

微分動的計画法に基づく貯水池群操作量の数値計算法

中村 修¹・迫 秋一²・河村 明³・小松利光⁴

¹正会員 西日本技術開発(株) 開発部システム担当部長(〒810 福岡市中央区渡辺通一丁目1-1)

²正会員 元西日本技術開発(株) 土木本部長(現昭和地下工業(株) 顧問)

³正会員 工博 九州大学助教授 工学部建設都市工学科

⁴正会員 工博 九州大学教授 工学部建設都市工学科

本稿は、貯水池群操作において流入量などが確定的に想定される基本的問題に対して、微分動的計画法に基づく2次の洪水被害関数を主要コスト関数とする貯水池群最適操作量の数値計算法を開発し、それについて述べたものである。この方法では、一次元探索付Newton法による微分動的計画法と特異制御問題に対処するためのペナルティ関数法を組合わせて、比較的多期にわたる貯水池群最適操作量決定の数値計算法を構成している。

Key Words : *optimal operation of reservoirs, DDP, computational method of optimal control problem*

1. はじめに

水資源の確保とその有効利用に関しては、ダム貯水池の効率的な運用法の策定が重要な役割をはたしている¹⁾。

本報告は、利水を目的とする貯水池群の合理的運用法策定に資するため、最適制御問題の有力な解法である微分動的計画法²⁾(DDP)に基づく、比較的多期にわたる貯水池群操作量の新しい数値計算法について述べたものである。

貯水池群の管理手法については、既に多くの研究が発表されている。そのなかで、本稿作成に当って参照したものを記すと、高樟らのDP論的研究³⁾、竹内のDCL法⁴⁾、吉川らの洪水持続曲線の性質とその応用^{5),6)}、S. J. JakowitzらのCDDP⁷⁾(Constrained DDP)及びW. W. G. Yehのレビュー論文⁸⁾などである。更に、ダム操作の諸側面に関する池淵の解説論文⁹⁾より重要な示唆を得た。なお、本稿の直接的な検討対象ではないが、確率的手法に関する論文も少なくない。

貯水池は、主としてあらかじめ策定された運用法に従って操作される。この運用法の策定に際しては、過去の長期間にわたる実績貯水池流入量曲線、或いはこれを統計的に処理して得た時間-流入量曲線、及び適切な手法¹⁾で推定された水の時間-需要量曲線を確定的なものとし、みなして貯水池操作量の最適化を行い、その結果より運用法の策定を行う場合が多い。また、数種の流入量・需要量曲線を想定してシミュレーションを行い、その結果より運用法あるいは操作量の決定を行うことも考えら

れる。本稿では、このように流入量などが確定的に想定された貯水池群のモデルに対し、操作量決定の数値計算法について検討した結果を報告する。

貯水池の状態の推移は、離散動的システム¹⁰⁾として表すことが多く、本稿も貯水池の状態方式を時間について離散的とした。

本稿の貯水池群操作量の計算法では、まず最初に、K. Ohnoが文献11)において状態制約条件付最適制御問題の数値計算法として提案したNewton法によるDDPを軽微な補足を加えた上で、一次元探索付Newton法¹²⁾によるDDP(DDP with Damped Newton's Method、以下ではDN-DDPと略称する)の形で使用して、収束可能な出発値の範囲拡大を図った。次いで、D. J. Bell & D. H. Jacobson¹³⁾の ϵ -アルゴリズムと類似の手法をこれと組合せて特異制御問題に対処し、実用的な数値計算法を工夫・構成した。本稿では、便宜上、この計算法をSDN-DDP(Sequential Regularized DDP with Damped Newton's Method; 逐次正則化DN-DDP)と略称する。

従って、本稿のSDN-DDPは、前述のMurray & YakowitzらのCDDPと同じくDDPに基づくものであるが、制約条件の処理などにはこれと異なった方法を採用して、計算の収束性をより頑健にすると共に、コスト関数に、より現実的な洪水被害関数を採用し、この結果として生じる操作量の特異制御問題に対処する機能を備えたものである。SDN-DDPにより解析可能となるモデルの範囲は広く、下流域の比較的簡単な水路網も対象とすることが可能で、実質上のコスト関数は殆んど水

路網中の需要点の水需要量のみで構成できる。よって、特殊な場合を除けば、コスト関数の一部を下流域の各水路の流量に直接担当させる必要はない。

次に、既存の計算法として、坂和の方法¹⁴⁾、及び前記 Yakowitz らの CDDP をとり、本稿の需給モデルに適用できるように若干の変更・拡張を施して、本稿の方法との比較を行った。

2. 貯水池群の方程式とコスト関数

貯水池群の最適運用計算において、下流域の水需要量のみからコスト関数を構成すると、貯水池群の操作量は、後述のように特異制御となる可能性があり、これを求める計算が著しく難しくなることがある。この難点を解消するため、以下では安定化関数と呼ぶ、ペナルティ関数を導入して本来のコスト関数に加えることとする。

(1) 貯水池群の方程式とその最適制御問題

貯水池群の方程式は、通常モデルと同じく、第 n 期での状態変数として貯水量 x_n を、操作量としては貯水池からの取水量もしくは放流量 u_n をとることとする。そして、これらの関係が次のように表されるものとする。

$$\left. \begin{aligned} x_{n+1} &= x_n + Bu_n \Delta t + q_n \Delta t \\ &= f_n(x_n, u_n, q_n, \Delta t), n=0, 1, \dots, N-1 \end{aligned} \right\} \quad (1)$$

$$x_0 = a_0 \quad (2)$$

$$x_N = a_N \quad (3)$$

ここに、

$$x_n \triangleq (x_{n1}, x_{n2}, \dots, x_{nm})^T$$

：貯水池群の n 期首の貯水量で状態変数である。なお、 m は貯水池の数であり、上付添字 T は転置を表す。

$u_n \triangleq (u_{n1}, u_{n2}, \dots, u_{nr})^T$ ：貯水池群の n 期の操作量で取水量もしくは放流量を表す。なお、 r は操作量の数であり、一つの貯水池は最小 1 個の操作量を持つ。 $m \leq r$ である。

a_0 ：初期状態すなわち初期貯水量

q_n ：貯水池への n 期の河川流入量

Δt ：時間間隔，単位期間長

n ：時点 $t_n = n\Delta t$ を示す整数で期数，段数

N ： n の最大値，最終時刻を示す整数

B ： x_{n+1} と u_n の関係を表す行列 ($m \times r$ 行列)

a_N ：指定最終状態すなわち最終貯水量

とする。

x_n, u_n に関する制約条件式は、 x_n, u_n に関する一次式のみを考え、制約条件式は必ず u_n の成分を含むものとする。例えば、 $x_{n+1,i} \leq x_{i \max}, u_{nj} \geq 0$ ，などは、

$$g_{n,s}(x_n, u_n) = f_{ni}(x_n, u_n) - x_{i \max} \leq 0$$

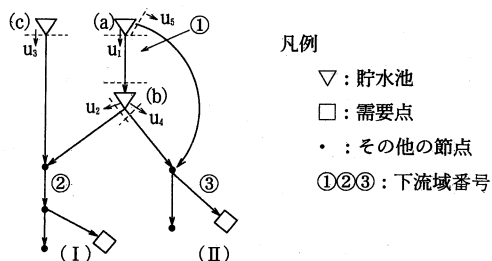


図-1 例題モデル図

$$g_{n,s+1}(x_n, u_n) = -u_{nj} \leq 0$$

と表す。等式制約条件の使用頻度は少ないが、同様に表すことが出来る。これらをまとめて次のように書く。

$$\left. \begin{aligned} g_n(x_n, u_n) &\leq 0, h_n(x_n, u_n) = 0 \\ n &= 0, 1, \dots, N-1 \end{aligned} \right\} \quad (4)$$

ここで、 g_n, h_n はそれぞれ m_n, r_n 次元ベクトルである。

終端制約条件は、 $x_N = f_{N-1}(x_{N-1}, u_{N-1})$ の固定端条件のみを取扱い、式 (3) で与える。

本稿では、式 (1)~式 (4) の 4 式を充たし、次項で述べるコスト関数 $J(l)$ を、所与の l の下で、最小化する $\{x_n^*, u_n^*\}^l$ を求める最適制御問題を、問題 (A) と呼ぶこととする。但し、 l は後述の逐次正則化過程の番号である。また、 $*$ は最適値を示すものとする。

ペナルティ関数の最終パラメータは、事前には明らかでないので、 $l=0, 1, 2, \dots$ と、各 l につき問題 (A) を解くとともにペナルティ関数のパラメータを更新し、 $J(l)^*$ を l について収束させて最終的な最適操作量 $\{u_n^*\}$ を求めることとする。なお、最終的な最適解 $\{x_n^*, u_n^*\}$ を求める問題を本稿では問題 (T) と呼ぶ。

(2) コスト関数と安定化関数

現実に即した貯水池運用の基準、すなわち操作量のコスト関数 (評価関数) を簡単に表すことは、非常に難しい問題である。

本稿では、文献 1), 15) などを参照して、コスト関数の主要成分は、各期 n の、各需要点の節水率の 2 乗に重み係数と需要量を掛けて集計した関数 L_{170} を、対象期間について累加した渇水被害関数とする。なお、ここで節水率とは、(不足水量 ÷ 需要量) × 100% のことであるが、本稿では、計算の便宜上 × 100% を落した節水比を代わりに用い、同時に重み係数を補正して渇水被害関数と同じ値を示す関数を採用して、同じ名称で呼ぶこととする。

L_{170} を具体的に求めるために、図-1 のような水の需要点と供給点 (貯水池) の関係図を考える。

貯水池地点、河川 (水路) の始点、終点、合分流点および需要点を節点とし、その間を流向の持つ水路 (有向

辺)で結ぶと有向グラフとなる。これを本稿では需給関係図と呼ぶ。

次いで、貯水池節点で水路間の連絡を切断すると、部分グラフが得られる。これを下流域と呼ぶこととする。

図-1のモデル図では、需給関係は3個の下流域より成り立っている。

このようにして、 u_n の第 j 成分 u_{nj} 、需要点 P での需要量 D_{nP} を各下流域に配分し、 u_{nj} は一下流域の水路始節点への流入量として取扱う。

各需要節点へ流出する水路は1本に限るものとし、この水路を需要水路 ld と呼ぶこととする。その流量 Q_{ld} は、需要量 D_{nP} に対する供給量とみなせる。

本稿の問題では、操作量 u_n を与えても、下流域の流量配分を定めなければ、前述の L_{1n0} が陽には求まらないことがある。しかし、通常の水路網については、与えられた u_n のもとで、 L_{1n0} を最小とする流量配分は常に可能である。但し、流量配分計算の都合上、水路の送水コストを必要とする場合もある。

以上より、 n 期の主コスト関数 L_{1n} を次のように定める。なお、本稿の計算法の目的は、この L_{1n} を対象期間について累加した渇水被害関数 J_1 の最小化を図ることである。

$$L_{1n}(u_n) = \min \left\{ \sum_R \left\{ \sum_P C_P D_{nP} (\Delta Q_{nP} / D_{nP})^2 + \sum_l C_l Q_l^2 \right\}_R \right\}_n \quad (5)$$

ここに、

- R : 下流域の番号
- L_{1n} : n 期の主コスト関数、評価関数
- C_P : 需要点 P における渇水被害関数の係数
- D_{nP} : 下流域 R の需要点 P における n 期の水需要量
- Q_l : 下流域 R の水路 l の n 期の流量
- C_l : 下流域の水路 l の流量に送水コストを付けた場合の係数。通常は0とする。
- $\Delta Q_{nP} \triangleq Q_{ld} - D_{nP}$: 需要点 P へ流出する需要水路 ld の n 期の不足流量

などである。

更に、洪水時などにおける余剰流量の処理のため、需給関係図には、適当な無効放流水路を設けることとする。ここで、需給関係図作成上の留意を若干述べる。

- ① 需給関係図は、もし可能ならば等価な変換を行って、なるべく単純化する。
- ② 一つの貯水池から、同一流域への操作量 u_{nj} は1個とする。更に、消去できる操作量は予め消去する。また、貯水池 i へ流入する河川は水路とはせず、その流量を q_{ni} で表す。
- ③ 上流貯水池より下流貯水池への流下量 u_{nj} は、もし需要点への分流がなければ、 q_{ni} および他の貯水池 e からの流下量 u_{ne} とは別個に取扱う。

④ ある節点 a より別の節点 b への経路が複数のときは、送水コストを設けて問題が不定となることを避ける。

⑤ 送水コストを設けても、 $(L_{1n})_R$ について、 $\{Q_l\}_R$ が一意に定まらない水路網は取扱わない。

Newton法においては、仮定した u_n を更新する場合に、 L_{1n} の値とともに $\nabla_u L_{1n}$ 、 $\nabla_u^2 L_{1n}$ が必要である。ただし、

$$\nabla_u L_n = (\partial L_n / \partial u_{n1}, \dots, \partial L_n / \partial u_{nr}),$$

$$\nabla_u^2 L_n = (\partial^2 L_n / \partial u_{ni} \partial u_{nj}), \text{ とする。}$$

u_n の一つの成分 u_{nj} は、複数下流域にわたって供給されることはないから、式(5)の \min と \sum が交換できる。従って、 L_{1n} とその微係数などは、下流域別にその成分 L_{1nR} などを求めて、これらを合成すれば得られる。なお、 L_{1n} と直接には関係しない下流域もあることに一応注意する。

L_{1nR} などは、与えられた $(u_n)_R$ に対し、形状が簡単な下流域においては直接これらを求め、下流域の形状がやや複雑なときは2次計画法によって求めることとする。

後者の場合には、水路の流量 Q_l などを変数にとり、 $(u_n)_R$ などの流入節点、水路の始点、合分流点における流量連続の式を制約条件式として、下流域の関数 L_{1nR} の最小化を行い、必要な諸量を求める。制約条件式は、例えば、

$$\left. \begin{array}{l} u_{nj} \text{の流入節点; } Q_l = u_{nj}, \\ \text{合分流節点; } \sum \pm Q_l = 0 \end{array} \right\} \quad (6)$$

などと表される。

本稿では、2次計画法の解法として、よく用いられていること、 $\partial L_{1nR} / \partial (u_n)_R$ 、 $\partial^2 L_{1nR} / \partial (u_n)_R^2$ が求め易いことなどから、Wolfeの方法¹⁶⁾(short-form)を採用する。この方法は、Kuhn-Tuckerの必要性の条件を直接使用し、求まる解のうち制約条件式のLagrange乗数 λ の値が含まれており、これが制約条件式の右辺による L_{1nR} の負の微係数となっている¹⁶⁾。また、 u_{nj} 流入節点の制約条件式の右辺 u_{nj} に摂動項を並置して掃き出しの範囲をここまで拡げて解けば、 $\partial^2 L_{1nR} / \partial (u_n)_R^2$ も解と同時に求まる。これらの結果より必要な諸量を合成する。

なお、Wolfeの方法が解けるためには、 $(u_n)_R$ に対し L_{1nR} を最小とする $\{Q_l\}_R$ が一意に定まるように、下流域の水路系が構成されていなければならないが、その判定は容易と思われる。

並列貯水池が在る場合や洪水時などでは、 u_n による L_{1n} の2次微分の逆行列が存在しないことがある。従って、 $L_{1n}(u_n)$ のみで問題(A)のコスト関数を構成すると、最適制御問題におけるHamilton関数 H_n ^{14),17)}にも同様な問題が起る場合がある。このとき、操作量は特異制御

となり、最適操作量は一意には定まらない。このような特異制御問題においても最適解は存在するが、解を求める手続きは複雑となる。数値計算による特異制御問題の解法として、Bell-Jacobson¹³⁾は、問題 (A) に即して云えば、本稿で ε -関数と呼ぶペナルティ関数、 $(1/2)\varepsilon_1 u_n^T u_n$ を L_{1n} に加えてコスト関数を構成して問題を解き、逐次、 $\varepsilon_l \rightarrow 0$ とする反復計算法を提案している。更に、ペナルティ関数は、 x_n と u_n の関数でもよいことを示唆している。

一方では、従来より貯水池群の運用計算に対し、空き容量を一つの指標とする方法、並びにルール・カーブ法がよく用いられてきた¹⁾。

これらを総合的に参考にして、本稿では、パイロット軌道と称する貯水池の基準の容量推移曲線 $\{X_n(PT)\}^l$ を設定し、これと計算容量 $\{x_n\}^l$ との差の 2 乗に比例する付加関数を導入する。更に、 ε_l を小さくする代りに、パイロット軌道を逐次更新してこれに対する最適解を求めて、最終的な最適解を得る方法を採用する。すなわち、まず各貯水池 d ごとに、

$$L_{2nd}^l(x_n, u_n) = \varepsilon_{nd} (f_{nd}(x_n, u_n) - X_{n+1}(PT)_d^l)^2 \quad (7)$$

で表される L_{2nd}^l を定める。

ここに、 l : 問題 (A) に対するパイロット軌道の更新回数、 $l=0, 1, \dots$, である。

次いで、需給関係図に一つの貯水池より複数流域へ供給する水路が含まれている場合は、その貯水池 d において、余水吐け放流水と同じ水路を通る操作量 u_{ndj} 以外の操作量 u_{ndk} に対して、前述の ε -関数を追加する。すなわち、

$$L_{2n}^l = \sum_d \{L_{2nd}^l + \sum_k \frac{1}{2} \varepsilon_{dk} u_{ndk} u_{ndk}\} \quad (8)$$

を付加関数とし、これを安定化関数または正則化関数と呼ぶこととする。次に、 L_{2n}^l を L_{1n} に加えて n 期のコスト関数を、

$$L_n^l = L_{1n} + L_{2n}^l \quad (9)$$

と定める。 L_n^l は明らかに、各 x_n, u_n に関する狭義凸関数である。これにより問題 (A) のコスト関数を、

$$J(l) = \sum_{n=0}^{N-1} L_n^l \quad (10)$$

と定義する。よって、 $J(l)$ は、各 x_n, u_n の狭義凸関数であり、また変数分離可能である。なお、簡単のため、 l は適宜省略するか添え字で表す。

初期のパイロット軌道 $\{X_n(PT)\}^0$ は、4 章 (4) 節で述べるようになるべく高水位の最適解 $\{x_n^*, u_n^*\}$ が得られるように設定する。但し、安定化関数を導入して解を得ることは、本来は無数にあるかも知れない最適解の中から有用と思われる一つの解を選出することに相当する。この解の特性およびこれに関連する事項については、4 章 (8) 節で述べる。

$\varepsilon_{dk} \neq 0$ となる場合は、最適解が僅かに歪む可能性はあるが、解析の必要上、これを許容することとする。

本稿の数値計算においては、本来の $\nabla_u L_{2n}$ 、 $\nabla_u^2 L_{2n}$ に次のような補正を行い、同じ記号で表して、以後これを使用する。このとき、 $\nabla_u^2 L_{2n}$ は適合近似 (consistent approximation¹⁸⁾) と考えられる。

$$\nabla_u L_{2n} = (\text{本来の } \nabla_u L_{2n}) \times s \quad (11)$$

$$\nabla_u^2 L_{2n} = (\text{本来の } \nabla_u^2 L_{2n} \text{ の対角項はそのままとし、それ以外の要素に } 0.5 \text{ を乗じた行列}) \times s^2 \quad (12)$$

ここに、 s は補正係数で、これについては後述する。

なお、本稿では、以上のように定める計算モデルを需給モデルと呼ぶ。もし、需給関係図に直列貯水池群が含まれるときなどでは、本論の方法とスペースルール¹⁾を併用する計算法も考えられるであろう。

3. 準備事項

(1) 記号等について

- ① 普通のベクトルは、列ベクトルで表す。
- ② 最適解に対する変数などは、* を付けて表す。
- ③ s 次元ベクトル a について、

$$\|a\|_p \triangleq (|a_1|^p + |a_2|^p + \dots + |a_s|^p)^{1/p}$$

とし、 $p=2$ のときは、添字を省略する。

- ④ その他の記号は、慣用のものを使用する。

(2) 実行可能操作量の仮定

実在の貯水池では、操作量 $u(t)$ は常に存在しているので、これをモデル化した貯水池の状態方程式 (1) の操作量 u_n は、初期条件、終端条件、計算対象期間および諸制約条件を無理なく設定すれば、 $n=0, 1, \dots, N-1$ について実現可能と仮定できる。本稿では、問題 (A) の実行可能解の存在を仮定する。

(3) 最適制御問題 (A) の解の性質について

各 l について、問題 (A) は、これと等価な数理計画法の問題に変換できて、そこで得られている諸結果、実行可能解の仮定およびコスト関数の狭義凸性から、大域的な一意の最適解の存在が云える^{19), 20)}。

従って、局所最適解を求める Newton 法などの数値解が収束すれば、その点が求める解である。

(4) 動的計画法 (DP) と利得関数^{2), 11)}

各 l について、問題 (A) は、状態方程式とコスト関数の性質より、DP が適用できる。よって、利得関数 $V_n(x_n)$ を $n=0, 1, \dots, N$ に対し、

$$V_n(x_n) = \min \{ \sum_{j=n}^{N-1} L_j + L_N(x_N) \mid x_{j+1} = f(x_j, u_j), g_j \leq 0, h_j = 0, j = n, \dots, N-1 \} \quad (13)$$

と定めると、

$$V_n(x_n) = \min \{L_n(x_n, u_n) + V_{n+1}(f_n(x_n, u_n)) | g_n \leq 0, h_n = 0\}, n=0, 1, \dots, N-1 \quad (14)$$

となる。更に、問題 (A) では終端条件式 (3) を変形し、

$$V_N(x_N) = L_N(x_N) = 0 \quad (15)$$

とすることが出来る。

定義より、各 l に対し、 $J(l)^* = V_0(x_0)$ である。周知のように、DP では、式 (15) より出発して式 (14) を再帰的に解いて最適解 $\{u_n^*\}$ を求める。一方、DDP では、次元の障害を緩和するため、式 (14)、(15) より求めた最適性の条件式を反復計算法で解く。

(5) 最適性の必要条件¹¹⁾など

最適制御の解析では、関数の性質などについて若干の仮定を設けるのが普通である。これを簡単に記す。

仮定 1 f_n, g_n, h_n は x_n, u_n について 2 階の導関数まで連続とする。 L_n は、 x_n について 2 階の導関数まで連続であり、 u_n については 1 階の導関数が連続で、2 階の導関数は区分的連続 (定数) である。 L_n は x_n, u_n について狭義凸とする。

仮定 2 活性化した g_n^* と h_n^* は、各 $\nabla_u g_{ni}^*, \nabla_u h_{nj}^*$ が一次独立である。なお、 g_n は活性化のもののみを考える。

仮定 3 g_n^* には強相補性¹⁴⁾が成り立つ。

以上の仮定は、問題 (A) の新しい制約とはならない。

次に、最適解 $\{u_n^*\}$ の計算に必要な条件式について記す。各 l ごとに、時点 $n(n=0, \dots, N-1)$ における式 (14) の $V_n(x_n)$ 最小化問題に対する Lagrange 関数¹⁴⁾ F_n は、次のように定まる。なお、 F_n 導入の目的は、各期 n での最適性の条件式を求めるためである。

$$F_n(x_n, u_n, \lambda_n, \mu_n) = L_n(x_n, u_n) + V_{n+1}(f_n(x_n, u_n)) + \lambda_n^T g_n(x_n, u_n) + \mu_n^T h_n(x_n, u_n) \quad (16)$$

仮定 2 より、2 次の制約想定が成り立つ²⁰⁾。

よって、 u_n^* が式 (14) の解となるためには、2 次の最適性条件を充たすことが必要である¹¹⁾。すなわち、Kuhn-Tucker の条件式 (17)、(18)、(19) を充たす λ_n^*, μ_n^* があって、式 (20) が成り立つことが必要である。

$$\nabla_u F_n^* = \nabla_u L_n^* + \nabla V_{n+1}^* \nabla_u f_n^* + (\lambda_n^*)^T \nabla_u g_n^* + (\mu_n^*)^T \nabla_u h_n^* = 0 \quad (17)$$

$$\text{diag}(\lambda_n^*) g_n^* = 0, h_n^* = 0 \quad (18)$$

$$g_n^* \leq 0, \lambda_n^* \geq 0 \quad (19)$$

活性化 g_{ni}^* に対し $\nabla_u g_{ni}^* Z = 0$ 、更に $\nabla_u h_{nj}^* Z = 0$ となるベクトル Z に対し、

$$Z^T \nabla_u^2 F_n^* Z \geq 0 \quad (20)$$

ここに、

$$\nabla_u^2 F_n^* = \nabla_u^2 L_n^* + \nabla V_{n+1}^* \nabla_u^2 f_n^* + (\lambda_n^*)^T \nabla_u^2 g_n^*$$

$$+ (\mu_n^*)^T \nabla_u^2 h_n^* + (\nabla_u f_n^*)^T \nabla^2 V_{n+1}^* \nabla_u f_n^* \quad (21)$$

である。

本稿の問題 (A) における L_n, g_n, h_n の性質、(2) 節、(3) 節、仮定 1~3 などより、 λ_n^*, μ_n^* の存在と式 (17)~(20) の成立が云えて、更に F_n^* の性質より、

$$Z^T \nabla_u^2 F_n^* Z > 0 \quad (22)$$

が成り立つ。式 (17)、(18)、(19)、(22) は、 u_n^* が式 (14) の最適解となる必要充分条件である。

いま、 $n=0, 1, \dots, N-1$ に対して、

$$y_n = (u_n^T, \lambda_n^T, \mu_n^T)^T \quad (23)$$

$$T_n(x_n, y_n) = (\nabla_u F_n, g_n^T \text{diag}(\lambda_n), h_n^T)^T \quad (24)$$

とおくと、式 (17)、(18) は、

$$T_n(x_n^*, y_n^*) = 0, n=0, 1, \dots, N-1 \quad (25)$$

と表される。終端条件を充たし、式 (25) を充たす $\{u_n^*\}$ が問題 (A) の最適解である。

もし、 $T_n(x_n^*, y_n)$ の y_n に関する Jacobi 行列 $J_n(x_n^*, y_n)$ が正則なら、 y_n^* は、 $T_n(x_n^*, y_n) = 0$ より求まる。

$J_n(x_n, y_n)$ は、次のように表される。

$$J_n = \begin{bmatrix} \nabla_u^2 F_n & \nabla_u g_n^T & \nabla_u h_n^T \\ \text{diag}(\lambda_n) \nabla_u g_n & \text{diag}(g_n) & 0 \\ \nabla_u h_n & 0 & 0 \end{bmatrix} \quad (26)$$

仮定 2 と式 (22) より、 J_n^* の正則性が得られる。また、 $T_n(x_n, y_n)$ の x_n に関する Jacobi 行列 $K_n(x_n, y_n)$ は、次のように表される。

$$K_n(x_n, y_n) = (\nabla_{xx}^2 F_n^T, \nabla_x g_n^T \text{diag}(\lambda_n), \nabla_x h_n^T)^T \quad (27)$$

ここに、

$$\nabla_{xx}^2 F_n = \nabla_{xx}^2 L_n + \nabla_u f_n^T \nabla^2 V_{n+1} \nabla_x f_n + \nabla V_{n+1} \nabla_{xx}^2 f_n + \lambda_n^T \nabla_{xx}^2 g_n + \mu_n^T \nabla_{xx}^2 h_n \quad (28)$$

である。次いで、陰関数定理¹⁸⁾より、

$$\nabla_x y_n^*(x_n) = (\nabla_x u_n^*(x_n))^T, \nabla_x \lambda_n^*(x_n)^T, \nabla_x \mu_n^*(x_n)^T)^T = -J_n^{-1}(x_n, y_n^*(x_n)) K_n(x_n, y_n^*(x_n)) \quad (29)$$

となる。

強相補性より、

$$g_n^T(x_n, u_n^*(x_n)) \nabla \lambda_n^*(x_n) = 0 \quad (30)$$

である。よって、 $V_n(x_n) = L_n$ を考慮すれば、 $n=0, 1, \dots, N-1$ に対して次式がなりたつ¹¹⁾。

$$V_n(x_n) = F_n(x_n, y_n^*(x_n)) \quad (31)$$

$$\begin{aligned} \nabla V_n(x_n) &= \nabla_x F_n(x_n, y_n^*(x_n)) \\ &= \nabla_x L_n + \nabla V_{n+1} \nabla_x f_n + \lambda_n^*(x_n)^T \nabla_x g_n \\ &\quad + \mu_n^*(x_n)^T \nabla_x h_n \end{aligned} \quad (32)$$

$$\begin{aligned} \nabla^2 V_n(x_n) &= \nabla_{xx}^2 L_n + \nabla_x f_n^T \nabla^2 V_{n+1} \nabla_x f_n \\ &\quad + \nabla V_{n+1} \nabla_{xx}^2 f_n + \lambda_n^*(x_n)^T \nabla_{xx}^2 g_n + \mu_n^*(x_n)^T \nabla_{xx}^2 h_n \\ &\quad + (\nabla_{xu}^2 L_n + \nabla_x f_n^T \nabla^2 V_{n+1} \nabla_u f_n \\ &\quad + \nabla V_{n+1} \nabla_{xu}^2 f_n + \lambda_n^*(x_n)^T \nabla_{xu}^2 g_n \end{aligned}$$

$$\begin{aligned}
& + \mu_n^*(x_n)^T \nabla_{x_u}^2 h_n \nabla_{x_u} \mu_n^*(x_n) + \nabla_{x_g} g_n^T \nabla_{x_n} \lambda_n^*(x_n) \\
& + \nabla_{x_h} h_n^T \nabla_{x_u} \mu_n^*(x_n) \quad (33)
\end{aligned}$$

上の式 (33) の右辺には、本稿の問題 (A) に対しては零となる項が若干含まれている。すなわち、状態方程式、制約条件式が x_n , u_n について一次式であるから、式 (33) の各項のうち次の因数をもつ項は 0 となる。すなわち、 $\nabla_{x_f}^2 f_n$, $\nabla_{x_g}^2 g_n$, $\nabla_{x_h}^2 h_n$, $\nabla_{x_u}^2 f_n$, $\nabla_{x_u}^2 g_n$, $\nabla_{x_u}^2 h_n$ を因数に持つ項が消える。その結果 $\nabla^2 V_n$ の数値計算式が少し簡単となる。

4. 貯水池群最適操作量の数値計算法

本稿における貯水池群最適操作量の数値計算法は、二重の反復計算過程よりなっている。第 1 のものは、Bell-Jacobson の ε -アルゴリズムに類似した、特異制御問題に対処するために導入した安定化関数のパイロット軌道 $\{X_n(PT)\}^l$ を更新し、第 l 次のコスト関数 $J(l)$ を再構成する計算過程である。計算手順ブロック・チャートの外側ループに相当するもので、これを逐次正則化過程 (Sequential Regularizing Process) と呼ぶこととする。

第 2 のものは、各番号 l のパイロット軌道に対する問題 (A) の DN-DDP による反復計算過程で、 $J(l)$ に対する最適操作量 $\{u_n^*\}^l$ を求めるものである。これを DN-DDP 過程 (DN-DDP Process) または、単に DN-DDP と呼ぶ。なお、 l 次の両計算過程を合せて l 次の逐次計算過程と呼ぶこととする。

既述のように、Ohno は、状態制約条件を持つ離散動的システムの最適制御問題を解く DDP のアルゴリズムとして、Newton 法によるものを提案し、その収束証明と R -収束率を示した。

Newton 法による DDP では、変数の出発値が或る程度最適値に近いことが前提となっている。本稿では、この方法を、一次元探索付 Newton 法による DDP、即ち DN-DDP として使用し、更に狭義凸のコスト関数を採用することにより、数値計算の安定を保つとともに出発値の収束範囲を拡げ、問題 (A) に対する有効な数値計算法とした。

以下では、これと逐次正則化過程をあわせたアルゴリズム (SDN-DDP) を示すとともに、その収束性と計算最適解の特性について簡単に述べる。次いでこの方法による貯水池群最適操作量の数値計算例を示す。

(1) DN-DDP についての補注

前述の様に、問題 (A) の最適解 $\{u_n^*\}^l$ が満たすべき必要充分条件は、式 (19), (25) と境界条件である。これを反復計算で解くに当たって、DDP では修正量 δy_n^k を、

$$\delta y_n^k = \delta y_{n0}^k + \partial y_n^k / \partial x_n \delta x_n^k \quad (34)$$

の形で求めることが多く^{11), 12)}、本稿もこれに従う。 δy_{n0}^k は、Newton 法などで得られる修正量で、逆時間方向に求め、一方、 δx_n^k , δy_n^k は順時間方向に求める。ここに、 k は反復の回数を表す整数である。

本稿の方法では、通常の Newton 法にステップサイズ α を持ち込んで δy_{n0}^k を、

$$\delta y_{n0}^k = -\alpha J_n^{-1}(x_n^k, y_n^k) T_n(x_n^k, y_n^k)$$

で求めるが、 J_n , T_n などには、未知量 $\nabla V_{n+1}(x_{n+1})$ および $\nabla^2 V_{n+1}(x_{n+1})$ を含むことになる。従って、これらの推定値 $\nabla \bar{V}_{n+1}$, $\nabla^2 \bar{V}_{n+1}$ を代用して計算式を作成する。すなわち、

$$\bar{y}_n^{k+1} = \bar{U}_n(x_n^k, y_n^k) = y_n^k - \alpha \bar{J}_n^k(x_n^k, y_n^k) \bar{T}(x_n^k, y_n^k) \quad (35)$$

とする。また、 $\partial y_n^k / \partial x_n$ は、式 (29) を近似して求める。

以下の計算手続は、アルゴリズムの項に示す。

なお、本稿では、 $\nabla \bar{V}_{n+1}$, $\nabla^2 \bar{V}_{n+1}$ を含む関数 \cdot_n を $\bar{\cdot}_n$ で表すこととする。また $[\bar{J}_n^{-1} \bar{K}_n]$ で \cdot_n に対応する $[\bar{J}_n^{-1} \bar{K}_n]$ の部分行列を示す。

一次元探索のステップサイズ α の探索と収束の判定は、式 (17) の誤差評価式を、

$$\begin{aligned}
E_n^k(\alpha) = & \|\nabla_u \bar{F}(x_n^{k+1}, y_n^{k+1}, (\nabla \bar{V}_{n+1}^k|_{\alpha=0} \\
& + \nabla \bar{V}_{n+1}^k|_{\alpha})/2, \alpha)\|
\end{aligned}$$

として、

$$E^k(\alpha) = \sum_{N-1} E_n^k(\alpha) \quad (36)$$

$$E^{*k}(\alpha) = \sqrt{\sum_{N-1} E_n^k(\alpha)^2} \quad (37)$$

などの大小で行う。

(2) 不等式制約条件の取扱いと J_n の近似式

本稿では、式 (26) のように Lagrange 乗数 λ_n を直接 J_n の中で使うことを避けて、数値計画法の一技法を参照し、活性な不等式を等式と同じように扱うこととする。

この方法の妥当性については、文献 14), 21) を参照されたい。なお、以下では簡単のため、記号 h_n は適宜省略し g_n で代表させることとする。

更に、本稿の SDN-DDP では、本来の J_n の代りに、修正した適合近似^{18), 21)}、

$$\begin{aligned}
J_{Mn} = & \begin{bmatrix} \nabla_u^2 F_n & \nabla_u g_n^T \\ \nabla_u g_n & \text{diag}(-C_{Mn})^{-1} \end{bmatrix} \quad (38) \\
C_{Mn} : & (C_{Mj1}, C_{Mj2}, \dots)
\end{aligned}$$

を J_n と表して使用する。数値計算の途中で、仮定 2 が一時的に破れた場合などに有効である。収束の速さは若干低下するが、安定性は良好である。通常、 C_{Mn} の要素は次節の終端境界条件の C_{Nd} より大きくなる。

(3) 終端状態制約条件の処理

最終端 $n=N$ における境界条件の式 (3) は、

$$\phi(x_N) = x_N - a_N = 0 \quad (39)$$

と表せる。本稿では、この形の境界条件に対し、坂和¹⁴⁾が示した以下のような手法を採用する。

コスト関数 J に対し、乗数法^{14), 21)}に従って $\phi(x_N)$ を含む Lagrange 関数 J_c を作り、その最小化を行う。すなわち、

$$J_c = J + b^k \phi(x_N) + \sum_d C_{Nd} \phi_d(x_N)^2 \quad (40)$$

と J_c を定めて乗数法を適用する。まず、

$$b^0 = (\text{diag}(C_N^0) \phi(x_N^0))^T \text{ とし、}$$

$$\nabla \bar{V}_N^k(x) = b^k \phi_x(x_N^k) \quad (41)$$

$$b^{k+1} = b^k + (\text{diag}(C_N^k) \phi(x_N^k))^T \quad (42)$$

により $\nabla \bar{V}_N^k(x)$ を求める。ここで、 b は横ベクトルである。また、 $\nabla^2 \bar{V}_N^k = \text{diag}(C_N^k)$ となる。本稿の計算例では、 C_N^k の成分 C_{Nd}^k の初期値を、 $C_{Nd}^0 = 0.1/x_{Nd \max}$ 程度とし、その後は、1ステップ当り 1% 前後で漸増させ、 $5C_{Nd}^0$ 程度を上限とする。

(4) パイロット軌道とその更新

式 (7) により、パイロット軌道 $\{X_n(PT)\}^l$ と安定化関数 L_{2nd} の関係が示されている。ここで、 ϵ_{nd} は L_{1n} の係数と計算の安定性を考慮して定める。例えば、 $\epsilon_{nd} = C_d/(x_{nd})_{\max}$ などとする。 d は貯水池番号で、 $(x_{nd})_{\max}$ は制限容量などとする。 C_d は、適当な正数である。

初期のパイロット軌道 $\{X_n(PT)\}^0$ は、なるべく高水位に、例えば各時点 n ごとに制限または満水位より数センチくらい低い水位に設定し、 x_n の境界点は避け、終端点 N 付近では終端指定水位にすりつける。第 l 次の逐次計算過程で得られた最適解 $\{x_n^*\}^l$ と $\{X_n(PT)\}^0$ より、パイロット軌道の更新を次式のいずれかで行う。

$$X_n^{l+1}(PT) = (x_n^*)^l \quad (43)$$

$$X_n^{l+1}(PT) = 0.5^{(l+1)} X_n^0(PT) + (1 - 0.5^{(l+1)}) \cdot (x_n^*)^l \quad (44)$$

$$n=1, \dots, N-1$$

ここに、 l : 逐次正則化過程の番号、 $l=0, 1, \dots$ とする。

(5) その他

① 2-(2) 節で述べた補正係数 s は、 $\nabla_x^2 L_n$ が $\nabla_x^2 L_n$ に比べて相当大きくなるように、例えば、 $s=1 \sim 5$ 位に選定する。

② $\nabla_x L_n$ に対しては、活性な g_n による小さいペナルティ関数 $\frac{1}{2} c g_{nd}^2(x_n)$ を L_n に追加して求めれば、計算効率が若干向上することがある。

(6) アルゴリズム (SDN-DDP)

計算手順の概要は次の通りである。

ステップ (0):

① 計算に必要な諸定数を定める。(本章 (8) 節および 5 章数値計算例を参照されたい。)

② パイロット軌道 $\{X_n(PT)\}^0$ を定め、 $l=0$ とする。

③ 式 (1), (2), (3), (4) を充たす各目上の操作量 $\{u_n^0\}$, 状態変数 $\{x_n^0\}$ を定め、 $\{\lambda_n^0\}$, $\{\mu_n^0\}$ を 0 とし、 b^0 を求め、 $k=0$ とする。

ステップ (1): 式 (41), および (3) 項より、 $\nabla \bar{V}_N^k(x_N^k)$ と $\nabla^2 \bar{V}_N^k(x_N^k)$ を求める。また、 b^k を更新する。
ステップ (2):

① ステップ (3)~(6) より、次元探索の各探索点 α_m に対する $\{y_n^{k+1}\}_m$, $\{x_n^{k+1}\}_m$, $E'(\alpha_m)^k$ を求めながら、 $E'(\alpha_{opt})^k$ を最小とする α_{opt}^k を定める。但し、 $0 < \alpha_{\min} \leq \alpha_{opt}^k \leq \alpha_{\max} \leq 1$ とし、 α_{\min} , α_{\max} は適当に与える。

② ステップ (3)~(6) より、 α_{opt}^k に対する $E'(\alpha_{opt})^k$, $\{y_n^{k+1}\}_{opt}^k$, $\{x_n^{k+1}\}_{opt}^k$ を求める。

③ もし、制約条件が所定の精度で充たされて $E(\alpha_{opt})^k \leq \epsilon_B$, または、 $k \geq k_{\max}$ ならばステップ (7) へ行く。ここで ϵ_B , k_{\max} は予め与えた数値である。

④ 活性な不等式を調べ、必要があれば制約式などを更新し、 $k \leftarrow k+1$ として、ステップ (1) へ行く。

ステップ (3): $n=N-1, \dots, 1$ および $\alpha = \alpha_m$ などに対し、次の計算を行う。

$$\begin{aligned} \bar{y}_n^{k+1} &= \bar{U}_n(x_n^k, y_n^k) \\ &= y_n^k - \alpha \bar{f}_n^1(x_n^k, y_n^k) \bar{T}_n(x_n^k, y_n^k) \end{aligned} \quad (45)$$

$$\begin{aligned} \nabla \bar{V}_n^k &= \nabla_x L_n + \nabla \bar{V}_{n+1}^k \nabla_x f_n \\ &\quad + \beta (\bar{u}_n^{k+1} - u_n^k)^T \nabla u_n^k(x_n^k, u_n^k)^T \nabla^2 \bar{V}_{n+1}^k \nabla_x f_n \\ &\quad + (\bar{\lambda}_n^{k+1})^T \nabla_x g_n + (\bar{\mu}_n^{k+1})^T \nabla_x h_n \end{aligned} \quad (46)$$

$$\begin{aligned} \nabla^2 \bar{V}_n^k &= \nabla_x^2 L_n + \nabla_x f_n^T \nabla^2 \bar{V}_{n+1}^k \nabla_x f_n \\ &\quad - (\nabla_{xu}^2 L_n + \nabla_x f_n^T \nabla^2 \bar{V}_{n+1}^k \nabla u_n^k) [\gamma \bar{f}_n^1 \bar{K}_n]_u \\ &\quad - \nabla_x g_n^T [\gamma \bar{f}_n^1 \bar{K}_n]_x \\ &\quad - \nabla_x h_n^T [\gamma \bar{f}_n^1 \bar{K}_n]_u + R(0) \end{aligned} \quad (47)$$

ここに、 $\beta = 0.4 \sim 1.0$

$$\gamma = \text{Min}(1, \text{Max}(0.5, 2\alpha))$$

$R(0)$; 問題 (A) に対し 0 となる項

である。

〔備考〕 式 (46) は、式 (32) を (x_n^k, \bar{y}_n^{k+1}) において近似したものである。式 (47) は、式 (33) の (x_n^k, y_n^k) における近似式である。なお、本稿のテスト計算では、式 (46) の β は 0.5 とした。

ステップ (4): $n=0$ では、 $\alpha = \alpha_m$ などに対し、

$$x_0^{k+1} = a_0, \quad y_0^{k+1} = \bar{U}_0(x_0^k, y_0^k) \quad (48)$$

である。

ステップ (5): $n=1, \dots, N-1$, $\alpha = \alpha_m$ などについて

て、次の計算を行う。

$$x_n^{k+1} = f_{n-1}(x_{n-1}^{k+1}, u_n^{k+1}) \quad (49)$$

$$y_n^{k+1} = \bar{y}_n^{k+1} - [\bar{f}_n^{-1}(x_n^k, y_n^k) \bar{K}_n(x_n^k, y_n^k)](x_n^{k+1} - x_n^k) \quad (50)$$

また、 $n=N$ について次の計算を行う。

$$x_n^{k+1} = f_{N-1}(x_{n-1}^{k+1}, u_n^{k+1}) \quad (51)$$

ステップ (6) : $\alpha = \alpha_m$ などに對し、以上で求めた $\{x_n^{k+1}\}$, $\{y_n^{k+1}\}$, $\{\nabla \bar{V}_n\}$ などと式 (37) より $E'(\alpha)^k$ を求め、ステップ (2) の該当箇所へ帰る。

ステップ (7) :

① $\{x_n^{*l}\} = \{x_n^{k+1}\}_{opt}^l$ とおいて、式 (43) または (44) より $\{X_n(PT)\}^{l+1}$ を求める。

② $l \geq l_{max}$ となるか、パイロット軌道の修正値が所定の範囲内に入ればステップ (8) へ行く。

ここに l_{max} は、予め与えた打ち回数である。

③ $\{x_n^{k+1}\}_{opt}$, $\{x_n^{k+1}\}_{opt}$ を名目上の出発値 $\{u_n^0\}^{l+1}$, $\{x_n^0\}^{l+1}$ とし、 $\{\lambda_n^0\}$, $\{\mu_n^0\}$ を更新し (例えば 0), b^0 を求め、 $l \rightarrow l+1$, $k=0$ などとしてステップ (1) へ行く。

ステップ (8) : 計算を止め、必要な結果を出力する。

以上。

(7) アルゴリズムの収束性について

a) DN-DDP の収束性

DN-DDP の収束性の証明は、標準の Newton 法による DDP の場合と殆ど同様であるので省略する。

代わりに、標準の Newton 法による DDP の収束証明の結果を文献 11) より引用すれば、概ね次の通りである。「いま、 $\delta_n^k = \|y_n^k - y_n^*(x_n^k)\|_1$ とし、 $\delta^k = (\delta_1^k, \dots, \delta_{N-1}^k)^T$ とすると、最適解に近い名目上の値 $\{y_n^0\}$, $\{x_n^0\}$ より出発すれば、DN-DDP で $\alpha=1$ とした計算法により、次式が得られる。

$$\delta^{k+1} \leq Q^k(\alpha)_{\alpha=1} \delta^k, \quad \|Q^k(\alpha)_{\alpha=1}\| < 1$$

ここに、 $Q(\alpha)$ は DN-DDP の計算過程を整理して得られる不等式の係数行列である。上式より、

$$\lim_{k \rightarrow \infty} \|\delta^k\| \rightarrow 0, \quad \text{となるので} \quad \lim_{k \rightarrow \infty} \{y_n^k\} \rightarrow \{y_n^*\}, \quad \text{が従う。}$$

同様の論法により、 $\|Q^k(\alpha)_{\alpha=1}\| \leq \|Q^k(\alpha)_{\alpha=1}\|$ 、と出来るので、DN-DDP では、標準のものに較べて広い範囲の δ^k に対し、上式と類似の関係式が成り立つ。よって、DN-DDP は、より広い範囲の出発値に対して収束する。

このことは、数値計算結果によって確認できる。すなわち、後述の数値計算例の問題は、 $\alpha=1$ のもとでは収束しないが、或る $\{\alpha^k\} \leq 1$ のもとでは収束する。

以下に、本稿の問題 (A) の特徴を列記する。採用した諸定数が適当であれば、

① 式 (47) のリカッチ型差分式で求まる $\nabla^2 \bar{V}_n^k$ は、対称正定かつ有界と推定される。

② $\nabla_n^2 L_n$ は、区分的に定数で有限の跳躍量を持ち、 $\nabla_n^2 L_n$ は定数である。

しかしながら、DN-DDP にも、取り扱い可能な貯水池数 m 、期間数 N に限界があることは明らかである。

なお、DN-DDP の収束率は R-1 次と推定される。また、実際の数値計算では、誤差 $\{E^k\}$ は、若干の振動を伴いながら次第に小さくなり、 ϵ_E 以下となる。

b) 逐次正則化過程 (Sequential R. Process) の収束性

コスト関数 $J(l)$ は、下流域の需要充足度を示す J_1^l と、パイロット軌道 $\{X_n(PT)\}^l$ と $\{x_n^*\}^l$ との差による J_2^l に分けられる。パイロット軌道の更新に当っては、 J_2 がより小さくなるように軌道が修正される。また、各 l について $\{x_n^*, u_n^*\}^l$ が求まることは、前項より明らかであるから、次の関係式がなりたつ。

$$\begin{aligned} & [J_1(\{x_n^*\}^{l+1}, \{u_n^*\}^{l+1}) + J_2(\{X_n\}^{l+1}, \{x_n^*\}^{l+1})] \\ & \leq [J_1(\{x_n^*\}^l, \{u_n^*\}^l) + J_2(\{X_n\}^{l+1}, \{x_n^*\}^l)] \\ & \leq [J_1(\{x_n^*\}^l, \{u_n^*\}^l) + J_2(\{X_n\}^l, \{x_n^*\}^l)] \end{aligned} \quad (52)$$

上式より $J^{*l+1} \leq J^{*l}$ となるので、 J^{*l} は単調に減少し、 l が大きくなれば $J^{*l} \rightarrow J^*$, $J_2^{*l} \rightarrow 0$ (或は ϵ_{E2}) となる。従ってパイロット軌道も収束する。

なお、 $J_2^{*l} \rightarrow \epsilon_{E2}$ となるのは、主として式 (8) 右辺の $\{ \}$ 内第 2 項がある場合である。

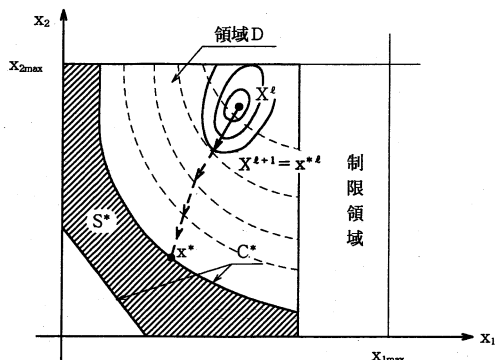
以上より、SDN-DDP により J^{*l} は J^* に収束し、 J^l の狭義凸性から問題 (T) の最適解 $\{x_n^*, u_n^*\}$ が得られることが云える。

(8) 計算最適解の特性

渇水被害関数のみでコスト関数を作り、操作量の最適化を図ると、既述のように特異制御問題となり、通常は無数の最適解を持つこととなる。この事は、例えば 2 個の貯水池よりなる並列モデルを想定すれば明らかである。すなわち、通常は二つの貯水池からの取水の組合せに任意性があり、また洪水時の予備放流の放流量曲線にも任意性がある。更に無数の最適解の中には、工学的に受け入れ難いものもあろう。

安定化関数を導入して特異性を解消し、SDN-DDP で問題を解くことは、無数の J_1 のみをコスト関数とする本来の最適解の中から一つの解を選択することに相当する。更に、その解は工学的に容認できるものでなければならない。この条件に適する解として、本稿では各貯水池がなるべく高い水位推移曲線を持つものを想定した。極めて概念的ではあるが、このような傾向を持つ最適解が SDN-DDP により得られることを述べる。但し、式 (8) 右辺の $\{ \}$ 内第 2 項は無視し得るものとする。

まず、 $M = m \times N$ とする。ここに m , N は貯水池の数と対象期間数である。M 次元空間 E^M を考え、座標成分は各貯水池・各期首の貯水容量とする。よって貯水池群の容量推移曲線および l 次のパイロット軌道 X^l は E^M 内の点で表され、貯水池運用の実行可能領域は E^M



図—2 計算最適解説明図

内の領域 D として表される。これを概念図で示せば、例えば図—2 のような説明図が描ける。

E^M 上の、本来の解集合 S^* の包絡面を C^* とする。いま、 X^0 を右側上方に設けるものとし、1 次のパイロット軌道 X^1 が得られたとする。第 l 次の逐次計算過程をへて、 $\{x^{*l}, u^{*l}\}$ が得られる。図示の点 x^{*l} はこの位置を示す。この段階で、 x^{*l} はコスト関数 $J(l)$ を最小とする点である。 E^M 上で x^{*l} を通り $J_2^*(l)$ の等高面 Z^l を描くと、 Z^l は楕円面であり、その形は $J_2(l)$ の係数 $\{\epsilon_{nd}\}$ によって定まる。 x^{*l} は、 D 内の Z^l 上で $J_1(l)$ を最小とする点ともなっている。

X^l の更新には、式 (43) を使用するものとする。各 l に対する x^{*l} は勿論 SDN-DDP の一段階として決定されるが、 X^l の点列を $l=0, 1, 2, \dots$ と追跡すれば、各 x^{*l} したがって次の X^{l+1} は、 X^l を原点とし x^{*l} を通る楕円面 Z^l と D の共通領域上で $J_1(l)$ を最少とする経路を通して D 上を J_1 の小さい方へ降下することになっている。式 (44) に対しても殆ど同様である。若し、初期値 X^0 が領域 S^* の外部にあれば、最終段階で、 x^{*l} は S^* の内部にまで達することはなく包絡面 C^* 上の一点 x^* に収束する。その位置は、問題 (T) の全係数が与えられると定まる。要約すれば、 X^l は E^M の距離を $\{\epsilon_{nd}\}$ で補正した距離 d_ϵ のもとで、 D 上を J_1 について最急勾配の経路を通して区分的に C^* 上に降下する。 x^* の位置は、定性的には、 C^* 上で X^0 から相対的に余り遠くない所にあると云えよう。 X^0 、 $\{\epsilon_{nd}\}$ は x^* を定めるパラメータの一種と考えられる。

従って、SDN-DDP により得られた解に工学的にみて不満があれば、主として $\{\epsilon_{nd}\}$ を操作して修正することが出来る。本稿では $\epsilon_{nd} = C_d / (x_{nd})_{\max}$ としているので入力データ C_d の修正を行う。 C_d を大きくすれば、 x^* における d 貯水池の対応する水位推移曲線は相対的に高くなり、小さくすれば低くなる。

本稿では、 C_d の決定は数回の計算結果の視察により行うこととした。

なお、数理計画法と SDN-DDP を組合せて、 C_d の探索を行うことも考えられるが、単純に行えば本稿の方法より数倍以上の計算量が必要と推定される。

5. 数値計算例

本稿で示した SDN-DDP の有効性を検証するため、図—1 で表される例題需給モデルについてテスト計算を行った。以下に計算の前提とした諸定数などおよび計算の結果得られた操作量などを簡単に示す。

(1) モデルの諸定数など

計算に使用したモデルの諸定数などを以下および表 1 に示す。

渇水被害関数の係数： $C_p = 10.0 (\text{m}^3/\text{s})^{-1}$ (共通)

$$\Delta t = 5 \text{ (日)}, s = 2.0, \beta = 0.5$$

計算対象期間 N : 216 期 (半旬単位, 3 年間)

操作量 u_{nj} の下限: 0 または $0.5 \text{ m}^3/\text{s}$

4 章 5 節②の係数 c : $0.4/x_{d \max}$

貯水量の初期, 終端条件: とともに, $(304.0, 214.1, 163.4)^T (\text{m}^3/\text{s} \cdot \text{day})$

出発値 $\{u_n^0\}$, $\{x_n^0\}$: $\{x_n^0\}$ ができるだけ $\{X_n(PT)\}^0$ に近くなるように選ぶ。

$X_n^l(PT)$ の更新には式 (44) を使用する。

なお、各貯水池の制限容量、各貯水池への流入量、各需要点の需要量は、計算結果図に示す。

(2) 計算結果

SDN-DDP で得られた各貯水池状態推移図を図—3 に示す。また需要点における需給の充足状態推移図を図—4 に示す。なお、供給率 = 100% - 節水率とする。

これらの結果図および行数の都合で表示を省略した諸結果を含めて、例題モデルについてのテスト計算で得られた数値解は、殆ど最適解に収束していると判断される。

式 (36) の E^k で云えば、最初は約 2300 程度であった $(E^k)_{k=0, l=0}$ が、各 l ごとに、DN-DDP の反復回数 k が最大 139 以下で 0.80 以下となっている。各 l に対する $|X_{nd}^{l+1}(PT) - X_{nd}^l(PT)| / x_{d \max}$ の最大値は、最初 0.298 であったが、 $l=13$ で 0.016 となった所で計算を打切った。各 l における DN-DDP の反復回数は、平均 105 回程度であった。なお、このテスト計算には、処理性能が約 30 MIPS の EWS で、2 時間 47 分の CPU 時間を要した。

6. 既存の方法との効率比較

(1) 効率比較の対象とする既存の方法

本稿の SDN-DDP と効率比較を行う既存の制約条件

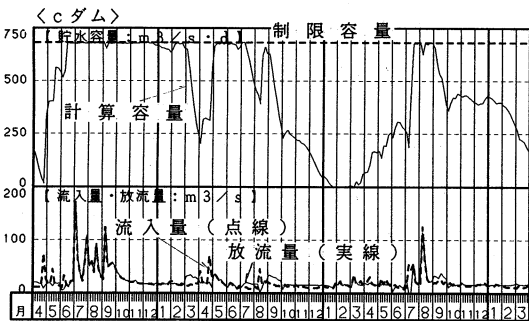
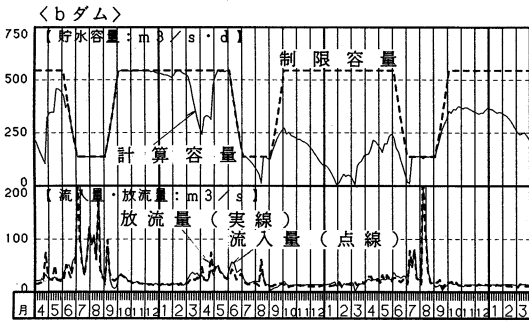
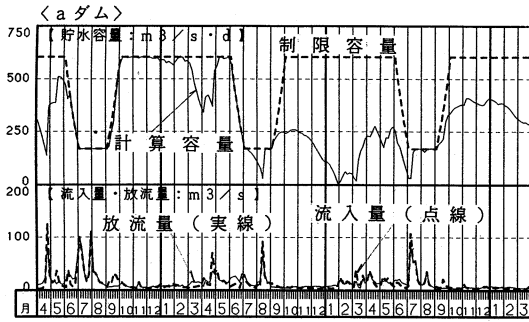


図-3 計算貯水容量推移曲線

表-1 各貯水池に関する定数

	容量 (m³/s·day)	L_{2n} の係数 ϵ_{nd}	終端条件の C_{2n}^0
貯水池 a	605	$0.16/x_{na \max}$	$0.1/x_{a \max}$
貯水池 b	545	$0.16/x_{nb \max}$	$0.1/x_{b \max}$
貯水池 c	680	$0.14/x_{nc \max}$	$0.1/x_{c \max}$

	J_{Mn} の要素 C_{Mnd} 注	制限水位	備考
貯水池 a	$20.0/x_{a \max}$	あり	ϵ_{nd}, C_{2n}^0 の単位 : (m³/s day) ⁻²
貯水池 b	$20.0/x_{b \max}$	あり	
貯水池 c	$20.0/x_{c \max}$	なし	

注: u_{ni} のみの制約式の係数 $C_{Mnu} = 5.0$

付最適制御の数値計算法として、参考文献 7), 8), 14), 22), その他を参照のうえ坂和の方法¹⁴⁾と Yakowitz らの CDDP⁷⁾を選定した。

坂和の方法は、最大原理による確定的最適制御問題の標準的な数値計算法の一つとされており、CDDP は貯水池群の最適制御の有効な数値計算法⁷⁾とされている。

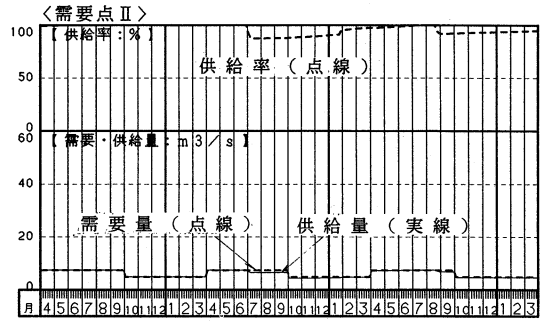
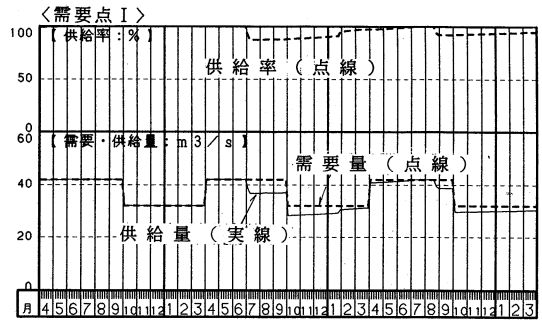


図-4 計算供給率推移曲線

両者ともに、本稿のモデルのような下流域の水路系への流量配分は、直接的には考慮されていないので、若干の変更・拡張を行って本稿のモデルが取り扱えるようにして、DN-DDP に対応する反復過程の範囲で SDN-DDP との効率比較を行った。

これらを以下では拡張坂和法および変形 CDDP と呼ぶこととし、夫々の概要を極く簡単に述べる。

a) 拡張坂和法

この方法は、最大原理に基づく反復計算法である。すなわち、まず逆時間方向に本稿の $\nabla \bar{V}_n$ に相当する随伴ベクトル \bar{d}_n を本稿の方法に準じて求めるが、 $\nabla^2 \bar{V}_n(x)$ を含む項と $(\bar{y}_n^{k+1} - \bar{y}_n^k)$ を係数とする項は考えない。次いで、活性な状態制約条件がある場合の Lagrange 常数 λ_n^k は、Hamilton 関数 H_n との解析的な関係より、

$$\lambda_n^k = -\nabla_u H_n^k (\nabla_u g_n^k)^T (\nabla_u g_n^k (\nabla_u g_n^k)^T)^{-1} \quad (53)$$

と求めるものとする。順時間方向の計算は、 $\{\bar{d}_n^k\}$ を使って、各時点 n ごとに Hamilton 関数 H_n^k を構成し、これを最小化する操作量 u_n^{k+1} を順次求めるものである。しかし、順時間方向の計算には、式 (50) と同様に $-\bar{f}_n^{-1} \bar{K}_n \delta x_n^k$ を加えた方が計算の安定がよいので、本稿ではこの項を加えることとする。また、一次元探索を行うこととする。

b) 変形 CDDP

Yakowitz らの CDDP は、文献 7) で提案されたものであるが、計算手法の細部に関しては明示されていない部分が含まれている。よって文献 23) 等により一部を

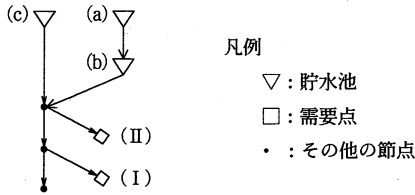


図-5 比較計算モデル図

表-2 比較計算結果表 (期間の単位: 半句)

		SDN-DDP	拡張坂和法	変形CDDP
72期間 (1年)	反復回数	29	43	26
	打切誤差	0.30	0.30	0.30
144期間 (2年)	反復回数	83	190	130
	打切誤差	0.50	0.50	0.50
288期間 (4年)	反復回数	146	508	361
	打切誤差	0.70	0.70	0.70

注: 打切り誤差は式(36)の E^k を対象とする。

補足し、必ずしもCDDPの特徴ではないと思われる部分を若干変更・拡張したものをここでは変形CDDPと呼ぶこととする。その要点は以下の通りである。ここでも次元探索を行うものとする。

① 活性な制約条件がない期間については、CDDPはSDN-DDPと殆ど同じである。

$\nabla \bar{V}(x)$ の計算式は、本来のCDDPの方が、SDN-DDPより、最適解の付近で級数展開した式に対し僅かに近似度は高いが、その数値計算結果は、本稿の式(46)を採用した場合より、必ずしも良好とは云えなかったもので、式(46)を使用することとする。

② 活性な制約条件が在る期間におけるLagrange定数 λ_n^k は、坂和の方法と同様に、式(53)で求める。

従って、CDDPでは、SDN-DDPのように、 λ_n^k を未知数として次の反復計算まで持ち越すことはない。この点がCDDPとSDN-DDPの大きな相異点である。

また、本稿では、順時間方向の計算においては、各 n について新しく活性となる制約条件のチェックは行わず、これを1回の反復計算の終了後にまとめて行う。

活性な制約条件が在る期間 n では、順時間方向の計算においてHamilton関数の最小化を行う。すなわち、安定化関数を含む全流域にわたる2次計画問題を解いて、 u_n^{k+1} を求め、これにステップサイズ α を考慮して u_n^{k+1} を決定する。従って、本来のCDDPでは、反復計算の途中でも変数値は実行可能領域内に取ることとなっているが、変形CDDPではこの規約を若干緩める。

(2) 計算効率比較用モデル

比較計算に用いた流域モデル図は、本稿の図-1例題モデル図を簡略化した図-5を主に用いた。次いで図-

1の例題モデル図についても比較計算を行った。なお、計算は、パイロット軌道の更新を含まないDN-DDPの範囲で行った。

a) 図-5のモデルの諸定数など

各貯水池の容量、流入量、需要量およびコスト関数の係数などは、例題モデルと殆ど同じとした。一方、終端条件計算用パラメータなどは各方法に適すると思われる値を採用した。すなわち拡張坂和法、変形CDDPでは、計算の安定上SDN-DDPより小さい値を採用したが、簡単のため数値の記載は省略する。

b) 比較計算の結果—その1

図-5のモデルについて、表-2の結果を得た。

c) 比較計算の結果—その2

図-1の例題モデルについて、拡張坂和法、変形CDDPによって216期間(3年)のテスト計算を行ったが、必ずしも十分な結果が得られなかった。一方SDN-DDPでは $k=139$ 回で誤差 E^k が0.8となった。更に、288期間(4年)の解析も可能であった。

d) 計算結果の検討

本稿の比較計算によれば、SDN-DDPは、他の二法に較べて、収束の速さにおいて同程度以上であり、長期間・複雑なモデルに対する計算の収束性についてより頑健である。従って適用可能範囲も広いと判断される。

なお、拡張坂和法、変形CDDPともに、SDN-DDPとの比較計算のため、オリジナルな方法に筆者らが若干の改変・拡張を施したものであるが、まだ改良の余地が残されていると考えられる。

7. む す び

本稿では、貯水池への流入量と下流域の需要量が確定的に想定される場合について、貯水池群と複数の需要点をもつ時間について離散的なモデルの最適操作量を求める計算法として、SDN-DDPとその適用結果について述べた。数値計算例の結果より、本方法は、或る程度現実的な問題に対しても充分適用可能である。

また、SDN-DDPと拡張坂和法および変形CDDPとの比較を行った結果、SDN-DDPは、後者2法と比べて、収束の速さにおいて同等以上であり、計算の収束性において頑健であり、適用可能な範囲がより広いと云う結果を得た。

SDN-DDPの特徴の一つは、式(4)の制約条件の処理に際して必要なLagrange定数を、数値計算上は独立な変数として扱ったことである。

貯水池の操作量決定に関する最優先事項は、コスト関数の選定であるが、極めて難解な問題で本稿では到底論じ得ない所である。しかし、本稿で採用した主コスト関数 L_{in} は、文献1)、15)などの記述より、相当の妥当

性を持つものと判断される。もっとも、その結果として特異制御問題が生じることとなった。本稿のSDN-DDPは、計算量はかなり増加するものの、これに充分対処し得ると思われる。

ところで、特異制御問題が生じないような合理的なコスト関数が設定できれば、非常に望ましいことである。但し、その設定に際しては、計算上の都合は副次的なものである。貯水池群操作量に関する一二の文献の計算例においては、予め特異制御問題が生じないようにコスト関数が選ばれているが、これらは実用上の見地からは選択の根拠が不充分と思われる。

本稿のSDN-DDPにおいては、終端境界条件の処理などに際し、パラメータの選定に若干の考慮が必要である。しかし、乗数法において類似のことが生じることを思えば、止むを得ないことと云える。4章(8)節で述べたように、並列貯水池がある場合などには本来の最適解は無数にあるのが普通である。安定化関数の係数の選び方により求まる解の特性が決まるので、工学的に望ましい最適解を得るためには安定化関数の係数の貯水池間のバランスに注意が必要である。

なお、本稿の方法においても、計算可能な期間数 N と貯水池の数 m には、当然その限度はあるが、試算の結果では従来のものより適用可能な範囲は大幅に広がっている。

最後に、本稿作成に際しては、九州大学工学部建設都市工学科阪本好史教授の懇切な御指導を頂いた。また、西日本技術開発(株)開発部長 津城正氏、同部長代理 矢次正彦氏、同 木寺佐和記氏より貴重な御意見を頂いた。更に、同システム開発課 黒木修身氏、中川裕氏には種々御協力を頂いた。ここに記して、各位に深く感謝の意を表する次第です。

参考文献

- 1) 土木学会：土木工学ハンドブック(第4版)，pp.2567-2574，技報堂出版株式会社，1989。
- 2) Jacobson, D.H. and Mayne, D.Q. : *Differential Dynamic Programming*, pp.99-132, American Elsevier, 1970.
- 3) 高棹琢馬，池淵周一，小尻利治：水量制御からみたダム群のシステム設計に関するDP論的研究，土木学会論文報告集，第241号，pp.39-50，1975。
- 4) 竹内邦良：貯水量の累加損失係数を用いた貯水池群の最適操作手法，土木学会論文報告集，第222号，pp.93-103，1974。
- 5) 吉川秀夫，竹内邦良：濁水持続曲線の性質とその応用，

- 土木学会論文報告集，第234号，pp.61-71，1975。
- 6) 竹内邦良，吉川秀夫：濁水持続曲線法とマスカーブ法，土木学会論文報告集，第303号，pp.53-63，1980。
- 7) Murray, D.M. and Yakowitz, S.J. : *Constrained differential dynamic programming and its application to multireservoir control*, *W. Res. Res.* Vol.15 No.5, pp.1017-1027, 1979.
- 8) Yeh, W. W-G. : *Reservoir management and operations models*, *W. Res. Res.*, Vol.21, No.12, pp.1797-1818, 1985.
- 9) 池淵周一：ダム操作，オペレーションズ・リサーチ，Vol.33, No.9, pp.11-15, 1988.
- 10) 土木学会：水理公式集(昭和60年版)，pp.181-186，土木学会，1985。
- 11) Ohno, K. : *A new approach to differential dynamic programming for discrete time systems*, *IEEE Trans. Automat. Control*, Vol.AC-23, No.1, pp.37-47, 1978.
- 12) Dyer, P. and McReynolds, S.R. : *The Computation and Theory of Optimal Control*, pp.68-78, Academic Press, 1970.
- 13) Bell, D.J. and Jacobson, D.H. : *Singular Optimal Control Problems*, pp.153-171, Academic Press, 1975.
- 14) 坂和愛幸：最適化と最適制御，pp.163-173，pp.112-140，pp.8-43，森北出版(株)，1980。
- 15) 建設省土木研究所：濁水時の水管理に関する計画学的研究，土木研究所資料第1508号，pp.103-134，1979。
- 16) 三根久：オペレーションズ・リサーチ上，pp.153-164，pp.121-122，朝倉書店，1966。
- 17) Ortega, J.A. and Leake, R.J. : *Discrete maximum principle with state constrained control*, *SIAM J. Cont & Opt.*, Vol.15, No.6, pp.984-990, 1977.
- 18) Ortega, J.M. and Rheinboldt, W.C. : *Iterative Solution of Nonlinear Equations in Several Variables*, pp.355-363, pp.128-129, Academic Press, 1970.
- 19) Canon, M.D., Cullum, C.D. and Polak, E. : *Theory of Optimal Control and Mathematical Programming*, pp.130-140, McGraw-Hill, 1970.
- 20) 福島雅雄：非線形最適化の理論，pp.70-80，産業図書(株)，1980。
- 21) Bertsekas, D.P. : *Constrained Optimization and Lagrange Multiplier Methods*, pp.248-252, pp.240-243, Academic Press, 1982.
- 22) Blank, D. and Shinar, J. : *Efficient combinations of numerical techniques applied for aircraft turning performance optimization*, *J. Guidance, Control and Dynamics*, Vol.5, No.2, pp.124-130, 1982.
- 23) Yakowitz, S. and Rutherford, B. : *Computational aspects of discrete-time optimal control*, *Appl. Math. & Computation*, Vol.15, pp.29-45, 1984.

(1994.8.22 受付)

A COMPUTATIONAL METHOD FOR OPERATIONS OF MULTI-RESERVOIR SYSTEMS BASED ON DIFFERENTIAL DYNAMIC PROGRAMMING

Osam NAKAMURA, Shuichi SAKO, Akira KAWAMURA
and Toshimitsu KOMATSU

The purpose of this paper is to report a computational method for finding optimal controls of multi-reservoir operations, based on the method of K. Ohno's new approach to DDP with Newton's Method. We added a line search technique to it for the stability of numerical computation and combined the method with modified Bell-Jacobson's ϵ -Algorithm to cope with singular control. We refer to it as SDN-DDP.

The simulation study shows that the method can accurately solve optimal control problems of multi-reservoir and water demand systems under a strictly convex cost function, with some computational quantities