International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Algorithms for Stochastic Inflow Nonlinear Objective Water Reservoir Control Problem

Casti, J.L.

IIASA Working Paper

WP-74-070

1974

ALGORITHMS FOR THE STOCHASTIC INFLOW-NONLINEAR OBJECTIVE WATER RESERVOIR CONTROL PROBLEM

J. Casti

November 1974 WP-74-70

Algorithms for the Stochastic Inflow-Nonlinear Objective Water Reservoir Control Problem

J. Casti

I. Introduction

In earlier IIASA WP's [1,2], algorithms to determine the optimal control of a water reservoir network with stochastic inflows and nonlinear utilities have been proposed. Both studies [1] and [2] utilize a dynamic programming-type approach, coupled with approximations of one type or another, in order to yield a computational algorithm in which the bulk of the calculation is carried out by efficient (and rapid) network flow algorithms. The purpose of this note is to present a synthesis of the work [1,2] and to spell out the precise steps of an algorithm in sufficient detail to enable a computer program to be constructed.

2. Basic Problem

Before considering our algorithm, let us briefly review the basic water reservoir control problem. We are given m reservoirs connected in some type of network configurations by various branches (rivers, tributaries, etc.). At the beginning of each time period t, reservoir i contains an amount of water s_i, i = 1,2,...,m and various demands for water for irrigation, flood control, drinking, navigation, etc. are placed upon the system. The problem is to determine the amount of water u_i, i = 1,2,...,m, to be released from each reservoir in order that

some measure of utility of the water released is maximized, subject to various constraints. For a single stage process, this problem is not too difficult; however, the real problem is complicated by being a multistage process with the added feature of having stochastic inflow at each reservoir at each time due to rainfall and underground water run-off. In addition, the various utility functions for each reservoir are often nonlinear, thereby precluding any direct application of linear programming procedures. Consequently, other approaches are required.

In order to formulate our problem in mathematical terms, let

$s_i(t)$ = amount of water available in reservoir i at time t,

$u_i(t)$ = amount of water released from reservoir i at time t,

$r_i(t)$ = external inflow to reservoir i at time t (stochastic quantity), $i = 1,2,\ldots,m$, $t = 0,1,\ldots,T$.

Clearly, the dynamics of each reservoir are described by the equation

$$s_i(t+1) = s_i(t) - u_i(t) + r_i(t) + \sum_{j\in I_i} \beta_j u_j(t) \quad , \quad (1)$$

where I_i is the subset of $\{1,2,\ldots,m\}$ consisting of those reservoirs which input water to reservoir i, and β_j represents the fraction of water released from reservoir j which is absorbed by the network before it reaches reservoir i, $0 \leq \beta_j < 1$, $i = 1,2,\ldots,m$.

Let us assume that there is a certain cost associated with having an amount of water s_i available at reservoir i. Following [1], we assume this cost is expressible by the convex function $\phi_i(s_i)$, i = 1,2,...,m, i.e. The total objective function is

$$J(s_1,s_2,\ldots,s_m) = \sum_{k=1}^{m} \phi_k(s_k) \quad . \tag{2}$$

Since the quantities $r_i(t)$ in (1) are random variables with distribution functions $dG_i(r)$, our optimization problem may be formulated as

$$\min \mathcal{E}[J]$$

over all control sequences $\{u_1(t),\ldots,u_m(t),\ t = 0,1,\ldots,T-1\}$, where s(t) and u(t) are related by (1) and the constraints

$$\mu_i(t) \leq u_i(t) \leq s_i(t) \quad , \tag{3}$$

are satisfied. Here $\mathcal{E}$ denotes the expected value relative to the distribution function $dG_i(r)$, while the quantities $\mu_i(t)$ represent certain minimal demands for water which must be met by release from reservoir i, i = 1,2,...,m.

We tackle this problem by dynamic programming. Let

$f_t(s_1,\ldots,s_m)$ = expected value of J when the process has T-t time periods remaining, is in state $(s_1,\ldots,s_m)$ and an optimal policy is pursued, t = 0,1,...,T.

Then it is an easy application of the principle of optimality to see that f_t satisfies the functional equation

$$f_t(s_1,s_2,\ldots,s_m) =$$

$$\min_{\substack{\mu_i(t) \le u_i(t) \le s_i(t) \\ i = 1,2,\ldots,m}} \left\{ \int \left[\sum_{k=1}^{m} \phi_k(s_k) + f_{t+1}\left(s_1 - u_1 + r_1 + \sum_{j\in I} \beta_j u_j \ , \right.\right.\right.$$

$$\left.\left.\left. s_2 - u_2 + r_2 + \sum_{j\in I_<} \beta_j u_j, \ldots, \right)\right] dG(r) \right\} \ ,$$

$$t = 0,1,\ldots,T-1 \ , \quad (4)$$

$$f_T(s_1,s_2,\ldots,s_m) = \sum_{k=1}^{m} \phi_k\left(s_k(T)\right) \quad (5)$$

3. Approximations

Our next objective is to make approximations in Eq. (4) so that it will be possible to utilize network flow algorithms to effect the minimization over the u's for fixed values of $s_1,\ldots,s_m$, $r_1,\ldots,r_m$. This means that both the individual reservoir costs $\phi_i(s_i)$ and the "next stage" return $f_{t+1}(\alpha_1,\alpha_2,\ldots,\alpha_m)$ must be judiciously approximated. The heart of our methods is in the selection of approximations for these quantities that not only preserve accuracy, but also enable us to apply network flow techniques for solution of the minimization over the u's.

The first approximation is to replace the individual reservoir costs by piecewise linear functions. Since we have

assumed each ϕ_i is a convex function of its argument with $\phi_i(0) = 0$, we have

$$\phi_i(s_i) = \begin{cases} m_{i1}\, s_i \quad , & 0 \le s_i \le s_i^{(1)} \quad , \\ m_{i2}\, s_i + s_i^{(1)}(m_{i1} - m_{i2}) \quad , & \\ \qquad \vdots & \\ & s_i^{(1)} < s_i \le s_i^{(2)} \quad . \end{cases} \tag{6}$$

Hence, in each segment of the form $s_i^{(j)} \le s_i \le s_i^{(j+1)}$, the function ϕ_i is linear.

Our second approximation is in "policy space", i.e. we guess an operating policy $u^o(s_1,\ldots,s_m;\, t)$ and use this policy to determine a return function from the relations (4) and (5). This is a type of approximation well-suited to taking advantage of experience and "seat-of-the-pants" operating rules for reservoir systems. In addition, it can be shown that the algorithm described below will monotonically improve (in the sense of the criterion (2)) the current policy as the iteration procedure progresses. Thus, we have a systematic method for improving any existing policy.

Having fixed an approximation to the policy u, our last approximation is to the optimal value function $f_{t+1}\Big(s_1 - u_1(s_1,..,s_m;t) + r_1 + \sum_{j\in I_1} \beta_j u_j(s_1,\ldots,s_m;t),$ $s_m - u_m(s_1,\ldots,s_m;t) + r_m + \sum_{j\in I_m} \beta_j u_j(s_1,\ldots,s_m;t)\Big)$. By virtue of the criterion (2) and the structure of the ϕ_i, it is not difficult to see that the function $f_{t+1}(\cdot,\ldots,\cdot)$ should be

separable in its arguments, i.e.

$$f_{t+1}(\alpha_1,\ldots,\alpha_m) = \gamma_1(\alpha_1) + \gamma_2(\alpha_2) + \cdots + \gamma_m(\alpha_m) \quad , \quad (7)$$

where the functions γ_i will be convex relative to the variable s_i (here we use $\alpha_i = s_i - u_i(s_1,\ldots,s_m;t) + r_i + {}_j\Sigma\beta_j u_j(s_1,\ldots,s_m;t)$. Again, we may approximate the γ's by piecewise linear functions, thereby giving $f_{t+1}(\cdot,\ldots,\cdot)$ the desired linear structure. Clearly, the previous approximation to the ϕ_i will be used to approximate $f_T(\alpha_1,\ldots,\alpha_m)$, while for $t < T - 1$, approximation algorithms in the DYGAM program may be employed.

4. The Algorithms

We shall present two alternative algorithms in this section. The first will be based directly upon the policy space idea presented above, while the second is based upon ideas introduced in [2]. In both cases, the primary objective is to reduce the calculation to a level at which almost all the work is done by the efficient network flow algorithms.

Alternative I (Policy Space Iteration)

The steps in this algorithm are the following:

1. Approximate the functions $\phi_i(s_i)$ by piecewise linear functions as in (6);

2. Guess an initial policy $u^o(s_1,\ldots,s_m;t)$ for all $s_1,\ldots,s_m$, $t = 0,1,\ldots,T-1$;

3. Determine the approximate optimal value functions $f_t^o(s_1,\ldots,s_m)$ by iterating the relation (4) for $t = 0,1,\ldots,T-1$, using the initial function (5);

4. Approximate each function $f_t^o(\alpha_1,\ldots,\alpha_m)$ by piecewise linear functions of $\alpha_1,\ldots,\alpha_m$ as in (7);

5. Determine the up-dated policy estimate $u^1(s_1,\ldots,s_m;t)$ as that function which minimizes

$$\int \Bigg[\sum_{k=1}^{m} \phi_k(s_k) + f_{t+1}^{(o)} \Big(s_1 - u_1 + r_1 + \sum_{j\in I_1} \beta_j u_j,\ldots,s_m - u_m + r_m + \sum_{j\in I_m} \beta_j u_j\Big)\Bigg] dG(r) \quad .$$

Notice that for each fixed set of values for $s_1,\ldots,s_m$, $r_1,\ldots,r_m$, t, this is a network flow problem. This step is carried out for all $t = 0,1,\ldots,T-1$, and all $s_1,s_2,\ldots,s_m$. (Remark: For computational purposes, it may be better to let the s_i vary only over the regions $(s_1,0,\ldots,0)$, $(0,s_2,0,\ldots,0)$, $\ldots$, $(0,0,\ldots,0,s_m)$ and then interpolate the values of $u^1(s_1,\ldots,s_m)$ for non-lattice points). Having obtained the next policy u^1, return to step 3 and continue until convergence.

Alternative II:

In this approach (which follows [2]), we note that the optimal release policy $\{u_i^*(t),\ t = 0,1,\ldots,T-1,\ T = 1,2,\ldots,m\}$ is equivalent to knowledge of the optimal levels $\{s_i^*(t)\}$. Hence, we reformulate the problem in terms of water levels only. That is, the function $f_t(s_1,\ldots,s_m)$ satisfies

$$f_t\Big(s_1(t),\ldots,s_m(t)\Big) = \int \Bigg[\sum_{k=1}^{m} \phi_k\Big(s_k(t)\Big) f_{t+1}\Big(s_1(t+1),\ldots,s_m(t+1)\Big)\Bigg] dG(r)$$

when the optimal policy is used at time t, or, equivalently, when we have optimal water levels at time t+1 (here the random quantities r_i are implicitly included in the term $s_i(t+1)$). The problem, of course, is that the optimal levels $s_i(t+1)$ (for fixed r_i) are not known and must be determined. To accomplish this task, the following algorithm is proposed:

0. Let t = T-1 and approximate $f_{t+1}\left(s_1(t+1),\ldots,s_m(t+1)\right) = \sum_{i=1}^{m} \phi_i\left(s_i(t+1)\right)$ as in (6);

1. Fix a value of the water levels, say $\bar{s}_1(t),\ldots,\bar{s}_m(t)$;

2. Fix a value of the random parameters $r_i(t)$;

3. Solve the network-flow problem of minimizing

$$\sum_{i=1}^{m} \phi_i\left(\bar{s}_i(t)\right) + f_{t+1}\, s_1(t+1),\ldots,s_m(t+1)$$

over all $\mu_i(t) \le u_i(t) \le \bar{s}_i(t)$, where $s_i(t+1)$ is given by (1);

4. Change the random variables to new levels and repeat steps 2-4, forming expected values according to the probability distribution dG(r);

5. Change to a new set of water levels $\bar{s}_i(t)$ and repeat step 3 until all levels have been considered;

6. Approximate the function $f_t(s_1,\ldots,s_m)$ by a piecewise multilinear form (using DYGAM subpackage), let $t \to t-1$, and return to step 1.

Remarks

i) At step 1, in view of the separable form of the objective function, it will again probably be best to use only water levels of the form $(s_1,0,\ldots,0)$, $(0,s_2,\ldots,0)$, etc. This will

save on computing time by cutting down the number of cases, while still yielding sufficient information to make the approximation in step 6 accurate if the s_i grid is fine enough;

ii) In the approximation of step 6, some experimentation will probably be necessary to determine how many pieces should be taken in the "piecewise" multilinear form. The usual trade-off between fewer pieces and high-order terms vs. more pieces and lower order approximations needs to be examined. Generally speaking, however, it is preferable to take several low-order pieces.

References

[1] Casti, J. "A Network Flow-Dynamic Programming Algorithm for Complex Water Reservoir Problems", IIASA WP-74-52, 1974.

[2] Dantzig, G. "Notes on Network Flow, Linear Programming, and Stochastic Programming Models for a River Basin", IIASA WP (to appear 1974(5)).