北見工業大学研究報告
第12巻 第2号 (昭和56年)

Memoirs of the Kitami Institute of Technology
Vol. 12 No. 2 (1981)

過負荷解消可能経路選択アルゴリズム*

奈　良　宏　一**
山　城　　迪**

(昭和55年9月24日受理)

Computing Algorithms of Selecting Over-load Removable Roots in Power System

by Koichi NARA and Susumu YAMASHIRO

In this paper, the authors deal with two algorithms concerned with graph theory. One of them is an algorithm to find all the elements which can be included in the loops containing designated elements using a fundamental loop matrix.

The other is an algorithm to find the minimal number of paths which is necessary in order to construct every possible pattern connecting one node to the other.

These patterns consist of all the combinations of paths which connect one node to the other without flowers.

Algorithms mentioned above are utilized for removing the over-load in a power system's automatic operation.

A simple example is given of the application of these algorithms to removing the over-load in a power system.

1. まえがき

電力系統の復旧過程において健全母線から脱落負荷母線へ送電する場合に，途中線路に過負荷を発生していないことを条件に，しゃ断器操作回数最小の経路を見つけたい場合がある。このためには過負荷解消操作をも含めて最小操作経路を検出しなければならず，通常の最短経路問題をそのまま適用することはできない。

そこで，筆者らは，健全系の基準母線から対象とする負荷母線へのすべての経路を見つけ，これに負荷潮流を流したとき過負荷があれば，これらの経路の組合せによって過負荷を解消する操作を行なった後，この中から最小操作の経路を見出す方法を開発し，報告してきた[1),2)]。

過負荷解消手法としては，

* 昭和53年度電気四学会北海道支部連合大会 (昭和53年10月)，昭和54年電気学会全国大会 (昭和54年4月) で発表

** 北見工業大学電気工学科

(i)　系統の切替，母線切替

(ii)　異系統運用

(iii)　発 電 調 整

(iv)　負 荷 制 限

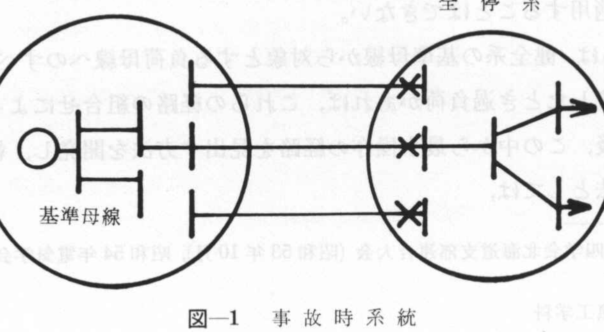
などが報告されているが[3]，筆者らは，並用またはループ投入による過負荷の解消も系統復旧時に有効であることを示し，これを利用することを提案してきた[4)~6)]。しかし，過負荷要素とループを構成する要素をグラフのスキャンによって検出する手順にかなりの演算時間を要し，最小操作などの条件を考慮せずに30母線程度の系統へ適用することがようやく可能な程度であった[6]。そのため，実規模系統で最小操作経路を見出すためには過負荷要素とループを構成する要素を高速に検出し，これを含む経路の組合せから過負荷を発生しない最小操作経路を高速に見出すアルゴリズムを開発する必要があった。

ループを構成する要素は記号積（ラテン積）によって高速に数え上げることが可能である[6),8)]。しかし，この方法はマトリクスの要素として記号列を扱う必要があり，通常のオンライン制御用計算機の記憶容量の範囲を超えた記憶容量を必要とするため，実規模系統での実用化は不可能である。

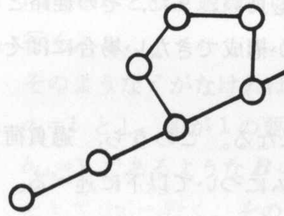
本稿では，まず，基本ループ行列を用いて過負荷要素とループを構成可能な要素を実用範囲の記憶容量で高速に検出する計算アルゴリズムについて述べる。次に，組合せから除外しても過負荷解消に影響を与えない経路があることを示し，この経路の検出アルゴリズムについて述べる。最後に，簡単な例題によってこれらの計算アルゴリズムを確認している。

2.　ループ投入による過負荷解消アルゴリズムの概要

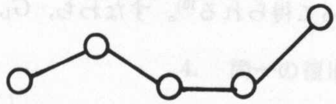
いま，事故後系統を分離系統の並列等によって図–1のように健全系と全停系に分離できるものとする。全停系の特定の負荷に健全系より電力を供給するためのあらゆる可能な復旧パターンは，健全系の1つの基準母線から目的の負荷に至る途中に花を作らないすべての経路の組合せによって構成することができる。



図—1　事 故 時 系 統

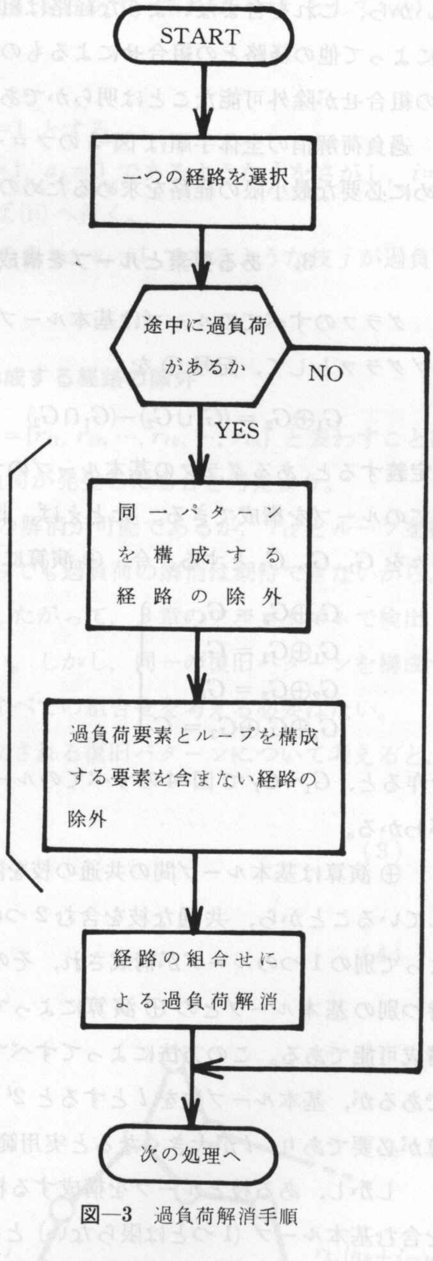


(a)　花を作る経路



(b)　花を作らない経路

図—2　花を作る経路と花を作らない経路



図—3　過負荷解消手順

花を作らない経路とは，ある1つのノードから出発して再びそのノードにもどるような閉路を作らない経路のことである[8)]。図-2に，花を作る経路と花を作らない経路の例を示している。以下，花を作らない経路を単に経路と呼ぶことにする。

2点間を結ぶ相異なる2つの経路はループを構成する。

また，ある装置（線路）の過負荷はその装置（線路）とループを構成する装置（線路）の投入によって解消可能であることは先に示した[4),5)]。

したがって，ある1つの経路をとった時途中に過負荷を発生するならば，別の経路を投入し，それによって構成されたループに過負荷要素が含まれているならば過負荷を解消できる可能性がある。もし，最小操作などの条件を満足する必要があれば，過負荷を発生した経路と他のすべての経路のあらゆる組合せの中から，過負荷を発生せずに条件を満足するパターンを選択すれば良い。ところが，いま，経路の数を m とすると，ある経路と他の経路を組合せる方法は 2^{m-1} 通りあり，m の増加によって組合せの数は極めて大きくなる。m が20を超えると実用上演算困難となり，実規模系統へ適用するには，この組合せの数を減らす必要がある。

実際，過負荷解消可能な経路には過負荷要素とループを構成する要素を含まなければなら

ないから，これを含まないような経路は組合せから除外しても良い。また，その経路との組合せによって他の経路との組合せによるものと同じパターンしか構成できない場合にはその経路との組合せが除外可能なことは明らかである。

過負荷解消の全体手順は図-3のフローチャートのようになる。このうち，過負荷解消のために必要な最小限の経路を求めるための2つのアルゴリズムについて以下に述べる。

3. ある要素とループを構成するすべての要素の検出アルゴリズム

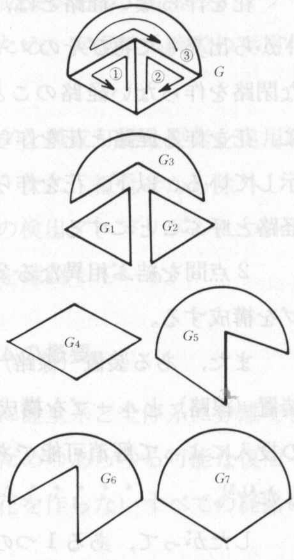
グラフのすべてのループは基本ループの組合せによって得られる[10]。すなわち，G_1, G_2 をサブグラフとして，記号 $\oplus$ を，

$$G_1 \oplus G_2 = (G_1 \cup G_2) - (G_1 \cap G_2) \tag{1}$$

で定義すると，あるグラフの基本ループのすべての組合せの $\oplus$ 演算によってそのグラフのすべてのループを構成できる。たとえば，図-4において基本ループを①, ②, ③と定め，この各々を G_1, G_2, G_3 とする。今，$\oplus$ 演算によって，

$$\left.\begin{aligned} G_1 \oplus G_2 &= G_4 \\ G_1 \oplus G_3 &= G_5 \\ G_2 \oplus G_3 &= G_6 \\ G_1 \oplus G_2 \oplus G_3 &= G_7 \end{aligned}\right\} \tag{2}$$

を作ると，$G_1 \sim G_7$ で図-4のすべてのループを尽していることがわかる。



図—4 グラフのすべての閉路

$\oplus$ 演算は基本ループ間の共通の枝を除去する操作を意味していることから，共通な枝を含む2つのループの $\oplus$ 演算によって別の1つのループが構成され，そのループと共通な枝を持つ別の基本ループとの $\oplus$ 演算によってさらに別のループが構成可能である。この方法によってすべてのループを構成可能であるが，基本ループ数を l とすると 2^l 通りの組合せの $\oplus$ 演算が必要であり，l が大きくなると実用範囲を超えてしまう。

しかし，ある枝とループを構成する枝をすべて求めるだけであるならば，それは，その枝を含む基本ループ（1つとは限らない）と共通の枝を持つ基本ループ，さらにその基本ループと共通の枝を持つ基本ループというように次々と関係する基本ループを選択していくことによって可能である。すなわち，それらの基本ループを構成している枝のすべてがもとの枝とループを構成することを利用しており，次のアルゴリズムによってある枝とループを構成するすべての枝を求めることができる。

(i) 基本ループマトリクス $\boldsymbol{B}(c \times m)$，最初にすべての要素が0であるような補助ベクトル $\boldsymbol{a}(c \times 1)$，結果ベクトル $\boldsymbol{e}(1 \times m)$ を準備する。$i'=0$ とする。

(ii) $i'=i'+1$ とし，過負荷の枝 j を含む $\boldsymbol{B}$ の行 i' をさがす。あれば $i=i'$ として (iv) へ行く。

(iii) そのような i' がなければ終了する。

(iv) $a_i=1$ とし，値が1の要素 b_{ij} に対して $e_i=1$ とする。

(v) $b_{ij}=1$ であるような $\boldsymbol{B}$ の列 j を見て，$b_{kj}=1$, $a_k=0$ であるような k をさがし，$i=k$ として(iv)へ行く。そのような k がなければ(ii)へ行く。

上記アルゴリズムで得られたベクトル $\boldsymbol{e}$ の要素のうち，$e_j=1$ であるような枝 j が過負荷の枝とループを構成する枝となっている。

4. 同一の復旧パターンを構成する経路の除外

基準母線から負荷母線に至る i 番目の経路を $\boldsymbol{R}_i=[r_{i1}, r_{i2}, \cdots, r_{ik}, \cdots, r_{in}]$ と表わすことにする。いま，この経路を投入したとき線路 r_{ik} に過負荷が発生した場合を考えよう。

この場合，別の経路との組合せによって過負荷の解消が可能であるが，r_{ik} とループを構成する要素を含まない経路は，その経路の投入によっても過負荷の解消は期待できないから，これらの経路は組合せから除外することができる。したがって，3章のアルゴリズムで検出した要素を含む経路についてのみ組合せを考えれば良い。しかし，同一の復旧パターンを構成するものが含まれているため，これらについてさえ，すべての組合せを考える必要はない。

すなわち，i 番目の経路との組合せによって構成される復旧パターンについて考えると，$j=j\sim n_i-1$ について，

$$r_{k,(n_k+j+1-n_i)} = r_{i,(j+1)} \tag{3}$$

を満足する $\boldsymbol{R}_k$ をさがし，$j=1\sim j-1$ について，

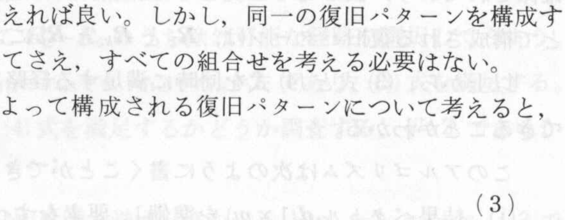
$$r_{lj} = r_{kj} \tag{4}$$

但し，n_i : $\boldsymbol{R}_i$ に含まれる要素の数

を満足する $\boldsymbol{R}_l$ $(l>k)$ をすべて除外しても，$\boldsymbol{R}_i$ との組合せによって構成可能なすべての復旧パターンを構成することができる。

これは次の理由による。

出発母線から分枝があるたびにそのすべての枝に枝を分けて各経路を描くと，基準母線から負荷に至るすべての経路は図-5のように描くことができる。図-5に示すように，(3)式を満足する $\boldsymbol{R}_k$ は，要素 $r_{k,(n_k+j-n_i)}$ において $\boldsymbol{R}_i$ の要



図−5 同一要素を含む経路

素 r_{ij} が至ると同じ母線に至っており，これ以降については r_{ij} に続く経路と，$r_{k,(n_k+j-n_i)}$ に続く経路が同一かまたは各々，お互いに相手の要素を通る経路しか存在しない。(線路を枝，母線をノードとしたグラフで考えれば，このグラフは無向性であるから，各々相手の線路を通る経路も考えられる。)

まず，$r_{k,(n_k+j+1-n_i)}$ 以降の経路が同じ場合について考えよう。この場合，$r_{k,(n_k+j-n_i)}$ までの経路のみが異っているので，ある1つの経路 $\boldsymbol{R}_k$ に $r_{k,(n_k+j-n_i)}$ 以前の経路を代表させると，それ以外の経路 $\boldsymbol{R}_l$ は $\boldsymbol{R}_i$ との組合せによっても，$\boldsymbol{R}_i$, $\boldsymbol{R}_k$ と r_{ij} 以降の経路が $r_{l,(n_l+j-n_i)}$ 以降の経路と等しい経路の組合せによって得られるパターンと同一のパターンしか構成しない。したがって，$\boldsymbol{R}_l$ を除外しても構成不可能なパターンが生じることはない。

次に，お互いに相手の線路を通る場合について考えてみる。

今，$\boldsymbol{R}_i$ を基準経路としているので，$r_{k,(n_k+j-n_i)}$ の後，r_{ij} を逆向きに通る場合についてのみ考えれば良い。$r_{k,(n_k+j-n_i)}$ の後，r_{ij} を通って至る母線は $\boldsymbol{R}_i$ で $r_{i,(j-1)}$ を通って至る母線に等しい。この母線からのすべての分枝は，この母線に至った時になされており，$\boldsymbol{R}_i$ もその1つにすぎない。したがって，$r_{i,(j-1)}$ を通って至った母線からのすべての経路はその時点で確保されており，仮にその1つを $\boldsymbol{R}'_i$ とすると，$r_{k,(n_k+j-n_i)}$ の後，r_{ij} を通る経路の1つと $\boldsymbol{R}_i$ とで構成される復旧パターンは，$\boldsymbol{R}_i$ と $\boldsymbol{R}_k$ と $\boldsymbol{R}'_i$ によって構成できる。

以上から，(3) 式と (4) 式を同時に満足する経路を除外してもすべての復旧パターンを構成できることがわかる。

このアルゴリズムは次のように書くことができる。

(i)　結果ベクトル $\boldsymbol{d}(1\times m)$ を準備し，要素をすべて0とする (mは経路の数。)。$i=1$ とする。

(ii)　$k=i+1$ より順に (3) 式を満足する $\boldsymbol{R}_k$ をさがす。なければ (iv) へ行く。

(iii)　あれば $l>k$ について (4) 式を満足する $\boldsymbol{R}_l$ をさがし，$d_l=1$ とする。(4) 式を満足する $\boldsymbol{R}_l$ がなくなるまで (iii) を繰り返す。

(iv)　$i=i+1$ として (ii) へ行く。$i=m$ ならば終了。

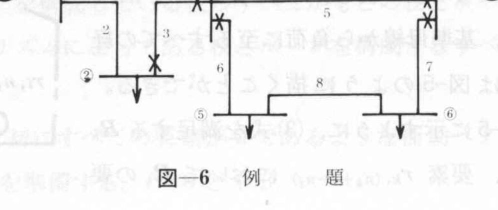
6. 例　　題

図-6のような系統を考える。母線 ① を無限大母線とし，故障によって図のように母線 ④, ⑤, ⑥ が停電している状態を考える。この状態で母線①のから⑥へ電力を供給したいと仮定すると，①～⑥ 間の花を作らない経路は次のようになる (図-7 参照)。

$\boldsymbol{R}_1=[1,\ 4,\ 7]$

$\boldsymbol{R}_2=[1,\ 5,\ 7]$

$\boldsymbol{R}_3=[1,\ 6,\ 8]$



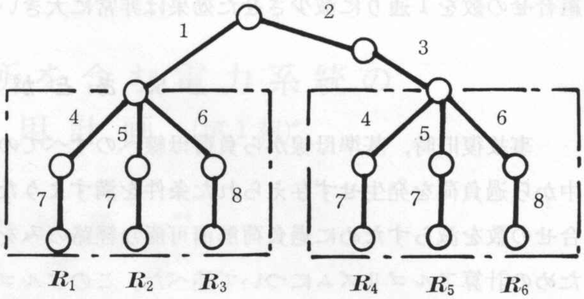
図—6　例　　題

$\boldsymbol{R}_4=[2,\ 3,\ 4,\ 7]$

$\boldsymbol{R}_5=[2,\ 3,\ 5,\ 7]$

$\boldsymbol{R}_6=[2,\ 3,\ 6,\ 8]$

今，$\boldsymbol{R}_1$を選択したとき，線路1が過負荷するとしよう。過負荷は$\boldsymbol{R}_2$～$\boldsymbol{R}_6$の経路のうち，線路1とループを構成する線路を含む経路との組合せによって解消される可能性がある。



図―7　例題の経路

さて，図-6の系統の基本ループ行列は，

$$
\boldsymbol{B}=\begin{array}{c}\\ (1)\\ (2)\\ (3)\end{array}\begin{array}{c}\begin{array}{cccccccc}1&2&3&4&5&6&7&8\end{array}\\ \left[\begin{array}{ccc|ccccc}1&1&1&&&&&\\ \hline &&&1&1&&&\\ &&&&1&1&1&1\end{array}\right]\end{array}\qquad(5)
$$

であるから，3章のアルゴリズムによって$\boldsymbol{B}$の第1行のみが線路1とループを構成する要素を含むことになり，$\boldsymbol{e}=[01100000]$から，線路2, 3が結果として検出される。

ところで，4章のアルゴリズムによって$\boldsymbol{R}_2$～$\boldsymbol{R}_6$のうち除外可能な経路を選択してみよう。まず，(3)式を満足する経路をさがすと，$r_{1,2}=r_{4,3}=4$，$r_{1,3}=r_{4,4}=7$であり，(3)式を満足する。したがって，$\boldsymbol{R}_4$を残し，$\boldsymbol{R}_5$, $\boldsymbol{R}_6$について(4)式を満足するかどうか調査する。$j=2$であるから，r_{l1}，r_{l2}について調査すれば良い。

$r_{4,1}=r_{5,1}=2$，$r_{4,2}=r_{5,2}=3$であり，$\boldsymbol{R}_5$は(4)式を満す。また，$r_{4,1}=r_{6,1}=2$，$r_{4,2}=r_{6,2}=3$であり$\boldsymbol{R}_6$も(4)式を満す。したがって，$\boldsymbol{R}_5$または$\boldsymbol{R}_6$を$\boldsymbol{R}_1$と組合せて得られるパターンは，$\boldsymbol{R}_1$～$\boldsymbol{R}_4$の組合せによって得られるはずである。たとえば，$\boldsymbol{R}_1$と$\boldsymbol{R}_5$の組合せによって得られるパターンは$\boldsymbol{R}_1$, $\boldsymbol{R}_2$と$\boldsymbol{R}_4$の組合せによって得られることは図-6から明らかである。したがって，$\boldsymbol{R}_5$と$\boldsymbol{R}_6$は除外してもさしつかえない。残った経路は$\boldsymbol{R}_2$～$\boldsymbol{R}_4$である。

一方，線路1とループを構成する線路は2, 3であったから，線路2, 3を含まない経路$\boldsymbol{R}_2$, $\boldsymbol{R}_3$はこれを$\boldsymbol{R}_1$と組合せても線路1の過負荷の解消には寄与できず，組合せから除外できる。

なお，この経路の除外によって構成不可能なパターンが生ずるが，これらのパターンによっても線路1の過負荷が残るので考慮する必要はない。

以上から，$\boldsymbol{R}_1$と$\boldsymbol{R}_4$との組合せによってのみ過負荷解消の調査をすれば良いことがわかる。もし，$\boldsymbol{R}_4$との組合せによっても過負荷の解消ができないならば，母線④の発電機の立上りを待たない限り過負荷なしに母線⑥の負荷への送電はできないことになる。

線路1が過負荷のとき$\boldsymbol{R}_2$～$\boldsymbol{R}_6$の経路のすべてを組合せる場合には，$2^5=32$通りの組合せが必要であり，さらに，ループを構成する要素を含まない経路を除いたとしても$\boldsymbol{R}_4$～$\boldsymbol{R}_6$の組合せ，つまり$2^3=8$通りの組合せが必要であることを考えると，本稿のアルゴリズムによって

組合せの数を1通りに減少させた効果は非常に大きいことがわかる。

5. あ と が き

事故復旧時，基準母線から負荷母線へのすべての経路を見つけ，それらの経路の組合せの中から過負荷を発生せず与えられた条件を満すような復旧パターンを選択する場合に，その組合せの数を減らすために過負荷解消可能な経路のみを，実用範囲の記憶容量で高速に選択するための計算アルゴリズムについて述べた。このアルゴリズムは，

(i) 過負荷線路とループを構成する線路の選択アルゴリズム

(ii) 同一の復旧パターンを構成する経路を除外するアルゴリズム

の2つであった。この2つのアルゴリズムによって，組合せの数を極端に減少できることがわかった。なお，本稿のアルゴリズムを操作回数最小の条件のもとで用いる場合には，すでに見つかっている最小操作回数よりも操作回数が多い組合せについては，それ以上の調査は不要であるので，過負荷検出のための潮流計算を省略できるなど，与えられた条件によってはより高速に求める組合せを得ることができる。

おわりに，一連の研究において御指導を賜る小池東一郎北見工業大学長に深謝申し上げます。

文 献

1) 奈良・小池：「ループ投入による過負荷解消を考慮した事故時自動復旧操作手法—操作回数最小化手法—」 昭和53年度電気四学会北海道支部連大 54 (昭和53年10月).
2) 奈良・小池：「ループ投入による過負荷解消を考慮した事故時自動復旧操作手法—操作回数最小化法(その2)—」 昭和54年電気学会全国大会.
3) 鈴木・幅：「事故時系統自動操作論理の開発」 電学論 B, Vol. 93-B, No. 8 (昭和48年).
4) 奈良・山城：「ループ投入による過負荷解消を考慮した事故時自動復旧操作手法について」 昭和52年電気四学会北海道支部連大 65 (昭和52年10月).
5) 奈良・山城：「ループ投入による過負荷解消を考慮した事故時自動復旧操作手法」 北見工大研報, Vol. 10, No. 1 (昭和53年).
6) 奈良・山城：「ループ投入による過負荷解消を考慮した事故時自動復旧操作手法—送電余力最大ループ法 (その2)—」 北見工大研報, Vol. 10, No. 2 (昭和54年).
7) たとえば，A. T. Berztiss 古川他訳：「データ構造 I」 p. 235 (昭和49年)，日本コンピュータ協会.
8) たとえば，小野寺：「グラフ理論の基礎」 p. 44–88 (昭和43年)，森北出版.
9) たとえば，Maeda：「GRAPH THORY」 p. 21 (1972), WILEY.
10) たとえば，前田・伊東：「現代グラフ理論の基礎」 p. 66 (昭和53年)，オーム社.