FEEDBACK VERTEX AND EDGE SETS OF A DIGRAPH
AND APPLICATIONS TO SPARSE MATRICES

by

L. K. Cheung and E. S. Kuh

Memorandum No. ERL-M419

February 11, 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

FEEDBACK VERTEX AND EDGE SETS OF A DIGRAPH AND APPLICATIONS TO SPARSE MATRICES

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 the problem of finding minimum feedback sets of a digraph. The problem is shown to be equivalent to transforming sparse matrices to certain optimum triangular forms, thus it is important in the analysis and solution of large systems by means of tearing.

It is further demonstrated that the minimum feedback vertex set problem and the minimum feedback edge set problem can be simply reduced to each other. Algorithms for finding minimum feedback sets of some important classes of digraph are proposed and their application in the branch-and-bound approach of solving the general problem for an arbitrary digraph is discussed. An iterative cardinality reduction algorithm is also presented for finding a local minimum feedback set of an arbitrary digraph.

Research sponsored by the National Science Foundation Grant GK10656x3 and the Joint Services Electronics Program Contract USAF-F44620-71-C-0087.

I. INTRODUCTION

Let G = (X, E) be a digraph [1] with a set of vertices X and a set of directed edges $E = \{(x_i, x_j) \mid x_i, x_j \in X\}$. A subset $S \subset X$ is a feedback vertex set*, denoted by FVS, of G if the subgraph defined on (X - S) contains no vircuits [1]. Similarly, $F \subset E$ is a feedback edge set, denoted by FES, of G if the subgraph (X, E - F) contains no circuits. A feedback set with a minimum cardinality is called a minimum feedback set.

Minimum feedback sets of a digraph have been treated in connection with signal flow graphs [2] - [4] and with computer program simplification [5], [6]. One way of finding a minimum feedback set is by solving an associated minimum cover problem [7], [8]. This involves the generation of elementary circuits in a digraph [2], [8], [9]. In [10], a new method is proposed in which only a minimal set of circuits is generated. Another approach to the minimum feedback set problem is to use topological transformations. Guardabassi [11] introduces some topological simplification rules and completes the solution with a branch-and-bound algorithm. Diaz, et al. [12], Cheung and Kuh [10] further extend the set of simplification rules. However, these rules fail to determine a minimum feedback set without resorting to some other algorithms.

In this paper, we shall present some results on the minimum feedback set problems and shall address ourself also to the problem of finding a good local minimum feedback set. However, before we proceed with these, we will point out in Section II that the problems of FVS and FES have their

*Some other names for feedback vertex set are essential vertex set and feedback node set. Terminology used here will be defined in the latter part of this section.

counter-parts in obtaining canonical forms of sparse matrices. More specifically, the FVS problem is equivalent to the problem of transforming a sparse matrix to an optimum bordered triangular form; while the FES problem is equivalent to that of an optimum pseudo triangular form. These forms are often desirable in analyzing and solving problems in large systems by means of the method of tearing [10, 13]. Section III shows that the FVS and FES problems are basically the same and can be reduced to each other in a simple manner. In Section IV, we study some classes of digraphs for which minimum feedback sets can be determined by simple topological transformations. Section V presents an iterative method of finding a local minimum feedback set. In the remaining part of this section, we shall introduce some relevant definitions.

A (simple) directed path $\mu(x_i, x_j)$ of length ℓ, $\ell > 0$, is an ordered set of (distinct) vertices:

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

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

$$k = 1, 2, \ldots, \ell,$$

$$x_i \neq x_j.$$

A (simple) circuit η of length ℓ, $\ell > 0$, is an ordered set of (distinct) vertices:

$$\eta = \{p_1, p_2, \ldots, p_{\ell+1}\}, \quad \text{such that}$$

$$p_1 = p_{\ell+1},\ (p_k, p_{k+1}) \in E, \quad k = 1, 2, \ldots, \ell.$$

A circuit of length 1, 2 is called a self-loop, doublet respectively.
A digraph is said to be cyclic (acyclic) if it has (does not have) circuits.
The section graph defined on a subset $Y \subset X$ is $G(Y) \triangleq \{Y, E(Y)\}$, where $E(Y) = \{(x_i, x_j) \in E \mid x_i, x_j \in Y\}$. The cardinality of a minimum feedback

set is called the index of the digraph. For simplicity, feedback vertex / edge set of a digrah G is denoted by FVS(G)/FES(G).

II. EQUIVALENT PROBLEMS IN SPARSE MATRICES

Consider an n X n non-singular sparse matrix $A^{\dagger}$ and its transformed matrix PAP^T where P is a permutation matrix. The transformed matrix is said to be in a Bordered Triangular Form (BTF) if the nonzero entries are restricted to the shaded area of Fig. 1a. If, for a particular permutation matrix $\hat{P}$, the number of the bordered columns, k, is a minimum, we say that $\hat{P}A\hat{P}^T$ is an optimum BTF, or

$$k(\hat{P}) = \min_P k(P) = k_{\min}.$$

In Fig. 1b, the transformed matrix PAP^T is called a Pseudo Triangular Form (PTF). The removal of nonzero elements designated by crosses in the upper triangular part of Fig. 1(b) makes the matrix triangular. If, for a particular transformation $\bar{P}$, the number of nonzero elements in the upper triangular part, q, is a minimum, we say that $\bar{P}A\bar{P}^T$ is an optimum PTF, or

$$q(\bar{P}) = \min_P q(P) = q_{\min}.$$

The advantages of these two optimum forms are quite obvious in solving large sparse matrix equations. Basically, the optimum forms yield optimal strategy in tearing, and, thereby one is capable of minimizing computation and storage.

In [10], we demonstrated that the problem of finding an optimum BTF is equivalent to that of finding a minimum feedback vertex set of a digraph. An associated digraph of an n X n matrix $A = [a_{ij}]$ with nonzero diagonal entries is denoted by $\mathcal{G}(A)$. $\mathcal{G}(A) = (X, E)$, with cardinality of the vertex

†We assume that A has a nonzero diagonal and that symmetric permutation in any order is numerically stable.

set, $|X| = n$. The edge, $(x_i, x_j) \in E$ iff $a_{ij} \neq 0$ for $i \neq j$; $i, j = 1, 2, \ldots, n$. Obviously, $\mathcal{G}(A)$ and $\mathcal{G}(PAP^T)$ are isomorphic. Thus deleting the element a_{ij} from the matrix can be interpreted as deleting the edge (x_i, x_j) from the associated digraph. Using the fact that a matrix is transformable by symmetric permutation to a triangular form if and only if the associated digraph is acyclic, we conclude that q_{min} is the edge index of the digraph. Thus to find an optimum PTF is equivalent to that of finding a minimum FES of the associated digraph $\mathcal{G}(A)$.

The algorithms and techniques for finding minimum feedback sets of a digraph are therefore useful in solving many large system problems. It should further be pointed out that the triangular matrix form is a special case of the block triangular form. Thus the bordered and pseudo triangular forms are, respectively, special cases of the bordered block triangular form and the pseudo block triangular form. These general forms occur frequently in physical problems such as large power systems and integrated circuits. Often we may consider the diagonal blocks in the general forms as modules or clusters in the physical problems, thus reducing the general problem to that of the bordered triangular form and pseudo triangular form which we treat in this paper.

III. CONVERSION OF FEEDBACK VERTEX AND FEEDBACK EDGE SETS

In this section, we show that the FVS and the FES problems are basically the same and can be reduced to each other in a simple way.

First we consider the conversion from FVS to FES. In this case, a minimum FVS of $G = (X, E)$ is identified as a minimum FES of $\bar{G}$ which is constructed as follows. $\bar{G} = (\bar{X}, \bar{E})$ has two vertices x' and x'' for each vertex x in G. There is an edge from x' to x'' in $\bar{G}$. All edges directed to x in G becomes edges directed to x' in $\bar{G}$. Similarly, for edges directed from x in G, we have a corresponding set of edges directed from x" in $\bar{G}$. Thus $|\bar{X}| = 2|X|$ and $|\bar{E}| = |E| + |X|$. The construction of $\bar{G}$ from G is illustrated in Fig. 2(a). We observe that there is a 1-1 correspondence between a circuit in G and a circuit in $\bar{G}$. We now show how to identify a FVS of G from a FES of $\bar{G}$, and vice versa, by means of the following theorem.

<u>THEOREM 1</u>. There is a correspondence between a minimum FVS of G and a minimum FES, with the same cardinality, of $\bar{G}$.

<u>Proof</u>. (i) Suppose $FVS_1(G) \triangleq \{x_1, x_2, \ldots, x_k\}$ is a minimum FVS of G, then $FES_1(\bar{G}) \triangleq \{(x_1', x_1''), \ldots, (x_k', x_k'')\}$ is a FES of $\bar{G}$. Suppose $FES_1(G)$ is not minimum, let $FES_2(\bar{G})$ be a minimum FES of $\bar{G}$. $FES_2(\bar{G})$ may contain two types of edges (a): (x_i', x_i'') and (b): (x_i'', x_j'). For each edge of the second type, we replace it by an edge (x_j', x_j''). It is obvious that the resulting set of edges, $FES_3(\bar{G})$, is also a FES of $\bar{G}$. Now, corresponding to each edge in $FES_3(\bar{G})$, we can identify a vertex in G. Let $FVS_2(G)$ be the set of vertices in G constructed in this way. Obviously, $FVS_2(G)$ is a FVS of G. Since $|FVS_2(G)| = |FES_3(\bar{G})| = |FES_2(\bar{G})| < |FES_1(\bar{G})| = |FVS_1(G)|$, so $FVS_1(G)$ is not a minimum FVS of G.

(ii) For a given minimum FES of $\bar{G}$, we can construct a $FES_1(\bar{G})$ having the same cardinality and containing only edges of the type (x_i', x_i''). Then as in case (i), we get a $FVS_1(G)$ from $FES_1(\bar{G})$. Parallel to case (i), we can show that $FVS_1(G)$ is a minimum FVS of G. This concludes the proof.

Next we show the conversion of FES to FVS. In this case, a FES of G is identified as a FVS of a new graph G*. G* for this purpose is the line graph of G [1]. A line graph can be constructed from a graph G in the following way. For each edge e_i of G, we define a vertex x_i^* in G*.

(x_i^*, x_j^*) is an edge of G* iff in G the terminal vertex of e_i coincides with the initial vertex of e_j. Fig. 2(b) illustrates this point. Again, we observe that a circuit in G, defined in terms of a set of edges, is 1-1 corresponding to a circuit in G*, defined in terms of a set of vertices. As a consequence of the construction, we have the following theorem.

<u>THEOREM 2</u>. There is a 1-1 correspondence between a (minimum) FES of G and a (minimum) FVS of its line graph G*, in particular, both sets have the same cardinality.

<u>Proof</u>. Let FES(G) = $\{e_1, e_2, \ldots, e_k\}$ be a FES of G. Obviously FVS(G*) $\triangleq$ $\{x_1, x_2, \ldots, x_k\}$ is a FVS of G*. The converse is also obvious.

<u>REMARKS</u>. (i) The above two theorems show the duality between the FVS and FES problems. So knowing an algorithm for one problem, we can solve the other by applying it to the derived digraph. The price we pay for this unified approach is that, in general, the derived digraph is more involved than the original graph.

(ii) Note that in THEOREM 1, the correspondence is not 1-1. Also if the FES of $\bar{G}$ is not minimum, we cannot construct a corresponding FVS in G, such that the the two feedback sets have the same cardinality. To see this point, consider the case when FES($\bar{G}$) contains two edges: (x_i'', x_j') and (x_k'', x_j'). (Note that this can happen only when FES($\bar{G}$) is not minimum.) To identify the corresponding FVS in G, we have to replace the above two edges by, say (x_j', x_j''). This implies that the FVS in G has a smaller cardinality.

(iii) 1-1 correspondence holds in THEOREM 2, even if the sets are not minimum sets.

(iv) In [10, 11], a set of topological transformation rules has been introduced for the FVS problem. Since the FVS and FES problems can be reduced to each other, it is obvious that there exist similar reduction rules in the FES problem [14].

(v) Since the FVS and FES problems are reducible to each other, we shall pursue only on the FVS problem from here on.

IV. MINIMUM FEEDBACK SETS OF SOME CLASSES OF DIGRAPHS

As pointed out in the introduction, existing topological transformation rules are insufficient for finding FES/FVS of an arbitrary digraph. In this section, we shall show that for some classes of digraphs, simple reductions rules can be used to find minimum FVS/FES. The importance of these results is that they enable us to find efficiently the lower bounds of the index of an arbitrary digraph under study. In addition, they can be used to iteratively reduce the cardinality of an initial estimate of a feedback set. First, let us define two basic topological transformation [11].

(1) Deletion of a vertex $x \in X$ from a digraph $G = (X, E)$:

Remove x and form the section graph $G(X - \{x\})$ of G.

(2) Elimination of a vertex $x \in X$:

Delete x according to (1) and add a set of edges $F \triangleq \{(p, q) | (p, x), (x, q) \in E, \forall p, q \in X\}$ to $G(X - \{x\})$, i.e., we form $\bar{G} \triangleq (X - \{x\}, (E - \mathrm{Inc}(x)) \cup F)$, where $\mathrm{Inc}(x)$ is the set of edges incident with x.

The first class of digraphs to be considered has the following properties:

$\mathcal{G}_1$: $G = (X, E) \in \mathcal{G}_1$ if

There exists a tree T and a co-tree E_1 such that $E = T \cup E_1$, $T \cap E_1 = \phi$ and for all $(u,v) \in E_1$, $u \neq v$, there exists a directed path from v to u in T.

Let $\mathcal{P}_1$ be an algorithm operating on $G \in \mathcal{G}_1$ defined as follows.

STEP 1 Delete all vertices with self-loops, if the resulting section graph is (ϕ, ϕ), go to STEP 3

STEP 2 Eliminate a vertex whose min(in-, out-degree) ≤ 1 and go to STEP 1

STEP 3 Set S = set of all vertices deleted in STEP 1. End.

Example. Consider the digraph shown in Fig. 3(a) where the solid lines represent tree branches. Obviously, $G \in \mathcal{G}_1$. We will demonstrate that by applying $\mathcal{P}_1$, we are able to reduce G to (ϕ, ϕ), and thus obtain a minimum FVS of G.

(a) Apply STEP 2 at vertex a. We obtain the graph of Fig. 3(b).

(b) Apply STEP 1 and delete vertex b which has a self loop. We obtain the graph of Fig. 3(c).

(c) Apply STEP 2 at vertex c. We obtain the graph of Fig. 3(d).

(d) Repeat STEP 2 at vertex f. We obtain the graph of Fig. 3(e).

(e) Apply STEP 1 and delete vertex d which has a self-loop. We obtain the graph of Fig. 3(f).

(f) Apply STEP 2 at vertex e. We obtain the graph of Fig. 3(g), which is nothing but a vertex.

(g) Repeat STEP 2. We have totally reduced the graph.

(h) According to STEP 3, a minimum FVS is {b, d}.

THEOREM 3. Let $G \in \mathcal{G}_1$. $\mathcal{P}_1$ will reduce G to (ϕ, ϕ). Furthermore S is a minimum FVS of G.

Proof. It is obvious that all vertices with self-loops are part of a minimum FVS, so remove all these vertices. First, we show that there always exists a vertex to which $\mathcal{P}_1$ can be applied. Since $G \in \mathcal{G}_1$, there exists a tree T satisfying the conditions of $\mathcal{G}_1$. By assumption, for each end vertex (a vertex in T with only one edge incident with it) x of the tree T, if $(u, x) \in T$, then all other edges incident at x must go out of x, and vice versa if $(x, u) \in T$. This means that min(in-, out-degree of x) = 1. Suppose we eliminate such a vertex (STEP 2) and obtain a reduced digraph with $T_1 \subset T$ as the tree, then vertex u can be in any one of the following states: (i) it becomes an end vertex with respect to T_1 without self-loop; (ii) it becomes a non-end vertex without a self-loop; (iii) it has a self-loop. If (i) or (ii) happens, the reduced digraph has more than two vertices, the tree T_1 has more than two end vertices. Hence STEP 2 can be applied on the reduced digraph. Now consider case (iii). On deleting x (STEP 1), we delete some edges and links of T_1. Let the reduced tree be $T_2 \subset T_1$. The resultant digraph is either decomposable to a set of strongly connected components each having the same property as G, or, a non-end vertex of T_1 becomes an end vertex of T_2. This end vertex again has min(in-, our-degree) = 1. Thus in all cases, the reduced digraph has the same property as G. Hence deletion and elimination can be used to reduce G to a null graph (ϕ, ϕ). The set of vertices deleted (STEP 1) forms a minimum FVS of G because STEPs 1 and 2 are index preserving [11]. Next, suppose we eliminate an arbitrary vertex x whose min(in-, out-degree) ≤ 1 from G, we want to show that the transformed graph is also a $\mathcal{G}_1$ graph. Since $G \in \mathcal{G}_1$, there exists a tree T having the property of $\mathcal{G}_1$. Eliminate x, form the transformed graph and construct a new tree as follows. The new tree contains all original tree edges except those removed in the elimination process, plus the new edges (y,z) where (y,x) and $(x,z) \in T$, $\forall y, z \in X$. It is obvious that this new tree has the desired $\mathcal{G}_1$ property. This completes the proof.

We now consider a generalization of the above to an arbitrary digraph. Let us partition the edges E into 4 disjoint sets: T, E_1, E_2, E_3 such that T = tree constructed using the depth-first search method with α: $\{1, 2, \ldots, n\} \leftrightarrow X$ being the ordering on X generated in the course of the depth-first search [15]; E_1 = set of edges each of which connects a vertex to its ancestor, where x is an ancestor of y (which is a <u>descendant</u> of x) if there exists a directed path from x to y in the tree T; E_2 = set of edges each of which connects vertices in different subtrees, where two subgraphs of a tree are called subtrees if there exists no directed path between any vertex in one subgraph and any vertex in the other subgraph; E_3 = set of edges each of which connects a vertex to its descendent. In Fig. 4, an example of this is shown. With this partitioning, we have the following obvious result.

<u>THEOREM 4</u>. Let $G = (X, E)$ and $G_1 \triangleq (X, T \cup E_1)$. Then (i) $\mathcal{P}_1$ can be applied to G_1 to find a minimum FVS of G_1, and (ii) $G_{23} \triangleq (X, T \cup E_2 \cup E_3)$ is acyclic.

<u>Proof</u>. (i) is a special case of Theorem 3 and (ii) is a consequence of the depth-first search.

Theorem 3 shows that for graphs in which a particular tree can be constructed, we can determine a minimum FVS simply by using two topological transformations. For an arbitrary graph, we may not be able to partition the set of edges into disjoint subsets having the properties defined in Theorem 3. In this case, one possible way of finding a local minimum FVS is to extract a subgraph G_1 from G such that this subgraph satisfies the conditions in Theorem 4. A minimum FVS is then determined for this subgraph. On deleting this FVS from G, we again extract a subgraph satisfying the conditions in Theorem 4. Repeating this process, we get a local minimum FVS of G. The results in the above two theorems can also be incorporated into a branch-and-bound method [11], [16] of finding minimum FVS. In this method, we pick an arbitrary circuit η (preferably one with a shortest length) and assume that a particular vertex $x_1 \in \eta$ is in a minimum FVS. We delete this vertex from G and obtain a section graph $G(X - \{x_1\})$. Similarly, we pick another vertex from the circuit, delete it from G and obtain another section graph. In this way we obtain k section graphs, where k is the number of vertices in the circuit η. We illustrate this operation by Fig. 5. This is the branching process. On repeating the process on each and subsequent section graphs, we will end up with null graphs which are the end-vertices of the branch-and-bound tree. Obviously, it is not efficient to construct the whole tree. The whole idea of the branch-and-bound method is where to branch. One way [16] is to branch at a vertex having a minimum lower bound of the index of G. This requires a quick estimate of the index of the section graphs. This can be done simply using the result of Theorem 4, namely, for the section graph $\bar{G}$ under study, we extract $\bar{G}_1$ which satisfies the conditions of a $\mathcal{G}_1$ graph. Since $\bar{G}_1 \in \mathcal{G}_1$, we obtain index of $\bar{G}_1$ immediately. Since $\bar{G}_1$ is a subgraph of G, index of $\bar{G}_1$ is a lower bound for the index of G. In conclusion, results in Theorems 3 and 4 can be used to find efficiently lower bounds in the branch-and-bound method of finding the index of a digraph.

The second class of digraphs $\mathcal{G}_2$ we are going to consider has the following property.

$\mathcal{G}_2$: G = (X, E) $\in \mathcal{G}_2$ if index (G) < 2.

For this class, we define the following operation $\mathcal{P}_2$.

STEP 1 Set G $\triangleq$ (X, E) = the given graph

STEP 2 Delete all vertices with self-loops.

STEP 3 In G, if there exists x $\in$ X such that min(in-degree, out-degree of x) $\leq$ 1, go to STEP 3, else go to STEP 4.

STEP 4 Eliminate x from G. Set G = the transformed graph. Go to STEP 2.

STEP 5 In G, <u>if</u> there exists a vertex x $\in$ X such that on removing edges that form doublets at x, min(in-, out-degree of x) = 0, then delete all edges incident at x, except those forming doublets at x. Set G = the transformed graph, go to STEP 2 <u>else</u> go to STEP 6.

STEP 6 Set S = all vertices deleted in Step 2. S is a minimum FVS of the original given graph.

END.

The following Theorem shows that $\mathcal{P}_2$ determines indices of $\mathcal{G}_2$ graphs.

<u>THEOREM 5</u>. Let G = (X, E) $\in \mathcal{G}_2$. Then index (G) = 0, 1 iff S generated by $\mathcal{P}_2$ has 0, 1 vertices respectively.

<u>Proof</u>. (i) Suppose index (G) = 0, then G is acyclic and $\mathcal{P}_2$ can always be applied at the "source" vertex (vertices with in-degree = 0) of G and its transformed digraph. Hence by repeated application of STEP 5 of $\mathcal{P}_2$, we can delete all edges of G. Hence S is empty. Conversely, if S is empty, then index (G) is 0 because STEPs 4 and 5 are index preserving.
(ii) Suppose index (G) = 1. It is obvious that the theorem holds for cases with $|X|$ = 1, 2, 3. Suppose it is true for $|X| \leq$ n. Consider G

with $|X| = n+1$. We want to show that $\mathcal{P}_2$ can be applied to at least one vertex of G. Since index (G) = 1, there exists a vertex s such that $G^* = G(X - \{s\})$ is acyclic. Partially order vertices in G* so that $x > y$ iff $(x, y) \in E$. In G*, let us call vertices with zero in-degree (out-degree) the source (sink) vertices. If a source vertex has a doublet with s (or has an edge going to s), we apply STEP 5 of $\mathcal{P}_2$. If the source vertex has an edge coming from s, we can apply STEP 4 to eliminate the vertex. In any case, we can eliminate at least one vertex of G. On doing so, we get a reduced digraph having at most n vertices and with the same index as G. Hence sufficiency is proved. Necessity is obvious because the single vertex (with a self-loop) in the reduced digraph is a minimum FVS with cardinality equal to 1. Finally, we note that steps 3-5 when applied to an arbitrary vertex of a $\mathcal{G}_2$ graph produces again a $\mathcal{G}_2$ graph. This is because these steps are index preserving. This completes the proof.

Thus, for graphs with index less than 2, we can determine their indices using $\mathcal{P}_2$. Though $\mathcal{G}_2$ is a limited class of digraphs, we can make use of its property in the determination of a local minimum FVS of an arbitrary graph as will be shown in the next section.

V. LOCAL MINIMUM FEEDBACK VERTEX SET

In this section, we propose an iterative reduction algorithm for finding a local minimum FVS of an arbitrary digraph. The algorithm is as follows. Let G = (X, E) be the graph under study.

LFVS (Algorithm for the generation of a local minimum FVS)

STEP 1 Initialization. Find a minimal FVS of G

STEP 2 Reduction. Reduce the minimal FVS found in STEP 1 to a local minimum FVS.

END.

The algorithm begins by generating a reasonably small initial estimation of FVS of G. For the purpose, we use a minimal FVS of G as our starting point. We define a minimal FVS of a graph G as a FVS such that no proper subset of it is also a FVS of G. In STEP 2, we reduce this minimal set to a local minimum set by means of some elementary augmentations. We shall elaborate on these steps. First consider the following algorithm for generating a minimal FVS of $G = (X, E)$.

MINIMALS (Algorithm for finding a minimal FVS of $G = (X, E)$)

STEP 1 Set $G_1 \triangleq (X_1, E_1) = G \triangleq (X, E)$, $n = |X|$.
$i = 1$, $S = \emptyset$. S is a minimal FVS of G.

STEP 2 In G_i, if there exists $x_i \in X_i$ with a self-loop, go to STEP 3, else, go to STEP 4.

STEP 3 Delete x_i from $G_i \triangleq (X_i, E_i)$ and form $G_{i+1} \triangleq (X_{i+1}, E_{i+1})$ where $X_{i+1} = X_i - \{x_i\}$, $E_{i+1} = E_i(X_{i+1})$.
Set $S = S \cup \{x_i\}$. Go to STEP 5.

STEP 4 Eliminate any x_i from G_i and form G_{i+1}.

STEP 5 Set $i = i+1$, if $i = n+1$, Stop, else, go to STEP 2.

We now show that MINIMALS does generate a minimal FVS.

THEOREM 6. Let MINIMALS operate on $G = (X, E)$ and produce a feedback vertex set S. Then S is minimal.

Proof. Let $S = \{x_{i_1}, \ldots, x_{i_s}\}$ and suppose MINIMALS deletes S from G in the order $x_{i_1}, \ldots, x_{i_s}$. MINIMALS deletes x_{i_1} because it has a self-loop in either G or the transformed graph $G^*_{i_1} = G^*(X - \{x_1, \ldots, x_{i_1-1}\})$, which is obtained from G as a result of eliminating $x_1, \ldots, x_{i_1-1}$. In the latter case, the self-loop in $G^*_{i_1}$ is in fact a circuit in G passing through only the eliminated vertices. Similarly, a self-loop at x_{i_2} is either a self-loop in G or a self-loop in $G^*_{i_2}$, which is obtained from $G^*_{i_1+1}$ as a result of eliminating vertices. Note that $G^*_{i_1+1}$ is obtained from $G^*_{i_1}$ by deleting x_{i_1}. Thus each vertex in S has a circuit which does not pass through any other vertices in S. Also, S is a FVS of G because any circuit in G must pass through some vertices of S, as a consequence of the vertex elimination process. Hence, S is minimal.

Once a minimal FVS is found, we then proceed to reduce this to a local minimum FVS. The following is the reduction algorithm.

<u>REDUCTION (Algorithm for reducing a FVS to a local minimum FVS)</u>

STEP 1 Let $S = \{s_1, s_2, \ldots, s_s\}$ be a FVS of G

STEP 2 For all $s_i, s_j \in S$, do: Determine index of

$G^* = G((X - S) \cup \{s_i, s_j\})$,

if index $(G^*) \leq 1$, set

$S = (S - \{s_i, s_j\}) \cup \{s^*\}$ (where s^* is tne new FVS of G^*,

note that $\{s^*\}$ may be empty) and go to STEP 2,

<u>else</u> continue the do loop.

END.

In the above algorithm, we need a subroutine to detemine the index of a graph whose index is known to be at most 2. $\mathcal{P}_2$ of Section IV is precisely a tool for this purpose. It can be seen that algorithm REDUCTION reduces an arbitrary FVS of a digraph to a FVS such that every section graph of the form $G^* = G(X - \text{FVS}) \cup \{s_i, s_j \in \text{FVS}\}$ has index equal to 1. Such a FVS is our local minimum FVS of G. Incidentally, a local minimum FVS is also a minimal FVS, as shown in the following theorem.

<u>THEOREM 7</u>. Let G = (X, E) be a digraph and S be a FVS of G. If S^* is the result of applying algorithm REDUCTION on S, then S^* is a minimal FVS of G.

Proof. Suppose not, let $\underline{S} < S^*$ be such that $\underline{S}$ is also a FVS of G. Take a vertex $s \in (S^* - \underline{S})$ and another vertex $y(\neq x) \in S^*$. Since $G((X - S^*) \cup \{s, y\})$ has index ≤ 1, REDUCTION would not terminate at S^*. Hence S^* is minimal.

Let us illustrate REDUCTION by an example.

Example. Consider the digraph shown in Fig. 6(a). Suppose we have obtained a minimal FVS $S^* = \{x_2, x_3, x_5, x_7\}$. Let us apply REDUCTION on S^*. Arbitrarily, we pick a pair x_2, x_3 from S^*. Next we form the section graph $G(\{x_1, x_2, x_3, x_4, x_6\})$ as shown in Fig. 6(b). Using $\mathcal{P}_2$ this section graph can be reduced to the graph shown in Fig. 6(c). So $\{s^* = x_1\}$ is a FVS of $G(\{x_1, x_2, x_3, x_4, x_6\})$. So we can replace x_2, x_3 by x_1 in the FVS S^*. The updated FVS becomes $s^* = \{x_1. x_5, x_7\}$. Now REDUCTION terminates at this FVS. So S^* is the desired local minimum FVS of G.

VI. CONCLUSION

In this paper, we study the problem of finding minimum feedback sets of a digraph. We have demonstrated the relation between the problem and that of transforming a matrix to suitable form for the purpose of tearing. We have also shown that the feedback vertex and feedback edge set problems can be reduced to each other in a simple manner. For graphs whose edge set can be partitioned into subsets having special properties and for graphs whose index is at most 2, we have proposed simple algorithms for finding indices. These algorithms are shown to be useful in the determination of the index of an arbitrary graph by the branch-and-bound method. Finally, an iterative cardinality reduction algorithm is proposed to find a local minimum FVS of a digraph.

The result obtained in this paper provide a stepping-stone towards the topological determination of minimum feedback sets of an arbitrary digraph. Further work could be done in extending the class of digraphs for which existing topological transformations can be used to find their indices.

REFERENCES

[1] Harary, F., Graph Theory, Addison-Wesley, Reading, Mass., 1969.

[2] Lempel, A. and Cederbaum, J., "Minimum Feedback Arc and Vertex Sets of a Directed Graph," IEEE Trans., CT-13, pp. 399 - 403, December 1966.

[3] Younger, D. H., "Minimum Feedback Arc Sets for a Direct Graph," IEEE Trans., CT-10, pp. 238 - 245, 1963.

[4] Riegle, D. E. and Lin, P. M., "Matrix Signal Flow Graphs and an Optimal Topological Method for Evaluating Their Gains," IEEE Trans., CT-19, pp. 427 - 435, September 1972.

[5] Allen, F. E., Program Optimization, Research Report RC-1959, IBM Watson Research Center, Yorktown Heights, N.Y., April 1966.

[6] Martin, D. and Estrin, D., "Models of Computational Systems - Cyclic to Acyclic Graph Transformations," IEEE Trans., EC-16, pp. 70 - 79, February 1967.

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

[8] Divieti, L. and Grasselli, A., "On the Determination of Minimum Feedback Arc and Vertex Sets," IEEE Trans., CT-15, pp. 86 - 88, March 1968.

[9] 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, pp. 77 - 94, November 1964.

[10] Cheung, L. K. and Kuh, E. S., "The Bordered Triangular Matrix and Minimum Essential Sets of a Digraph," to be published IEEE Trans. Circuits and Systems. Also, ERL Report ERL-M375, University of California, Berkeley, February 5, 1973.

[11] Guandabassi, G., "A Note on Minimal Essential Sets," IEEE Trans., CT-18, pp. 557 - 560, September 1971.

[12] Diaz, M., Richard, J. P. and Courvoisier, M., "A Note on Minimal Essential Sets in Complex Directed Graphs," IEEE Trans., CT-19, pp. 512 - 513, September 1972.

[13] Steward, D. V., "Partitioning and Tearing Systems of Equations," SIAM J. Numer. Anal., Vol. 2, pp. 345 - 365, 1965.

[14] Cheung, L. K., Sparse Matrix Techniques: Theory and Application, Ph.D. Thesis, University of California, Berkeley, California, August 1973.

[15] Tarjan, R., "Depth-First Search and Linear Graph Algorithms," SIAM J. Comput., Vol. 1, pp. 146 - 160, 1972.

[16] Lawler, E. L. and Wood, D. E., "Branch-and-Bound Methods: A Survey," Operations Research, Vol. 14, #4, pp. 699 - 719, July-August 1966.

Figure Captions

Fig. 1. The bordered triangular form and the pseudo triangular form.

Fig. 2. Illustration for the conversion between FVS and FES.

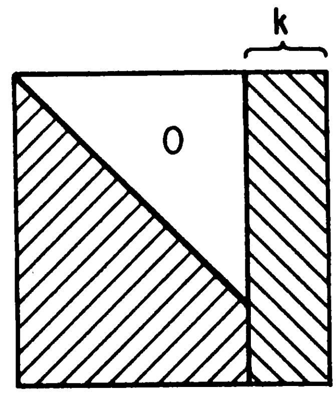
Fig. 3. Example of P_1 algorithm.

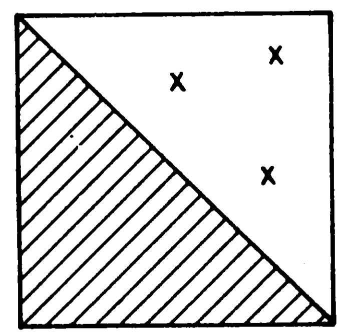
Fig. 4. Illustration of edge set partition according to Theorem 4.

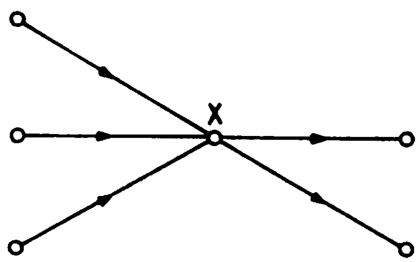
$$\begin{cases} E_1 = \{(4,1),\ (5,1)\} \\ E_2 = \{(5,4)\} \\ E_3 = \{(2,4)\} \end{cases}$$

Fig. 5. Illustration for the branch-and-bound method.

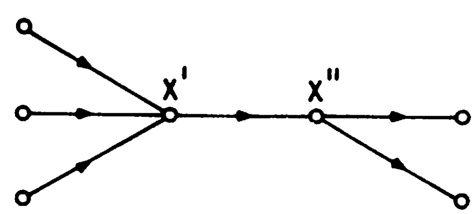
Fig. 6. Example for Reduction.

$PAP^T =$ 

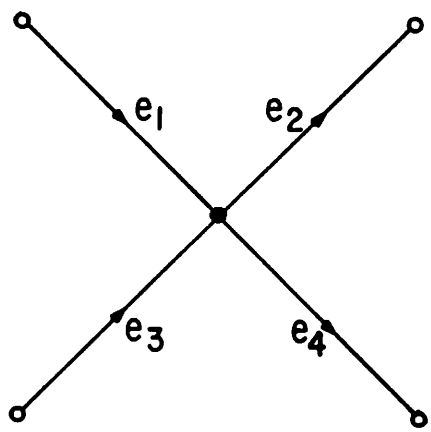
$PAP^T =$ 



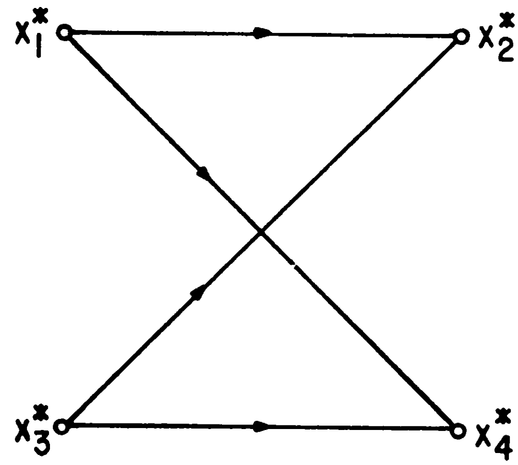
G



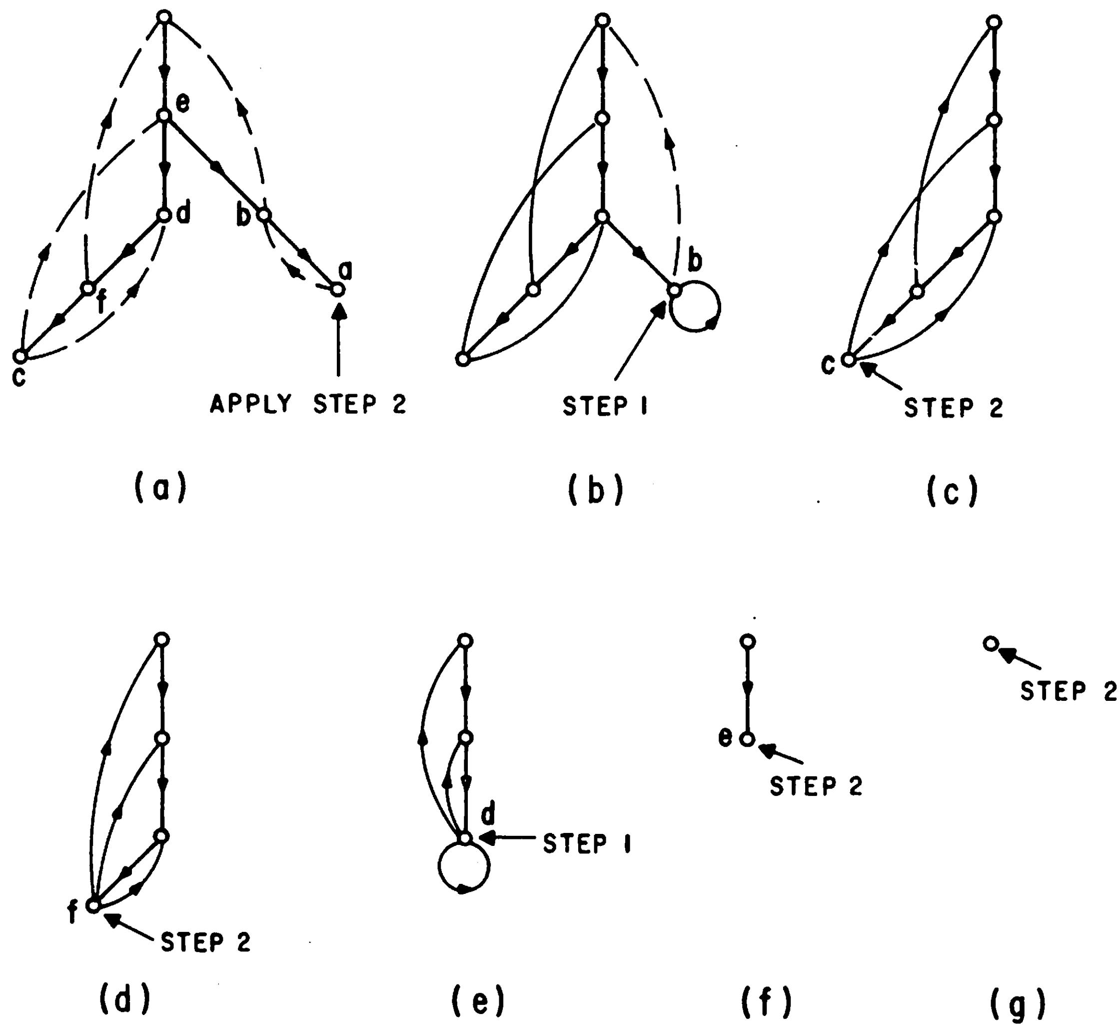
$\bar{G}$

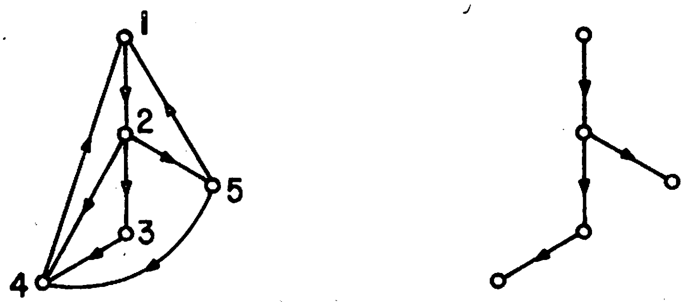


G

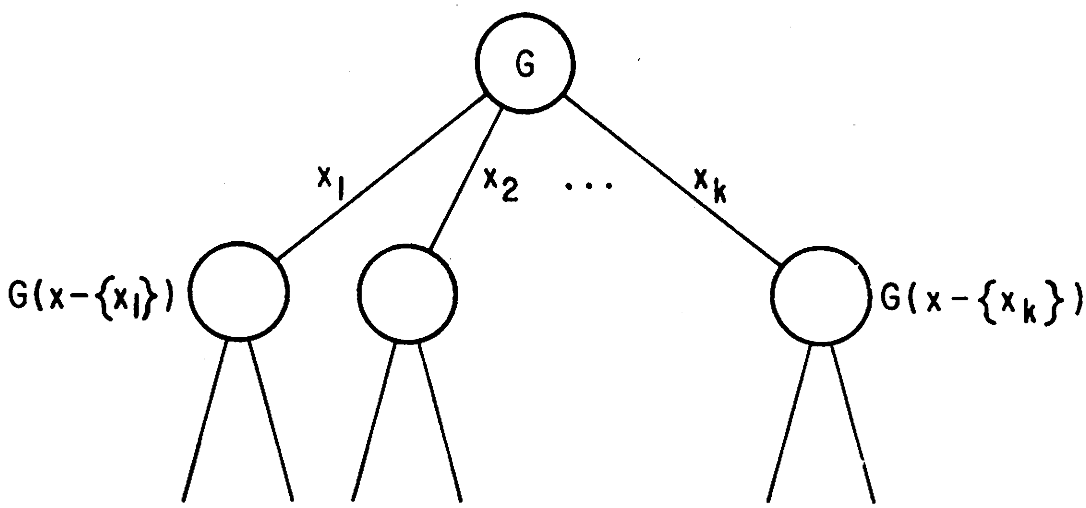


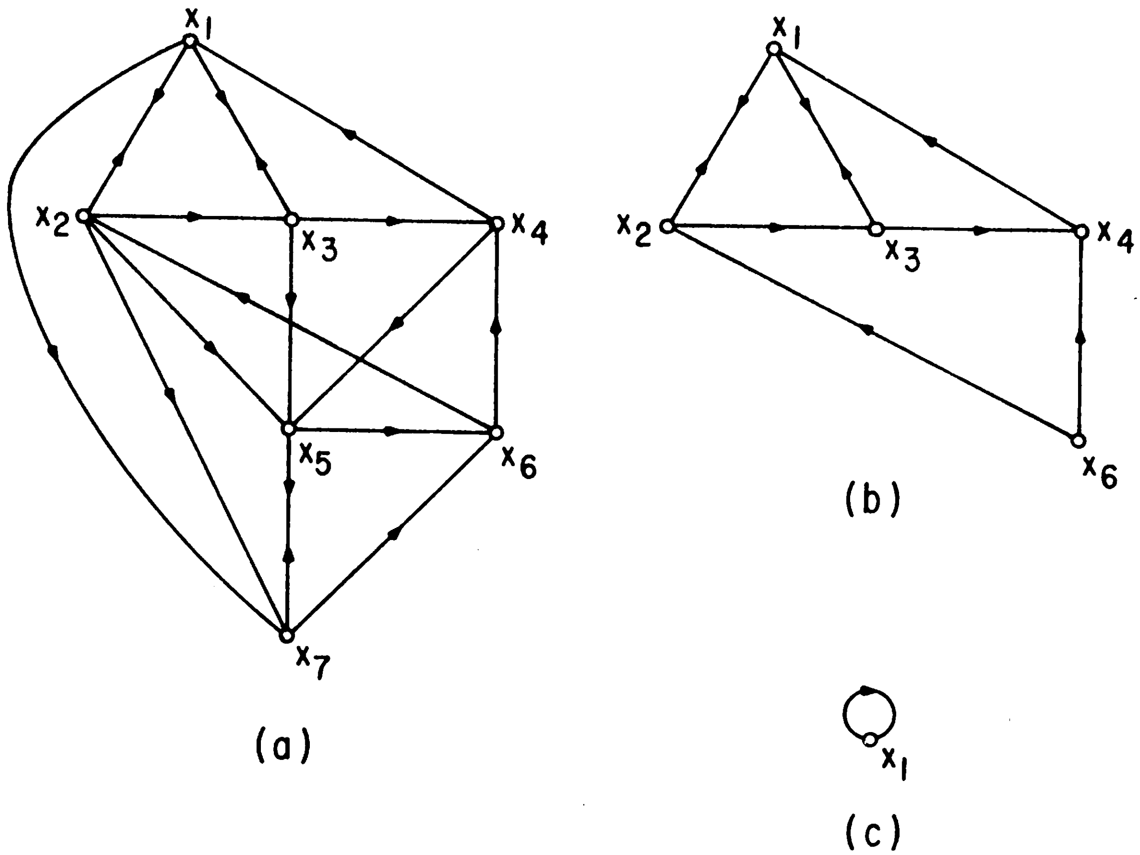
G^*


e
d
b
a
f
c
APPLY STEP 2
(a)
b
STEP I
(b)
c
STEP 2
(c)
f
STEP 2
(d)
d
STEP I
(e)
e
STEP 2
(f)
STEP 2
(g)



(a) G (b) T


G
x1
x2
...
xk
G(x-{x1})
G(x-{xk})


x1
x2
x3
x4
x5
x6
x7
(a)
x1
x2
x3
x4
x6
(b)
x1
(c)