数理解析研究所講究録
第427巻 1981年 112-130

112

A HEURISTIC ALGORITHM FOR THE MINIMUM FEEDBACK ARC SET PROBLEM

Hiromu ARIYOSHI† and Youichi HIGASHIYAMA††

† Department of Electrical Engineering
Faculty of Engineering, Ehime University

††Computer Center, Ehime University
Matsuyama 790, JAPAN

Abstruct: In this paper, we propose an algorithm for the identification of near optimal feedback arc sets of a directed graph. Several authors have proposed heuristic algorithms but no algorithm guaranteed a solution in running time proportional to a constant power of size of a graph. To simplifying the design of algorithms we have used a depth-first search algorithm as the basic search. Exploiting properties of depth-first search, we have constructed an algorithm with the time complexity of $O(n^2m^4)$ and the space complexity of $O(n \cdot m)$, where n and m are the number of vertices and arcs of a graph, respectively. It is also shown that the algorithm has been successfully applied to test graphs, so called directed star polygons.

1. Introduction

One of the interesting problem about a digraph is that of finding a minimum set of arcs whose removal leaves the remaining graph free of directed circuits(cycles). This problem was originally suggested by Runnyon in connection with the analysis of sequential switching circuits with feedback paths[1]. Recently, much effort has been devoted to solving systems of linear equations where the matrix of coefficients is sparse. Finding

minimum feedback vertex and arc sets of a digraph associated with the matrix has been shown quite useful in providing tearing methods for the solution of such large systems[2∿5].

An earliest graph-theoretic approach for the minimum feedback arc set problem is that of Younger who has established a relationship between the problem and a sequential ordering of the vertices, and has proposed a branch and bound algorithm[6]. The best known and frequently used algorithms are due to Lempel and Cederbaum[7], and Divieti and Graselli[8]. Their techniques are basically composed of three steps; the first step is the generation of all cycles, the second step is to form a covering table on which simplification techniques are applied, and the last step is to use reduction rules and column branching technique to obtain the solution. To reduce the amount of computational work, Guardabassi has proposed a completely topological branch and bound algorithm using topological reduction rules[9]. Smith and Walford have described an algorithm based on a 2-subgraph partition implied by an arbitrary set of vertices[10].

In practice, an exact optimal solution may not always be required. Suboptimal solutions for the problem may suffice[5]. The alogorithms mentioned so far yield suboptimal solutions[2∿5]. However these methods, involving a systematic but exhaustive search, seem to be quite inefficient in terms of execution time. An algorithm guaranteeing a suboptimal solution in running time proportional to a constant power of the size of digraph is desirable, but no algorithm has been discovered.

This paper describes an approximate algorithm for the mini-

mum feedback arc set problem using entirely different approach. To design an efficient algorithm, we must avoid the use of exhaustive searches. The approach presented in this paper is mainly based on a familiar depth-first search due to Tarjan[11] so that the estimation of run-time is possible.

In section 2, we describe a depth-first search modified with respect to the selection of an arc to traverse during a DFS. In sections 3 and 4, we present algorithms for finding suboptimal feedback arc sets. Section 5 devotes the description of directed star polygons for which the algorithm is run to test the efficiency. Experimental results are demonstrated in Section 6.

2. Modified Depth-first Search

The depth-first search(DFS) on a directed graph $G=(V,E)$ explores the graph as follows. When we are visiting a vertex $v \in V$, always choose an arc (v,w), oriented away from v to w. If the vertex w has been previously visited(this arc is refered as a link), we return to v and choose another arc. If the vertex w has not been previously visited, we visit it and apply the process recursively w, this chosen arc contributes a spanning tree on the resulting graph $\vec{G}$. If all the outgoing arcs from v have been examined, we go back along the arc (u,v) that led to v, This step is called backtracking, and continue exploring the arcs incident on u.

During a DFS, we assign a vertex, say v, two serial numbers $M(v)$ and $N(v)$, so that $M(v)=i$ if v was the i-th vertex to be visited and $N(v)=j$ if v was the j-th vertex to be backtracked

during the traversal. The links of $\vec{G}$ are classified into three types, called backward links(b-links for short), forward links (f-links), and cross links(c-links) which, say (v,w)'s, are identified from the relations $N(v) < N(w)$, $M(v) < M(w)$, and $N(v) > N(w)$ and $M(v) > M(w)$, respectively. We denote the sets of tree-arcs, b-links, f-links, and c-links by Ω_t, Ω_b, Ω_f, and Ω_c, respectively. Let $F(e)$ be the fundamental cutset with respect to $e \in \Omega_t$, and let $F_b(e)$ and $F_{cf}(e)$ be proper subsets of $F(e)$ consisting of b-links, and c- and f-links, respectively. $F^+_{cf}(e)$, a subset of $F_{cf}(e)$, denotes the set of arcs whose directions are coincide with that of e(see Fig.2(a)).

As is wellknown, DFS is extremely useful in simplifying many graph-theoretic algorithms. Advantages exploiting a DFS in connection with the design of heuristic algorithms for the problem are

(1) A set Ω_b is a minimal feedback arc set(FAS),

(2) The graph under test is partitioned into maximal strongly connected components, and then one can consider only one component at a time,

(3) A depth-first search is a natural approach to use in generating fundamental cutsets, and

(4) It is possible to improve the size of a particular minimal feedback arc set Ω_b by use of selection-rules on vertices during DFS and by use of interchange-rules between $F_b(e)$ and $F^+_{cf}(e) \cup \{e\}$ mentioned later.

In what follows we explain selection-rules and illustrate an example to show the effectiveness of a modified depth-first

search(MDFS). Fig. 1 shows an intermediate state during a DFS on a graph in which V_0 is a set of vertices that have not yet been visited and V_1 is a set of vertices that have been visited. Let $G[V_k]$ be a section graph defined by $V_k \subset V$.

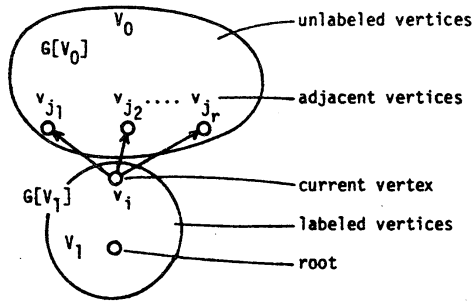

Fig.1. An intermediate state during DFS.

When we are visiting $v_i \in V_1$, we may select any of the adjacent vertices $v_{j1}, v_{j2}, \ldots, v_{jr} \in V_0$. In this process, if we select a vertex having less incomming arcs on $G[V_0]$, the resulting graph $\vec{G}$ may have Ω_b of smaller size than that obtained by an ordinary DFS. Let $d_0^+(j)$ and $d_0^-(j)$ be the out-degree and the in-degree of v_j on $G[V_0]$, respectively.

Definition 1. Selection-rules: A vertex v_j to be visited during a DFS is selected under the following rules:

(1) The vertex v_j has the minimum in-degree $d_0^-(j)$ among the adjacent vertices v_{jk}'s such that $(v_i, v_{jk}) \in E$, and

(2) If there are two or more candidates, v_j has the maximum out-degree $d_0^+(j)$.

Fig.2(a) illustrates G obtained by an ordinary DFS in which

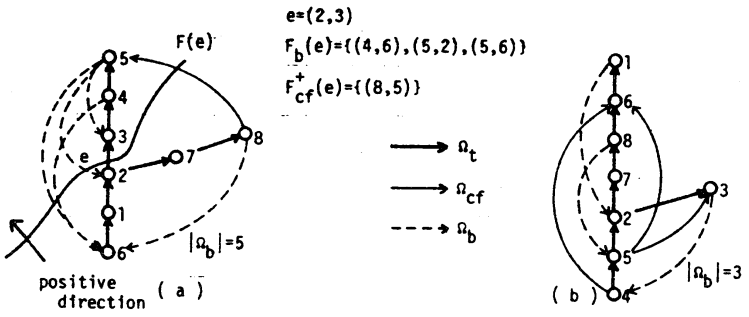

Fig.2. Graphs obtained by DFS and MDFS.

the size of Ω_b, the number of dotted arcs, is found to be five. The vertices v_1,v_4,v_7, and v_8 have the minimum in-degree on G[V], among which v_4 and v_8 have the maximum out-degree. Then we can select either vertex, say v_4, as the root. The adjacent vertices of v_4 are found to be v_5 and v_6 among which v_5 has the minimum in-degree on G[V-{v_4}]. Consequently, we can select v_5 as the second vertex to be visited. Applying the selection-rules to G of Fig.2(a), we get another graph G as shown in Fig.2(b). It should be noted that the resulting graph has Ω_b of smaller size than that of G.

Fig.3 shows a description of the MDFS algorithm in which $E_b(\cdot)$ appeared on lines 7,12, and 15 will be stated later.

```
procedure MDFS(graph G=(V,E))
 begin
  integer array N(|V|),M(|V|)
  arc set Ω_T,Ω_B,Ω_CF,E_B
  integer value C1,C2
1 procedure BACKTRACK(vertex set V_0,vertex v_i,
                                    vertex v_k)
    begin
2     if there exists a vertex v_i∈V_0 for which
        the selecting rule of definition 1 holds
        then begin
3         M(v_j):=C1:=C1+1 ;
4         V_0:=V_0-{v_j} ;
5         Ω_T:=Ω_T∪{<v_i,v_j>} ;
6         BACKTRACK(V_0,v_j,v_i) ;
7         E_B(v_i):=E_B(v_i)∪E_B(v_j) ;
8         N(v_j):=C2:=C2+1
        end ;
      else begin
9         for each vertex v_j∉V_0,<v_i,v_j>∈E do
            begin
10            if N(v_j)=0 then begin
11                Ω_B:=Ω_B∪{<v_i,v_j>} ;
12                E_B(v_i):=E_B(v_i)∪{<v_i,v_j>}
              end ;
13            else if <v_i,v_j>∉Ω_T then begin
14               Ω_CF:=Ω_CF∪{<v_i,v_j>} ;
15               E_B(v_i):=E_B(v_i)∪E_B(v_j)
              end
            end
      end
    end BACKTRACK ;
16 V_0:=V ;
17 while there exists a vertex v_i∈V_0 for which the
    selecting rule of definition 1 holds do begin
18    M(v_i):=C1:=C1+1 ;
19    V_0:=V_0-{v_i} ;
20    BACKTRACK(V_0,v_i,Dummy)
    end
 end MDFS ;
```

Fig.3. Procedure MDFS.

Lemma 1. Procedure MDFS requires $O(n+m+m^2)$ operations and $O(n \cdot m)$ memory spaces.

Proof: As compared with a DFS, entirely different parts are selecting operations appeared on lines 2 and 17, and computations of $E_b(\cdot)$ appeared on lines 7,12, and 15.

The first part will requires $O(n+m)$ operations since when we are visiting v_i as shown in Fig.1, we need to $d^+(i)$ comparisons in addition to the searching of the adjacency list. For the second part, it is clear that we need to prepare $n \cdot |\Omega_b|$ memory spaces for storing $E_b(\cdot)$ and that the union operations are executed at most $d^+(i)$ times for each v_i. Since the total number of union operations during a MDFS is equated as m, this part requires $O(m^2)$ operations. Q.E.D.

3. Minimaization Technique of Ω_b

Given a graph $\vec{G}$, if there exists $e \in \Omega_t$ such that $|F_b(e)| > |F^+_{cf}(e)|+1$, Ω_b is not an optimal one because we can find fewer feedback arcs which form $(\Omega_b - F_b(e)) \cup F^+_{cf}(e) \cup \{e\}$. In this section we consider an improvement in terms of fundamental cutsets.

Before introduce effective cf-links, we illustrate the following example. Fig.4(a) shows a subgraph of $\vec{G}$ obtained by a DFS in which the labeled number corresponds to DFS number $M(\cdot)$.

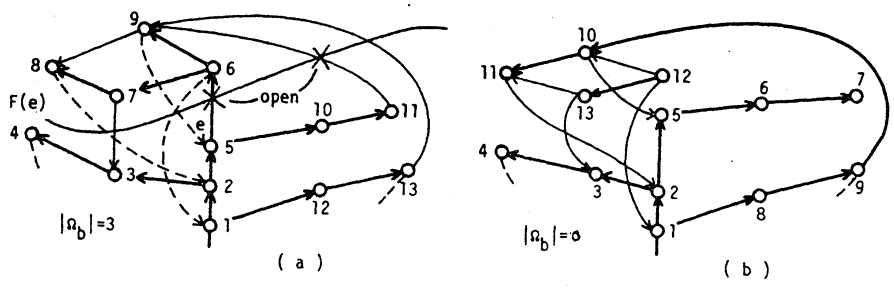

Fig.4. Introductory example for an effective cf-link.

It is easily seen that $|\Omega_b|=|F^+_{cf}(e)|+1=3$. Now deleting $e=(5,6) \varepsilon \Omega_t$ and $(11,9) \varepsilon F^+_{cf}(e)$, and then applying a DFS we may get an acyclic graph shown in Fig.4(b). This implies the set of arcs (5,6) and (11,9) is optimal one than that of Ω_b of Fig.4(a).

This observation leads the following definition.

Definition 2. Effective cf-link: For an arc $e_i \varepsilon F^+_{cf}(e)$, if there exists a cycle containing e_i and exactly one b-link which belongs to $F_b(e)$, e_i is called an effective cf-link with respect to e, and the set of such links is denoted by $F_{cf,eff}(e)$.

Definition 3. The first traversed b-link: Let P_v be the directed path from a specified vertex v of $\vec{G}$. The b-link of P_v whose initial vertex has the minimum distance from v, if exists, is called the first traversed b-link with respect to P_v. The set of such links with respect to all P_v's is denoted by $E_b(v)$.

The set $E_b(\cdot)$ is easily identified during MDFS as shown in Fig.3. The conditions that an arc $e_i \varepsilon F^+_{cf}(e)$ is to be the effective cf-link with respect to e are as follows.

(1) There is a b-link which belongs to $F_b(e) \cap E_b$(the terminal vetex of e_i), and

(2) The initial vertex e_i is a decendant of the terminal vertex of the b-link.

In order to optimize Ω_b, we need to compute k-value, written k(e), for every arc $e \varepsilon \Omega_t$ of $\vec{G}$. We define k-value as

$$k(e)=|F_b(e)|-|F_{cf,eff}(e)|-1 \qquad (1)$$

Definition 4. Reference tree-arc: For a given $\vec{G}$, let E_1 and E_2 be $E_1=\{e \varepsilon \Omega_t | k(e) > 0\}$ and $E_2=\{e \varepsilon \Omega_t | k(e) \leq 0, F_b(e) \neq \phi\}$, respectively. An arc e_r whose k-value is maximum, if $E_1 \neq \phi$, and minimum

,if $E_1=\phi$ and $E_2\neq\phi$, is called the reference tree-arc with respect to an optimization.

Now we can show in Fig.5 how to optimize Ω_b by interchanging $F_b(e)$ with $F_{cf,eff}(e)\cup\{e\}$. When the reference tree-arc is found in $\vec{G}'$, either $F_b(e_r)$ or $F_{cf,eff}(e_r)\cup\{e_r\}$ is refered as a candidate set of feedback arcs, and then these arcs are deleted from $\vec{G}'$. This process is repeated for the resulting graph and terminates when Ω_b of the graph is empty.

```
procedure LEASTFAS(graph G=(V,E))
 begin
   arc set MFAS,Ω_B
   integer array k(|V|)
1   procedure REFTREE(graph G'=(V,E'))
      begin
2       MDFS(G') ;
3       if Ω_B≠φ  then
          compute k(e) for each eεΩ_T
          and identify RT-edge e_r ;
4       if k(e_r)>0  then
          MFAS:=MFAS U F_CF,eff(e_r) U {e_r} ;
5       else MFAS:=MFAS U F_B(e_r) ;
6       E':=E'-MFAS ;
7       REFTREE(G')
      end REFTREE ;
8   MFAS:=φ ;
9   REFTREE(G)
 end LEASTFAS ;
```

Fig.5. Procedure LEASTFAS.

```
procedure APFAS#1(graph G=(V,E))
 begin
   integer array N(|V|),M(|V|)
   arc set MFAS,MFAS'
1   procedure DFS(graph G"=(V,E"),vertex v_i,
                                    vertex v_k)
      begin
2       for each vertex v_j,<v_i,v_j>εE"  do begin
3           if M(v_j)=0  then begin
4             M(v_j):=1 ;
5             DFS(G",v_j,v_i) ;
6             N(v_j):=1
            end ;
7         else if N(v_j)=0 then
8                 M(•):=0; N(•):=0; go to 13;
                  comment at this point
                          a cycle is found
        end
      end DFS ;
9   LEASTFAS(G) ;
10  MFAS':=MFAS ;
11  MFAS:=φ ;
12  delete every edge eεMFAS' from a given
    graph G and let the resultant graph be G' ;
13  while MFAS'≠φ  do begin
14      select and delete any edge e_m from MFAS' ;
15      MFAS:=MFAS U {e_m} ;
16      E":=E' U {e_m} ;
17      while any vertex v_i,M(v_i)=0  do begin
18          M(v_i):=1 ;
19          DFS(G",v_i,Dummy) ;
          end
20      MFAS:=MFAS-{e_m}; E':=E'U{e_m}
      end
 end APFAS#1 ;
```

Fig.6. Procedure APFAS#1.

Although a feedback arc set that results from procedure LEASTFAS seems to be optimal than that obtained from a MDFS, its minimality is not quaranteed. To ensure this property, procedure LEASTFAS may be followed by a minimality check routine. Procedure APFAS#1, shown in Fig.6, has a check routine based on a DFS.

Lemma 2. Procedure LEASTFAS has the time complexity of $O(n^2m^2)$.

Proof: To identify the reference tree-arc for a particular MDFS, we must form fundamental cutsets with respect to Ω_t. This formulation requires O(m) operations for each tree-arc using M and N numbers. In addition to the generation of cutsets, the identification of effective cf-links must be executed. To do this, $O(|\Omega_b| \cdot |F_{cf}(e)|)$ comparisons are required for each cutsets, Then procedure REFINE requires $O(n+m+m^2+n.m^2)$ operations. Since the number of calls of REFINE is at most (n-1) times, the number of tree-arcs, this algorithm has the time complexity of $O(n^2m^2)$.

From Lemmas 1 and 2, we can get the following lemma. Q.E.D.

Lemma 3. Procedure APFAS#1 has the time complexity of $O(n^2m^2)$ and space complexity of $O(n \cdot m)$.

4. An Approximate Algorithm

Algorithms stated above are themselves heuristic ones for the problem. However it is possible to refine the solution so as to be more and more optimal one by recursive calls of procedure APFAS#1. Fig.7 shows such an algorithm in which procedure REFINE is executed on the graph G' obtained from G by the deletion of a feedback arc, one by one, and then the number of

calls of APFAS#1 is equal to |MFAS| (the size of FAS) for each call of REFINE.

As a simple example, consider a graph with 10 vertices and 22 arcs as shown in Fig.8. During LEASTFAS, the resulting graphs of MDFS in the first, second, and third passes of REFINE are shown in Fig.'s 8(a), 8(b), and 8(c), respectively. In Fig.8(a), we can identify the reference tree-arc as (8,3) whose k-value is 2-1=1, then G_1 is obtained from G_0 by deleting this arc, for which the second MDFS is executed. $\vec{G}_1$ has all effective cf-links with respect to e_r=(4,6) for which

```
procedure APFAS#2(graph G=(V,E))
 begin
   arc set MFAS,MFAS'
   integer value K
1   procedure REFINE(graph G=(V,E))
      begin
2       MFAS':=MFAS ;
3       while MFAS'≠φ  do begin
4           select and delete any edge e from
            MFAS', delete e from a given graph
            G and let the resultant graph be G' ;
5           APFAS#1(G') ;
6           if K>|MFAS|+1  then begin
7               K:=|MFAS|+1 ;
8               REFINE(G)
              end
          end
      end REFINE ;
9   APFAS#1(G) ;
10  K:=|MFAS| ;
11  REFINE(G)
 end APFAS#2 ;
```

Fig.7. Procedure APFAS#2.

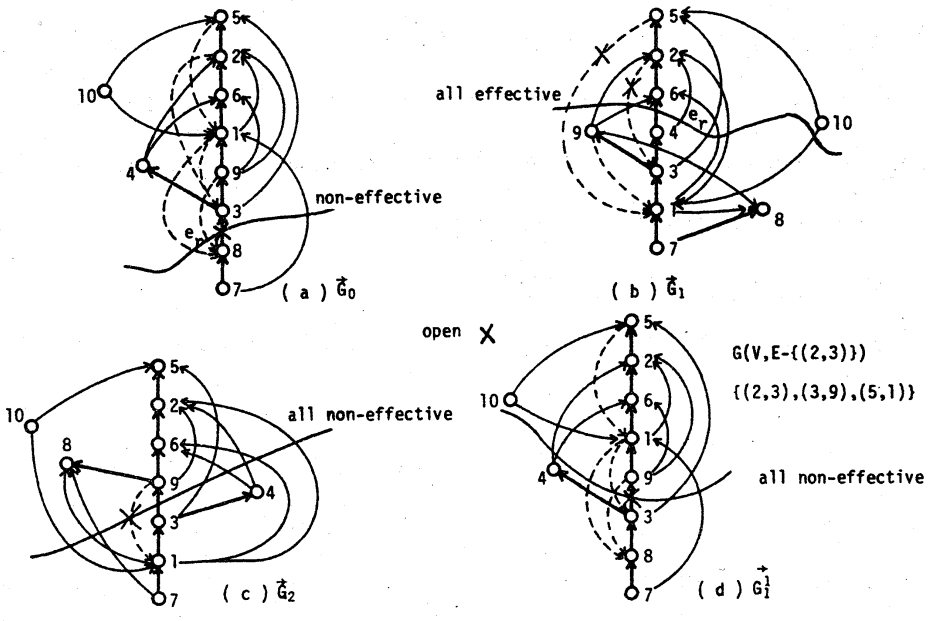

Fig.8. An illustrative example.

the k-value is minimum as shown in Fig.8(b). Deleting arcs (5,1) and (2,3) we get G_2 and $\vec{G}_2$ as shown in Fig.8(c). Finally arc (9,1) is deleted. A set of these deleted arcs {(8,3),(5,1), (2,3),(9,1)} is found to be minimal in the check routine of APFAS#1. Fig.8(d) shows an intermediate stage of APFAS#1 at line 5 of Fig.7 for G[V,E-{(2,3)}], from which we can identify the reference tree-arc as (3,9) with k-value 3, and then (3,9) is deleted. After the completion of APFAS#1, we can get a FAS {(3,9),(5,1)} of G_1^1 which leads an optimal FAS {(2,3),(3,9), (5,1)} of G_0.

To estimate the time complexity, we need to know the number of calls of REFINE. For a complete graph, an output of APFAS#1 at line 9 is a minimum FAS, so only once a call is occured. Hence we need $O(n^2m^3)$ operations for the completion of APFAS#2. In general case, it is difficult to estimate the exact number, but the following statement is true. Let the size of FAS at line 10 be k, and suppose that the refinement is always occured at the last trial arc of the pertinent FAS and the decrement is one. The number of calls of APFAS#1 is estimated as k+(k-1)+(k -2)+····, which is smaller than k^2.

From Lemma 3 and discussions above, we get Theorem 1.

Theorem 1. Procedure APFAS#2 has the time complexity of $O(n^2m^4)$ and the space complexity of $O(n\cdot m)$.

5. Test Graphs

Usually, to test the efficiency of such a heuristic algorithm, its behaviours on random graphs are analyzed, where a problem occurs as to how sample graphs are constructed. On the

other hand, associated with evaluation of heuristic algorithms for the minimum feedback arc set problem, a certain type of graphs, called directed star polygons(DSP's), satisfy the following conditions.

(1) There exists a simple algorithm for graph generation, and

(2) A minimum feedback arc set can be easily identified.

Several properties of DSP's have been investigated, and the procedure for finding minimum feedback arc set has been discussed[12,13]. In this section, we shall state the results in short.

Definition 5. Directed star polygon: A graph G=(V,E) is a directed star polygon if vertices v_i are labeled in such a way that $(v_i,v_j) \varepsilon E$ if and only if $(v_{i+k},v_{j+k}) \varepsilon E$, k=1,2,$\cdots$,n-1, where each of subscripts is expressed as one of the numbers 1,2, $\cdots$,n modulo n.

Definition 6. Symbol of DSP: Let G=(V,E) be a DSP. Let J= $\{j \mid 2 \leq j \leq n, (v_1,v_j) \varepsilon E\}$ and $S=\{s \mid s=j-1, j \varepsilon J\}$. Then a DSP is uniquely specified by n and symbol S, henceforce denoted by G(n,S).

Fig.9 shows a simple example of DSP. It is seen that J={3,4} and then S={2,3}.

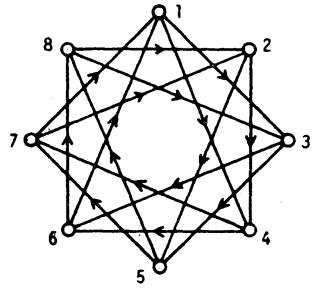

Fig.9. A DSP G(8,{2,3}).

Let f(G) be the size of minimum feedback arc sets. We have the following theorems.

Theorem 2. G(n,S) is a connected graph if and only if (n,S)=1 where (n,S) denotes the greatest common measure of n and all $s \varepsilon S$.

Theorem 3. Two DSP's G(n,S) and G(n,S') are isomorphic if and only if there exists an integer p such that (n,p)=1 and $S'=p\cdot S \triangleq \{sp \pmod n \mid s \in S\}$.

Theorem 4. Any DSP of degree 4 has a minimum feedback arc set whose size f(G) is equal to the maximum number of arc disjoint cycles.

Theorem 5. A minimum feedback arc set of any DSP G(n,S) of degree 4 is identified from S'_m which belongs to a set $\{S' \mid S' = p\cdot s,\ 1 \le p \le n-1\}$ and has a minimum sum of elements.

Theorem 6. Let $\{1,s_2,s_3,\cdots,s_m\}$ be a symbol of G(n,S). Define $k_i=\lfloor n/s_i \rfloor$ and $r_i=n-s_ik_i$ for $i \ge 2$. If $\sum_{i=2}^{m} r_is_i \le n$ then

$$f(G)=1+\sum_{i=2}^{m} s_i. \tag{2}$$

Theorem 7. For a given G(n,S), if $\sum_{i=1}^{m} s_i=n$ then

$$f(G)=n \tag{3}$$

where $m=|S|$.

6. Experimental Results

The algorithm has been implemented on a FACOM M-200 and has been applied to several types of DSP's. In this experiments, the algorithm was many times run for a particular DSP and its modified graph which is generated from the DSP by deletion of arcs that form one of f(G) arc disjoint cycles of the original graph, it should be noted that the size of minimum FAS of the resulting graph becomes f(G)-1. Furthermore, to get meaningful trials we used the following random permutations for the labeling of vertices[14].

Random permutations perform a series of n-1 transformations. Starting any permutation $\Pi=\{\pi_1,\pi_2,\cdots,\pi_n\}$, the element π_n is

interchanged with one of $\pi_1,\pi_2,\cdots,\pi_n$, choosen randomly. Then π_{n-1} is interchanged with one of $\pi_1,\pi_2,\cdots,\pi_{n-1}$, choosen randomly, and so on. This formation yields that each of n! permutations is equaly probable.

Table 1 shows the results for G(15,S), |S|=3. There are 49 different connected DSP's among which f(G)'s of 42 graphs were known a priori. The algorithm has been applied twice to each of these graphs. Table 1 is a list of 39 symbols for which both trials have been in successful, that is , f(G)=f*(G) where f*(G) denotes the solution. For the remaining three symbols {1,5,12},{1,2,14}, and {1,6,9}, we have got $f^*(G)-f(G)\leq 2$ [15].

Table 1. Experimental results for G(15,S), |S|=3.

S	f(G)	S	f(G)	S	f(G)
1,2,3	6	1,3,7	11	1,5,10	16
1,2,4	7	1,3,9	11	1,5,11	11
1,2,5	8	1,3,11	15	1,5,14	20
1,2,6	9	1,3,12	16	1,6,10	11
1,2,7	10	1,3,13	12	1,6,11	9
1,2,9	9	1,3,14	18	1,6,12	12
1,2,10	11	1,4,5	10	1,6,14	18
1,2,11	12	1,4,7	12	1,9,10	10
1,2,12	15	1,4,10	15	1,10,12	12
1,2,13	16	1,4,11	16	1,11,12	11
1,3,4	8	1,5,6	11	3,5,6	14
1,3,5	9	1,5,7	13	3,5,10	18
1,3,6	10	1,5,9	15	3,5,12	20

Tables 2(a) and 2(b) show the results for G(n,{1,4,7}), which satisfys Theorem 6, and its modified graph with various n, respectively, for each of which ten trials have been executed.

Table 2. Experimental results for G(n,{1,4,7}) and its modified graph.

(a) DSP

n	f(G)+0	f(G)+1	f(G)+2	CPU time (msec)	No. of REF.
30	10	-	-	2078	2.4
35	9	1	-	2719	2.0
40	9	-	1	3387	2.6
45	10	-	-	3921	1.8
50	10	-	-	5617	2.2
55	10	-	-	5729	1.6
60	10	-	-	5645	2.4
70	10	-	-	6362	1.6
80	10	-	-	8941	1.4
90	10	-	-	10327	1.2
100	10	-	-	11926	1.5

(b) Modified graph

n	f(G)+0	f(G)+1	f(G)+2	CPU time (msec)	No. of REF.
30	8	2	-	2689	2.6
35	4	3	-	3154	2.7
40	9	1	-	3124	1.7
45	9	-	1	4315	2.1
50	6	3	1	6357	2.8
55	10	-	-	4029	1.7
60	9	1	-	6847	2.8
70	7	2	1	11887	2.4
80	8	2	-	11983	3.1
90	10	-	-	9683	1.6
100	8	1	1	26072	2.7

The number on a column labeled f(G)+i,i=0,1,2, represents the number of trials which gave us the solution being f(G)+i. The last column represents the average number of calls of REFINE during the search.

Table 3. Experimental results for other DSP's which satisfy Theorem 6.

n	S	f(G)+0	f(G)+1
30	1,3,4	9	1
31	1,2,5	10	-
40	1,3,6	10	-
50	1,5,7	3	7
51	1,5,7	1	9

Table 3 shows the results for G(n,S)'s, $S \neq \{1,4,7\}$ which satisfy Theorem 6. The frequency giving f(G)+0 is decreased compared with the preceding results, but it should be noted that $f^*(G)-f(G) \leq 1$.

Tables 4(a) and 4(b) show the results for G(n,S)'s which satisfy Theorem 7. The appearance of the results is similar with that of Table 1.

Table 4. Experimental results for G(n,S), $|S|=3$, which satisfys Theorem 7.

(a) DSP

n	S	f(G)+0	f(G)+1	CPU time (msec)	No. of REF.
30	1,2,27	10	-	20734	2.1
35	1,10,24	10	-	17997	2.0
40	1,3,36	10	-	36830	1.3
45	1,7,37	10	-	44744	1.4
50	1,14,35	10	-	58849	2.1
55	1,15,39	10	-	85019	2.6

(b) Modified graph

n	S	f(G)+0	f(G)+1	f(G)+2	No. of REF.
30	1,2,27	10	-	-	1.6
35	1,10,24	7	3	-	2.2
40	1,3,36	10	-	-	1.5
45	1,7,37	7	3	-	3.0
50	1,14,35	3	6	1	3.2
55	1,15,39	7	2	1	3.5

Furthermore, to check the effectiveness of applications of selection-rules and interchange-rules, we have examined two algorithms; one is the algorithms without use of selection-rules, called APFAS#3, and the another is the algorithm which is obtained by replacing $F_{cf,eff}(e)$ with $F^{+}_{cf}(e)$ in Eq.(1) and LEASTFAS, called APFAS#4. The results of APFAS#3 on G(n,{1,4,7})with n=30,40,50,60, have been shown that the frequency giving $f(G)=f^*(G)$ was 21/40. The algorithm APFAS#4 have been applied to

G(30,{1,3,4}). The solutions were nearly equal to $|\Omega_b|$ obtained during the first pass in procedure LEASTFAS.

These results demonstrate that procedure APFAS#2 generates efficiently good solutions.

7. Conclusions

In this paper, polynomial-time algorithms for generating suboptimal feedback arc sets have been considered. In particular, introductions of a modified depth-first search with vertex-selection-rules and of interchange-rules acting on links of a graph, have been shown to be important to improve the size of a feedback arc set. Basing on these rules, we have constructed an efficient algorithm with time complexity $O(n^2m^4)$ and space complexity $O(n \cdot m)$. Finally, the algorithm has been applied to directed star polygons and their modified graphs with various n to test the efficiency. Experimental results demonstrate that the proposed algorithm, procedure APFAS#2, generates almost optimal feedback arc sets.

We feel that this behavior on test graphs would be hold for general graphs, but further work would be necessary to verify the fact.

Acknowledgements

The authors express their appreciation to Y. Ueki and A. Kagami, Ehime University, for their assistance in the preparation of test programs and the collection of performance statistics.

This paper was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, and

Culrure of Japan under Grant: Cooperative Research(A) 435013 (1980).

References

[1] S. Seshu and M. B. Reed, Linear Graphs and Electrical networks, Addison-Wesley Publishin Co., Inc., Reading, Mass.(1961).

[2] A. Kevorkian and J. Snoek,"Decomposition in solving large sets of nonlinear simultaneous equations", in Decomposition of large scale problems, D. M. Himmemblau, Ed., Amsterdam, The Netherlands, Northland(1973).

[3] L. K. Cheung and E. S. Kuh,"The bordered triangular matrix and minimum essential sets of a digraph", IEEE Trans. Circuits and Systems, Vol. CAS-21, 5, September(1974).

[4] G. Guardabassi and A. Sangiovanni-Vincentelli,"A two levels algorithm for tearing", IEEE Trans. Circuits and Systems, Vol. CAS-23, 12, December(1976).

[5] K. V. S. Bhat and B. Kinariwala,"Optimum tearing in large scale systems and minimum feedback cutsets of a digraph", J. Franklin Inst., Vol. 307, 9(1979).

[6] D. H. Younger,"Minimum feedback arc sets for a directed graph", IEEE Trans. Circuit Theory,Vol.CT-10,2, June(1963).

[7] A. Lempel and I. Cederbaum, "Minimum feedback arc and vertex sets of a directed graph", IEEE Trans. Circuit Theory, Vol. CT-13, 4, December(1966).

[8] L. Divieti and A. Grasseli," On the determination of minimum feedback arc and vertex sets", IEEE Trans. Circuit Theory, Vol. CT-15, 1, March(1968).

[9] G. Guardabassi, "A note on minimal essential sets", IEEE Trans. Circuit Theory, Vol. CT-18, 5, September(1971).

[10] G. W. Smith and R. B. Walford, "The identification of a minimal feedback vertex sets of a directed graph", IEEE Trans. Circuits and Systems, Vol. CAS-22, 1, January(1975).

[11] R. E. Tarjan, "Depth-first search and linear graph algorithms", SIAM J. on Computing, Vol. 1, 2, Sept.(1972).

[12] H. Ariyoshi, "Feedback arc sets of directed star polygons", Proceedings of the fourteenth Asilomar Conference on Circuits, Systems and Computers,November(1980) to appear.

[13] H. Ariyoshi and Y. Higashiyama, "On minimum feedback arc sets of directed star polygons", IECE of Japan, to appear.

[14] E. M. Reingold, J. Nievergelt,and N. Deo, Combinatorial algorithms theory and practice, Prentice-Hall, Inc., Englewood Clifs, N.J.(1977).

[15] Y. Higashiyama, A. Kagami, Y. Ueki, and H. Ariyoshi, "An approximate algorithm of the minimum feedback arc set problem", Mono. IECE of Japan, CAS 80-123, Feb.(1981).