ON THE MODELING AND COMPUTATIONAL ASPECTS OF DYNAMIC PROGRAMMING WITH APPLICATIONS IN RESERVOIR CONTROL

by

Moshe Sniedovich[1] and Sidney J. Yakowitz[2]

ABSTRACT

A discrete stochastic multistage decision model is developed, in the framework of which the dynamic programming algorithm is defined. It is demonstrated that the algorithm may be used as a solution procedure for non-routine reservoir control problems. The study demonstrates that analytical considerations may be used to significantly reduce the amount of computation needed for the implementation of the algorithm. Two reservoir control problems are studied: The "reliability problem" in which one seeks a minimum-cost operation rule under a constraint which specifies the maximum probability of shortage allowed during the life-time of the project, and the "range problem" in which the objective is the minimization of the expected value of the range of fluctuation around a critical storage level.

INTRODUCTION

Although the dynamic programming is frequently used in the context of reservoir control problems there are indications (Askew, 1974b) that certain difficulties are involved in its implementation.

More specifically, there seems to be no modeling framework for complex reservoir control problems (as far as the hydrologic literature is concerned). An attempt is made in this paper to develop such a framework. For this purpose Yakowitz's (1969) model is modified so as to represent decision processes with non-additive reward functions.

The dynamic programming algorithm is defined in the context of the model, under the hypothesis of certain monotonicity properties of the reward function. The potential use of analytical consideration in reducing the number of iterations involved in the implementation of the algorithm is demonstrated.

Contrary to certain statements made recently in the hydrologic literature (Askew, 1974a, 1974b, 1975) it is shown that the dynamic programming algorithm can be used as a solution procedure for reservoir control problems in which reliability constraints are imposed on the process.

In particular, it is shown that reliability constraints imposed on the process in the form of the maximum probability of shortage during the life of the process can be incorporated into the dynamic programming approach.

It is also demonstrated that the dynamic programming algorithm can be used as a solution procedure in which the objective is the minimization of the expected value of the range of fluctuation of the storage levels around a critical value.

DISCRETE STOCHASTIC MULTISTAGE DECISION MODEL

The discrete stochastic multistage decision model under consideration is a tuple $Z = (N,\Omega,D,p_0,P,r)$ where:

(1) N is a positive integer indicating the number of *decision stages* involved in the decision-making process. The decision stages are denoted by n, $1 \le n \le N$. For example, N may define the number of periods (month, season, year, etc.) to be considered while operating a reservoir, and n indicates the n-th period.

(2) Ω is a finite set of elements x called *states*. The elements x are used for the description of the state of the process at the different decision stages. In the reservoir control problem x_n may indicate the storage level in the reservoir at the beginning of the n-th period.

(3) $D = \{D_n\}_{1 \le n \le N}$ is a sequence of maps identifying the decision elements that are available at the different decision stages when the states x_n are observed. More specifically, $D_n(x_n)$ is the *set of*

[1]Research Associate, Department of Systems & Industrial Engineering, University of Arizona, Tucson, Arizona 85721

[2]Associate Professor, Department of Systems & Industrial Engineering, University of Arizona, Tucson, Arizona 85721

feasible decisions d_n available at the n-th decision stage when the state x_n is observed. For example, in the reservoir control problem $D_n(x_n)$ includes all the feasible target releases associated with the n-th decision stage when the storage level x_n is observed.

(4) p_0 is a counting density on Ω called the *initial distribution*, i.e., $0 \leq p_0(x) \leq 1$, $\forall x \in \Omega$ and $\sum_{x \in \Omega} p_0(x) = 1$. $p_0(x)$ indicates the probability of starting the process at n=1 with the state x. In the context of the reservoir control problem p_0 is the probability mass function of the initial storage level. Notice that for processes for which the first state of the process is given, by x° for example, the function p_0 has the following form: $p_0(x°) = 1$, $p_0(x) = 0$, $\forall x \neq x°$.

(5) $P = \{p_n\}_{1 \leq n < N}$ is a sequence of *transition laws*. More specifically, $p_n(\cdot|x_n, d_n)$ is a counting density on Ω, i.e., $0 \leq p_n(x_{n+1}|x_n, d_n) \leq 1$ and $\sum_{x_{n+1} \in \Omega} p_n(x_{n+1}|x_n, d_n) = 1$. In other words, $p_n(x_{n+1}|x_n, d_n)$ may be viewed as the conditional probability of the state x_{n+1} at the (n+1)st decision stage given that the state x_n is observed at the n-th decision stage and the decision d_n is made.

(6) r is a real-valued function defined on the sequences $(x_1, d_1, x_2, d_2, \ldots, x_{N-1}, d_{N-1}, x_N, d_N)$, called the *reward function*.

The decision-making procedure is determined by a *strategy* $\delta = (\delta_1, \delta_2, \ldots, \delta_N)$ where δ_n is a function which associates decisions with states at the n-th decision stage. More specifically, $\delta_n(x_n) = d_n$ is the decision made at the n-th decision stage when the state x_n is observed.

The strategy $\delta = (\delta_1, \delta_2, \ldots \delta_N)$ is said to be *feasible* if $\delta_n(x_n) \in D_n(x_n)$, $\forall n = 1, 2, \ldots, N$, $x_n \in \Omega$. The set of all such strategies is denoted by Δ and is assumed to be non-empty.

The application of the strategy $\delta \in \Delta$ generates a process which schematically may be described as follows: the process starts at the first decision stage n = 1 where the state x_1 is selected from Ω according to the initial distribution p_0; then the decision $d_1 = \delta_1(x_1)$ is made whereupon the process moves to the second state, n = 2 where the state x_2 is selected from Ω according to the transition law $p_1(\cdot|x_1, d_1)$; then the decision $d_2 = \delta_2(x_2)$ is made and the process moves to the third decision stage, n = 3 where the state x_3 is selected from Ω according to the transition law $p_2(\cdot|x_2, d_2)$ etc. Finally the process reaches the last decision state, n = N where the state x_N is observed and the decision $d_N = \delta_N(x_N)$ is made. As a result of observing the sequence $(x_1, d_1, \ldots, x_N, d_N)$ the reward $r(x_1, d_1, \ldots, x_N, d_N)$ is realized and the process terminates.

The preference order used for the evaluation of the elements δ of Δ is specified by $E_\delta[r]$ where

$$E_\delta[r] = \sum_{(x_1, x_2, \ldots, x_N) \in \Omega^N} r(h) \cdot p_0(x_1) \cdot p_1(x_2|x_1, d_1) \ldots p_{N-1}(x_N|x_{N-1}, d_{N-1}) \tag{1}$$

and

$$d_n = \delta_n(x_n) \ .$$

$E_\delta[r]$ is called the *expected value of the reward* associated with the strategy δ. The strategy δ^* is said to be *optimal feasible* if $\delta^* \in \Delta$ and

$$E_{\delta^*}[r] = \max_{\delta \in \Delta} E_\delta[r]. \tag{2}$$

The set of all the optimal feasible strategies is denoted by Δ^* and its elements by δ^*. It is assumed that Δ^* is not-empty, i.e., that there exists at least one optimal feasible strategy. For example, if $D_n(x_n)$ consists of finitely many elements for all $1 \leq n \leq N$ and $x_n \in \Omega$ then it is guaranteed that Δ^* is non-empty.

TYPE-M PROCESSES

In many situations the process under consideration is such that there exist sequences $\{r_n\}_{1 \leq n \leq N}$, and $\{\rho_n\}_{1 \leq n < N}$ of real valued functions such that

$$r_n = r_n(x_n, d_n, \ldots, x_N, d_N) \tag{3}$$

$$r_1(x_1,d_1,\ldots,x_N,d_N) = r(x_1,d_1,\ldots,x_N,d_N) \tag{4}$$

and

$$E_\delta[r_n|x_n,x_{n+1}] = \rho_n(x_n,d_n,E_\delta[r_{n+1}|x_{n+1}]),\ \forall\delta\varepsilon\Delta, n=1,2,\ldots N-1 \tag{5}$$

where

$$E_\delta[r_n|x_n] = \sum_{(x_{n+1},\ldots,x_N)\varepsilon\Omega^{N-n}} r_n(x_n,d_n,\ldots,x_N,d_N)\cdot p_n(x_{n+1}|x_n,d_n)\ldots\cdot p_{n-1}(x_N|x_{N-1},d_{N-1}) \tag{6}$$

and $d_n = \delta_n(x_n)$.

Moreover, in many processes ρ_n is such that $\rho_n(x_n,d_n,\cdot)$ is monotonically increasing. This type of processes will be referred to as *type-M processes*.

Examples:

(1) Consider the reward function r for which

$$r(x_1,d_1,x_2,\ldots,x_N,d_N) = \sum_{n=1}^{N} \ell_n(x_n,d_n)$$

where ℓ_n is a real valued function.

Define:

$$r_n(x_n,d_n,\ldots,x_N,d_N) := \sum_{m=n}^{N} \ell_n(x_n,d_n)\ ,\quad 1\leq n\leq N$$

and

$$\rho_n(x_n,d_n,a) = \ell_n(x_n,d_n) + a\ ,\quad 1\leq n<N$$

for any real number a.

Thus,

$$r_1(x_1,d_1,\ldots,x_N,d_N) = r(x_1,d_1,\ldots,x_N,d_N)$$

and

$$E_\delta[r_n|x_n,x_{n+1}] = \rho_n(x_n,d_n,E_\delta[r_{n+1}|x_{n+1}])\ ,\quad 1\leq n<N$$

and hence any process having the reward function r is a type-M process.

(2) Consider the reward function r for which

$$r(x_1,d_1,\ldots,x_N,d_N) = \exp\{\sum_{n=1}^{N} \ell_n(x_n,d_n)\}$$

where ℓ_n is a real valued function.

Define:

$$r_n(x_n,d_n,\ldots,x_N,d_N) := \exp\{\sum_{m=n}^{N} \ell_n(x_n,d_n)\}\quad 1\leq n\leq N$$

and

$$\rho_n(x_n,d_n,a) := a\cdot\exp\{\ell_n(x_n,d_n)\}\quad ,\ 1\leq n<N$$

for any positive real number a.

Thus,

$$r_1(x_1,d_1,\ldots,x_N,d_N) = r(x_1,d_1,\ldots,x_N,d_N)$$

and

$$E_\delta[r_n|x_n,x_{n+1}] = \rho_n(x_n,d_n,E_\delta[r_{n+1}|x_{n+1}]),\ \forall\delta\varepsilon\Delta$$

and hence any process having the reward function r is a type-M process.

An algorithm for the construction of optimal-feasible strategies for type-M processes is introduced in the next section.

THE DYNAMIC PROGRAMMING ALGORITHM FOR TYPE-M PROCESSES

Let Z be any type-M process, $\{r_n\}_{1\leq n\leq N}$ and $\{\rho_n\}_{1\leq n\leq N}$ any sequences associated with Z which satisfies equation 3 and 5 respectively. Consider the following procedure for the construction of the subset Δ° of Δ:

Step 1 For n = N and any arbitrary $x_N \varepsilon \Omega$ let $\Gamma_N(x_N)$ be the set of all the elements $d_N^\circ \varepsilon D_N(x_N)$ satisfying the equation

$$r_N(x_N,d_N^\circ) = \max_{d\varepsilon D_N(x_N)} \{r_N(x_N,d)\} \tag{7}$$

Set: $$f_N^*(x_N) = r_N(x_N,d_N^\circ),\ d_N^\circ \varepsilon D_N(x_N).$$

Step 2 For $1\leq n<N$ and any arbitrary $x_n \varepsilon \Omega$ let $\Gamma_n(x_n)$ be the set of all the elements $d_n^\circ\ \varepsilon D_n(x_n)$ satisfying the equation:

$$f_n(x_n,d_n^\circ) = \max_{d\varepsilon D_n(x_n)} f_n\ (x_n,d)\ ,\quad d_n^\circ \varepsilon \Gamma_n(x_n) \tag{8}$$

where

$$f_n(x_n,d) = \sum_{x_{n+1}\varepsilon\Omega} \rho_n(x_n,d,f_{n+1}^*\ (x_{n+1}))\ p_n(x_{n+1}|x_n,d) \tag{9}$$

Set: $f_n^*(x_n) = f_n(x_n,d_n^\circ),\ d_n^\circ\ \varepsilon\Gamma_n(x_n)$

Step 3 Set:

$$\Delta^\circ = \{\delta^\circ:\ \ \delta_n^\circ(x_n)\varepsilon\Gamma_n(x_n),\ \forall n=1,2,\dots,N,\ x_n\varepsilon\Omega\} \tag{10}$$

The recursive procedure defined above is called the *dynamic programming (DP) algorithm* and the strategies $\delta^\circ\varepsilon\Delta^\circ$ the *dynamic programming solutions*.

It was shown (Sniedovich 1976) that the dynamic programming solutions are optimal-feasible for all type-M processes.

Consider the type-M process $Z = (N,\Omega,D,p_0,P,r)$ where

$$\Omega = \{x:\ \ x = 0,1,2,3,\dots,M\},\ D = \{D_n\}_{1\leq n\leq N} \text{ with } D_n(x_n) = \{d_n:\ \ d_n = 0,1,\dots x_n\}.$$

The total number of equations T to be solved using the DP algorithm is given by

$$T = N\cdot(M + 1) \tag{11}$$

If each of these equations is solved by searching over the entire set $D_n(x_n)$, the total number of iterations required is given by

$$TT = \frac{N\cdot(M+1)\cdot(M+2)}{2} \tag{12}$$

which for N = 100, M = 20 yields TT = 23,100.

The question is whether certain properties of the process allow a reduction in the number of iterations needed for the implementation of the DP algorithm.

ANALYTICAL CONSIDERATIONS IN THE IMPLEMENTATION OF THE DP ALGORITHM

Suppose that while implementing the DP algorithm in the context of a certain type-M process it is found that $\Gamma_n(x_n) = \{d^\circ\}$ for some $1\leq n<N$ and $x_n\varepsilon\Omega$. The question is whether while solving the equation for some other element x_n' of Ω it is possible to make use of d° by restricting the search only to a

certain subset of $D_n(x_n')$ determined by d°.

It will be demonstrated that in certain processes there exists a very close relationship between the sets $\Gamma_n(x_n)$ and $\Gamma_n(x_n')$, $x_n, x_n' \varepsilon \Omega$.

Examples

(1) Consider the process $Z = (N,\Omega,D,p_o,P,r)$ where:

(i) $\Omega = 0,1,2,\ldots M$, $D = \{D_n\}_{1\leq n\leq N}$, $D_n(x_n) = \{0,1,2,\ldots,x_n\}$,

(ii) $P = \{p_n\}_{1\leq n<N}$, where

$$p_n(x_{n+1}|x_n,d_n) = \begin{cases} \text{Prob } (Q = x_{n+1} - x_n + d_n) \text{ , } 0\leq x_{n+1}<M \\ \\ \text{Prob } (Q\geq M - x_n + d_n) \text{ , } \quad x_{n+1} = M \end{cases}$$

where Q is a discrete random variable for which Prob (Q = negative integer) = 0.

(iii) $r(x_1,d_1,\ldots,x_N,d_N) = \sum_{n=1}^{N} \ell_n(d_n)$

where ℓ_n is a monotone increasing concave function.

For the above process Sniedovich (1976) had shown that there exists an optimal feasible strategy δ^* with the property that

$$\delta_n(x_{n+1})\varepsilon\{\delta_n(x_n),\ \delta_n(x_n) + 1\} \text{ , } \forall n = 1,2,\ldots N$$

Thus, while implementing the dynamic programming algorithm only two decisions are to be considered while solving the equation for $x_n\neq 0$. More specifically, for all $1\leq n\leq N$ the first step of the algorithm implies $\Gamma_n(0) = \{0\}$ and the set $\Gamma_n(x_n)$ is constructed from the set $\Gamma_n(x_n-1)$ by checking only two decisions at a time, those included in the set $\Gamma_n(x_n)$ and those which are greater from the elements of $\Gamma_n(x_n)$ by one.

For example, if for n = 21 and $x_n = 6$ it was found that $\Gamma_{21}(6) = \{4\}$ then while constructing the set $\Gamma_{21}(7)$ only the decisions $d_{21} = 4$ and $d_{21} = 5$ are to be considered.

(2) The same as the previous example with ℓ_n being a monotone increasing convex function. For this case Sniedovich (1976) has shown that there exists an optimal feasible strategy with the property that $\delta_n(x_n)\varepsilon\{0,x_n\}$ so that also in this case only two decisions are checked while solving the equations associated with the DP algorithm.

In order to compare the relative efficiency of the results mentioned above notice that for the two examples the total number of iterations needed for the implementation of the dynamic programming algorithm is given by

$$TT' = N\cdot(1+2M) \tag{13}$$

Let e be the ratio (in percent) of TT' and TT, i.e.,

$$e = \frac{TT'}{TT} \cdot 100 \tag{14}$$

Thus,

$$e = \frac{(1+2M) \cdot 2}{(M+1) \quad (M+2)} \cdot 100 \tag{15}$$

which for M = 20 yields e≈18.

In other words, the properties of the reward function and the other elements of the process enable one to reduce the number of iterations involved in the DP algorithm by 82%. Notice that for large M e can be approximated by

$$e \simeq \frac{4}{M} \tag{16}$$

so that the reduction in the number of iterations is proportional to M.

APPLICATION TO RESERVOIR CONTROL PROBLEMS

The two examples introduced in the previous section can be used for the modeling of the reservoir control problem for which

- N = number of periods (years, months, seasons) under consideration.
- x_n = the storage level at the beginning of the n-th period with M being the maximum capacity of the reservoir.
- d_n = the release from the reservoir during the n-th period.
- Q = the inflow to the reservoir (a random variable)
- ℓ_n = the reward associated with the release during the n-th period, and the objective in the maximization of the expected value of the reward.

It should be noted that concave reward functions are often used in the context of reservoir control problems (Dorfman, 1962; Arunkumar, 1975; Askew, 1974b).

THE RELIABILITY PROBLEM

In the previous examples it was assumed that the release is not allowed to exceed the storage level observed at the beginning of the period under consideration. If this constraint is relaxed then the set Δ of feasible strategies contains elements which generate shortages with positive probabilities.

There are indications (Askew, 1974a, 1974b, 1975) that even if a penalty is imposed whenever a shortage is realized the optimal feasible release strategy may still generate shortages with positive probability.

While the penalty imposed on the process whenever a shortage is realized reflects the direct consequences associated with the shortage (including its magnitude) it does not reflect the non-economic aversion to failure (shortage) based on practical and socio-political considerations.

In order to guarantee a certain level of reliability as far as shortages are considered one can impose reliability constraints on the release strategies. For example, Askew (1974a) introduces the following three reliability constraints:

(1) The probability that the system will fail to supply its target release in a given year must not exceed a fixed maximum.

(2) The probability that the target release will not be supplied in one or more years in the life of the project must not exceed a given maximum.

(3) The average number of occasions during the life of the project on which it will be unable to supply its forecast release must not exceed a given maximum.

Askew (1974b) also indicates that in many cases optimal release strategies designed to maximize the expected net benefit allow the system to fail on an appreciable number of occasions. Thus, if the probability of such failures is greater than the maximum allowed, a procedure is needed for deriving release strategies that are optimal under the reliability constraint under consideration.

While the incorporation of the first constraint indicated above in the dynamic programming algorithm seemed to be a straightforward procedure (Askew, 1974b, p. 1100) the other two constraints seem to cause a modeling problem. More specifically, when discussing the potential use of the dynamic programming algorithm when one of these constraints is imposed on the process Askew (1974b, p. 1100) argues the following:

"These constraints limit the magnitude of parameters that are function of system variables computed over the entire life of the system; therefore they cannot be introduced as normal constraints.... Our inability to control directly the probability of system failures may therefore be compensated for partly by our ability to compute its magnitude...."

Askew's statement is strange in two ways. First, since the system variables (state elements) are determined subjectively (to a certain degree), their "normality" depends on the definition of the state elements and, as will be demonstrated below, by modifying the definition of the state elements, the constraints become normal. Secondly, once the state elements are appropriately redefined it is possible to control the probability of system failure, as will be demonstrated below.

It seems as if the difficulty in incorporating the above constraints in the dynamic programming algorithm is due to the defects in the modeling stage of the problem. The discrete stochastic multistage decision model introduced above will be shown to be a convenient framework for solving the reliability problem.

Let C2 and C3 denote the second and third constraint introduced above, respectively. It will be shown now that the DP algorithm may be used to solve the reservoir control problems associated with C1 and C2.

THE RESERVOIR CONTROL PROBLEM UNDER C2

Let β be maximum allowed probability of at least one failure (shortage) during the life of the system. The constraint C2 can be written then as follows:

$$\text{Prob [at least one shortage during the N years]} \leq \beta \tag{17}$$

or alternatively,

$$\text{Prob [no shortage during the N years]} > 1-\beta = \alpha \tag{18}$$

Also let q denote the values the random inflow Q takes with positive probability and assume that $q \varepsilon \{0,1,2,...,MQ\}$.

The objective is then to formulate the reservoir control process as a stochastic multistage decision process using the model introduced above and to show that the DP algorithm can be used as a solution procedure. In other words, the elements N,Ω,D,p_o,P,r associated with the above problem are to be specified.

The positive integer N obviously indicates the number of years (periods) the reservoir is to be operated. As far as the state space Ω is considered, its elements should specify the following: the storage level in the reservoir at the beginning of any of the periods under consideration, the probability of having no shortage during the first (n-1)st periods, n-1,2,...N and the degree of shortage (if any) during any of the periods under consideration.

It seems then as if the elements x of Ω should be three dimensional vectors. However, it is possible to describe the storage levels at the beginning of the n-th period and the degree of shortage during the (n-1)st period by the same variable so that only two coordinates are needed for the construction of the state elements. More specifically, since the shortage cannot exceed the quantity M+MQ, Ω can be formally defined as follows:

$$\Omega = \{x: \ x = (x(1),\ x(2))\} \text{ , where}$$

(i) $x(1)\varepsilon\{-(M+MQ),\ -(M+MQ-1),...,0,1,2,...,M+MQ\}$

indicates the storage level in the reservoir with the convention that $x(1)\leq 0$ indicates that the reservoir is empty.

(ii) $x(2)\varepsilon[\alpha,1]$ indicates the probability of no shortage during the (n-1)st periods, where n is the period under consideration.

Notice that strictly speaking the state space Ω does not satisfy the requirement of the model since Ω (through x(2)) contains uncountably infinitely many elements. In other words, the interval $(\alpha,1]$ should be discretized in order for Ω to be properly defined. This subject will be discussed later.

In order to guarantee that the constraint C2 is satisfied by all the feasible strategies, the set $D_n(x_n)$ is defined as follows:

$$D_n(x_n) : = \{d_n: \ d_n\varepsilon\{0,1,2,...x_n+MQ\}, \ \ x_n(2) \cdot \text{Prob } (Q>d_n-x_n(1)) > \alpha \}$$

Notice that the relation between x_n,d_n and $x_{n+1}(2)$ is specified (deterministically!) by:

$$x_{n+1}(2) = x_n(2) \cdot \text{Prob } (Q>d_n - x_n(1))$$

and thus the above structure of $D_n(x_n)$ guarantees that all the feasible strategies indeed satisfy C2.

The initial condition of the process, specified by p_o can be expressed as follows:

$$p_o(x_i) = 1,\ x_1(1) = \text{initial storage level},\ x_2(2) = 1.$$

Often it is assumed that $x_1(1) = M$, i.e., it is assumed that the reservoir is full at the beginning of the first period.

The transition law P can be written then as follows:

$$p_n(x_{n=1}|x_n,d_n) = \begin{cases} \text{Prob } (Q=x_{n+1}(1) = d_n - \Psi\cdot(x_n(1))),\ x_{n+1}<M,\ x_{n+1}(2) = x_n(2) \cdot \text{Prob } (Q>d_n-\Psi(x_n(1))) \\ \\ \text{Prob } (Q\geq M + d_n - \Psi\cdot(x_n(1))),\ x_{n+1}(1) = M,\ x_{n+1}(2) = x_n(2) \cdot \text{Prob } (Q>d_n-\Psi(x_n(1))) \\ \\ 0 \qquad \text{, otherwise} \end{cases}$$

where $\Psi(x_n) := \max\{0, x_n\}$.

The reward function r is assumed to have the following structure:

$$r(x_1,d_1,x_2,\ldots,x_N,d_N) = \sum_{n=1}^{N-1} \ell_n(x_n,d_n,x_{n+1}) + \ell_N(x_N,d_N)$$

where $\ell_n(x_n,d_n,x_{n+1})$ depends on $x_n(1)$, d_n, and $x_{n+1}(1)$ and ℓ_N depends on $x_N(1)$ and d_N. Notice that $x_{n+1}(1)$ indicates the shortage during the n-th period, and if the reward associated with the last year is affected by the shortage during the last year $\ell_N(x_N,d_N)$ may be viewed as the expected value of the last reward given the storage level $\Psi(x_N(1))$ and the target release d_N.

It is obvious that the process $Z = (N,\Omega,D,p_o,P,r)$ is a type-M process and thus the dynamic programming algorithm can be used as a solution procedure.

It is still left to be determined how the range $(\alpha,1]$ is to be discretized. It should be noted that in principle the range $(\alpha,1]$ can be discretized by choosing some positive integer, say I and constructing the set $A = \{\alpha+\varepsilon, \alpha+2\varepsilon,\ldots,1\}$, $\varepsilon=\frac{1-\alpha}{I}$. Then the dynamic programming algorithm can be applied using one of the standard interpolation techniques (Casti, 1975). Although the real optimal solution may not be found, the integer I may be increased so as to guarantee that the solution provided by the DP algorithm is in a pre-specified range from the real optimum.

THE RESERVOIR CONTROL PROBLEM UNDER C3

The constraint C3 can be formally expressed as follows:

$$E\,[\text{number of shortages during the N periods}] \le \hat{N}$$

where $\hat{N}$ is the maximum expected number of shortages allowed during the life of the system.

As far as modeling is concerned, this problem is similar to the previous one with the following minor modifications:

(1) $x_n(2)\varepsilon[0,\hat{N}]$ is defined as the expected value of the number of shortages during the first (n-1)st periods.

(2) $x_{n+1}(2) = x_n(2) + \text{Prob}\,(Q<d_n - \Psi(x_n(1)))$

(3) $D_n(x_n) = \{d_n : d_n\varepsilon\{0,1,2,\ldots,x_n+MQ\},\ x_n(2) + \text{Prob}\,(Q<d_n-\Psi(x_n(1))<\hat{N}\}$

(4) $x_1^o(2) = 0$

(5)
$$p_n(x_{n+1}|x_n,d_n) = \begin{cases} \text{Prob}\,(Q=x_{n+1}(1) + d_n - \Psi(x_n(1))),\ x_{n+1}(1)<M,\ x_{n+2}(2) = x_n(2) \\ \qquad + \text{Prob}\,(Q<d_n-\Psi(x_n(1))) \\ \text{Prob}\,(Q\ge M+d_n-\Psi(x_n(1))),\ x_{n+1}(1) = M,\ x_{n+2}(2) = x_n(2) \\ \qquad + \text{Prob}\,(Q<d_n-\Psi(x_n(1))) \\ 0 \quad \text{, otherwise} \end{cases}$$

The same methods used for the discretization of the range $[\alpha,1]$ is the previous problem can be used in this case for discretizing the range $[0,\hat{N}]$.

THE RANGE PROBLEM

Consider the reservoir control problem where the objective is to minimize the expected value of the range of fluctuation of the storage levels around some critical value y°. This problem may represent situations where the reservoir is operated for recreational purposes.

The process $Z = (N,\Omega,D,p_o,P,r)$ may be used as a framework for solving the above problem where:

(1) N is the number of periods under consideration

(2) $\Omega = \{X:\ X = (X(1), X(2), X(3)),\ X(i)\varepsilon\{0,1,2,\ldots M\},\ i = 1,2,3\}$ with:

$x_n(1)$ = the minimal storage level observed during the first (n-1)st periods

$x_n(2)$ = the maximal storage level observed during the first (n-1)st periods

$x_n(3)$ = the storage level observed at the beginning of the n-th period.

(3) $D = \{D_n\}_{n \le N}$, $D_n(x_n) = \{d: d \varepsilon \{0,1,2,\ldots x_n\}\}$.

It is assumed then (for simplicity) the release is not greater than the storage level.

(4) $p_o(X_1^o) = 1, X_1^o(1) = X_1^o(2) = X_1^o(3) = y^o$.

It is assumed that the initial storage level is equal to the critical storage level y^o.

(5) $P = \{p_n\}_{1 \le n < N}$, where

$$p_n(x_{n+1}|x_n,d_n) = \begin{cases} \text{Prob }(Q = x_{n+1}(1) + d_n - x_n(3)),\ 0 \le x_{n+1}(3) < M,\ x_{n+1}(1) = \max\{x_n(1),\ x_{n+1}(3)\} \\ \qquad\qquad x_{n+1}(2) = \min\{x_n(2),\ x_{n+1}(3)\} \\ \text{Prob }(Q > M + d_n - x_n(3)),\ x_{n+1}(3) = M,\ x_{n+1}(1) = \max\{x_n(1),\ x_{n+1}(3)\} \\ \qquad\qquad x_{n+1}(2) = \min\{x_n(2),\ x_{n+1}(3)\} \\ 0 \quad , \text{otherwise.} \end{cases}$$

(6) $$r(x_1,x_2,\ldots,x_N) = \max\{x_N(3), \max_{1 \le n \le N}\{x_n(3)\}\} - \min\{x_N(3), \min_{1 \le n \le N}\{x_n(3)\}\}$$

In order to use the dynamic programming algorithm there is a need to find the sequences $\{r_n\}_{1 \le n \le N}$ and $\{p_n\}_{1 \le n \le N}$ that will preserve the type-M properties of the process.

Let $r_N(x_N,d_N) = \max\{x_N(1),\ x_N(3)\} - \min\{x_N(2),\ x_N(3)\}$

$$r_n(x_N,d_n,\ x_{n+1},\ldots,x_N,d_N) = r_N(x_N,d_N)\ ,\ 1 \le n < N$$

thus,

$$E_\delta[r_n|x_N,x_{n+1}] = E_\delta[r_{n+1}|x_{n+1}],\ \forall \delta \varepsilon \Delta$$

so that $\rho_n(x_n,d_n,a) = a$, $a = E_\delta[r_{n+1}|x_{n+1}]$ and r_n preserve the monotonicity property of the process, and hence the reservoir control problem defined above is a type-M process for which the DP algorithm provides optimal feasible solution.

The recursive equation associated with the DP algorithm are as follows:

Step 1

$$f_N^*(x_N) = r_N(x_N,d_N^o) = \max\{X_N(1),\ X_N(3)\} - \min\{X_N(2),\ x_N(3)\}$$

Step 2

$$f_n^*(x_n) = \min_{d_n \varepsilon D_n(x_n)} \{\sum_{x_{n+1} \varepsilon \Omega} f_{n+1}^*(x_{n+1}) \cdot p_n(x_{n+1}|x_n,d_n)\},\ 1 \le n < N$$

which in terms of the density $P_Q(q)$ can be written as follows:

$$f_n^*(x_n) = \min_{d_n \varepsilon D_n(x_n)} \{\sum_{q=0}^{MQ} f_{n+1}^*(x_{n+1}) \cdot P_Q(q)\} \quad 1 \le n < N$$

where

$$x_{n+1}(1)|_q = \begin{cases} \max\{x_n(1),\ x_n(1) - d_n + q\},\ x_n(1) - d_n + q < M \\ M, \qquad x_n(1) - d_n + q \ge M \end{cases}$$

$$x_{n+1}(2)|_q = \min\{x_n(2),\ x_n(1) - d_n + q\}$$

$$x_{n+1}(3)|q = \begin{cases} x_n(1) - d_n + q, & x_n(1) - d_n + q<M \\ M & , x_n(1) - d_n + q\geq M \end{cases}$$

with $x_1^\circ(1) = x_1^\circ(2) = x_1^\circ(3) = y^\circ$.

COMPUTATION EXAMPLE

Consider the following values for the elements introduced in the above problem:

M = maximum storage capacity at the reservoir = 10 units

MD = maximum release capacity = 3 units per period

N = number of years of operation = 15

X° = critical storage level = 7 units

P = the probability mass function of the inflow Q

$P_Q(0) = 0.20$; $P_Q(1) = 0.30$; $P_Q(2) = 0.30$; $P_Q(3) = 0.20$

The multistage decision problem associated with the above values was solved by the DP algorithm. The optimal value of the reward function was found to be 2.92. Portion of the optimal feasible strategy is presented in Table 1.

SUMMARY

A discrete stochastic multistage decision model is developed and demonstrated to be a suitable framework for the solution of reservoir control problems. The dynamic programming algorithm is defined and shown to be a potential solution procedure for a variety of reservoir control problems. Analytical techniques are demonstrated to be useful for the reduction of the number of iteration needed for the implementation of the algorithm. It is shown that the model and the dynamic programming algorithm presented in this paper can handle non-routine reservoir control problems.

REFERENCES CITED

Arunkumar, S., Characterization of Optimal Operating Policies for Finite Dams, Journal of Mathematical Analysis and Applications, Vol. 49, No. 2, February, 1975, pp. 267-274.

Askew, A. J., 1974a, Optimum Reservoir Operating Policies and the Imposition of a Reliability Constraint, Water Resources Research, Vol. 10, No. 1, 1974, pp. 51-56.

Askew, A. J., 1974b, Chance-Constrained Dynamic Programming and the Optimization of Water Resources Systems, Water Resources Research, Vol. 10, No. 6, 1974, pp. 1099 - 1106.

Askew, A. J., 1975, Use of Risk Premium in Chance-Constrained Dynamic Programming, Water Resources Research, Vol. 11, No. 6, December, 1975, pp. 862-866.

Casti, J., and R. Kalaba, Imbedding Methods in Applied Mathematics, Addison-Wesley Publishing Company, Inc., 1975.

Dorfman, R., Design of Water Resources Systems (Arthur Man, ed.), Harvard University Press, Cambridge, Massachusetts, 1962, pp. 88-158.

Sniedovich, M., On the Theory and Modeling of Dynamic Programming with Applications in Reservoir Control, Ph.D Dissertation, Department of Hydrology and Water Resources, University of Arizona, Tucson, AZ, 1976.

Yakowitz, S. J., Mathematics of Adaptive Control Processes, American Elsevier Publishing Co, New York, 1969.

Table 1. Portion of the Optimal Feasible Strategy $\delta_n^*(x)$ Associated with the Range Problem.

Sufficient Statistic			n														
x(1)	x(2)	x(3)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	4	5	-	-	1	1	1	1	1	1	1	1	1	1	1	1	1
7	4	6		-	2	2	2	2	2	2	2	2	2	2	2	2	2
7	4	7		-	3	3	3	3	3	3	3	3	3	3	3	3	3
7	5	5		0	0	1	0	0	1	0	1	0	0	0	0	0	0
7	5	6		-	1	2	1	1	2	1	2	1	1	1	1	1	1
7	5	7		-	2	3	2	2	3	2	3	2	2	2	2	2	2
7	6	6		1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	6	7	-	-	2	2	2	2	2	2	2	2	2	2	2	2	2
7	7	7	1	1	1	1	1	1	1	2	1	1	1	2	1	1	1
8	5	5	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0
8	5	6		-	1	1	1	1	1	1	1	1	1	1	1	1	1
8	5	7		-	2	2	2	2	2	2	2	2	2	2	2	2	2
8	5	8		-	3	3	3	3	3	3	3	3	3	3	3	3	3
8	6	6		-	0	1	0	0	1	0	1	0	0	0	0	0	0
8	6	7		-	1	2	1	1	2	1	2	1	1	1	1	1	1
8	6	8		-	2	3	2	2	3	2	3	2	2	2	2	2	2
8	7	7		-	1	1	1	1	1	1	1	1	1	1	1	1	1
8	7	8		2	2	2	2	2	2	2	2	2	2	2	2	2	2
9	6	6		-	0	0	0	0	0	0	0	0	0	0	0	0	0
9	6	7		-	1	1	1	1	1	1	1	1	1	1	1	1	1
9	6	8		-	2	2	2	2	2	2	2	2	2	2	2	2	2
9	6	9		-	3	3	3	3	3	3	3	3	3	3	3	3	3
9	7	7		-	0	1	0	0	1	0	1	0	0	0	0	0	0
9	7	8		-	1	2	1	1	2	1	2	1	1	1	1	1	1
9	7	9		2	2	3	2	2	3	2	3	2	2	2	2	2	2
10	0	0		-	0	0	0	0	0	0	0	0	0	0	0	0	0
10	5	5		-	0	0	0	0	0	0	0	0	0	0	0	0	0
10	7	7		-	0	0	0	0	0	0	0	0	0	0	0	0	0
10	7	10	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0