有向グラフの最小帰還点集合をもとめる問題の時間的複雑さについて

（昭和54年5月23日　原稿受付）

情報工学教室　永　松　正　博

On the Time Complexity of the Problem of Finding Minimum Feedback Vertex Sets of Directed Graphs

by Masahiro NAGAMATSU

Abstract

For a directed graph $G=(V, E)$, a subset V' of V is called a feedback vertex set of G if the subgraph of G generated by $V-V'$ has no circuit. It is well-known that the problem of minimum cardinarity feedback vertex set is NP-complete for general cases. As a consequence, this result is tantamount to showing that the problem of finding a minimum cardinarity feedback vertex set should not be solved in less than exponential time.

We show that this problem can be solved, i.e. there exists an algorithm that solves the problem, in time $O(|E|)$ for reducible flow graphs, and that this problem is NP-complete even for the flow graphs each node of which has outdegrees and indegrees less than or equal to 2.

1．まえがき

本論文では，有向グラフの最小帰還点集合をもとめるアルゴリズムについて議論する。このアルゴリズムは多くの場面で使用される。たとえば，デッドロックの生じたシステムに対して，いくつかのプロセスを中止させてデッドロックから回復させるとき，中止させるプロセスの個数を最小にしたい場合，[1]相互に呼び合う複数個のプロセデュアからなるプログラムが与えられたとき，それと等価で，プロセジュア間の呼び合う関係が最も簡潔なプログラムをもとめる場合，[11]それから，枝に述語をつけることにより，プログラムの正当性を証明するときに，述語をつける枝の個数を最小にしたい場合[7]などである。一般的な有向グラフのクラスに対して，最小帰還点集合をもとめる問題はＮＰ完全であることが知られている。[10]また，可約な流れ図のクラスに対しては，$O(|V|\cdot|E|)$ の時間で最小帰還点集合をもとめるアルゴリズムが存在する。[2]本論文では，可約な流れ図のクラスに対して $O(|E|)$ の時間で最小帰還点集合をもとめるアルゴリズムが存在することを示し，また，対象とする流れ図の各点の入次数，出次数を制限したとき，NP 完全性がどのようになるかについて述べる。

2．諸概念の説明

2.1．グラフに関する定義

有向グラフを2項組 (V, E) で表わす。ここで V は点とよばれるものの有限集合であり，E は枝とよばれるものの有限集合であり $E\subset\{(u,v)\mid u,v\in V\}$ である。枝 (u,v) を点 u から出て点 v に入る枝とよび，点 u を枝 (u,v) の始点，点 v を終点とよぶ。$G=(V,E)$ を有向グラフとする。$(u,v)\in E$ のとき，v を u のサクセッサ，u を v のプレディセッサとよぶ。点 u のすべてのサクセッサを一列に並べたものを点 u のサクセッサリストとよぶ。すべての点のサクセッサリストが与えられれば，もとの有向グラフを完全に表わすことができる。これを有向グラフのサクセッサリスト表現とよぶ。点 v のサクセッサの個数を v の出次数とよび $d_G^+(v)$ で表わす。また点 v のプレディセッサの個数を v の入次数とよび $d_G^-(v)$ で表わす。

$G=(V,E)$ を有向グラフとする。$V'\subset V$ かつ $E'=$

$\{(u,v)\mid u,v\in V',(u,v)\in E\}$であるとき，有向グラフ $G'=(V',E')$ を点集合 V' による G のサブグラフという。また，$E''\subset E$ であるとき，有向グラフ $G''=(V,E'')$ を枝集合 E'' による G のパーシャルグラフという。点の列 $(v_1,v_2,\cdots,v_m)$ が道であるとは，$1\leqq i\leqq m-1$ を満たすすべての i に対して $(v_i,v_{i+1})\in E$ であることである。点 v_1 をこの道の始点，点 v_m を終点といい，このとき点 v_1 から点 v_m に到達できるという。$v_1=v_m$ であるとき，この点の列を有向閉路という。$\Delta\subset V$ とする。G の中のすべての有向閉路が Δ の元を含むとき，Δ を G の帰還点集合（以下，FVS と略す）という。FVS のうちで点の個数が最小のものを最小帰還点集合（以下，MFVS と略す）という。グラフに関する詳しい諸定義は(4)を参照されたい。

2.2. デプスファーストサーチと可約な流れ図

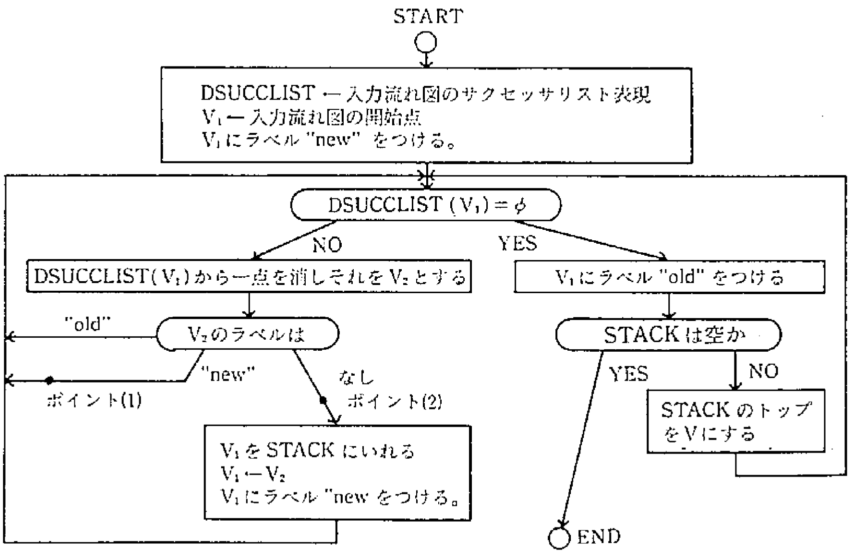
流れ図とは，開始点とよばれる特別な点をもつ有向グラフで，開始点からその有向グラフの任意の点に到達する道が存在するようなものである。流れ図とは，言葉どおり，計算機プログラムの命令を点に，コントロールの流れを枝とみなして有向グラフで表わしたものである。

次に，流れ図に対して，デプスファーストサーチをおこなうアルゴリズムを図-1に示す。このアルゴリズムの入力は，サクセッサリスト表現された流れ図である。アルゴリズムではプッシュダウンスタック STACK をひとつ使用する。サーチの途中でラベル "new" をつけられた点がちょうど STACK 上にならんでいる点になっている。

根とよばれる一点をもち，すべての点から根へ（根からすべての点へ）至る道が正確に一本だけであるような有向グラフを流入木（流出木）とよぶ。

$F=(V,E)$ を流れ図，V_s を F の開始点，SUCLIST を F のサクセッサリスト表現とする。SUCCLIST に対してアルゴリズム DFS を適用しているとき，ポイント(1)†を通過する時点での枝 (V_1,V_2) を F の SUCCLIST に対する逆方向枝という。またポイント(2)を通過する時点での枝 (V_1,V_2) の集合を E' とすると，F のパーシャルグラフ (V,E') は v_s を根とする流出木になる。この木を F の SUCCLIST に対するデプスファーストスパニングツリーとよぶ。

デプスファーストサーチ DFS を実行しているとき，あ



図―1　アルゴリズム DFS

† 本論文ではアルゴリズムを図-1のようにフローチャートの形で示すことにするが，アルゴリズム中のある場所を指定するとき，図中に示してある "ポイント(i)" を使用することにする。

る点にラベル“new”をつけるとき，その点にサーチが到達したといい，ラベル“old”をつけるとき，その点のサーチが終了したという。

可約な流れ図の定義は〝インターバル〟という言葉を使ってなされるのが普通である[8]が，ここではそれと等価な[9]次の定義をつかう。流れ図 F が可約であるとは，F のいかなるサクセッサリスト表現に対しても，F にアルゴリズム DFS を適用して得られる逆方向枝の集合が一意に定まることである。したがって F が可約な流れ図である場合，

BACK $(F)=\{v\mid v$ は F の逆方向枝の終点$\}$

という定義をすることができる。

2.3. NP 完全性

Σ をアルファベット，$L_1, L_2 \subset \Sigma^*$ とする。決定性チューリング機械で多項式時間で計算できる関数 $f: \Sigma^* \to \Sigma^*$ が存在して，$w \in L_1 \to f(w) \in L_2$ が成立するとき，L_1 は多項式時間で L_2 に変換可能であるという。$\mathcal{NP}$ を非決定性チューリング機械で受理できる言語のクラスとする。$L_0 \in \mathcal{NP}$ であり，かつ，すべての $L \in \mathcal{NP}$ に対し，L が L_0 に多項式時間で変換可能である場合 L_0 はNP完全であるという。

以上では，言語について述べたが，たとえば，任意のグラフ G に対して，G が k 個の点からなる FVS (以下では k−FVS と略す) を持つか否かを判定する問題は，k, G を適当に Σ 上に語にコーディングして，言語 $L=\{k, G$ をコーディングして得られる語 $\mid G$ は k−FVS を持つ$\}$ を受理するチューリング機械をつくる問題にすることができる。このようにして，組合せ問題に対しても NP 完全性を定義することができる。

NP 完全な問題に対して多くの人が，多項式時間で解くアルゴリズムをつくろうとしているがまだ成功していない。このことは，ある問題がNP完全であることがわかったならば，その問題は実行時間が入力の大きさのベキ乗で増加するような，非常にあつかいにくい問題であることを意味する。

L_0 がNP 完全であることがわかっているとき，$L_1 \in \mathcal{NP}$ が NP 完全であることを示すには，L_0 が L_1 に多項式時間で変換可能であることを示せばよい。NP 完全性に関する詳しいことは(3)，(5)，(10)を参照されたい。

3．最小帰還点集合をもとめるアルゴリズム

この章では，可約な流れ図に対して，$O(|E|)$ の時間で最小帰還点集合をもとめるアルゴリズムを示す。

〔補題 1〕[(2)] F を可約な流れ図とすると，BACK (F) の元のみからなる F の MFVS が存在する。

$F_0=(V_0, E_0)$ を流れ図，v_s を F_0 の開始点とする。$v_0 \in V_0$ とし，$E_1=E_0-\{(v_0, v)\mid v \in V_0\}$ からなる F_0 のパーシャルグラフを $F_1=(V_0, E_1)$ とする。点集合 $V_2=V_0-\{v\mid v \neq v_s, d_{F_1}^-(v)=0\}$ からなる F_1 のサブグラフを $F_2=(V_2, E_2)$，点集合 $V_3=V_2-\{v\mid v \neq v_s, d_{F_2}^-(v)=0\}$ からなる F_2 のサブグラフを $F_3=(V_3, E_3)$ ……と順次 F_4, F_5，……と作ってゆき，F_k と F_{k+1} が同じ流れ図になったとする。このとき F_k を $\langle F_0-\{v_0\}\rangle$ で表わすことにする。

〔補題 2〕 F を流れ図，Δ_1 を F の MFVS，$v \in \Delta_1$，Δ_2 を $\langle F-\{v\}\rangle$ の MFVS とする。このとき，$\Delta_2 \cup \{v\}$ は F の MFVS である。

（証明） F の有向閉路のうちで，v を含まないものは $\langle F-\{v\}\rangle$ の有向閉路である。したがって，Δ_2 の元を含む。したがって $\Delta_2 \cup \{v\}$ は F の FVS である。つぎに MFVS であることを示す。$\Delta_1-\{v\}$ は $\langle F-\{v\}\rangle$ の FVS である。したがって

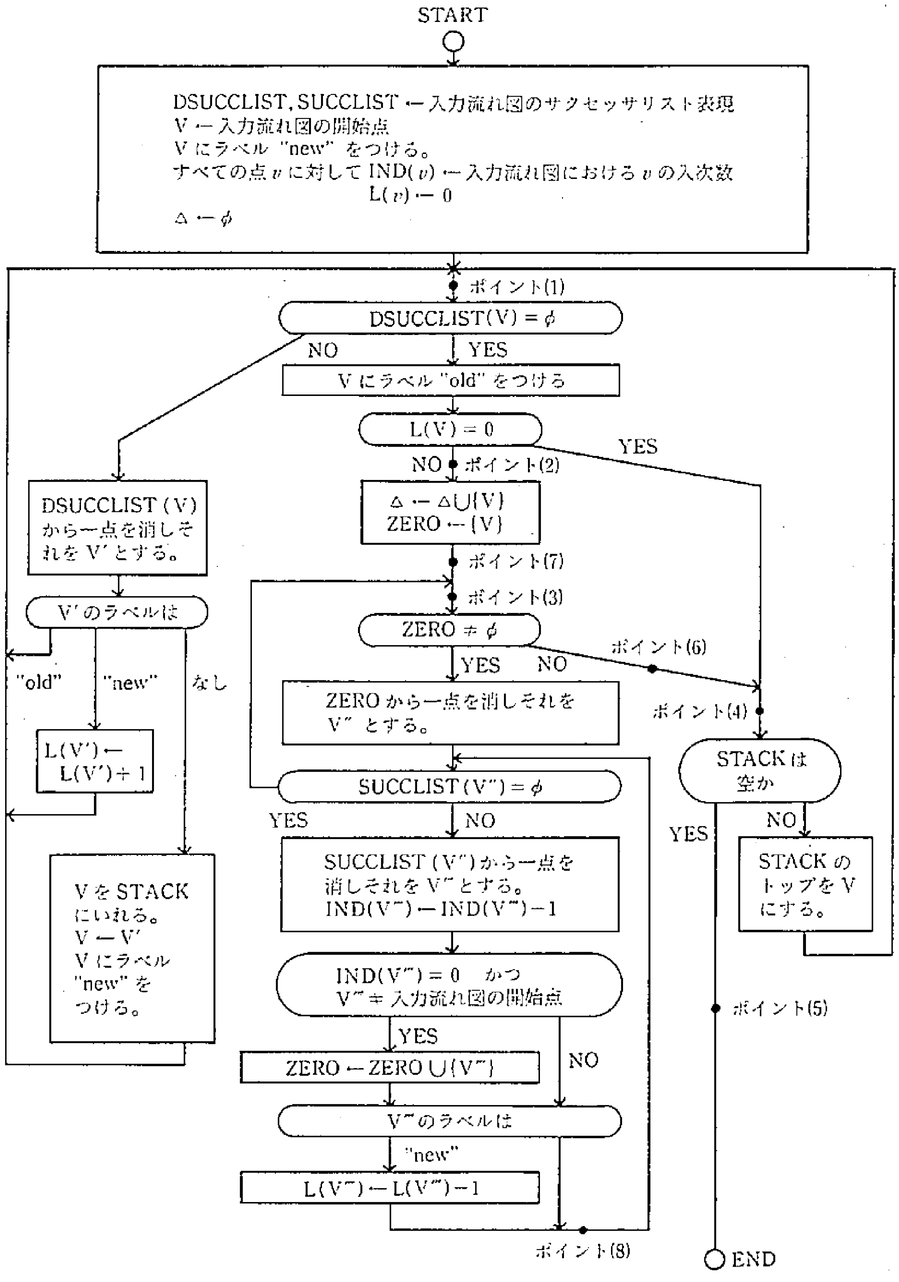
$|\Delta_2 \cup \{v\}|=|\Delta_2|+1 \leqq |\Delta_1-\{v\}|+1=|\Delta_1|$ であり，$\Delta_2 \cup \{v\}$ は F の MFVS である。

（証明終わり）

〔アルゴリズム MINFVS〕

図-2にこのアルゴリズムを示す。これは，可約な流れ図に対して MFVS をもとめるアルゴリズムである。入力流れ図が可約でないとき，このアルゴリズムは FVS を出力するが，MFVS であることは保証されない。また，任意の有向グラフに対しても，任意の点を選び，それを開始点として MINFVS をおこない。まだサーチされていない点が残っていれば，残りの点から任意の点を選び，それを開始点として MINFVS をおこなうことを繰り返せば FVS をつくることができる。

MINFVS の動作は，以下の補題，定理で正確に述べられるが，その前に，おおまかな動きを説明する。MINFVS の動作は DFS が基本となっている。DFS は BACK (F) の要素をさがし出す役目をする。MINFVS は DFS に補題 2 の前に述べた $\langle F-\{v\}\rangle$ の操作をする部分をつけ加えたものである。すなわち，MINFVS は，すでに



図—2 アルゴリズム MINFVS

MFVS の要素とわかっている点および,それらの点を通過することなしには開始点から到達できない点を除きながら，BACK (F) の要素をさがしてゆく。

流れ図に対して MINFVS を適用すると，流れ図はアルゴリズムの実行にしたがって変化してゆく。ある時点での SUCCLIST の表現している流れ図を，その時点での時点流れ図とよぶ。

〔補題3〕ポイント(1), (2), ……, (5)を図-2に書きこんである場所とする。可約な流れ図にアルゴリズムMINFVSを適用している過程において，任意の時点において，ポイント(1), (2), ……, (5)ではつぎの命題が成立する。

(i) ポイント(1), (4)において

(a) 時点流れ図は可約な流れ図であり，時点流れ図の点につけられているラベルと,STACKの内容は,時点流れ図にアルゴリズム DES をほどこしたものになっている。

(b) BACK (時点流れ図) の要素にはラベル "old" がついているものはない。

(c) すでての点 v に対して，DSUCCLIST (v)⊂SUCCLIST (v) である。

(d) すべての点 v に対して，L (v) は次に示す集合の要素の個数と等しい。

$$\left\{(u,\ v)\left|\begin{array}{l} v \in \text{DSUCCLIST}(u) \\ v \in \text{SUCCLIST}(u) \\ (u,\ v)\text{は時点流れ図の逆方向枝} \end{array}\right.\right\}$$

(ii)ポイント(2)において

(a) (i)の(a)と同じ

(b) BACK (時点流れ図)∩{ラベル "old" をもつ点} = {V}

(c) (i)の(c)と同じ

(d) (i)の(d)と同じ

(iii) ポイント(3)において

時点流れ図において，ZERO の要素を通らずには，開始点から到達できないような点の集合を FROMZERO とする。

(a) FROMZERO の要素にはすべてラベル "old" がついている。

(b) 時点流れ図から FROMZERO の要素をすべて除いた残りの点からなるサブグラフを時点非零流れ図とすると時点非零流れ図は可約である。また,時点非零流れ図の点につけられているラベルと STACK の内容は，時点非零流れ図にアルゴリズム DFS をほどこしたものになっている。

(c) BACK (時点非零流れ図) にはラベル "old" がついている点はない。

(d) (i)の(c)と同じ

(e) すべての点 $v \neq V$ に対して L (v) は次に示す集合の要素の個数と等しい。

$$\left\{(u,\ v)\left|\begin{array}{l} v \in \text{DSUCCLIST}(u) \\ v \in \text{SUCCLIST}(u) \\ (u,\ v)\text{は時点非零流れ図の逆方向枝} \end{array}\right.\right\}$$

(f) L (V) は次に示す集合の要素の個数と等しい。

$$\left\{(u,\ V)\left|\begin{array}{l} u \in \text{FROMZERO} \\ (u,\ V)\text{は時点流れ図の枝} \end{array}\right.\right\}$$

(iv) ポイント(5)において

(a) 時点流れ図は有向閉路を含まない流れ図である。

(証明略)

〔定理1〕アルゴリズム MINFVS が停止したとき，Δには入力流れ図の FVS が入っている。入力流れ図が可約である場合は，MFVS が入っている。

(証明) アルゴリズムが停止したとき，FVS がもとまることは明らかである。入力流れ図が可約な場合，MFVS がもとまることを証明する。F_0 を可約な流れ図とする。F_0 にアルゴリズム MINFVS を適用している過程を考える。ポイント(6)を i 回目に通過する時点での時点流れ図をF_iとする。また最後にポイント(6)を通過するのが m 回目であるとする。ポイント(7)を i 回目に通過する時点を考える。この時点での時点流れ図は F_{i-1} であり，時点非零流れ図は F_i である。この時点での Vの値を v_i とすると，ZERO = $\{v_i\}$ であるので，$F_i = \langle F_{i-1}-\{v_i\}\rangle$ である。また補題3のポイント(3)における命題と L (v_i) ≠ 0 であることより，FROMZERO の要素から v_iに枝が出ている。したがって，F_{i-1} において，v_i を含む有向閉路で，その上の点がすべてラベル "old" をもつようなものが存在する。一方，ポイント(2)における命題より，BACK (F_{i-1}) の要素で，その有向閉路上にあるものは v_i のみである。したがって補題1より，F_{i-1}のMFVS でv_iを含むものが存在する。補題2より，(F_i のMFVS) ∪

$\{v_i\}$ はF_{i-1} の MFVSになる。したがって，(F_m のMFVS) $\cup\{v_1,v_2,\cdots,v_m\}$ はF_0 の MFVS である。補題3のポイント(5)における命題より，F_m は有向閉路をもたない。$\{v_1,v_2,\cdots,v_m\}$ ＝（アルゴリズムが停止したときの Δ の値）は F_0 の MFVS である。　　（証明終わり）

アルゴリズム MINFVS は入力流れ図が可約でない場合, MFVS を出力しない場合もある。最悪の入力とそのサクセッサリスト表現を図-3に示す。開始点は v_n である。この流れ図の MFVSは $\{v_0\}$ であるが, 図-3に示したサクセッサリスト表現に対してアルゴリズム MINFVS を適用すると $\{v_1,v_2,\cdots,v_n\}$ を出力する。サクセッサリスト表現を SUCCLIST(v_n) = $\{v_0,v_{n-1}\}$ とすると $\{v_0\}$ を出力する。

〔定理2〕入力流れ図を $F=(V,E)$とするとアルゴリズム MINFVS は $O(|E|)$ の時間で停止する。

（証明）デプスファーストサーチの性質よりポイント(1)は高々 2 | E | 回通過する。ポイント(8)を含むループの中には SUCCLIST (V''') から一点を消せという命令があるので, このループは高々 | E | 回しかまわらない。次に、ポイント(3)を含むループについて考える。前に述べたことよりアルゴリズムが動いている間において ZERO には通算して高々 3 | E | 個しか要素が入らない。このループ中には ZERO の要素を消す命令があるので, ループも高々 3 | E | 回しかまわらない。以上のことよりこのアルゴリズムの実行時間が $O(|E|)$であることがわかかる。　　（証明終わり）

4. NP 完全性

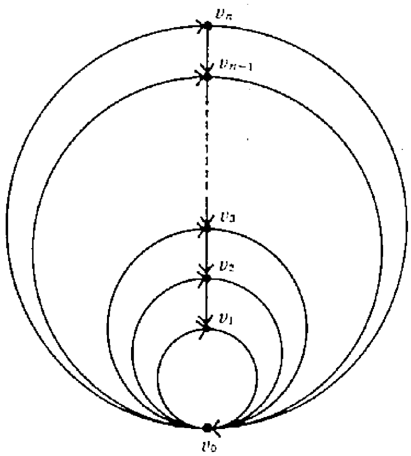
一般的な流れ図のクラスを対象にすると，MFVS をもとめる問題は NP 完全であることはすでに知られている。[10] この章では，流れ図の各点の入次数, 出次数を制限したときに MFVS をもとめる問題の NP 完全性がどのようになるかを述べる。各点の入次数（もしくは出次数が 1 以下であるような流れ図のクラスに対しては明らかに $O(|V|)$ の時間で MFTS をもとめることができる。つぎに各点の入次数，出次数が 2 以下である流れ図のクラスについて考える。

F を任意の流れ図とする。T_iを葉の個数が i 個の 2 分流入木, S_i を葉の個数が i 個の 2 分流出木とする。F の点 v の入次数，出次数がそれぞれ k,l のとき，図 4 に示すように,点 v を T_k とS_l が根でくっついたものでおきかえる。この変換を F のすべての点に対して実行して得られる流れ図を F' とする。F' のすべての点の入次数, 出次数はともに 2 以下である。しかも, F の MFVS をもとめる問題は，F'の MFVS をもとめる問題に帰着できる。したがってつぎの定理が成立する。

〔定理3〕各点の入次数，出次数がともに 2 以下であるような流れ図のクラスに対して， MFVS をもとめる問題は NP 完全である。

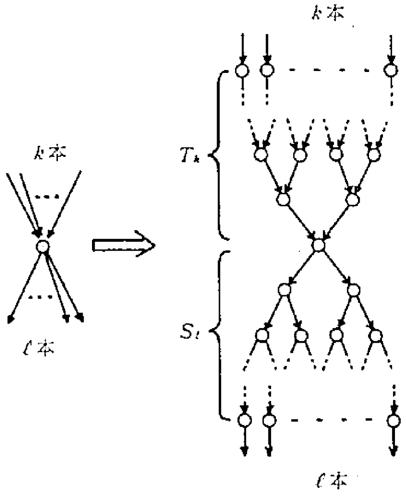
5. むすび

本論文では，3章において，可約な流れ図のクラスに対して $O(|E|)$ の時間で最小帰還点集合をみつけるアルゴリズム MINFVS を示した。MINFVS は可約でない流れ図に対しては，帰還点集合はみつけるが最小にならない場合もある。しかし、現実に計算機のプログラムを流れ図として見た場合，そのほとんどが可約である。また，近年さかんに言われている構造的プログラム[6]ではその傾向はさらに大きい。たとえば，ホワイルプログラムに対応する流れ図（D チャート）は可約である。このことは，アルゴリズム MINFVS が実用という面から



SUCCLIST(v_n) = $\{v_{n-1}, v_0\}$
SUCCLIST(v_{n-1}) = $\{v_{n-2}, v_0\}$
⋮
SUCCLIST(v_3) = $\{v_2, v_0\}$
SUCCLIST(v_2) = $\{v_1, v_0\}$
SUCCLIST(v_1) = $\{v_0\}$
SUCCLIST(v_0) = $\{v_1, v_2, \cdots, v_n\}$

図—3　アルゴリズム MINFVS に対する最悪の入力



図—4　次数を減少させる変換

見ても十分に意味があることを示している。

また，すべての枝を一度はサーチしなければ最小帰還点集合をみつけることはできないから，原理的に $O(|E|)$ よりも短い時間で動くアルゴリズムは存在しない。このことは，アルゴリズム MINFVS が最良のものであることを意味する。また現実のプログラムに対応する流れ図の場合，一般に出次数は2以下であるので，$O(|E|)$ も $O(|V|)$ も同じことになる。

有向グラフの出次数，入次数が最小帰還点集合をもとめる問題の NP 完全性に影響を与えないという4章の結果より，有向閉路の構造のみが NP 完全性にきいていると考えられる。3章で述べた可約性も，有向閉路の構造を特性づけるひとつの方向と考えられる。このことから，最小帰還点集合の問題を考える場合，有向閉路の構造の上手な特性づけを考えることが，多項式時間で解ける有向グラフのクラスを拡大するうえにも，また NP 完全性の本質をより深く理解するうえにも重要であると思われる。

参考文献

1）杉山，藤井，嵩，奥井；"分散型システムにおけるデッドロックの検出と回復"，信学技報 AL 76-47, 1976.

2）山下，本多；"可約なフローグラフの最小帰還点集合"，信学論 D Vol. J60-D No. 10, 1977.

3）Aho, A. V., J. E. Hopcroft, and J. D. Ullman ; "The Design and Analysis of Computer Algorithms" Addison-Wesley 1974.

4）Berge, C ; "Graphs and Hypergraphs" North-Holland Publishing Company 1973.

5）Cook, S. A. ; "The Complexity of Theorem Proving Procedures" Proc. 3rd Annual ACM Symp. on Theory of Computing 1971.

6）Dijkstra, E. W. ; "Note on Structured Programming" in Structured Programming, Academic Press 1972.

7）Floyd, R. W. ; "Assigning Meaning to Programs" in Proceedings of a Symp. in App. Math. 19 Mathematical Aspects of Computer Science 1967.

8）Hecht, M. S., and J. D. Ullman ; "Flow Graph Reducibility" SIAM J. Comput. 1 1972.

9）Hecht, M. S., and J. D. Ullman ; "Characterization of Reducible Flow Graphs" Journal of the ACM Vol. 21 No. 3 1974.

10）Karp, R. M. "Reducibility amoug Combinatorial problems" in Complexity of Computer Computations, Plenum Press 1972.

11）Strong, H. D., JR. A. Maggiols-Schettini, and B. K. Rosen ; "Recursion Structure Simplification" SIAM J. Comput. Vol. 4 No. 3 1975.