program, Contract F44620-71-C-0087.

1. INTRODUCTION

It is essential that, in solving a system of simultaneous linear equations

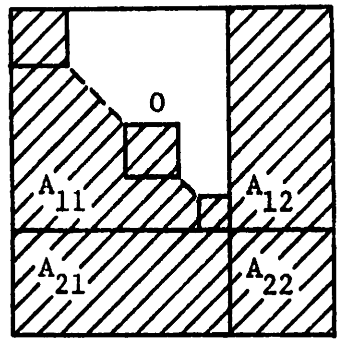
$$Ax = b \tag{1}$$

where A is a large n×n, nonsingular, sparse matrix, we take full advantage of the zero-nonzero structure of A in order to economize computer storage and time. This is especially important if (1) is to be solved many times with the sparsity structure of A remaining the same. Suppose A is reducible, that is, it can be transformed into the Block Triangular Form, by means of row and column permutations, as in (2),

PAQ = 

(2)

where P and Q are permutation matrices, we may use block Gaussian elimination on PAQ; and the original problem is reduced to that of solving a set of subproblems of the form $A_i \hat{x}_i = \hat{b}_i$. One of the advantages of doing this is that fill-ins are generated only in the shaded area. In addition, since the order of A_i is usually much smaller than that of A, so a good pivoting order within each block A_i is easier to find. In this paper, we are considering matrices which are irreducible, and our aim is to transform the given matrix, by row and column permutations, to a form which inherits the advantages of the Block Triangular Form. One obvious extension of the Block Triangular Form is the bordered Block Triangular Form in (3)

PAQ =  (3)

Notice that A_{11} is a square Block Triangular submatrix. It is clear that if we take pivots within the diagonal blocks, we have the same advantages enjoyed by the Block Triangular Form. Thus if the shaded areas are rather dense in nonzeros, the bordered Block Triangular Form obviously represents a kind of optimal partition.

The problem we are facing is how to define an optimal bordered Block Triangular Form. This is unlike the Block Triangular Form, which is canonically unique [1] if we specify that each diagonal block corresponds to an irreducible matrix. Here we have some difficulty in specifying a more meaningful criterion of defining the diagonal blocks. It is because of this difficulty that results in this area are rather scattered. Tewarson [2] has discussed this form and indicated that an iterative scheme had been proposed by Dickson [3]. Steward [4] studied a related problem of tearing and Nathan et al. [5] indirectly attempted to solve the problem.

In this paper, we shall confine ourself to a special case of the bordered Block Triangular Form, namely: the bordered Triangular Form, i.e. A_{11} in (3) is lower triangular. We shall further make two assumptions, namely: (i) we use symmetric permutations only, i.e. $Q = P^T$, and (ii) we assume that diagonal pivoting of A in any order is numerically stable. Our aim is to find a permutation matrix $\hat{P}$ such that the number of the bordered columns in A_{12} is a minimum. Thus, if we denote k(P) the number of columns in A_{12} corresponding to a permutation matrix P, we need to find $\hat{P}$ such that

$$k(\hat{P}) = \min_{P} k(P) = k_{min} \tag{4}$$

This formulation of the problem can be related to the problem of optimal tearing of a large system with the dual objective of minimizing the coupling of a torn system and simultaneously prescribing the form of the principal part of the system being triangular. Thus the theory and techniques to be described have applications in various large system problems.

In Section 2, we show that the problem formulated above can be stated precisely in terms of a graph associated with the given matrix A. In Section 3, we present an algorithm for finding $\hat{P}$. This amounts to the determination of a minimum essential set of a directed graph defined by A. An efficient new algorithm which combines new techniques and existing ones is given and illustrated with a complete example. Section 4 indicates some possible extensions and generalizations.

2. RELATION BETWEEN A MATRIX AND A DIGRAPH (DIRECTED GRAPH)

We now turn to the problem of finding a permutation matrix P which transforms A into an optimal bordered Triangular Form. We shall use a graph-theorectic approach. We begin by introducing the following definitions and remarks.

A <u>simple digraph</u>, G = (X,E), consists of a set of vertices X, and a set of directed edges $E = \{(x_i,x_j) | x_i,x_j \in X\}$. (x_i,x_j) is an edge directed from x_i to x_j. A (simple) directed path $\mu(x_i,x_j)$ of length ℓ is an ordered set of (distinct) vertices:

$$\mu(x_i,x_j) = \{p_1,p_2, \cdots, p_{\ell+1}\}, \text{ such that}$$

$$p_1 = x_i,\ p_{\ell+1} = x_j,\ (p_i,p_{i+1}) \in E$$

$$i = 1, 2, \ldots, \ell.$$

A (simple) circuit of length ℓ is a closed (simple) directed path of length ℓ with the initial and terminal vertices being identical. A digraph is said to be cyclic (acyclic) if it has (does not have) circuits. Let $G = (X,E)$ be a digraph and $Y \subset X$. The section graph $G(Y) \triangleq \{Y,E(Y)\}$, where $E(Y) = \{(x_i,x_j) \in E | x_i,x_j \in Y\}$. $S \subset X$ is an essential set of $G = (X,E)$ if $G(X-S)$ is acyclic. An essential set having the minimum number of vertices is called a minimum essential set, and the cardinality of a minimum essential set is called the index of the digraph.

For a matrix A with nonzero diagonal entries, we define an associated digraph $\mathcal{G}(A)$ as follows. $\mathcal{G}(A) = (X,E)$, with $|X| = n$, the order of A, and $(x_i,x_j) \in E$ iff (if and only if) $a_{ij} \neq 0$ for $i \neq j$; $i, j = 1, 2, \cdots$, n, where $A = [a_{ij}]$. Similarly, the adjacency matrix $B(G) = [b_{ij}]$ of a digraph $G = (X,E)$ is defined as an n×n matrix, where $n = |X|$, such that $b_{ij} \neq 0$ iff $(x_i,x_j) \in E$, $i \neq j$, and $b_{ii} = 1$, $i = 1, 2, \ldots, n$. It is obvious that $\mathcal{G}(A)$ and $\mathcal{G}(PAP^T)$, where P is a permutation matrix, are topologically identical. Likewise, A and $B(\mathcal{G}(A))$ are structurally the same to within symmetric row and column permutations.

Now deleting row i and column i from a matrix A with nonzero diagonal can be interpreted as the deletion of x_i and all edges incident with x_i from $\mathcal{G}(A)$. Since if the digraph $\mathcal{G}(A) \triangleq G = (X,E)$ is cyclic, by deleting an essential set S, G(X-S) becomes acyclic. Therefore if we delete rows and columns of matrix A corresponding to S, we have a submatrix whose associated digraph is acyclic. Using the fact [6] that a matrix is transformable, by row and column permutations, to a triangular form if and only if the associated digraph is acyclic, we conclude that k_{min} = index of $\mathcal{G}(A)$. Once a minimum essential set of $\mathcal{G}(A)$ is found, the corresponding rows and columns of A can be put in the bottom-most and right-most position to form the bordered Triangular Form. The appropriate permutation matrix is also determined immediately. Therefore the remaining problem is to determine a minimum essential set of a digraph. We illustrate the above idea with the following example.

Example 1

In Fig. 1(a), we have a matrix A of order seven. The associated digraph $\mathcal{G}(A) \triangleq G = (X,E)$ is shown in Fig. 1(b). From this digraph, we can determine a minimum essential set $S = \{x_1, x_5, x_7\}$. The section graph G(X-S) is shown in Fig. 1(c). G(X-S) is seen to be acyclic. By putting columns and rows {1,5,7} corresponding to S last, and with some appropriate permutations on the remaining rows and columns, we get PAP^T which is in bordered Triangular Form as shown in Fig. 1(d). The pertinent matrix P is shown in Fig. 1(e).

3. THE MINIMUM ESSENTIAL SET ALGORITHM

The problem of finding a minimum essential set of a digraph has been treated mainly in connection with signal flow graphs [5], [7]-[9]. The basic approach consists of two parts. The first part requires the generation of all circuits. The second part is to form a covering table [10] with columns and rows corresponding to all circuits and vertices pertinent to the circuits, respectively. The table is then simplified and reduced by some reduction rules similar to those used in minimization of switching functions. If the table cannot be reduced any further with these rules, column branching is used to obtain a minimum essential set [11]. Theoretically speaking, the problem can be considered solved. On the other hand, the amount of work involved in both parts can be excessive for certain graphs. Therefore much work needs to be done to increase the efficiency of the algorithm. The following are the major steps of our new algorithm.

MINIMUMS (Algorithm for Minimum Essential Set)

Step 1. Perform preliminary reduction on the digraph by means of a set of topological rules. If graph is completely reduced, go to Step 5.

Step 2. Generate all pertinent circuits of the reduced digraph.

Step 3. Construct a covering table and perform reduction on this table. If the table is completely reduced, go to Step 5.

Step 4. Use column branching to determine a minimum essential set.

Step 5. End. ¤

Remarks

In step 1, topological rules in [12], [13] and our new rules are used. These rules make use of "local information" of the digraph. In

Step 2, we introduce a circuit generation algorithm based on the method of Hohn, Seshu and Auffnkamp in [14]. Further topological rules are employed to keep the number of generated circuits small. Steps 3 and 4 are standard and we shall not persue in this paper. It will be clear after Section 3.2 that part of the "covering table reduction" in Step 3 is carried out in Step 2.

Let us discuss Steps 1 and 2 in detail.

3.1 Step 1: Preliminary simplification

To begin, let us define the local information at vertex x of a digraph $G = (X,E)$ as the complete topological knowledge of the section graph $G(\{x\} \cup \mathrm{Adj}(x))$, where $\mathrm{Adj}(x) \triangleq \{y \in X \mid (x,y) \text{ or } (y,x) \in E\}$. Next we define three types of local transformation at x for a digraph $G = (X,E)$ as:

T1 Deletion of vertex x:

Remove vertex x and its incident edges. The result is the section graph $G(X-\{x\})$.

T2 Elimination of vertex x:

Delete vertex x and add new edges to the section graph $G(X-\{x\})$ in the following way: we add (z,y) to $G(X-\{x\})$ iff (z,x) and $(x,y) \in E$. Note that if $y \equiv z$, (z,z) is a self-loop.

T3 Deletion of an edge incident with x:

Remove $(x,y) \in E$ or $(y,x) \in E$ from $G = (X,E)$ and form a new digraph $(X, E-\{(x,y) \text{ or } (y,x)\})$.

What Step 1 does is to make use of the local information at $x \in X$ and then perform alternatively various local transformations at x. In performing local transformations, we must keep track of the index of the reduced graph. If, after a local transformation, the reduced graph has the same index as the original graph, the transformation is called <u>index preserving</u>. In the following, the conditional transformations R2 to R5 are index preserving.

R1: Delete vertex x when x has a self-loop. The reduced graph has an index which is one less than the original.

R2: Eliminate vertex x when min(in-degree, out-degree of x) ≤ 1.

R3: Eliminate vertex x when $G(\{x\} \cup Adj(x))$ is a complete digraph, i.e., (p,q) and $(q,p) \in E$ for all $p, q \in \{x\} \cup Adj(x)$.

R4: Delete all edges incident at x except those forming doublets*, if, after removing those edges in the doublets, min(in-degree, out-degree of x) = 0.

R5: Delete edge $(y,x) \in E$ if $(y,z) \in E$ whenever $(x,z) \in E$. Likewise for $(x,y) \in E$.

<u>Remarks</u>

(i) R1, R2 and R4 were introduced by Guardabassi [12]. They depend only on the degree of x.

(ii) R3 and R5 are new, and they require the knowledge of the topology of the section graph G(Adj(x)). The proofs that they are index preserving are simple.

For R3, since $G^* \triangleq G(\{x\} \cup Adj(x))$ is a complete digraph, we must remove at least (k-1) vertices to break all circuits in G^*, where $k = |\{x\} \cup Adj(x)|$. Obviously, the set Adj(x) is the best choice of essen-

---

*A <u>doublet</u> is a circuit of length two.

tial vertices for G(X).

For R5, circuits through (y,x) and (x,z) are dominated[†] by circuits through (y,z). Other circuits through (y,x) are dominated by the doublets at x. Thus, no circuits through (y,x) need be considered in determining the index of the digraph. Hence, (y,x) can be deleted.

(iii) Each vertex deleted by R1 is a vertex in some minimum essential set [12].

(iv) R1-R5 are by no means exhaustive as far as local-information transformation is concerned. It is possible that more rules can be developed by exploiting the local information.

Summarizing, in Step 1, we test all vertices in the digraph and its transformed digraphs in order to perform local transformations whenever the conditions in the topological rules R1-R5 are satisfied. Afterwards, we will end up with a digraph to which the rules R1-R5 cannot be applied. At this stage, we go to Step 2. Let us illustrate Step 1 by an example.

<u>Example 2</u>

Consider the digraph G = (X,E) in Fig. 1(b). Rules R1, R2 and R4 fail to apply at any vertex. Using R5 at x_4, we can delete (x_3,x_4) because (x_3,x_1) and (x_3,x_5) are in E. Again, applying R5 at x_3, we delete (x_2,x_3). At this stage the transformed digraph is shown in Fig. 2(a). Now applying R2 at x_3 and x_4, we get Fig. 2(b). By R1, we delete x_1. The result is shown in Fig. 2(c). Using R2 at x_2 and <u>then</u> R2 at x_6, we get Fig. 2(d). By R1, we delete x_5 and x_7. Thus, G is completely re-

---

[†]Circuit η_1 is said to <u>dominate</u> circuit η_2 if $\eta_1 \subset \eta_2$. In other words, if a vertex breaks η_1, it also breaks η_2.

duced and we get a minimum essential set $S = \{x_1, x_5, x_7\}$.

3.2 Step 2: Generation of pertinent circuits

In this step, we generate all circuits of the reduced digraph that are pertinent to the determination of the index. The algorithm is basically a vertex elimination procedure [14]. Let us define the notion of a labelled digraph. Let $G = (X,E)$ be a digraph. The associated labelled digraph $\tilde{G} = (X,\tilde{E})$ is topologically identical to G except that each $(x_i,x_j) \in \tilde{E}$ carries a label or weight $\{x_i,x_j\}$ which represents a path of length one from x_i to x_j in $\tilde{G}$. The vertex elimination procedure is essentially the same as that discussed in the previous section with the following modification:

T4: Delete vertex x and add new edges to the section graph $\tilde{G}(X-\{x\})$ in the following way: we add (z,y) to $\tilde{G}(X-\{x\})$ iff (z,x) and $(x,y) \in \tilde{E}$. Then a label $\{z,\cdots,x,\cdots,y\}$ is assigned to this new edge, where $\{z,\cdots,x\}$, $\{x,\cdots,y\}$ are the labels of (z,x) and (x,y), respectively.

If $y = z$, (z,z) is a self-loop with a label defined in the same way. In other words, the new edge (z,y) in the elimination graph $\tilde{G}_x \triangleq (X-\{x\}, \tilde{E}_x = \tilde{E} \cup \{\text{all added edges}\} - \{\text{all deleted edges}\})$ represents a directed path in $\tilde{G}$ from z to y via x. Repeated use of T4 will eventually eliminate all vertices of $\tilde{G}$. It should be noted that in each elimination graph, there may be some parallel edges. For simplicity, we do not differentiate these parallel edges. In case of ambiguity, we append a subscript to the edge. To identify a particular edge, we use a bar over it. The following is a direct consequence of the procedure T4.

Proposition

Each self-loop in each elimination graph corresponds to a circuit in

$\tilde{G}$, and vice versa. ⌑

Since by the vertex elimination procedure, all circuits are generated. It is obvious that some of the circuits are dominated by others. To reduce the number of dominated circuits, we do the following. Let $\tilde{G}_* = (X_*, \tilde{E}_*)$ be some elimination digraph as obtained after eliminating $X - X_*$ from $\tilde{G} = (X, \tilde{E})$. Suppose we want to eliminate $x \in X_*$ next. Before doing this, we perform at x, transformation T3, namely, deletion of edge incident with x. We want to know under what conditions will T3 be index preserving. The following are some index preserving rules.

R6: For every $y \in Adj(x)$, delete all edges (y,x) and $(x,y) \in \tilde{E}_*$ whenever there is a simple doublet* $\{y,x\}$ in $\tilde{G}_*$. When this rule is applied, we record the simple doublet $\{y,x\}$ and store it as a column in the "covering table".

R7: For every $y \in Adj(x)$, delete all edges $(x,y) \in \tilde{E}_*$ except $(\overline{x,y})$, whenever there is an edge $(\overline{x,y}) \in \tilde{E}_*$ with a simple label.†

R8: For every $y \in Adj(x)$, delete all edges $(x,y) \in \tilde{E}_*$ whenever there is an edge $(\overline{x,y}) \in \tilde{E}_*$ whose label is a subset of those of the deleted edges (x,y). Similarly for all edges $(y,x) \in \tilde{E}_*$.

R9: For every $y \in Adj(x)$, delete all edges $(y,x) \in \tilde{E}_*$ whenever there is an edge $(\overline{x,y}) \in \tilde{E}_*$ with a simple label. When this rule is applied, we record all the doublets with each constituent edges $(\overline{x,y})$ and (y,x), and store them as columns in the "covering table".

Remarks

(i) By rules R6-R9, we perform T3 at a vertex x before eliminating

*A doublet in a labelled digraph is simple if the label of each constituent edge contains only two vertices.

†A label is simple if it contains only two vertices.

it by T4. In general, this checks the propagation of circuits.

(ii) R6-R9 should be applied in the order indicated. Otherwise, if we apply R9 first, say, we may record some circuits which are dominated by simple doublets.

(iii) It is easy to show that R6-R9 are index preserving.

For R6: It is clear that circuits through deleted edges are dominated by the simple doublet $\{y,x\}$.

For R7 and R8: Again, circuits through the deleted edges are dominated by circuits through $(\overline{x,y})$.

For R9: Circuits through each deleted edge (y,x) are dominated by circuits through $(\overline{x,y})$ and the deleted edge (y,x).

The complete algorithm for generating circuits pertinent to the determination of the index of a digraph $G = (X,E)$ is as follows.

CIRCUIT (Algorithm for generating pertinent circuits)

Step 1 Let $G = (X,E)$, $X = \{x_1,x_2,x_3,\cdots,x_n\}$

Define labelled digraphs:

$\tilde{G}_1 = (X_1,\tilde{E}_1) \triangleq (X,\tilde{E})$.

Step 2 Set $i = 1$.

Step 3 Consider $\tilde{G}_i$. Remove self-loops at $x_i \in X_i$. Record and store these loops as columns in the covering table. If $i = n$, go to

Step 7.

Step 4 Perform T3 at x_i using rules R6-R9. Record and store circuits as appropriate.

Step 5 Perform vertex elimination T4 at x_i, and form the elimination graph $\tilde{G}_{i+1}$.

Step 6 Set i = i+1, go to Step 3.

Step 7 End.

Let us illustrate this algorithm by an example.

Example 3

Consider the digraph in Fig. 1(b). The corresponding digraph $\tilde{G}_1 = (X_1, \tilde{E}_1)$ is shown in Fig. 3(a). All labels in $\tilde{G}_1$ are simple. For the sake of clarity, only some labels are shown. Consider x_1, it does not have any self-loops. Rule R6 applies, so we delete the two simple doublets $\{x_1, x_2\}$ and $\{x_1, x_3\}$ at x_1 and store them in the set $\mathcal{C}$ of circuits. On eliminating x_1 using T4, we get the elimination graph $\tilde{G}_2$ in Fig. 3(b). Note that in $\tilde{G}_2$, we introduce a new edge (x_4, x_7) with a label $\{x_4, x_1, x_7\}$. Labels of all other remaining edges are the same as in $\tilde{G}_1$. From $\tilde{G}_2$, we eliminate x_2 right away because rules R6-R9 do not apply at x_2. Similarly, we eliminate x_3 and form $\tilde{G}_4$ which is shown in Fig. 3(c). At x_4, R7 applies, so we delete (x_6, x_4) whose label is $\{x_6, x_2, x_3, x_4\}$. Eliminating x_4, we get $\tilde{G}_5$ in Fig. 3(d). At x_5, R6 applies, so we delete (x_5, x_7) and (x_7, x_5) and store the circuit $\{x_5, x_7\}$ in $\mathcal{C}$. R8 again applies at x_5, so we delete $(x_6, x_5)_b$ because its label $\{x_6, x_2, x_3, x_5\}$ is a superset of that of $\overline{(x_6, x_5)}_a$. Next R9 applies at x_5, so delete $\overline{(x_6, x_5)}_a$ and $(x_6, x_5)_c$ and store the circuits $\{x_5, x_6, x_2\}$ and $\{x_5, x_6, x_4\}$. After this sequence of transformation at x_5, $\tilde{G}_5$ becomes $\tilde{G}'_5$ in Fig. 3(e). On elimin-

x_5, x_6, x_7, we get two more circuits $\{x_2,x_6,x_7\}$ and $\{x_1,x_4,x_6,x_7\}$. Thus, altogether we have

$$\mathcal{C} = \left\{ \begin{matrix} \{x_1,x_2\}, \ \{x_1,x_3\}, \ \{x_5,x_7\}, \ \{x_2,x_5,x_6\} \\ \{x_4,x_5,x_6\}, \ \{x_2,x_6,x_7\}, \ \{x_1,x_4,x_6,x_7\} \end{matrix} \right\}$$

We complete this section by illustrating Steps 3 and 4 in MINIMUMS briefly with Example 3. We construct a covering table with columns η_j and rows x_i corresponding to circuits in $\mathcal{C}$ and vertices in X respectively, and we mark with a cross X in column η_j and row x_i if $x_i \in \eta_j$. This is shown in Fig 4. Using column branching, we will get a minimum essential set $\{x_1,x_5,x_7\}$ which is the same as obtained in Example 2.

4. CONCLUSION

The algorithm of finding a minimum essential set in the previous section could still be ineffective in dealing with a very large general graph. Like many other practical problems, [16], what is needed is an efficient algorithm which does not aim at the global minimum but rather gives a good feasible solution. In this connection, a useful concept to employ is the minimal essential set. A minimal essential set is defined as an essential set in which no proper subset is also essential. In a forthcoming paper, methods for generating minimal essential set will be given.

The results obtained for the bordered Triangular Form can be generalized and made applicable for the transformation of a matrix to the bordered Block Triangular Form. Using some criteria, we may identify the set of vertices corresponding to each diagonal block of A_{11} in eq. (3). By merging vertices in the same block to a single vertex and deleting all self-loops, we get a condensed graph [6]. The problem is then reduced to finding a minimum or minimal essential set of the condensed graph. Usually, the criteria of defining a block is problem-oriented. For example, if the entries of some rows and columns in A varies in the course of computation while others remain constant, it is obvious that we would put these rows and columns in the same block.

Another example is in the calculation of pole and zero of a transfer function of a linear time invariant network using the tableau approach [15]. Suppose the linear network has an (A,B,C,d) system representation, then the poles of the network is given by the zeros of det(sI-A) = det $\hat{U}$, where $\hat{U}$ has the form

$$\hat{U} = \left[\begin{array}{c|c} P & Q \\ \hline R & sI-T \end{array}\right] \tag{5}$$

$\hat{U}$ is obtained from a tableau whose rows corresponds to the Ohm's Laws, Kirchhoff's Laws and differential equations describing the network. It is shown in [15] that P can be put into a block triangular form. Each diagonal block corresponds to a subnetwork containing the same type of elements, for example, resistors. Thus, for this specific problem, we have a meaningful criterion of defining diagonal blocks.

In conclusion, in this paper we have related the problem of finding an optimal bordered Triangular Form by symmetric permutation of a given matrix to the problem of finding a minimum essential set of an associated digraph. We have also introduced an efficient algorithm in generating a minimum essential set of a digraph. A still open question is to generalize the problem to include nonsymmetric permutations in the transformation.

The question is then what is the graph-theoretic interpretation of k_{min} and how is it determined?

References

[1] Dulmage, A.L., Mendelsohn, N.S., "A Structure Theory of Bipartite Graphs of Finite Exterior Dimension," Trans. Roy. Soc. Canada, Section III, vol. 53 (1959), pp. 1-13.

[2] Tewarson, R. P., "Computations with Sparse Matrices," SIAM Rev. 12 (1970) pp. 527-543.

[3] Dickson, J.C., "Finding permutation operations to produce a large triangular submatrix," National Meeting, ORSA, Houston, Texas, 1965.

[4] Steward, D.V., "Tearing analysis of the structure of Disordering Sparse Matrices," Sparse Matrix, Yorktown Conference Proceedings (ed. Willoughby, R.A.) (Sept. 1968) pp. 65-74.

[5] Nathan, A., Even, R.K., "The inversion of sparse matrices by a strategy derived from their graphs," Comput. J. 10 (1967) pp. 190-194.

[6] Harary, F., Graph Theory, Addison-Wesley, Reading, Massachusetts, 1969.

[7] Divieti, L., "A method for the determination of the Paths and the index of a signal flow graph," Proc. 7th Int. Conv. on Automation and Instrumentation, Milan, (Nov. 1964) pp. 77-94.

[8] Lempel, A., Cederbaum, I., "Minimum Feedback arc and vertex sets of a directed graph," IEEE Trans. Circuit Theory vol. CT-13 (Dec. 1966) pp. 399-403.

[9] Riegle, D.E. & Lin, P.M., "Matrix signal flow graphs and an optimum topological method for evaluating their gains," IEEE Trans. CT-19 (Sept. 1972) pp. 427-435.

[10] Givone, D.D., Introduction to Switching Circuit Theory, McGraw Hill, N.Y., 1970.

[11] Diviete, L. and Grassalli, A., "On the determination of minimum feedback arc and vertex sets," IEEE Trans. Circuit Theory, vol. CT-15 (1968) pp. 86-88.

[12] Guardabassi, G., "A Note on Minimal Essential Sets," IEEE Trans. Circuit Theory (corresp.) vol. CT-18 (Sept. 1971), pp. 557-560.

[13] Diaz, M., Richard, J.P. and Courvoisier, M., "A Note on Minimal and Quasi-Minimal Essential Sets in Complex Directed Graphs," IEEE Trans. Circuit Theory (corresp.), vol. CT-19, (Sept. 1972), pp. 512-513.

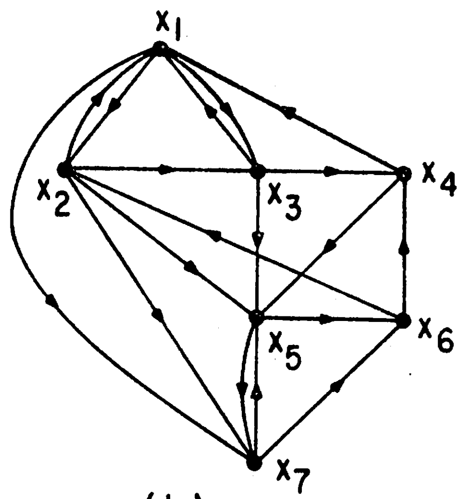
[14] Hohn, F.E., Seshu, S. and Auffnkamp, D.D., "The Theory of Nets," IRE Trans. on Electronic Computers, (Sept. 1957) pp. 154-161.

[15] Ohtsuki T. and Cheung L.K., "A matrix Decomposition-Reduction Procedure for the Pole-Zero Calculation of Transfer Functions," IEEE Trans. Circuit Theory, vol. CT-20, (May 1973).

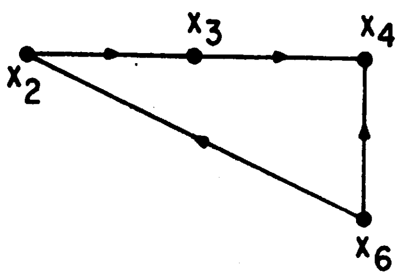
[16] Ogbuobiri, E.C., Tinney, W.F., and Walker, J.W., "Sparsity-directed decomposition for Gaussian elimination on Matrices," IEEE Trans. PAS-89 (1970) pp. 141-150.

	1	2	3	4	5	6	7
1	x	x	x				x
2	x	x	x		x		x
3	x		x	x	x		
4	x			x	x		
5					x	x	x
6		x		x		x	
7					x	x	x

(a)



(b)



(c)

	4	3	2	6	1	5	7
4	x				x	x	
3	x	x			x	x	
2		x	x		x	x	x
6	x		x	x			
1		x	x		x		x
5				x		x	x
7				x		x	x

(d)

			1			
		1				
	1					
					1	
1						
				1		
						1

(e)

Fig. 1. Illustrations for Example 1: (a) the given matrix, A, (b) the associated digraph $\mathcal{G}(A) \triangleq G = (X,E)$, (c) the section graph G(X-S), (d) the bordered Triangular Form, (e) the permutation matrix $\hat{P}$.

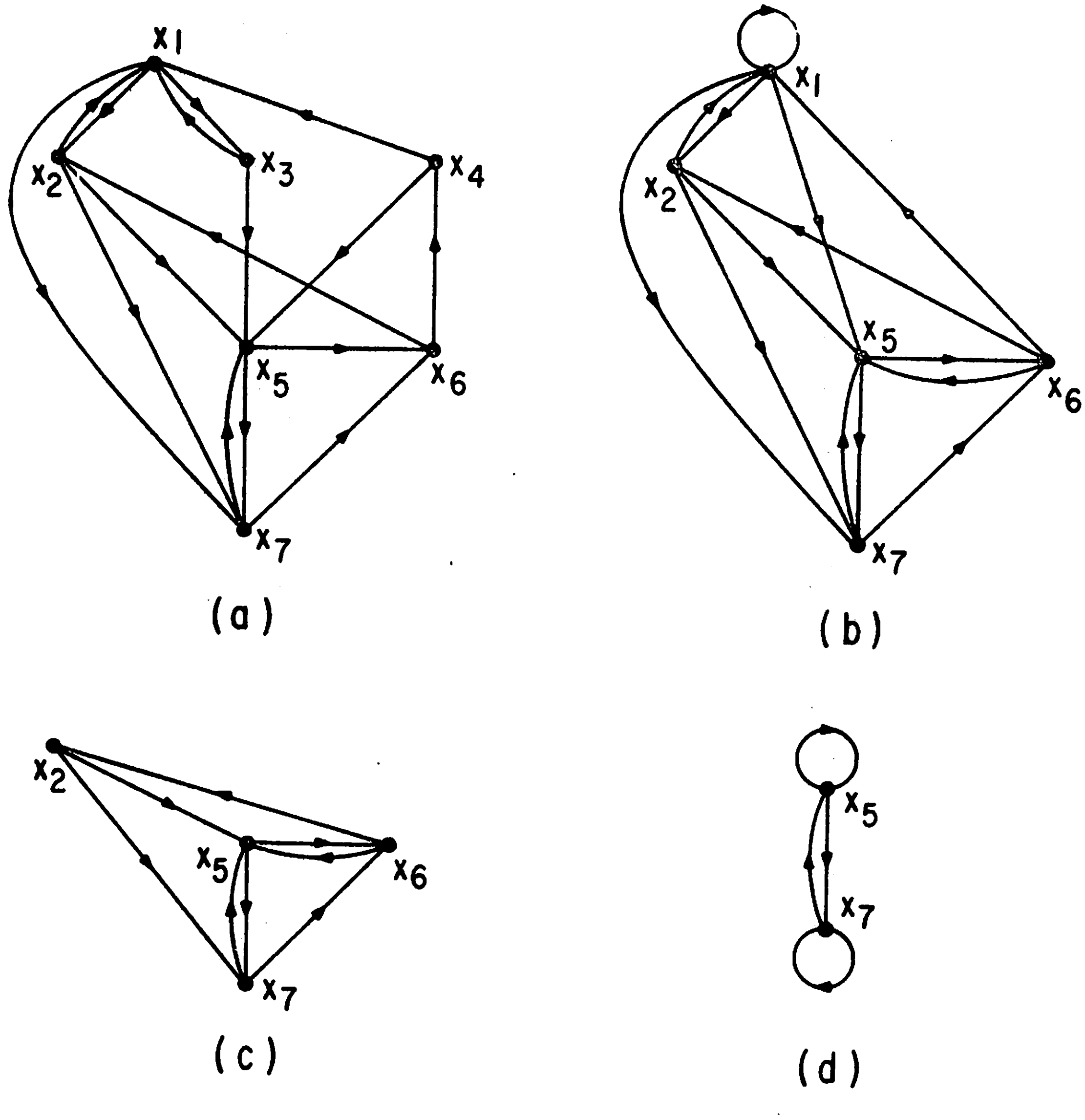
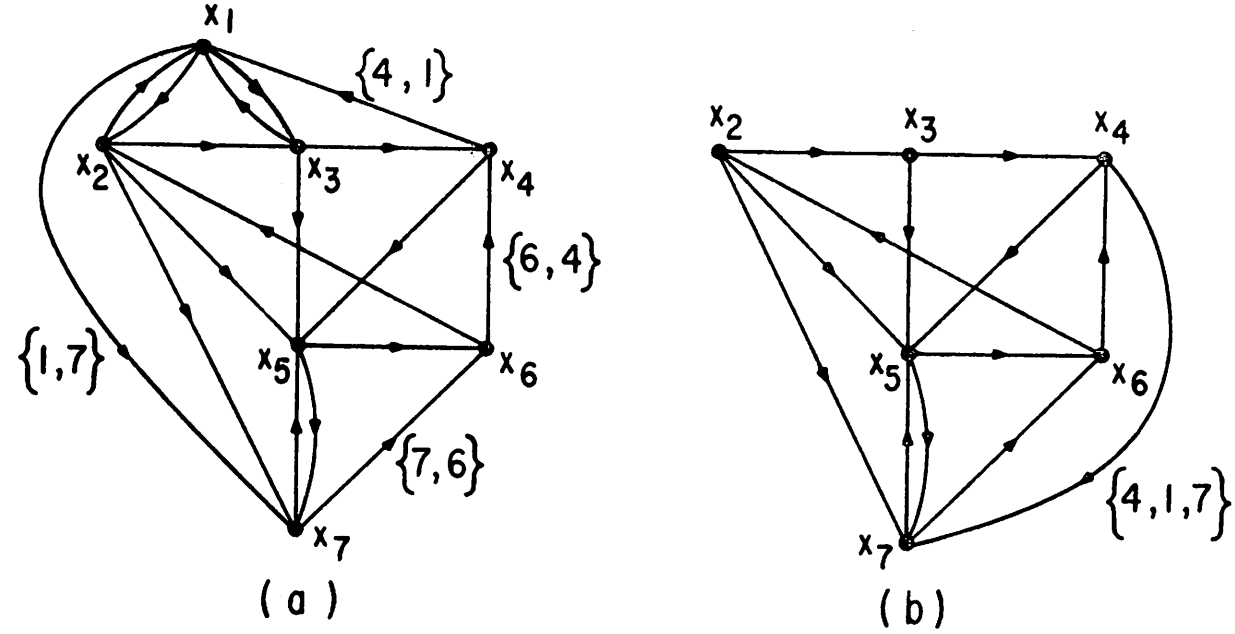
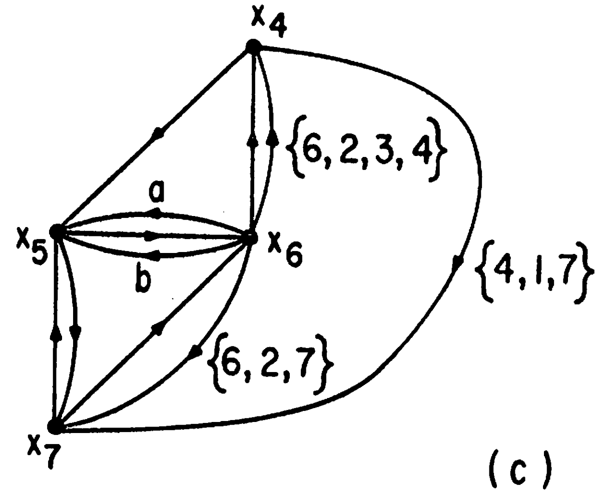
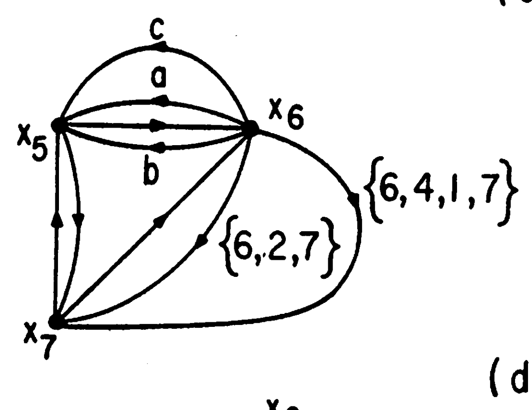


Fig. 2. Illustrations for Example 2: a succession of reduced digraphs.





edge	label
(x_5, x_6)	$\{5, 6\}$
$(x_6, x_5)_a$	$\{6, 2, 5\}$
$(x_6, x_5)_b$	$\{6, 2, 3, 5\}$



edge	label
$(x_6, x_5)_a$	$\{6, 2, 5\}$
$(x_6, x_5)_b$	$\{6, 2, 3, 5\}$
$(x_6, x_5)_c$	$\{6, 4, 5\}$

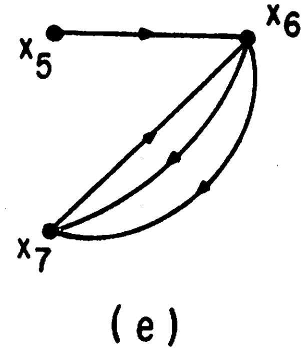


Fig. 3. Illustrations for Example 3: generation of circuits, (a) G_1, (b) $\tilde{G}_2$, (c) $\tilde{G}_4$, (d) $\tilde{G}_5$, (e) $\tilde{G}_5'$.

	η_1	η_2	η_3	η_4	η_5	η_6	η_7
x_1	X	X					X
x_2	X			X		X	
x_3		X					
x_4					X		X
x_5			X	X	X		
x_6				X	X	X	X
x_7			X			X	X

Fig. 4. A covering table for the example.