새로운 미분동적 계획법에 의한 저수지군의 최적제어

Applications of New Differential Dynamic Programming to the Control of Real-time Reservoir

선 우 중 호*
Sonu, Jung Ho
이 재 형**
Lee, Jae Hyoung

요 지

수자원 부족과 개발비의 앙등은 관리상의 문제를 제기하고 있다. 일반 관리가 그러하듯이 물의 관리도 목적하는 바의 계량화 작업이 따르게 되고 수학적인 모형의 복잡성 때문에 컴퓨터를 사용하게 된다. 그 모형이 비선형 함수관계를 갖고 있으며 복잡한 제약조건이 부가될때 컴퓨터를 사용하기 위해 선행되어야할 작업은 통일된 알고리즘을 정하는 일이다.

위와 같은 문제의 해결에 이용되어온 기존의 동적계획법은 두가지의 제한점을 가지고 있다. 즉, 변수를 이산화 해야하며 제약조건의 처리가 불가능 하다는 점이다. 현재까지 애용되고 있는 미분동적계획법에 의해 개발된 방법들 조차도 제약조건의 처리가 미흡하다.

본 논문에서는 위와 같은 어려움을 극복하고 저수지군의 다목적 다단계 제어에 응용할 수 있는 새로운 동적최적화 모형을 제안하였으며 본 논문에서 제안한 방법이 다른 대안들에 비해서 우월함을 입증하였다(즉, 기존의 문제를 본 방법에 의해 수치실험한 결과 기존 동적최적화 기법의 제약이 해소되었고 더 좋은 목적함수 값을 얻었다).

Abstract

The complexity and expensiveness of water resources projects have made optimum operation and design by computer-based techniques of increasing interest in recent years. Water resources problems in real world need many decisions under numerous constraints. In addition there are nonlinearities in the state and return function.

This mathematical and technical troublesome must be overcome so that the optimum operation polices are determined. Then traditional dynamic optimization method encountered two major cruxes: variable discretization and appearance of constraints. Even several recent methods which based on the Differential Dynamic Programming (DDP) have some difficulties in handling of constraints.

This paper has presented New DDP which is applicable to multi-reservoir control. It is int-

*정회원 · 서울대학교 공과대학 부교수 토목공학과
**정회원 · 전북대학교 공과대학 조교수 토목공학과

ended that the method suggested here is superior to abailable alternatives. This belief is supported by analysis and experiments(New DDT does not suffer course of dimensionality and requires no discretization and is able to handle easily all constraints and nonlinearity).

1. Introduction

The purpose of this paper is to describe and analyze the method, which gives a clue to the difficult problem, to determine optimum operating policies of real reservoirs. As water resource systems grow large and become more complex, the optimum use of multipurpose reservoir uses become important. The investment cost and operating expenses of projects are so large that even small improvements in system utilization can involve substantial amounts of money. Also complication arises in interaction between the objectives(the various control points, power generators, irrigation outlets, pumping station etc.) which cause difficulties in obtaining an optimum design or operating policy using an empirical approach. Thus, the potential benefits of using optimization techniques in these problems are very great indeed.

Optimization of such a system is still a formidable task when the number of state and control variables and the number of constraints is large. In the study of T.V.A.[1], for instance, the Tennessee Valley Authority, which manage a system of 40 water reservoirs, reported that the maximum size of the problem of controlling a system of water reservoirs solved with the existing numerical methods involved a system of 6 reservoirs.

If a general analytic solution of the discrete-time problem does not exist, then numerical methods should be available. Dynamic programming, although impressive in comparison with direct enumeration, is effective only when the number of variables is small. Consequently, many iterative methods have been developed.

One of them is the succesive approximation method which is originated from Larson[2] in TVA projects formed to be most effective for multi-reservoir control. After this study Larson and Lerkler[3], Korsak and Larson[4] improved this method. Consequently a convergence proof for this technique covering the multi-reservoir problem was given by Larson and Korsak[5]. Trot and Yeh[6] applied this tool to determine optimum reservoir size and operating policy. It was refined and exploited by Yeh et al.[7], Nopmongcol and Askew[8]. Since it has been termed the Dynamic programming with Successive Approximation(DPSA), it is basically a procedure for successive approximation on state trajectory space. This technique has some disadvantage, that is, suffering the course of dimensionality.

Heidary et al.[9] and Chow et al.[10] have developed what that call a discretized version of Differential Dynamic Programming in connection with their studies of the multireservoir control problem(IDP). However, because of discretization they lose many advantage of Differential Dynamic Programming. Above all Turgeon[11] has shown that Incremental Dynamic Programming may yield nonoptimal solutions.

To avoid this disadvantage Murray and Yakowitz[12] devised a linearly constrained LQP(performance is quadratic and state equation is linear) which is based on the Differential Dynamic Programming procedure of Jacobson and Mayne[13]. They approximated objective function by quadratic function in order to avoid second order derivative that appears in DDP. Although their device is very effective, they couldn't find a tool that modify solution and made some mistakes. In approximation procedure for objective function they regarded state variable as constant, but in solution model it as variable. For resticting bounds of state variable and control variable, in every state they have constructed some feasible region which all variable may not violated by using Fletcher[14] algorithm. In this method there are two procedures. One is constructing feasible region and the other is modifying intermediate solution, but it is not sure how to modify trial solution.

Difficulties in constructing feasible regions was tackled by Jae Hyoung[15,16]. The author devised the conceptual algorithm which is based on "Discrete Maximum Principle with State Constrained Control"[17] and Nedeljkovic's algorithm[18,19]. But the approach is not clear whether it is easy to obtain numerically optimal solutions of the mathematical model based on Ortega[17]. The second alternative[16] has a measure to modify a trial solution and is a numerical model. It is successful to find a true optimal path. However difficulties in handling constraints remains still.

It is sure that difficulties in handling nonlinearity can not be avoided. For this Yeh et al.[20] have developed practical procedure by adopting two phases that LP is adapted to DPSA. They adopted two phases method a approximated solution in Phase I by using linear programming(LP) and the true solution in phase II by using DPSA. In this project they fitted nonlinear benefit curves in linear for LP model. Nonlinearity in benefit or cost function is ordinary phenomena. The algorithms that is recently developed and first order algorithm, have difficulties in handling of constraints and non-linearity. Yeh applied two phase method to this problem. Yeh's two phase model is regarded as a method to reduce the burden of nonlinearity. This method seems to have two drawbacts. One of them is dimensionality in LP system. The order is hard to fit terminal constraints.

It is clear that more general and suitable method would substantially increase ability to solve this kind of problems. Ohno[21] who proposed a possibility of them modified the second order discrete-time DDP for problems with and without constraints and proved its local convergence(i.e. when the starting point is sufficiently close to the optimum)[18].

As his theory seems to be more suitable to this problem than any other[15](i.e. no discretization, no restriction on the relation of variables, Quadratic convergence), here it is proved that it is possible to develop discret-time multipurpose and multireservoir optimum control version of Ohno's algorithm. Also this paper presents how to overcome local convergence problem and how to deal with constraints. It is shown that the model gives better result than any other existing methods.

2. General Formulation and Definition

The multi-reservoir control process is defined to be a discrete time control problem such that the set of decision is the finite set $1, 2, \cdots, N$ of integers and the state and control u are m and m-tuples, respectively. Consider a system composed of m reservoirs with both series and parallel connections. The flow of water between the reservoirs can be described by a set of difference equation(referred to sta te equation, physically speaking continuity equation).

$$x_{n+1}=f(x_n, u_n, y_n) \qquad (1)$$

in which $X=(x_n^1, \cdots, x_n^m)$, in which x_n^m represents the storge level; $U=(u_n^1, \cdots, u_n^m)$, in which u_n^m represents the release policy for reservoir m during time horizon n; $Y=(y_n^1, \cdots, y_n^m)$, in which y_n^m represents the total inflow into reservoir m during time horizon n, which includes the natural stream flow and release from the upstream; m=total number of reservoirs or reservoir index; f is an m-dimensional vector of functions.

The net benefits are defined as follow

$$V=\sum_{n=0}^{N-1} L_n(x_n, u_n)+L_N(x_N) \tag{2}$$

in which $L_n(x_n, u_n)$=the return by selling water and/or power produced by the entire system during planning period through optimal operation in the existing reservoirs.

It is necessary to maximize V under all constraints imposed upon the system. Assume that each of the m reservoirs has a maximum allowable storage capacity, x_{max}^m, a minimum pool of x_{min}^m, and a maximum allowable release, u_{max}^1, a minimum release, u_{min}^1(for example, during flood season it is necessary that reservoir is in the low level and during day time power generated maximally, but in spring season reservoir level is reserved for recreation etc.) for the 1 controlers. Then the optimal operating policy associated with this given set of reservoirs capacity and constrained release is the sequence of decisions $u_0, u_2, \cdots, u_{N-1}$ which maximize return. The objective function for optimal operation is to

$$\max \sum_{n=0}^{N-1} L_n(x_n, u_n)+L_N(x_N, u_N) \tag{3}$$

subjected to

$$x_{min} \leq x_n \leq x_{max} \tag{4}$$

$$u_{min} \leq u_n \leq u_{max} \tag{5}$$

and state equation

$$x_{n+1}=f(x_n, u_n, y_n) \tag{6}$$

in which the initial state of the system, x_0, is given. If a final reservoir level would be restricted in order to schedule periodically, then terminal level could be considered as a final value. In this case x_n is also given. The problem is two boundary balue problem(is referred TBVP). These system equtations not only describe the movement of water through the system but also describe the transition of the state variable(x) from n stage to $n+1$ stage. This problem is conceptually formulated by using Dynamic Programming(DP) or Maximum Principle(MP). The constraints are arranged so that conceptual mode of this problem is changed into solvable mode. If equation(4) and (5) is invertible, then

$$a(x_n) \leq u_n \leq b(x_n) \tag{7}$$

or $a(x_n)-u_n=g_1(x_n, u_n) \leq 0$

$u_n-b(x_n)=g_2(x_n, u_n) \leq 0$

more generally $g(x_n, u_n) \leq 0$ and the terminal constraints is formulated as follows

$$h_N(x)=x_N-\hat{x}_N \tag{8}$$

or $h(x_n, u_n)=0$

where $\hat{x}_N$ means a given value. This problem can be formulated by mathematical programmings. But such methods require computational amount proportional to N, while stage wise methods require it linear in N. These consideration leads us to a stage wise method for control problem (for example, DP or MP).

The Dynamic Programming is a stagewise procedure which, in principle, enable one to determine the optimal solution. This method requires the construction of $u_n(x)$ for $n=N-1, N-2, \cdots 0$, where for each x,

$$u_n(x)=\arg\min_u[L_n(x_n, u_n)+V_{n+1}(f_n(x_n, u_n))] \quad \cdots(9)$$

the optimal value functions V being determined recursively by $V_N(x)=0$ and

$$V_n(x)=L_n(x_n, u_n(x))+V_{n+1}(f_n(x_n, u_n)) \tag{10}$$

The function $u_n(x)$ having been computed for $n=N-1, \cdots 0$. The basic trouble with this prototypical DP algorithm is that with the exception of certain rare instances the functions

$u_n(x)$ and $V_n(x)$ cannot be conveniently represented in the computer. It is well known that this conventional DP requires the storage of $V_{n+1}(x_{n+1})$ for appropriate latice points of x_{n+1} and the comparisons of values of expression $L_n(x_n, u_n)+V_{n+1}(f_n(x_n, u_n))$ at all admisible controls u_n for all lattice points of x_n. Since this approach is computationally impossible when $m \geq 5$, we following the line of Jacobson and Mayne[13] in the reservoir control problem, this implies that all other factors are remaining the same, while the memory and computational burden grows exponentially with the number of reservoirs in a system.

The effort toward overcoming this limitations are techniques known collectively as successive approximation methods which orginated from Bellman and Drey fus[22]. This kind of a traditional DDP can hardly solve optimal control problems with inequality constraints on state variables. The new DDP[21] algorithm is based upon Kuhn-Tucker conditions and is composed of iterative methods for solving systems of nonlinear equations. It is shown in following section that the new DDP with Newton's method can overcome "discritization and constraints" problem.

3. Optimal Conditions

Let $\{x_n^0;\ n=0, \cdots N\}$ be the optimal trajectory corresponding to the optimal control $\{u_n^0;\ n=0,\ N-1\}$. All the functions $f_n, g_n, L_n (n=0, \cdots N-1)$ and L_N are twice differentiable. For scalar functions L_n, $\nabla_u L_n$ and $\nabla_u^2 L_n$ mean the gradient row vector and the Hessian matrix of L_n with respect to u_n, respectively. And for vector functions f_n, $\nabla_v f_n$ and $\nabla_u^2 f_n$ mean the Jacobian matrix and the second derivative of f_n with respect to u_n, respectively. That is, $\nabla_u L_n=(\partial L_n/\partial u_{n1} \ldots,\ \partial L_n/\partial u_{nl})$, $\nabla_u^2 L_n=(\partial L_n/\partial u_{ni}\,\partial u_{nj})$, $\nabla_v f_n=(\partial f_{ni}/\partial u_{nj})$ and for any m-dimensional column vector z, $z^T \nabla_u^2 f_n=\sum_{i=1}^{m} z_i \nabla_u^2 f_{ni}$, where z^T denotes the transpose of z. Let L_n^0 and $\nabla_u L_n^0$ stand for $L_n(x_n^0, u_n^0)$, $\nabla_u L_n(x_n^0, u_n^0)$ and so on respectively.

Define the Lagrangian fnctions $F(n=0, \ldots, N-1)$ for one variable minimization problem of equation (10) as

$$F_n(x_n, u_n, \lambda_n, \mu_n)=L_n(x_n, u_n) + V_{n+1}(f_n(x_n, u_n)) + \lambda_n^T g_n(x_n, u_n) + \mu_n^T h_n(x_n, u_n) \qquad (11)$$

where Lagrange multipliers λ_n and μ_n are m_n and l_n dimensional column vectors. If V_{n+1} is twice continuously differentiable, then the following Kuhn-Tucker conditions hold as second-order necessary conditions that u_n^0 be an optimal solution of equation (10). There exist λ_n^0 and μ_n^0 such that

$$\nabla_u F_n^0=\nabla_u L_n^0+\nabla V^0_{n+1}\nabla_u f_n^0+\lambda_n^{0T}\nabla_u g_n^0 + (\mu^0)^T \nabla_u h_n^0=0 \qquad (12)$$

$$\mathrm{diag}(\lambda_n^0) g_n^0=0,\quad h_n^0=0 \qquad (13)$$

$$g_n^0 \leq 0,\quad \lambda_n^0 \geq 0 \qquad (14)$$

and such that for every vecto z satisfying $\nabla_v g_n^0 z=0$ and $\nabla_u h_n^0 z=0$,

$$z^T \nabla_u^2 f_n^0 z \geq 0 \qquad (15)$$

where $\mathrm{diag}(\lambda_n)$ denotes the diagonal matrix with the i-th diagonal element λ_n^i and

$$\nabla_u^2 F_n^0=\nabla_u^2 L_n+\nabla V_{n+1}^0 \nabla_u^2 f_n^0 + (\lambda_n^0)^T \nabla_u^2 g_n^0+(\mu_n^0)^T \nabla_u^2 h_n^0 + (\nabla_u f_n^0)^T \nabla^2 V_{n+1}^0 \nabla_u f_n^0 \qquad (16)$$

4. New Differential Dynamic Programming Algorithm

Put $w_n=(u_n^T, \lambda_n^T, \mu_n^T)^T$ for $n=0, \cdots N-1$ and define $T_n(x_n, w_n)$

$$=(\nabla_u F_n, g_n^T\ \mathrm{diag}(\lambda_n), h_n^T)^T \qquad (17)$$

For fixed x_n, $T_n(x_n, w_n)=0$ is a system of $(1+m_n+l_n)$ equations for the same number of unknowns (for example unknowns of $u=l\times n$, $\lambda=m\times n$, $\mu=1$). In addition, conditions (12) and (13) can be rewritten as $T_n(x_n^0, w_n^0)=0$. Therefore, if w_n^0 is an isolated solution of $T_n(x_n^0, w_n)=0$, that is, if there exists a neighborhood of x_n^0 which contains no other solutions of $T_n(x_n^0, v_n)=0$, then w^0 can be obtained by solving $T_n=0$, in its appropriate neighborhood without taking into consideration inequality(14). From inverse function theorem[23] it follows that if the Jacobian matrix of T_n with respect to w_n is nonsingular at w_n^0, then w_n^0 is an isolated solution of $T_n(x_n^0, w_n)=0$.

The Jacobian matrix of T_n, denoted J_n, is given by

$$J_n(x_n, w_n)=\begin{pmatrix} \nabla_u{}^2F_n & \nabla_u g_n{}^T & \nabla_u h_n{}^T \\ \mathrm{diag}(\lambda_n)\nabla_u g_n & \mathrm{diag}(g_n) & 0 \\ \nabla_u h_n & 0 & 0 \end{pmatrix} \quad (18)$$

The Jacobian matrix of T_n with respect to x_n, denoted by K_n, is given by

$$K_n(x_n, w_n)=(\nabla_{ux}{}^2F_n{}^T, \nabla_x g_n{}^T \mathrm{diag}(\lambda_n), \nabla_x h_n{}^T)^T \quad (19)$$

where

$$\nabla_{ux}{}^2F_n=\nabla_{ux}{}^2L_n+\nabla_u f_n{}^T\nabla^2 V_{n+1}\nabla_x f_n +\nabla V_{n+1}\nabla_{ux}{}^2 f_n+\lambda_n{}^T\nabla_{ux}{}^2 g_n+\mu_n{}^T\nabla_{ux}{}^2 h_n \quad (20)$$

if we assume J_n^0 is nonsingular, then $w_n^0(x_n)=(u_n^0(x_n)^T\lambda_n^0(x_n)^T, \mu_n^0(x_n)^T)$ is an isolated solution of $T_n(x_n, w_n)=0$ for fixed x_n belonging to a neighborhood x_n of x_n^0. Moreover $V_n(x_n)$ is twice continuously differentiable in x_n and

$$\nabla V_n(x_n)=\nabla_x L_n+\nabla V_{n+1}\nabla_x f_n+\lambda_n^0(x_n)^T\nabla_x g_n +\mu_n^0(x_n)^T\nabla_x h_n \quad (21)$$

$$\begin{aligned}\nabla^2 V_n(x_n)=&\nabla_x{}^2L_n+\nabla_x{}^2 f_n\nabla^2\nabla_{n+1}\nabla_x f_n \\ &+\nabla V_{n+1}\nabla_x{}^2 f_n+\lambda_n^0(x_n)^T\nabla_x{}^2 g_n \\ &+\mu_n^0(x_n)^T\nabla_x{}^2 h_n+(\nabla_{xu}{}^2L_n \\ &+\nabla_x f_n{}^T\nabla^2 V_{n+1}\nabla_u f_n+\nabla V_{n+1}\nabla_{xu}{}^2 f_n \\ &+\lambda_n^0(x_n)^T\nabla_{xu}{}^2 g_n+\mu_n^0(x_n)^T\nabla_{xu}{}^2 h_n) \\ &\cdot\nabla u_n^0(x_n)+\nabla_x g_n{}^T\nabla\lambda_n^0(x_n) \\ &+\nabla_x h_n{}^T\nabla\mu_n(x_n) \end{aligned} \quad (22)$$

where all functions assume values at $(x_n, u_n^0(x_n))$ and $w_n^0(x_n)$ is given by

$$\nabla w^0(x_n)=\begin{pmatrix} \nabla u_n^0(x_n) \\ \nabla\lambda_n^0(x_n) \\ \nabla\mu_n^0(x_n) \end{pmatrix} =-J_n^{-1}(x_n, w_n^0(x_n)) K_n(x_n, w_n^0(x_n)) \quad (23)$$

This direction matrix leads us to a optimal solution and was proved by Ohno[21]. He suggested the following numerical scheme. Denote an arbitrary iteration procedure for solving the system of nonlinear equations $T_n(x_n, w_n)=0$ with fixed x_n by

$$w_n{}^{k+1}=U_n(x_n, w_n{}^k),\quad k=0, 1, 2, \cdots \quad (24)$$

For example, Newton's method is described as

$$U_n(x_n, w_n{}^k)=w_n{}^k-J_n{}^{-1}(x_n, w_n{}^k) T_n(x_n, w_n{}^k) \quad (25)$$

Let $\{m_n^0;\ n=0, \cdots, N-1\}$ be given, and let $\{x_n^0; n=0\ldots, N\}$ be the trajectory corresponding to $\{u_n^0\}$. Then a conceptual algorithm is as follows. Calculate $w_n{}^{k+1}$ by $w_n{}^{k+1}=U_n(x_n{}^k, w_n{}^k)$ for $n=N-1, \cdots, 0$. It is to be noted, however, that T_n, J_n, and $K_n(n=0, \cdots, N-2)$ include unknown values $\nabla V_{n+1}(x_{n+1}{}^k)=\nabla^2 V_{n+1}(x_{n+1}{}^k)$. Consequently it is essential to obtan their approximate values which guarantee that $\{w_n^0\}$ is a point of attraction of the following Ohno's DDP algorithm, there exist open neighborhoods W_n of $w_n^0(n=0, \cdots, N-1)$ such that for any $w_n^0\in W_n$, $w_n{}^k(k=0, 1, \cdots)$ generated by the algorithm remain in W_n and converge to $w_n{}^2$(23). Since exact values $\nabla V_n(x_n)$, $\nabla^2 V^u(x_n{}^k)$ and $\nabla w_n^0(x_n{}^k)$ are given, such approximate values will be obtained by approximating appropriately (21) through (23). Denote by $\nabla\tilde{V}_n{}^k$ and $\nabla^2\tilde{V}_n{}^k(n=1,\ N-1\ldots, k=0, 1, 2, \cdots)$ the approximate values of $\nabla V_n(x_n{}^k)$ and $\nabla^2 V_n(x_n{}^k)$, respectively.

New DDP alforithm proposed by Ohno is as follows,

Let w; $n=0, \cdots N-1$ and x; $n=0, \cdots, N$ be given, and set $k=0$.

Step 1; By using $\nabla V_N = \nabla L_N$ and $\nabla^2 V_N = \nabla^2 L_N$, calculate $\tilde{w}_{N-1}{}^{k+1}$, $\nabla \tilde{V}_{N-1}{}^k$ and $\nabla^2 \tilde{V}_{N-1}{}^k$ by

$$\tilde{w}_{N-1}{}^{k+1} = U_{N-1}(x_{N-1}{}^k, w_{N-1}{}^k) \quad (26)$$

$$\nabla \tilde{V}_{N-1}{}^k = \nabla_x L_{N-1} + \nabla V_N(f_{N-1}) \nabla_x f_{N-1} + (\lambda_{N-1}{}^{k+1})^T \nabla_x g_{N-1} + (\tilde{\mu}_{N-1}{}^{k+1})^T \nabla_x h_{N-1}, \quad (27)$$

and

$$\begin{aligned}\nabla^2 \tilde{V}_{N-1}{}^k = {} & \nabla_x{}^2 L_{N-1} + \nabla_x f_{N-1}{}^T \nabla^2 f_{N-1}{}^T \nabla^2 V_N \nabla_x f_{N-1} \\ & + \nabla V_N \nabla_x{}^2 f_{N-1} + (\lambda_{N-1}{}^k)^T \nabla_x{}^2 g_{N-1} \\ & + (\mu_{N-1}{}^k)^T \nabla_x{}^2 h_{N-1} - (\nabla_{xu}{}^2 L_{N-1} \\ & + \nabla_x f_{N-1}{}^T \nabla^2 V_N \nabla^2 V_N \nabla_u f_{N-1} \\ & + \nabla V_N \nabla_{xu}{}^2 f_{N-1} + (\lambda_{N-1}{}^k)^T \nabla_{ux}{}^2 g_{N-1} \\ & + (\mu_{N-1}{}^k)^T \nabla_{xu}{}^2 h_{N-1}) [J_{N-1}{}^{-1} K_{N-1}]_u \\ & - \nabla_x g_{N-1}{}^T [J_{N-1}{}^{-1} K_{N-1}]_\lambda \\ & - \nabla_x h_{N-1}{}^T [J_{N-1}{}^{-1} K_{N-1}]_\mu \end{aligned} \quad (28)$$

where all functions except $\nabla_u f_n$ in (27) assume values at $(x_{N-1}{}^k, \tilde{u}_{N-1}{}^{k+1})$, all functions except ∇V_N and $\nabla^2 V_N$ in (28) assume values at $(x_{N-1}{}^k, u_{N-1}{}^k)$ and ∇V_N & $\nabla^2 V_N$ assume values at $x_N = x_N$.

Step 2: For $n = N-2, \cdots, 1$, calculate $\tilde{w}_n{}^{k+1}$, $\nabla \tilde{V}_n{}^k$ and $\nabla^2 \tilde{V}_n{}^k$ by

$$\tilde{w}_n{}^{k+1} = \tilde{U}_n(x_n{}^k, w_n{}^k) \quad (29)$$

$$\begin{aligned}\nabla \tilde{V}_n{}^k = {} & \nabla_x L_n + \nabla \tilde{V}_{n+1}{}^k \nabla_x f_n \\ & + (u_n{}^{k+1} - u_n{}^k)^T \nabla_u f_n(x_n{}^k, u_n{}^k)^T \nabla^2 \tilde{V}_{n+1}{}^k \nabla_x f_n \\ & + (\lambda_n{}^{k+1})^T \nabla_x g_n + (\tilde{\mu}_n{}^{k+1})^T \nabla_x h_n \end{aligned} \quad (30)$$

and

$$\begin{aligned}\nabla^2 \tilde{V}_n{}^k = {} & \nabla_x{}^2 L_n + \nabla_x f_n{}^T \nabla^2 \tilde{V}_{n+1}{}^k \nabla_x f_n + \nabla \tilde{V}_{n+1}{}^k \nabla_x{}^2 f_n \\ & + (\lambda_n{}^k)^T \nabla_x{}^2 g_n + (\mu_n{}^k)^T \nabla_x{}^2 h_n \\ & - [\nabla_{xu}{}^2 L_n + \nabla_x f_n{}^T \nabla^2 \tilde{V}_{n+1}{}^k \nabla_u f_n + \nabla \tilde{V}^k{}_{n+1} \nabla_{xu}{}^2 f_n \\ & + (\lambda_n{}^k)^T \nabla_{xu}{}^2 g_n + (\mu_n{}^k)^T \nabla_{xu}{}^2 h_n] [\tilde{J}_n{}^{-1} \tilde{K}_n]_u \\ & - \nabla_x g_n{}^T [\tilde{J}_n{}^{-1} \tilde{K}_n]_\lambda - \nabla_x h_n{}^T [\tilde{J}_n{}^{-1} \tilde{K}_n]_\mu \end{aligned} \quad (31)$$

where all functions except $\nabla_u f_n$ in (30) assume values at $(x_n{}^k, \tilde{u}_n{}^{k+1})$ and all functions in (31) assume values at $(x_n{}^k, u_n{}^k)$.

Step 3: By using the initial condition $x_0 = \tilde{x}_0$, calculate w^{k+1} by

$$w_0{}^{k+1} = \tilde{U}_0(\hat{x}_0, w_0{}^k) \quad (32)$$

Step 4: For $n = 1, \cdots, N-2$, calculate $x_n{}^{k+1}$ and $w_n{}^{k+1}$ by

$$x_n{}^{k+1} = f_{n-1}(x_{n-1}{}^{k+1}, u_{n-1}{}^{k+1}) \quad (33)$$

and

$$w_n{}^{k+1} = \tilde{w}_n{}^{k+1} - [\tilde{J}_n{}^{-1} \tilde{K}_n(x_n{}^k, w_n{}^k)] (x_n{}^{k+1} - x_n{}^k) \quad (34)$$

Step 5: Calculate $x_{N-1}{}^{k+1}$, $w_{N-1}{}^{k+1}$, and $x_N{}^{k+1}$ by

$$x_{N-1}{}^{k+1} = f_{N-2}(x_{N-1}{}^{k+1}, u_{N-2}{}^{k+1}) \quad (35)$$

$$w_{N-1}{}^{k+1} = w_{N-1}{}^{k+1} - [J_{N-1}{}^{-1} K_{N-1} (x_{N-1}{}^k, w_{N-1}{}^k)] \quad (36)$$

and

$$(x_{N-1}{}^{k+1} - x_{N-1}{}^k)$$

if $\max_m \|w_n{}^{k+1} - w_n{}^k\| < \varepsilon$, then stop; otherwise set $k = k+1$ and go back to Step 1, where ε is a small given number and $\| \cdot \|$ denotes l_1 norm.

5. Computational Studies in Multireservoir Control

Ohno's DDP algorithm has been applied to multireservoir control problems. The first of these problems was introduced into the literature by Larson[2] and subsequently served as an illustrative example for discrete differential dynamic programming in an investigation by Heidari et al.[9] as well as an example of multilevel incremental dynamic programming in the work of Nopmongcol and Askew[8]. The second example was presented by Chow and Cortes Bivera[10] to illustrate DDP with adaptive corridor with selection. Also Murray and Yakowitz[13] dealt with this example to illustrate DDP with Flectcher[14] algorithm. But this example was not presented here because it is same as first example except inflows which is constant variable in the state equation. Our final computational study provides the solution of a control problem as same as second example except having nonlinear objective function.

In order to understand the characteristics of the models of three problems are tabulated in Table 1. Model A and B is called "Singular arcs Control" problem which is difficult to

Table 1. Characteristics of three problems

Problems / Characteristic of Model	A : Larson Problem	B : Chow and Cortes Rivera's Problem	C : W. Yeh type Objetcive function
State equation	$x_{n+1}=x_n+Mu_n+y_n$	$x_{n+1}=x_n+Mu_n+y_n$	$x_{n+1}=x_n+Mu_n+y_n$
Inflows	y_n=Constant	y_n=Variable	y_n=Variable
Constraints	$a(x_n)\leq u_n\leq b(x_n)$	$a(x_n)\leq u_n\leq b(x_n)$	$a(x_n)\leq u_n\leq b(x_n)$
Initial and terminal value	Given	Given	Given
Objective function	Linear	Linear	Nonlinear

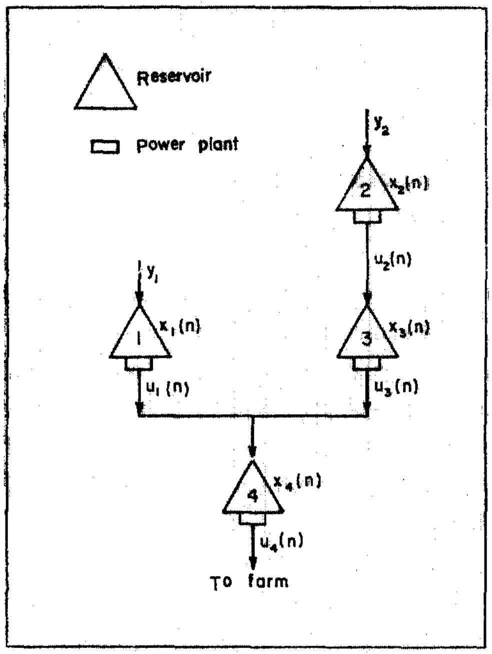

Fig. 1. Reservoir network of a simplified system.

solve. There are several methods to relaxe this difficulty. The popular one is the trans formation of the original problem into linear state, Quadratic objective function problem. New DDP can be applied to this modified model.

In multireservoir control problem, the discussion of this first part is particularly lengthy because here some fine points (applicable to remaining problem) of New DDP implementation are described. It is presumed that the four reservoirs comprising the system have the configuration shown in Figure 1. For the problem at hand the law of motion (equation (1))

can be written,

$$\begin{pmatrix} x_{n+1}^1 \\ x_{n+1}^2 \\ x_{n+1}^3 \\ x_{n+1}^4 \end{pmatrix} = \begin{pmatrix} x_n^1 \\ x_n^2 \\ x_n^3 \\ x_n^4 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} u_n^1 \\ u_n^2 \\ u_n^3 \\ u_n^4 \end{pmatrix} + \begin{pmatrix} y_n^1 \\ y_n^2 \\ y_n^3 \\ y_n^4 \end{pmatrix} \qquad (37)$$

or

$$x_{n+1}=x_n+Mu_n+y_n, \quad 1\leq n\leq 12$$

Additionally state constants are imposed to keep the nonegativity is storage and lies within the reservoir capacity. The state space constraints were taken as

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} x_n^1 \\ x_n^2 \\ x_n^3 \\ x_n^4 \end{pmatrix} \leq \begin{pmatrix} 10 \\ 10 \\ 10 \\ 15 \end{pmatrix} \quad 1\leq n\leq 12 \qquad (38)$$

The initial state is taken to be $(5, 5, 5, 5)^T$, and the terminal state is constrained to be $(5, 5, 5, 7)^T$. The final decision time step N is 12. Furthermore, for each n, $y_n^1=2$, and $y_n^2=3$. The release constraints are determined by the condition that

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} u_n^2 \\ u_n^2 \\ u_n^3 \\ u_n^4 \end{pmatrix} \leq \begin{pmatrix} 3 \\ 4 \\ 4 \\ 7 \end{pmatrix} \quad 1\leq n\leq 12 \qquad (39)$$

The control and state constraints have been combined by Murray and Yakowitz as follows.

$$Bx_n+b_n\leq u_n\leq Ax_n+a_n \qquad (40)$$

$$g_1(x_n, u_n) = -Iu_n + Ax_n + a_n \quad (41)$$

$$g_2(x_n, u_n) = Iu_n - Bx_n - b_n \quad (42)$$

where I is $m \times m$ identity matrix, A and B $m \times m$ matrix, a_n and b_n $m \times 1$ vector respectively and these constants was given by Murray and Yakowitz.

The one procedural fine point remaining to be discussed, is how to obtain positive definate quadratic approximations to the linear loss function $L(x, u, n)$ defined by

$$L(x, u, n) = \sum_{j=1}^{4} C_{j,n} u_{j,n} \quad (43)$$

The loss coefficients $C_{j,n}$ are given Table 2. It is sufficient to describe the method used to obtain an approximation to function-u. It is approximated by

$$Q(x, u, n) = -C^T u + u^T D u \quad (44)$$

where D is an sufficient small diagonal constant matrix.

Finally for our experiments terminal constriants are transformed into

$$h(x_N, u_N) = x_{N-1} + Mu_{N-1} + y_{N-1} - \hat{x}_N \quad (45)$$

The functional with respect to the equations (41)~(45) is

$$F(x_n, \lambda_n^1, \lambda_n^2) = Q(x_n, u_n) + v_{n+1}(x_{n+1}) + \lambda_n^{1T} g_1(x_n, u_n) + \lambda_n^{2T} g_2(x_n, u_n) \quad \text{for } n = 1, \cdots, N-1 \quad (49)$$

$$F_N(x_N, \lambda_N^1, \lambda_N^2) = Q(x_N, u_N) + v_{N+1}(x_{N+1}) + \lambda_N^{1T} h(x_N, u_N) + \lambda_N^{2T} g_2(x_N, u_N) \quad (47)$$

$$\text{for } n = N$$

Test run for analysis of influence of D:

Fig. 2 indicates changes in total benefit during the iteration. The circled line represents increase in total benefit when d is selected as $d^* = 0.87d$ in every step of iteration. In this test run, $\in$ was set as $\in = 10^4$ and the nominal control $u_n^0 = \{2, 3, 3, 5\}$ and the initial value $\lambda_1^0 = \{5, 5, 5, 5\}$, $\lambda_2^0 = \{500, 500, 500, 500\}$. In this computation we found that d^* increase the value of objective function until singular points appeared in the Jacobian. Unfortunately state variables are violating constraints before objective function reaches an optimum value (Fig. 3). The dotted line with $d = 0.3350$ are quadratically converging in Figure 2. Also the effect has been understood so that constraints of objective function is more dominant than state constraints during the period of large value C, otherwise states do not cut the boundaries. This test run give us two clues to the problem, that is, how to select "d" and how to evalute the constraints relatively.

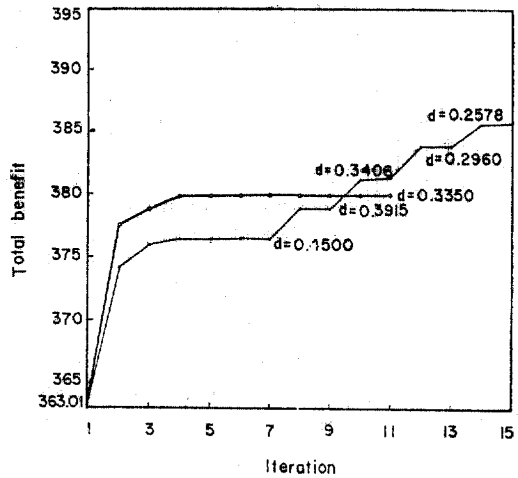

Fig. 2. Total benefit for the test A5

Table 2. Time functions used in calculating benefit

k	$c_1(k)$	$c_2(k)$	$c_3(k)$	$c_4(k)$	$c_5(k)$
0	1.1	1.4	1.0	1.0	1.6
1	1.0	1.1	1.0	1.2	1.7
2	1.0	1.0	1.2	1.8	1.8
3	1.2	1.0	1.8	2.5	1.9
4	1.8	1.2	2.5	2.2	2.0
5	2.5	1.8	2.2	2.0	2.0
6	2.2	2.5	2.0	1.8	2.0
7	2.0	2.2	1.8	2.2	1.9
8	1.8	2.0	2.2	1.8	1.8
9	2.2	1.8	1.8	1.4	1.7
10	1.8	2.2	1.4	1.1	1.6
11	1.4	1.8	1.1	1.0	1.5

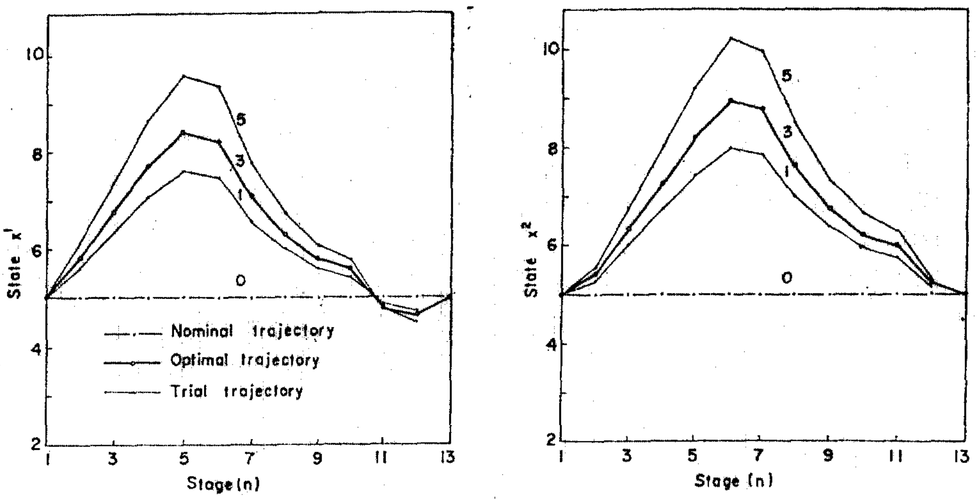

Fig. 3(1). Optimal trajectory for the test A5

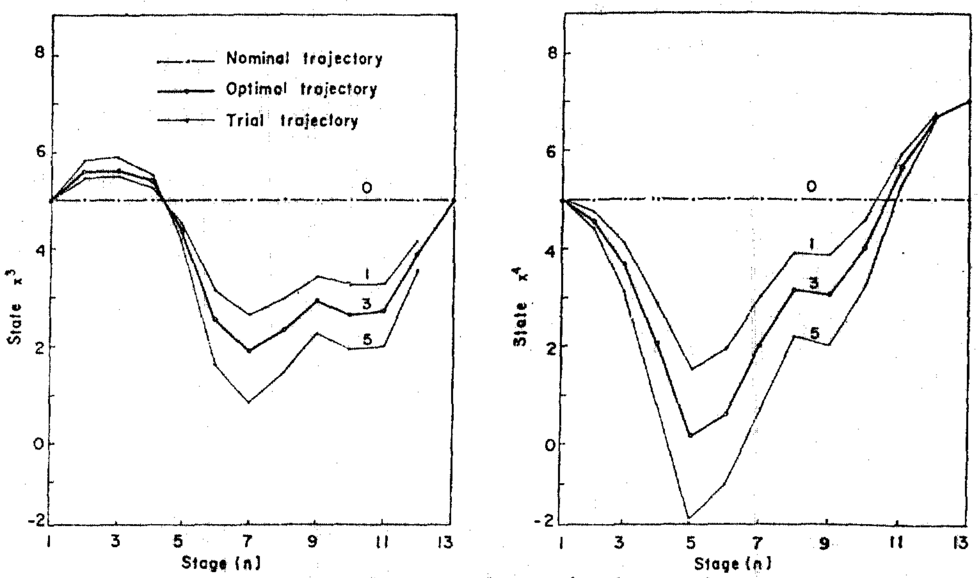

Fig. 3(2) Optimal trajectory for the test A5

Multireservoir control problem(A)

In this example the same nominal values and initial values and d with the same value as test run have been used. But the priniciple of the relative constraints has been applied to the model not to violate some states constraints. This technical operation is very usual way. If the program succedes in running without violation, then it plays the role that controls force to release water from reservoir even in low benefit season. So we can obtain more benefit in this modified model than in unadaptable methods. The computation results for this example are summerized in Fig. 4, 5. But it has not succeded in running the model of this problem with Murray and Yakowitz's objective function (12).

In Table 3 Murray and Yakowitz have compared the performance of his DDP and DDDP. Problem 1 was also solved by a DPSA algorithm by Larson[2]. His computations took 30s on a B-5500 computer.

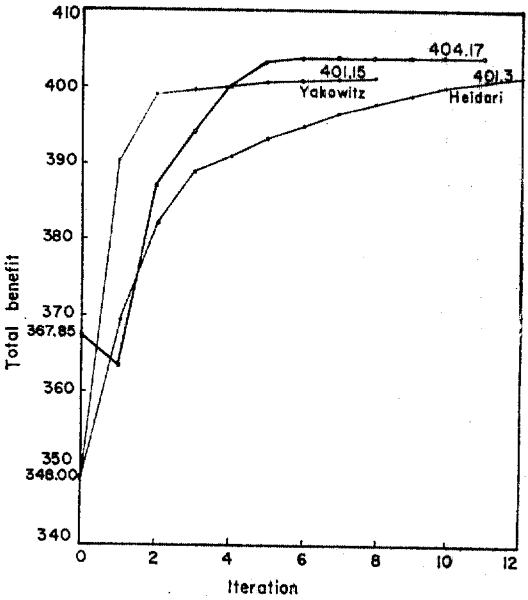

Fig. 4. Total benefit for the problem A

It should be pointed out that problem 1 is particularly well suited to methods such as DDDP and DPSA which require discretization of state space. It is reported by Heidari et al.[8] that when noninteger corridor width was used, 18 DDP iterations were needed to obtain a return of 399.06. However only 4 New DDP iterations has been required to reach a return of 400 without discretization of state space. Consecution time of every iteration in this study took about 300 400milli second on FACOM.

Fig. 5 displays comparison of trajectories with Larson's. Any other model didn't present their state trajectory. In this figure Larson's model might yield nonoptimal state trajectory because they use the discretized state value as

Table 3. Computational effort for problem 1.

Initial trajectory	D D P			D D D P		
	Time,s	Iterations	Final value	Time,s	Iterations	Final value
1	9.63	3	401.197	35.32	8	401.3
2	19.30	6	401.274	48.39	13	401.3
3	25.7	8	401.151	31.04	8	401.3

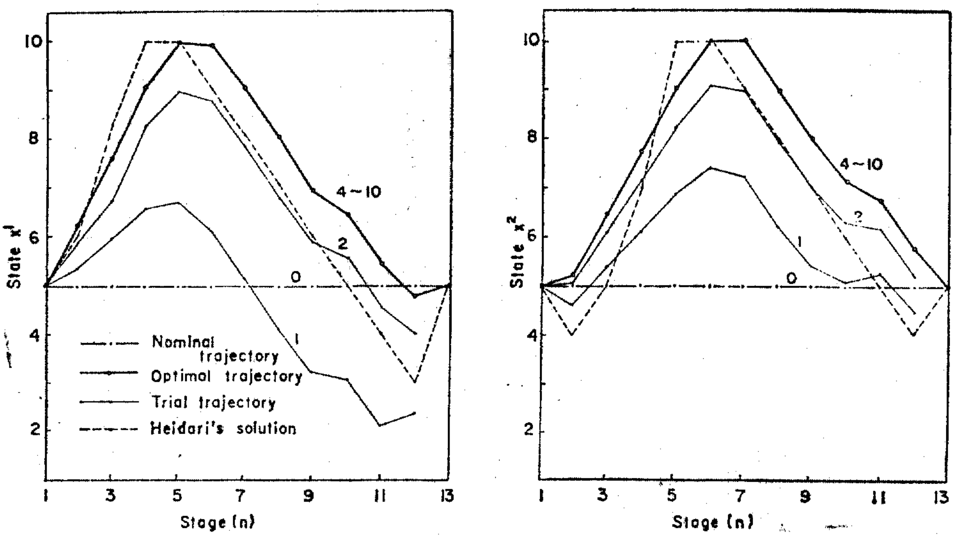

Fig. 5(1) Optimal trajectory for the problem A

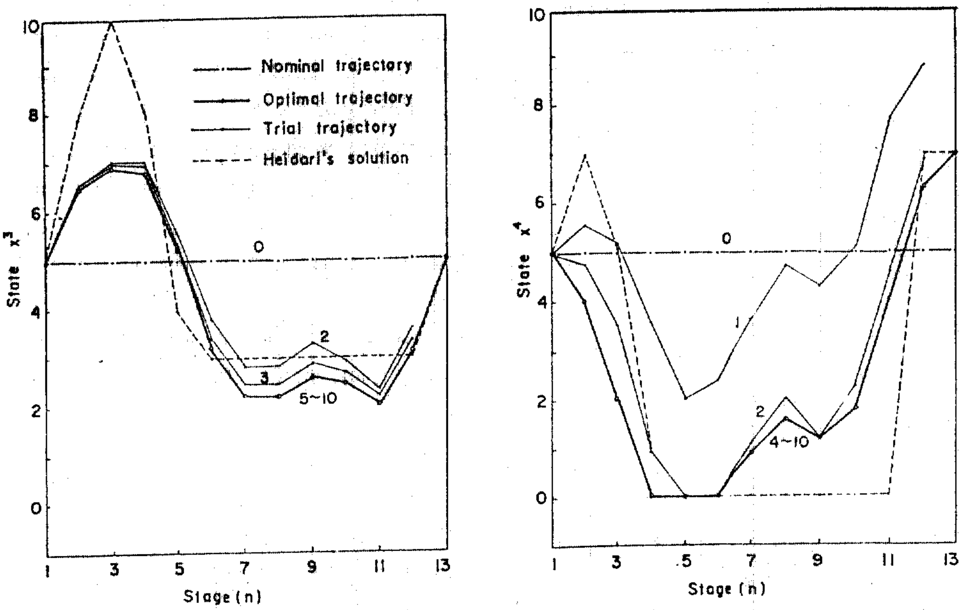

Fig. 5(2). Optimal trajectory for the problem A

integer. The trajectories can't be compared with the results of Murrary and Yakowitz which is very recent study, because they didn't refer to that.

Multireservoir control problem (C)

This problem, introduced by Chow and Cortes-Rivera[10], is essentially the same as problem 1, the difierence being that the inflow and constraint parameters are chosen so that consequently for methods(such as DDDP and DP-SA) which use state discretization, adaptive step-size selection will have to be employed. For Murrary and Yakowitz's DDP and new DDP state variable is not discretized and it makes no difference whether the solution componets are integer valued. The inflow parameters are given in Table 4 a, the maximum storages

Table 4 a. Inflow values y_n^j

n	Value of j	
	1	2
1	0.5	0.4
2	1.0	0.7
3	2.0	2.0
4	3.0	2.0
5	3.5	4.0
6	2.5	3.5
7	2.0	3.0
8	1.25	2.5
9	1.25	1.3
10	0.75	1.2
11	1.75	1.0
12	1.0	0.7

Table 4 b. Maximum permissible storage values for x_n^j

n	Value of j			
	1	2	3	4
2	12.0	15.0	8.0	15.0
3	12.0	15.0	8.0	15.0
4	10.0	15.0	8.0	15.0
5	9.0	12.0	8.0	15.0
6	8.0	12.0	8.0	15.0
7	8.0	12.0	8.0	15.0
8	9.0	15.0	8.0	15.0
9	10.0	17.0	8.0	15.0
10	10.0	18.0	8.0	15.0
11	12.0	18.0	8.0	15.0
12	12.0	18.0	8.0	15.0

aregiven in Table 4 b, the minimum storage is 1.0 for all n 1. The nominal values are same as in problem (A). We don't need d any more because objective function is nonlinear. The minimum permissiable storage in any reservoir during times 2 through 12 is 1.0 for each n, and releases u_n are constrained by

$$\begin{pmatrix} 0.005 \\ 0.005 \\ 0.005 \\ 0.005 \end{pmatrix} \leq u_n \leq \begin{pmatrix} 4.0 \\ 4.5 \\ 4.5 \\ 8.0 \end{pmatrix} \quad 1 < n < 12 \quad (48)$$

For this problem the initial state and final states are required to be $(6, 6, 6, 8)^T$ and $(6, 6, 6, 8)^T$, respectively. In this study the objective function of Cortes-Rivera'a model into model C have been modified, in order to test the feasibility of this method about arbitrary nonlinear system. As an example the type of objective function have been selected which appeared in Yeh study[20]. This objective function is

$$L(x, u, n) = \sum_{j=1}^{4} C_j u_{j,n}{}^a \qquad (49)$$

here $a<1$, in the computation "a" has been taken by 0.7. In Yeh's study there were not any informations about inflows and cost coefficients. Therefore the results have not been compared with theirs.

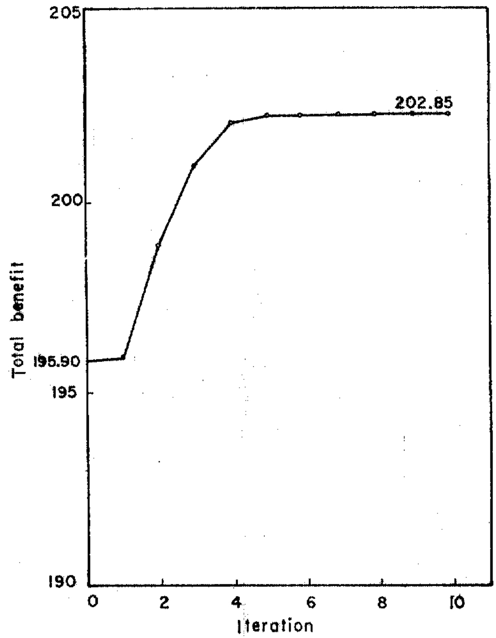

Fig. 6. Total benefit for the problem C

Fig. 6 depicts the convergence rate of the total benefit and Fig. 7 displays the behavior of state variable and optimal state trajectory.

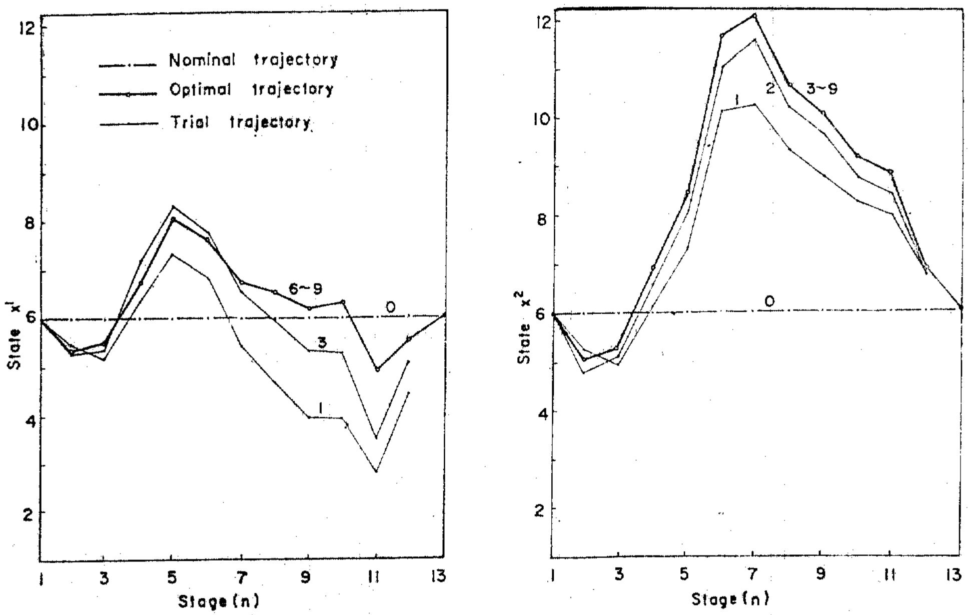

Fig. 7(1). Optimal trajectory for the problem C

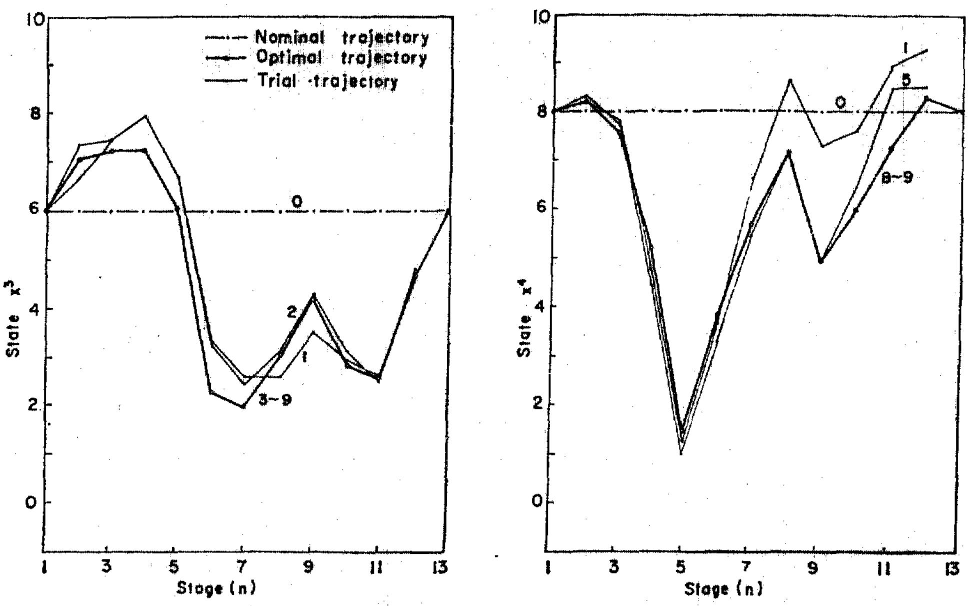

Fig. 7(2). Optimal trajectory for the problem C

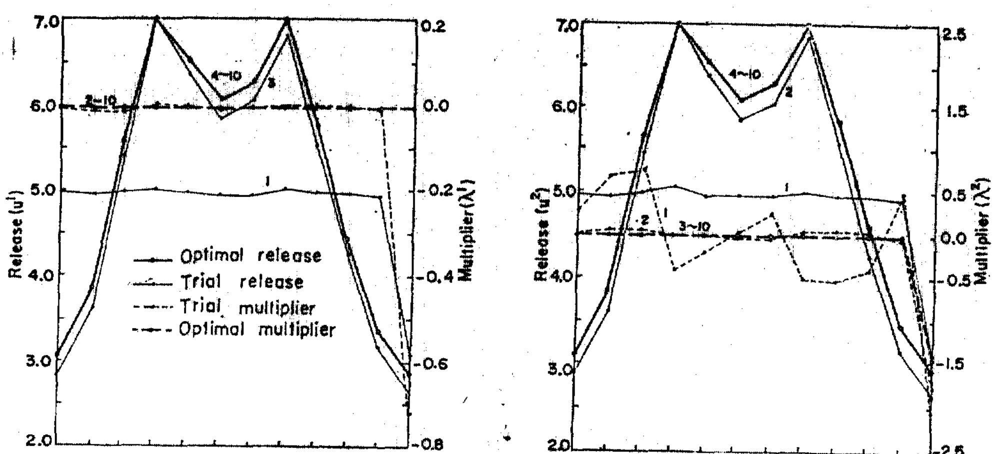

Fig. 8. Behavior of Lagrangian multipliers

Finally in order to show that Lagrangian multipliers play the role to-restrain the violation of the constrants, that is, how Kuhn-Tucker conditions are satisfied (Fig. 8). But there are so many multipliers that they can't be depicted all. In Figure 8 arbitrary two trajectories of Lagrangian multipliers have been presented. This figure indicates that Lagrangian multiplier is non-negative. Therefore it can be assured that all the solution is optimal.

6. Conclusion

The new DDP algorithm has been applied to solving discrete time optimal control problems with equality and inequality constraints on both control and state variables. Its convergence has been proved. The present algorithm makes full use of the structure of discrete time optimal control problems, and in this sense it is natural algorithm for solving discrete time constrained or unconstrained optimal control problems.

For theoretical reasons and numerical experiments the differential dynamic programming method seems to have the most attractive properties of any available numerical methods for multivariate dynamic programming problems. From a practical point of view, clearly the most interesting aspects of the New DDP method are the following;

(1) The memory and computational amounts are relatively small;

(2) State and control spaces need not to be discretized;

(3) No state variable decoupling(which characterizes the DPSA technique) is done;

(4) No taking care wether the relations between state and control variables are linear or non linear;

(5) No need to adopt two phase technique (as in Yeh[20]), whether constraints exist or not.

It is to be admitted that the structure of the problem studied here, with nonlinear function, is convenient. On the other hand experience in linear system is that if the requisite derivatives of the loss function, law of motion and constraint are easily obtained, there is no essential difficulty in solving more complicated discrete time optimal control problem. Convergence rate of the modified objective function model proposed in this papaer is superior to Murray and Yakowitz's. In Murray and Yakowitz's model there were not the measure to confine control and state variable to constraints. But in this study, Lagrangian multiplier which is the function of state variable play the role of control not to violate constraints.

In order to start the present algorithm, it is essential to obtain an initial guess $\{{}^0w_n\}$ belonging to the neighborhood $\{w_n\}$ of $\{w_n^0\}$. This is a main drawback of the new DDP. However in this experience this difficulty can be easily covered. For example, nominal releases are selected so that water level stored in reservoir is always equal to initial value.

Finally this study has proved that terminal equality constraints has been successfully applied by adopting equality constraints instead of one of inequality constraints at the final stage.

REFERENCES

1. Tennessee Valley Authority, "Development of Water Resource Management Methods for the TVA Reservoir System, Project Status", June 1976, *A report for a short course, Div. of Water Manage., Water Syst. a Develop.* Br., Knoxville, Tenn., 1976.
2. R.E. Larson, "*State Increment Dynamic Programming*", American Elsevier Press, New York, 1676.
3. R.E. Larson and A.J. Korsak, "A Dynamic Programming Successive Approximations Technique with Convergence Proofs", *Automatica, vol. 6*, pp. 245~252, 1970.
4. A.J. Korsak and R.E. Larson, "A Dynamic Programming Successive Approximations Technique with Convergence Proofs(2)", *Automatica, Vol. 6*, pp. 253~260, 1970.
5. R.E. Larson and W.G. Keckler, "Applications of Dynamic Programming to the Control of Water Resource Systems", *Automatica, vol. 5*, pp. 15~26, 1969.

6. Trott, W.J., and Yeh, W. W-G., "Optimization of Multiple Reservoir Systems," *Journal of the Hydraulics Division, ASCE, Vol. 99, No. HY 10*, October 1973, pp. 1865~1884.
7. Yeh, W. W-G., "Real-Time Reservoir Operation: The California Central Valley Preojct Case Study," *Proceedings of the National Workshop on Reservoir Systems Operation*, University of Colordo, Boulder, Colorado, August 13~17, 1979, pp. 217~227.
8. P. Nopmongcol and A. Askew, "Multilevel Incremental Dynamic Programming", *Water Resources Research. 12(6)*, 1291~1297, 1976.
9. M. Heidari, V.T. Chow, P.V. Kokotovic and D.D. Meredith, "Discrete Differential Dynamic Programming Approach to Water Resources Systems Optimization", *Water Resources Research, vol. 7, No. 2*, April 1971.
10. V.T. Chow and G. Crotes-Rivera, "Applications of DDDP in Water Resources Planning", *Res. Rep. 78*, Univ. of Ill. Water Resour. Center, Urbana, 1974.
11. A. Turgeon, "Incremental Dynamic Programming may Yield Nonoptimal Solutions", *Water Resources Research, vol. 18, No. 6*, pp. 1599~1604, Dec. 1982.
12. D.M. Murray and S.J. Yakowitz, "Constrained Differential Dynamic Programming and Its Applications to Multireservoir Control", *Water Resources Research, vol. 15, No. 5*, Oct 1979.
13. D H Jacobson and D Q Mayne, *Differential Dynamic Programming*", Elsevier, New York, 1970.
14. R Fletcher, "A General Quardratic Programming Algorithm", *J. Inst Math. Its Appl., 7*, 76~91, 1971.
15. J H Lee, "The Necessity of Multireservoir Management and an Alternative of Release Rule", 전북 대학교 공학연구, vol 13, 1983.
16. Lee Jae-Hyoung, "A Basic Study for Optimum Opfrations of Multireservoirs for Water Utilizations", 충북 대학교 국문집, vol 123, 1982.
17. N B Nedeljkovic, "New Algorithms for Unconstrained Nonlinear Optimal Control Problems", *IEEE Transactions on Automatic Control, Vol. AC-26, No. 4*, Aug, 1981.
18. N.B. Nedeljkovic, "New Algorithms for Discrete-time Optimal Control Systems", *School of Mathematical and Physical Sciences Murdoch University*, Western Australia, Nov. 1982. 19.
19. D.Q. Mayne and E. Polak, "First-order Strong Variation Algorithms for Optimal Control", *J. of optimization Theory and Appl., vol. 16, No. 3/4*, 1975.
20. W.W-G. Yeh, A.M. ASCE, Leonard Becker, and Wen-Sen Chu, "Real-Time Hourly Reservoir Operation", *J. of the water Resources Planning and Management Division*, Sep. 1979.
21. K. Ohno, "A New Approach to Differential Dynamic Programming for Discrete Time Systems", *IEEE Transactions on Automatic Control, vol. AC-23, No. 1*, Feb. 1978.
22. Bellman. R. and S. Dreyfus, "*Applied Dynamic Programming*", Princeton University Press, Princeton, N. J., 1962.
23. J.M. Ortega and W.C. Rheinboldt, "*Iterative Solution of Nonlinear Equations in Several Variables,*" New York, Academic, 1970.

(接受：1984. 4. 28)