ダイナミック・プログラミングの応用（I）*

深　尾　　毅**　　比留間玲子**

1. はじめに

1次元の変分問題として昔から多くの問題（たとえば最短曲線の問題とか，力学系あるいは物理学上の諸変分原理など）が取扱われて来たが，その多くはそのEuler の方程式によって解くことがしばしば困難である．むしろ逆に，系の動作を示す複雑な微分方程式を解く代りに，それに対する変分原理から“直接に”解を求めることがしばしば行われ，また必要とされていることは周知のとおりである．さらに近年は，“解析性”をもっていないで，昔流には定式化も困難な種々の型の変分問題だとか，確率的要素をふくむ変分問題だとかが，計画，制御の問題の必要から続々と登場して来た．

こういう種々雑多な問題には，それぞれの特徴に応じて取扱い方が工夫されるのは当然であるが，そういう段階に達するには，まずいろいろな試みによって問題の基本的な諸性質をみることが必要であろう．そのような試み，あるいは実験には高いレベルのものも，また低いレベルのものもあり，高いレベルのものによれば少ない試行で問題の全貌をとらえることができるが，問題の性質が未知であればあるほど，少なくとも初段階ではそれは困難であろう．

一方，最も低いレベルの試みとしては，ありとあらゆる場合を片端からあたり，それらを整理することによって諸性質の大凡そを知ろうとするものがある．しかし，こういう全く機械的な方法では現在の計算機では扱いかねるものが多く，比較的わずかの高度の情報をうるには余りにも効率が悪い．組織的な整理をしながら，そしてできればその整理の仕方も得られた情報にもとづいて，一層高度化あるいは指向性をもつような方法がのぞまれる．

我々が問題にする広い意味での変分問題あるいはシーケンシャルに多段的に連った形の最適化問題では，その多くが持つ特徴を利用すると，もっと少ない努力によって問題の解に関する必要な情報のすべてを能率よく取りそろえることができる．しかも努力をふやせばふやすほど精度が上るという単調増加の形で問題の性質が知られ，機械的にあらゆる場合を端から順に当るという低いレベルの試みがしばしばおかす欠点，すなわち中途で努力をやめると，ほとんど性質がわからないとか，あるいは努力の割りには僅かの情報しか得られないという欠点をふせぐことができる．このような方法がダイナミックプログラミングである．

一般にいって，我々が直面する問題の性質を計算機によって実験的に調べようとする際，

（1）　できるだけ少ない努力で誤りなく必要な情報をうること，

と同時に，

（2）　一部の努力にとどめても中途半端あるいは局部的でなく，大局的な情報が得られ，努力がふえると共にその精度を向上しうること，

が重要である．ダイナミック・プログラミングはある型に限るにしても，その範囲では（1），（2）をよく満すもので，ナイーブな方法であるというそしりはあるにせよ，そういう意味で実験手段として十分評価してよいと考える．最適化問題に限って考えた場合，それに対して種々の mathematical programming が工夫されているが，それらは多くの場合，基本的性質の知れた，型のきまった問題に対する有効な“計算方法”ではありえても，未知の問題に対する“実験方法”ではありえない．

以上において，性質の未知な変分問題に対してダイナミック・プログラミングが有効な実験手段であることをのべたが，そのことは同時に変分問題のきわめて一般的な計算法を与えていることに他ならない．したがって基本的性質はある程度わかっていても諸制限等があって複雑な構成の，扱いにくい変分問題の計算に対して有力な出発点あるいは中心的役割を果しうる．これが第2の応用分野である．ただ，それのみでは計算法としては一般的でありすぎるため，実際的に，能率よく計算を行うには，問題に応じて幾多の工夫が必要となる．特に，制御要素の多い（多変関数の）変分

* Applications of Dynamic Programming (I).
by Takeshi Fukao, Reiko Hiruma, (Electrotechnical Laboratory)

** 電気試験所

問題でそのことがいえる．

そこで本文では，第2の計算法という観点に立って，実際問題で必ず問題になる多変関数の場合に対する計算上の工夫を，貯水池運用問題を例にして多少実施してみたので報告したい．貯水池運用問題は典型的変分問題の一つであり，在庫管理の問題などと密接な対応関係がある．特に stochastic な場合の計算法は興味深く重要であるが，本文ではその手はじめに deterministic な場合についてのべる．

2. 例題とそれに対するダイナミック・プログラミングの適用

例にとりあげる問題は，電力系統における deterministic な貯水池運用，しかも越流（貯水池よりのあふれ）効果を考慮しない場合とする．この問題は普通，貯水池の水をたくわえうる能力を利用して，天然に与えられる河川流量にコントロールを加えて水力発電し，ある一定期間にわたる火力発電の総燃料費を最小にする問題と考え，次の変分問題であらわされる．

$$\text{Minimum}\int_0^T F\{G(t)\}dt,$$

$$\frac{dS}{dt}=J(t)-Q(t),$$

$$G(t)+\sum_i P(t)=P_R(t),$$

$${}_iP={}_iP({}_iS,\ {}_iQ),$$

$${}_iS_{\min}\leqq {}_iS(t)\leqq {}_iS_{\max},$$

$${}_iQ_{\min}\leqq {}_iQ(t)\leqq {}_iQ_{\max},\quad \text{その他の制約．}$$

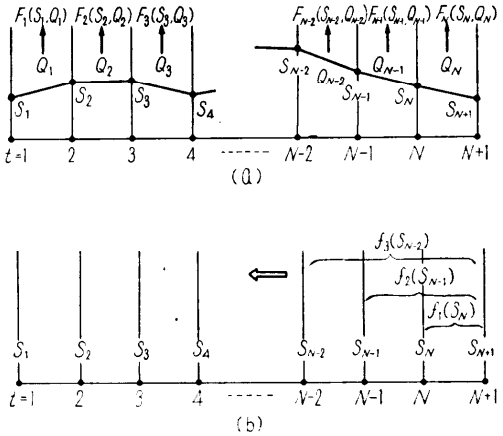
ここで　は単位時間当り燃料費，G：火力出力，P：水力出力，P_R：負荷，S：貯水量，Q：使用流量，J：貯水池への流入流量であり，FはGについてconvex，PはQについてconcaveである．（iは発電所番号）

これを離散的にあらわすと**第1図**（a）のようになるが，「ある時刻 K での貯水量を定めれば，その時刻以後の最適なオペレーションには，その時刻以前のオペレーションは直接には影響を及ぼさない」という性質があることから，

$f_{N-K+1}(S_K)$＝K-時刻でS_Kなる状態を初期条件としたそれ以後の$N-K+1$ stagesにわたる期間で，最適なオペレーションを行った時の最適コスト（S_Kの関数である）．これを最小コスト関数とよぶ．

を定義すると

$$f_{N-K+2}(S_{K-1})=\text{Min}[F_{K-1}+f_{N-K+1}(S_K)]\quad(1)$$



第 1 図

なる関係が成り立つ．したがって，たとえば**第1図**(b)にあるように，$f_1(S_N)$＝Min$[F_N(S_N,\ Q_N)]$を求め，（1）から$f_2(S_{N-1})$を求め，……とやると最後に$f_N(S_i)$が求まって，どのような初期条件を指定しても最適なオペレーションが求められる態勢になる．これがダイナミック・プログラミングの一つの型であるが，これを実行するのに最小コスト関数の計算が必要で，もし貯水池がn個あると（変関数がn個），それはn変数の多変数関数になってnが大きいと大変厄介になる．

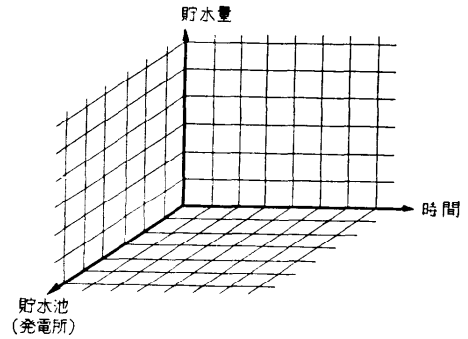
3. 逐次近似法

前にのべたように変関数の数（我々の問題では貯水池の数）がふえると，計算が実用上急激に膨大になり，困難になるという欠点がある．このような多次元性のうまい処理法としてユニークなものは現在まだ見出されていない現状であるが，我々は“逐次近似”によって，この計算の困難を取除くことにする．

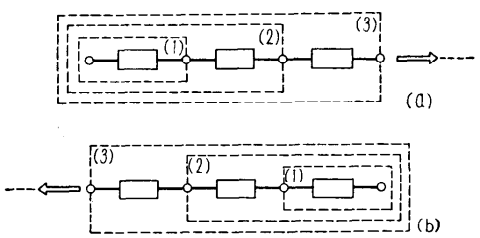
この場合，まず問題になることは，逐次近似によって到達する極小が真の最小に一致するかどうかであるが，そのためには，許容される領域と評価関数（あるいはコスト）の凸条件（convexity）が必要である．我々の貯水池運用問題では，貯水池の落差変動の出力に及ぼす影響の故に，モデルのあらわし方，近似の仕方によっては評価関数が必ずしもこの条件を満さない場合がある．しかし後にのべるような諸注意によってほぼ実用的には十分，極小と最小とが一致するものと考えてよい．したがって逐次近似法が有効になる．

さて，逐次近似法にはいくつもの型があり，ダイナ

ミック・プログラミング（以下，D.P. と略称する）の考え方そのものも原理的には逐次近似解法である．すなわち，**第2図**での概念図でいうと，時間軸にそって問題を分割し，**第3図**（a）あるいは（b）に示すように，問題（1）より（2），（3）……の順序にインフォーメーションを累積して逐に全期間にわたる最適解をうるという逐次近似解法である．



第 2 図



第 3 図

第2図に示してあるように，時間軸にそった問題分割の他に，貯水量（あるいは流量など）の軸にそって分解する逐次近似法がありうる．すなわち，一時に最適状態に到達する代りに“少しずつ”最適状態に近づこうとする逐次近似で，この型を我々は“incremental successive approximation”と称することにし，ISA と略称することにしよう．

一方，貯水池あるいは発電所軸に沿って分解する逐次近似法，すなわち，いくつかの発電所が系統内にふくまれているとき，適当な順序で，発電所一つずつ，あるいは適当なグループ単位で，一層経済的なオペレーションに逐次近づけようとする近似法があり，これを“relaxation successive approximation”とよび，RSA と略称することにする．

さて，普通，D.P. は時間軸にそった分解のみを考えて ISA も RSA も考えない．この場合には，多数貯水池をふくむ運用問題になると，時間軸にそって分割された問題間の連結が非常に大掛りとなって計算が厄介になる．すなわち，それら問題間の連結は，前にのべたように，最小コスト関数によって行われるが，それは貯水池の数だけの変数の多変数関数であり，その決定に多大の手間を要する．

そこで ISA を導入すると，各変数の変動領域がせばめられ，したがって最小コスト関数はあるせまい領域での local な表現ですむから簡単になり，さらに RSA を導入すると最小コスト関数が1変数化（1次元の問題化）して，その決定は極めて簡単になる．

さて，上記の三つの逐次近似法の成立条件は，D.P. だけだと最もゆるく，ISA を導入するとややきつくなり，さらに RSA を入れると一層苛酷になる．すなわち一般的には，

（1） D.P. はマルコフ型の多段決定過程に対して成り立ち，

（2） ISA はさらにその問題の local な極小と真の最小とが一致すれば成り立ち，

（3） RSA は ISA の範囲内で一般には成り立つ．

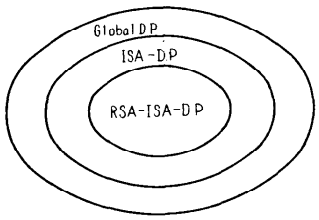
したがって，以上の三つの逐次近似法のコンビネーションを考えるとき，

（1） D.P. そのもの（global D.P. と称す），

（2） ISA なる D.P. 法（incremental D.P. と称す），

（3） ISA にして RSA なる D.P. 法（increamental relaxation D.P. と称す）．

の三つが考えられ，その包含関係を示すと**第4図**のようになる．



第 4 図

具体的な計算の実行に際して問題になるもう一つの事柄は最小コスト関数の表現の仕方であって，それには，

（1） 適当な連続関数の集り（代数関数，特に多項式あるいは一組の多項式群）で近似するもの，

（2） 有限個の点でのみあらわすもの，

が考えられる．（1）の場合を関数近似（functional

approximation)，略して FA，（2）の場合を多点近似（multi-points approximation)，略して PA と称することにする．もちろん，（1）でも，どのような関数近似をするかによって類別が行われ，（2）でも multi-points の数によって差異を生ずるわけではある．

以上のことから我々は，次の6種類の計算法を挙げることができよう．

（I） Global PA-D.P.
（II） ISA-PA-D.P.
（III） ISA-RSA-PA-D.P.
（IV） Global FA-D.P.
（V） ISA-FA-D.P.
（VI） ISA-RSA-FA-D.P.

例えば，（III）は multi-points approximation の incremental relaxation D.P. の意味である．なお（I）と（IV）とに global を附したのは incremental と区別して original な D.P. を陽に示すためである．

4. 各逐次近似法における計算型と計算量の比較

D.P. のやり方は**第3図**の如く，小問題解決を累積して全問題の解決に至る．その個々の小問題を sub-optimization の問題と称することにする．そうすると各逐次計算法における計算の型と計算の量を比較すると**第1表**のようになる．おおよそ

第 1 表　計算方式の比較

方　式	sub-optimization の型	sub-optimization の回数
（I）Global PA（M-points）	M-points search technique	$M^n \cdot N$
（II）ISA-PA（M_1-points）	M_1-points search technique	$M_1{}^n \cdot N \cdot m_1$
（III）ISA-RSA-PA（M_2-points）	M_2-points search technique	$M_2 \cdot N \cdot n \cdot m_2$
（IV）Global FA	n 次元 optimization	$2(M')^n \cdot N$
（V）ISA-FA	n 次元 optimization	$2(M_1')^n \cdot N \cdot m_1'$
（VI）ISA-RSA-FA	1 次元 optimization	$2M_2' \cdot N \cdot n \cdot m_2'$

N=stage 数
n=貯水池数
M=Global PA 方式における，1貯水池当りの格子点の数
M_1=ISA-PA 方式における，〃
M_2=ISA-RSA-PA 方式における，〃
m_1=ISA-PA 方式における収斂までの1貯水池当りの iteration の回数
m_2=ISA-RSA-PA 方式における，〃
M'=Global FA 方式において必要な points 数
M_1'=ISA-FA 方式において〃
M_2'=ISA-RSA-FA 方式において〃
m_1'=ISA-FA 方式における，収斂までの1貯水池当りの iteration の回数
m_2'=ISA-RSA-FA 方式における，〃

$$M \geq M_1 \geq M_2$$
$$M' \geq M_1' > M_2'$$

である．

実際問題には貯水量，流量その他の制約があって，第1表の sub-optimization の回数はある程度減る傾向にある．その傾向は（I），（II），（III）の多点近似のグループでは（I）において最も著しく，（II），（III）の順で弱まる．同様に（IV），（V），（VI）の関数近似のグループでは，（IV）において sub-optimization の回数の，諸制約によって減る度合いが最も大きい．sub-optimization 問題の型についても，諸制約がきついほど一層簡単化する傾向がある．そういう意味において第1表は，本来の D.P.（global D.P.）に対してはやや苛酷な比較ではあるが，おおよその目安としては十分役立つと考える．

まず多点近似のグループ，（I），（II），（III）での比較を考えてみる．例えば Global PA において $M=10$ とし，貯水池の数を $n=5$ とすると，sub-optinization の回数は実に $10^5 \cdot N$ に及ぶ．ところで，一つの sub-optimization は実は M^n 個の状態のコスト計算と，それらの間の大小関係比較より成り立っている（M-points search technique）から，コスト計算の回数でいうと $M^{2n} \cdot N$ 回，すなわち，今の例では $10^{10} \cdot N$ 回行わねばならず，実用上ほとんど不可能に近い．多くの場合，経済的な運用問題は相当細いコスト・ゲインを問題にするのであるから，$M=10$ では不十分で，もっと大きな M を取る必要がある．このようなわけで，最もまともな D.P. すなわち（I）の global D.P. は高々2乃至3貯水池の系統の場合程度に適用しうるにすぎないであろう．

（II）の ISA-PA では M_1 を十分小さくえらぶことができる．しかし余り小さくえらぶと，収斂までの iteration の回数 m_1 がふえるので，最適な M_1 のえらび方（並びに points の間隔）がある筈である．例えば $M_1=3$ にえらべば，sub-optimization の回数は $3^5 \cdot N \cdot m_1$（ただし $n=5$ の場合）となり，m_1 がある程度大きくても十分実用になりうる．コスト計算の回数としては $3^{10} \cdot N \cdot m_1 \cong 20{,}000\, N \cdot m_1$ 回におよぶ．

（III）の ISA-RSA-PA では，例えば $M_2=3$ のとき，sub-optimization の回数としては $15 \cdot N \cdot m_2$ に激減する．m_2 はある程度大きいにしても m_1 と大差はなく，極めて実用的になる．コスト計算の回数でいっても $45 \cdot N \cdot m_2$ 回にすぎない．

以上から，貯水池の数がある程度大きいときには，sub-optimization の回数からも（Ⅲ），（Ⅱ），（Ⅰ）の順序でのぞましく：

$$M^n \cdot N \gg M_1^n \cdot N \cdot m_1 \gg M_2 \cdot N \cdot n \cdot m_2,$$

一つの sub-optimization 自身も，（Ⅲ），（Ⅱ），（Ⅰ）の順序に簡単である：

$$M^n \gg M_1^n \gg M_2$$

すなわち，二重の意味で（Ⅲ）がのぞましいことになる．ただ問題は，（Ⅱ），（Ⅲ）という近似がどの程度に正しいかであるがこれについては後にふれる．

ほとんど同様なことが，関数近似のグループ（Ⅳ），（Ⅴ），（Ⅵ）に対してもいえる．M', M_1', M_2' はもちろん M, M_1, M_2 より一般には小さくてすむから，その意味では多点近似より sub-optimization の回数は少い（ただし（Ⅴ），（Ⅵ）に移るにつれ（Ⅱ），（Ⅲ）と差がなくなる）．しかし一方，sub-optimization の型自身は複雑になり，（Ⅳ），（Ⅴ）では n 次元 optimization が必要になって，諸制限が伴なう場合には，かなり大掛りな non-linear programming の問題になって，一つの sub-optimization を行うこと自身が相当大変な仕事になる．このような場合には（Ⅵ）の ISA-RSA-FA が非常に意味をもってくる．

5. フロート・チャートの典型

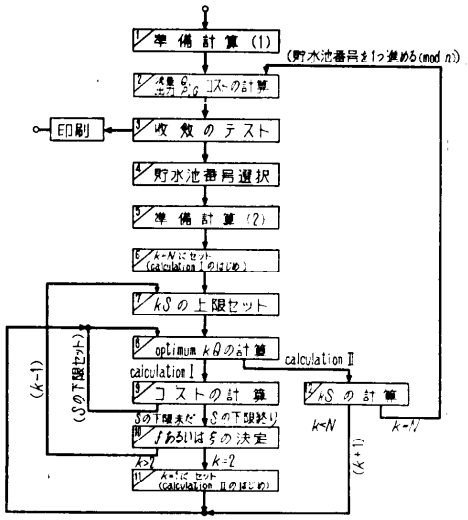
ISA-RSA-FA と ISA-RSA-PA の典型的なフロー・チャートを示すとそれぞれ**第5図**，**第6図**の如くである．ここで，ISA-RSA-FA としては特に最小コスト関数を線型関数近似した場合である．なお，貯水池より無駄に流す越流の効果は無視されているが，それに対しては容易に変えることができる．以下簡単にこれらを説明する．

(A) RSA-ISA-FA，Ⅰ次関数近似の場合（第5図）．

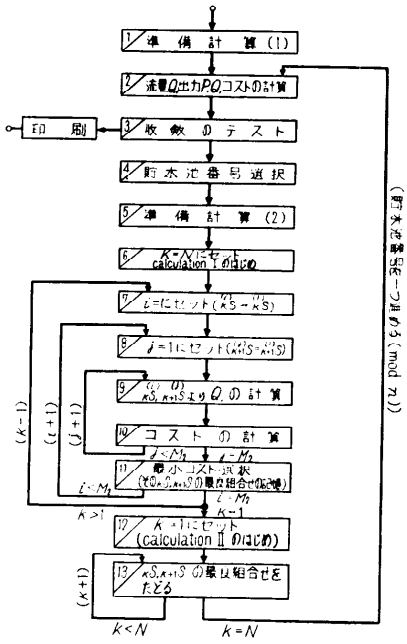
ステップ（2）よりはじめて同じステップ（2）にもどる大きなループを廻ってくると，一つの貯水池（発電所）についての1回の correction が完了する．そして次の順番の発電所の correction に進む．すべての発電所について一わたり correction が完了すると，再び最初の発電所の correction に入る．以下同様である．

ステップ（1）：

下流発電所の影響，制御不能分の差引など，一般的な環境条件の計算である．なお，初期想定状態としては，実行可能な（feasible；諸制限を満すもの）各貯水池の貯水量変化状況が与えられている．



第5図 ISA-RSA-FA のフロー・チャート



第6図 ISA-RSA-PA のフロー・チャート

ステップ（2）：

水力発電所，火力発電所，並びに総コストの，最新のインフォーメーションにもとづく計算．

ステップ（3）：

総コストの下降が，すべての貯水池につきおさまったかどうかによる収斂判定．我々の方法は単調減少性を有している．

ステップ（4）:

いま何番の発電所について計算するかを指定する．プログラム自身はどの発電所にも共通に用いられるものであるから，インフォーメーションの授受の場所，特性の差の指定などを行わねばならない．

ステップ（5）:

ある水力発電所のオペレーションの correction を考えている時には，他の水力発電所は知りうる限りの最も新しいオペレーションに固定して，それらを本来の負荷から差引いた等価負荷を当該水力発電所と火力系とが経済的に配分し合う形をとる．この等価負荷を計算することや，またRSA-ISAであるから，1回当りの correction を小さくおさえる必要があるが，そのため全期間にわたり変動をゆるされる貯水量変動範囲を求めたりする．

ステップ（6）:

ここから D.P. の計算に入る．D.P. の計算は，終りの時刻からはじめて順次前の時刻へ移る最小コスト関数 f の計算（これを calculation I と称す）および calculation I ですべての時刻の最小コスト関数がそろって，前の時刻から後の時刻へ順次移る最適運転の決定（これを calculation II と称す）とからなる．ステップ（6）で“$K=N$ にセット”とは，calculation I を終りの時刻からはじめるという指定である．

ステップ（7）:

一つの correction に際する，貯水量 S の許容変動幅（$\pm\Delta s$）を十分小さくえらぶと，貯水池運用の問題では，最小コスト関数 f は，その範囲ではほとんど S の線型関数に近づく（f の変化は一般にきわめておとなしい）．その S の二つの値を，S の許容変動範囲の上限および下限にとる．コスト最小化計算を，その上限からはじめることの指定がステップ（7）である．

ステップ（8）:

最小にすべきコストは ${}_KQ$ について相当複雑な関数になる．そこで我々は一種の区間縮小法によって ${}_KQ$ を決定することとした．その原則は incremental cost の符号組み合わせ判定による区間縮小過程のくりかえしである．

ステップ（10）:

最小コスト関数の傾斜 ${}_K\xi$ の決定．ステップ（8）ではコストそのものでなく，コストの ${}_KQ$ についての微分のみを用いるから，最小コスト関数そのものを求める必要はなく，その傾斜のみが知られれば十分である．

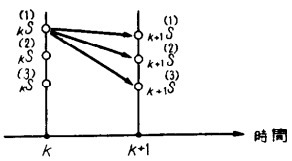
ステップ（11）:

ここから calculation II に入る．calculation I で，各時刻から最終時刻に至る最小コスト関数，あるいはその傾斜が出そろったから，初期時刻での初期条件から出発して順次，各時刻での最適状態を決定して行く．（11）→（8）→（12）→（8）→（12）→……と回る．

(B) RSA-ISA-PA の場合（第6図）．

ステップ（6）までは（A）と全く同じである．ステップ（6）よりはじまってステップ（11）までが calculation I，ステップ（12），（13）が calculation II を構成する．

たとえば 3-points 表現の場合（$M_2=3$）について説明する．3-points 近似であるから，各時刻での状態は高々三つに限られる（二つあるいは一つに reduce することもある）．たとえば第 K-時刻と第 $K+1$ 時刻



第 7 図

の関係をとりあげると**第7図**のとおりで，いま第 K-時刻での最小コスト関数を求めるとする．そうすると，${}_K\overset{(1)}{S}$ に対する最小コストは，$({}_K\overset{(1)}{S}, {}_{K+1}\overset{(1)}{S})$，$({}_K\overset{(1)}{S}, {}_{K+1}\overset{(2)}{S})$，$({}_K\overset{(1)}{S}, {}_{K+1}\overset{(3)}{S})$ の三つの組み合わせに対して行い（j をかえること），その中から最小のものをえらび出せばよい．同様に，${}_K\overset{(2)}{S}$ に対しては $({}_K\overset{(2)}{S}, {}_{K+1}\overset{(1)}{S})$，$({}_K\overset{(2)}{S}, {}_{K+1}\overset{(2)}{S})$，$({}_K\overset{(2)}{S}, {}_{K+1}\overset{(3)}{S})$ の組み合わせから（$i=2$），${}_K\overset{(3)}{S}$ に対しては $({}_K\overset{(3)}{S}, {}_{K+1}\overset{(1)}{S})$，$({}_K\overset{(3)}{S}, {}_{K+1}\overset{(2)}{S})$，$({}_K\overset{(3)}{S}, {}_{K+1}\overset{(3)}{S})$ の組み合わせから（$i=3$）それぞれ最小のものをえらび出すことになる．ただ，それらの組み合わせの中，流量 Q の制限によって不可能なものがありうるが，その場合には人為的に大きなコストになるとして，他の組み合わせと比較する型とすればよい．また，どの組み合わせが最適（最小コスト）かというインフォーメーションをとっておかなければならない．（一つの関数表現）．それがステップ（11）で行われている．

第 2 表

	方式	(1) 1次近似 σ (VI)に属す	(2) 2次近似 σ (VI)に属す	(3) 3-points σ (III)に属す	(4) 5-points σ (III)に属す	(5) 9-points σ (II)に属す	(6) 3-points $\bar{\sigma}$ (III)に属す	(7) 5-points $\bar{\sigma}$ (III)に属す	(8) 7-points $\bar{\sigma}$ (III)に属す	(9) 5-points $\bar{\sigma}$, loss 考慮 (III)に属す	(10) 1次近値 $\bar{\sigma}$, loss 考慮 (VI)に属す
	1 iteration 当りの所要時間	1 分	1分余	4 分	4分余	18 分	4分余	4分余	6 分	5 分	3 分
	word数 命令	480	550	500	550	580	530	580	600	630	670
	word数 数値	320	330	380	430	380	380	430	480	530	380
Δs 大	iteration 数 $2m_2$ (A-初期状態)	17	16	15	16	10	15	17	—	—	17
	収斂値 (A-初期状態)	百万円 4416	4417	4426	4421	4426	4437	4434	—	—	4702
	iteration 数 $2m_2$ (B-初期状態)	8	8	7	13	6	7	9	—	—	11
	収斂値 (B-初期状態)	4434	4434	4437	4435	4436	4436	4434	—	—	4702
Δs 小	iteration 数 $2m_2$ (A-初期状態)	59	—	55	—	—	52	39*	38*	40*	39
	収斂値 (A-初期状態)	4416	—	4429	—	—	4433	4428*	4428*	4697*	4702
	iteration 数 $2m_2$ (B-初期状態)	—	—	19	—	—	18	—	—	—	19
	収斂値 (B-初期状態)	—	—	4437	—	—	4433	—	—	—	4702

* ΔS 値が多少異る. $N=12$

ステップ(12)よりはじまる calculation II では，ステップ(11)で記憶されたインフォーメーションを頼りにして，初期条件から順次最適運転を決定して行く．

6. 逐次近似計算法の検討

各種方式の表面的な特徴と結果を示したのが**第2表**である．使用計算機は電気試験所 MARK-5 計算機(4,000 words ドラム・メモリー，平均5.2 ms access time)である．取り扱った問題は2貯水池よりなる系統である．

(1)より(5)までと，(6)以下とでは水力発電所の出力特性の表し方にやや差があり，さらに(9)，(10)では送電損失の影響をふくんでいる．以上の諸方式に対して，第2表では許容貯水量変動幅 ΔS をかなり大きく取った場合と，小さく取った場合につき，それぞれ2種類の初期想定状態 A, B から出発した場合の結果が示されている(A-初期状態よりも B-初期想定状態の方が最適状態に近い)．紙面の都合上，細いことは抜きにして次のような諸問題点について，得られた結果から極く簡単に説明しよう．ISA-D.P. あるいは ISA-RSA-D.P. で実際問題になることがらは

(1) 凸性条件(convexity)の問題

(2) 分割 stage 数

(3) ΔS の大きさのえらび方

(4) 貯水池の順序組み合わせの仕方(relaxation の型)

(5) 初期想定状態の取り方

(6) FA 方式と PA 方式の特徴とそれらの比較

などである．

(1) 凸性条件の問題:

前にのべたように，ISA のたて前からは，極小と最小とが一致しないと困る．貯水池運用の問題では領域の凸性はたもたれるが，評価関数は，落差変動によって水力発電所の出力が強く影響をうけると凸性が成り立たなくなる．離散的に考えた場合には，一つの stage での出力が，その stage の初期時刻の貯水量のみの影響をうけるとした場合に特に著しい．第2表における(1)より(5)の各方式はこのような特性の場合(第2表では σ で表わしている)で，このことが，初期想定状態 A と B とから出発した計算が，わずかではあるが異る所に収斂しているという結果にあらわれている．評価関数の凸性を強調するには，したがって，例えば落差変動の影響をゆるめるため，stage の初期および終期の貯水量の平均値が水力発電所出力に

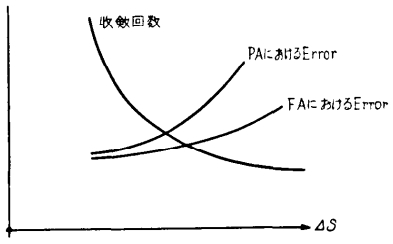
影響を与えるとすることで，（6）以下の方式ではそれが採用されており（第2表では $\bar{\sigma}$ で表わされている），初期想定状態がかなり変っても同一値に収斂している．さらに，水力発電所出力特性の凸性や送電損失を考慮すると，評価関数の凸性が強調され，実際計算も安定化することが示された．

（2）分割 stage 数:

第2表には示していないが，stage 数を12から36にふやして計算を行ったが，精度の問題を別とすれば，問題の本質的性質には差異はほとんどみられなかった．

（3）ΔS の大きさのえらび方:

一般には ΔS を小さく取る程，一層の cost reduction を期待できる．このことは第2表からも多少わかるように，PA 方式において特に著しく，FA では ΔS の大小は比較的ゆるやかな影響があるにすぎない．一方，収斂回数は ΔS がある程度以下に小さくなると急激に増大する．図示すると**第8図**の如くで，これから optimum な ΔS をえらぶことができようが，貯水池の容量，水力発電所の出力特性が効いてくる．



第 8 図

（4）貯水池の順序組み合わせの仕方

特に大きな ΔS をとらない限りは順次1回ずつ correction をくり返す方式で十分であり，したがって correction の順序は問題にならなかった．しかし扱った例題が2貯水池系であり，もっと多数の貯水池をふくむ（したがって変関数の多い）問題にはある程度能率のよい順序がある筈であり，また連立方程式解法における relaxation 法と同様，以上のような systematic relaxation 法の代りに，適当な目安による relaxation のやり方が考えられる．なお同じ貯水池を2回つづけて correction をくり返すということは RSA の立場からのぞましくなく，また RSA からはゆるされたとしても，それはほぼ，ΔS が適当でなかった（小さすぎた）ということに相当するから，一般にはのぞめない．

（5）初期想定状態の取り方:

もちろん，最適状態に近い初期想定状態から出発することがのぞましいのは当然である．しかしそれ以外に，特に PA 方式において，初期想定状態の取り方が格子点（とりうる状態点）の分布に影響を与えるという題題があり，これが収斂に影響を与え，また場合によって計算の不安定を生ずる原因になる．こういうわけで，上述の最適状態に“近い”初期想定状態とは，コスト的に近いばかりでなく，それよりはむしろパターン（あるいは変関数型）が最適状態のそれに“近い”ものであることが重要である．このことは多くの計算結果が示している．

（6）FA 方式と PA 方式の特徴とそれらの比較:

以上の諸結果，並びに第2表から次のことがいえる．

（a） PA 方式は相当多くの点をとるか，あるいは ΔS を小さく取らないと精度の向上が期待できず，また1回の correction 当りの時間がややかかる．したがって全体として計算時間は長びく．しかし，収斂はきっぱりと行われる．

（b） FA 方式はある程度大きな ΔS（もちろんこれは RSA が許す限度内で）でもかなり良好な cost reduction を行う．そして1回 correction 当りの speed も早い．しかし，収斂に十分近づいたところで，一般には多少の fluctuation を生じやすいので之に対する考慮が必要である．また近似関数として，高次多項式になるほどのぞましいと一般には考え勝ちであるが，計算の安定性などの考慮からは必ずしもそうではない．我々の例では，1次関数近似の方が2次関数近似より安定していた．それは，非常に細い，しかもわずかに凸性をもつという状態の場合に，2次関数近似で逆に凹性としてしまう可能性があるからである．

参 考 文 献

1) R. Bellman: Dynamic Programmin, Princegton U.P. (1957)
2) R. Bellman: Adaptive Control Process; A Guided Tour, Princeton U.P. (1961)
3) T.C. Koopmans: Water Storage Policy in a Simplified Hydroelectric System, Cowles Foundation Paper No. 115 (1958)
4) 深尾，山崎，木村：電力系統経済運用への動的計画法の応用，電気学会誌，79巻845号（1959年2月）
5) 深尾，比留間：電気四学会連合大会予稿（1961年4月）
6) 深尾：貯水池運用問題への DP の応用，オペレーションズ・リサーチ5巻1号，（1960）