JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 24, 80–109 (1968)

Dynamic Programming Applications to Water Resource System Operation and Planning

WILLIAM G. KECKLER AND ROBERT E. LARSON

Stanford Research Institute, Menlo Park, California

Submitted by Richard Bellman

I. INTRODUCTION

Dynamic programming provides an extremely powerful approach for solving the optimization problems that occur in the operation and planning of water resource systems. However, its applicability has been somewhat limited because of the large computational requirements of the standard algorithm. This paper summarizes some of these new computational procedures and discusses some recent applications.

As water resource systems have grown larger and more complex, the importance of optimum operation and planning of these systems has increased. The investment costs and operating expenses of projects are so large that even small improvements in system utilization can involve substantial amounts of money. Also, the various control points—power generators, irrigation outlets, pumping stations, etc.—interact in such a complicated manner that it is difficult to obtain an optimum design or operating policy using an intuitive approach. Thus the potential benefits of using optimization techniques in these problems are very great indeed.

The major reason dynamic programming is so attractive for these problems is the great generality of the problem formulation to which it can be applied. Nonlinearities in the system equations and performance criterion can easily be handled. Constraints on both decision and state variables introduce no difficulties. Stochastic effects can be explicitly taken into account.

In Section II the theory of dynamic programming is briefly reviewed and a number of computational procedures are described, including the standard computational algorithm, forward dynamic programming, the standard computational algorithm for stochastic control problems, iteration in policy space, successive approximations, and state increment dynamic programming.

In Section III some specific problems to which dynamic programming has been applied are discussed. Most water problems fall into one of the following three categories:

(1) Optimum operation during a short period, such as 24 hours, when all quantities are deterministic;

(2) Monthly or yearly policy optimization when some system parameters, such as stream inflows, must be treated as stochastic variables;

(3) Long-range planning or resource allocation where demands may or may not be treated as deterministic quantities.

Four illustrative examples are discussed, including at least one from each of the above categories. The first problem is the optimum short-term operation of a combined pumped hydro and irrigation storage facility involving two reservoirs; forward dynamic programming was used for this example. The second problem is the optimum short-term operation of a multipurpose four-reservoir system, where power generation, irrigation, flood control, and recreation are all considered; the technique of successive approximations was applied in this case. The third problem is the optimum management of a single reservoir over a one-year period, where stochastic variations of input stream-flows are considered; iteration in policy space was applied here. The fourth problem is the optimum planning of additions to a system over a 30-year period; forward dynamic programming was again used for this example.

II. Dynamic Programming

A. *Problem Formulation for the Deterministic Case*

Most of the problems for which dynamic programming has been used to obtain numerical solutions can be formulated as deterministic discrete-time variational problems [1-3]. The general case of this problem is formulated as follows:

Given:

(i) A system described by the nonlinear difference equation

$$x(k+1) = \Phi[x(k), u(k), k], \tag{1}$$

where

x = state vector, n-dimensional

u = control vector, m-dimensional

k = index for stage variable

Φ = n-dimensional vector functional;

(ii) A variational performance criterion

$$J = \sum_{k=0}^{K} L[x(k), u(k), k], \tag{2}$$

where

$$J = \text{total cost}$$

$$L = \text{cost for a single stage;}$$

(iii) Constraints

$$x \in X(k) \tag{3}$$

$$u \in U(x, k) \tag{4}$$

where

$$X(k) = \text{set of admissible states at stage } k$$

$$U(x, k) = \text{set of admissible controls at state } x, \text{ stage } k;$$

(iv) An initial state

$$x(0) = c. \tag{5}$$

FIND:

The control sequence $u(0),..., u(K)$ such that J in Eq. (2) is minimized subject to the system equation (1), the constraint equations (3) and (4), and the initial condition (5).

Continuous-time variational control problems can be treated by assuming that the control is piecewise constant in time and making appropriate transformations to the discrete-time case [1, 3]. Extensions to problems involving uncertainty can be made as in Part C of this section.

The dynamic programming solution to the above problem is obtained by using an iterative functional equation that determines the optimal control for any admissible state at any stage. A minimum-cost function is defined for all $x \in X$ and all k, $k = 0, 1,..., K$, as

$$I(x, k) = \min_{j=k,k+1,...,K} \left\{ \sum_{j=k}^{K} L[x(j), u(j), j] \right\}, \tag{6}$$

where

$$x(k) = x.$$

Abbreviating $u(k)$ as u, the iterative functional equation becomes

$$I(x, k) = \min_{u}\{L(x, u, k) + I[\Phi(x, u, k), k + 1]\}. \tag{7}$$

This equation is a mathematical statement of Bellman's principle of optimality [1-3]. It states that the minimum cost for state x at stage k is found by choosing the control that minimizes the sum of the cost to be paid at the present stage and the minimum cost in going to the end from the state at stage $k + 1$ which results from applying this control. The optimal control at state x and stage k, denoted as $\hat{u}(x, k)$, is directly obtained as the value of u for which the minimum in Eq. (7) is attained.

Since Eq. (7) determines $I(x, k)$ and $\hat{u}(x, k)$ in terms of $I(x, k + 1)$, it must be solved backward in k. As a terminal boundary condition

$$I(x, K) = \min_{u}[L(x, u, K)]. \tag{8}$$

The optimization over a sequence of controls is thus reduced to a sequence of optimizations over a single control vector.

A direct-search method for solving Eq. (7), based on quantizing the admissible values of x and u to a finite number of discrete values, is described in Refs. [1-4]. This computational procedure is appealing for a number of reasons. In the first place, thorny questions of existence and uniqueness are avoided. As long as there is at least one feasible control sequence, then the direct-search procedure guarantees that the absolute minimum cost is obtained. Furthermore, extremely general types of system equations, performance criteria, and constraints can be handled; constraints actually reduce the computational burden by decreasing the admissible sets X and U. Finally, the optimal control is obtained as a true feedback solution in which the optimal control for any admissible state and stage is determined.

The major limitation of this algorithm is the computational requirements associated with it (Bellman's "curse of dimensionality" [1]). The difficulty that is generally most severe is the amount of high-speed storage required to store $I(x, k + 1)$ during the computation of $\hat{u}(x, k)$ and $I(x, k)$. The number of storage locations required is one for each quantized state, a quantity that increases exponentially with the number of state variables. Another major difficulty is the amount of computer time required to carry out the calculations. A third consideration is the amount of off-line storage required to store the results. Many of the computational procedures discussed in the remainder of this section reduce one or more of these requirements.

B. *Forward Dynamic Programming*

If the minimum cost function is redefined to be the minimum cost to reach a given state and stage from the initial state, an iterative equation analogous to Eq. (7) can be derived. In this case the calculations proceed forward in k rather than backward; hence, the term *forward dynamic programming* is used to describe the computational procedure for this case.

For many applications this form of solution is more desirable than the dynamic programming solution. This is particularly true when the initial state is fixed and the terminal state and/or stage is free; most real-time dispatching problems fall into this category. The optimum final state can be selected by searching over the minimum costs for all admissible final states and, if desired, adding a terminal cost function. The terminal cost function can be quite flexible, and the effect of using different functions can easily be seen without repeating the dynamic programming calculation. No matter which final state is finally selected, the optimum trajectory from the initial state to this final state is obtained; this property of the solution is analogous to the feedback control property of the normal dynamic programming solution. It should be noted that for a given initial state and final state the same optimum trajectory is determined by both methods.

The iterative equation is derived by defining $I'(x, k)$ as the minimum cost to reach state x at stage k from the initial state. Formally,

$$I'(x, k) = \min_{u(0),u(1)\ldots u(k-1)} \left\{ \sum_{j=0}^{k-1} L[x(j), u(j), j] \right\}, \tag{9}$$

where

$$x = \Phi[x(k-1), u(k-1), k-1].$$

If the inverse functional to Φ is defined as θ, so that

$$\Phi\{\theta[x, u(k-1), k-1], u(k-1), k-1\} = x \tag{10}$$

then the iterative equation becomes

$$I'(x, k) = \min_{u(k-1)} (L\{\theta[x, u(k-1), k-1], u(k-1), k-1\} \\ + I'\{\theta[x, u(k-1), k-1], k-1\}). \tag{11}$$

As a boundary condition for this equation, $I'(x, 0)$ is specified. If, as is often the case, the initial state is fixed at one particular value, $I'(x, 0)$ is set to zero for this state and no other initial state is considered admissible. Several computational procedures using Eq. (11) are described in Ref. [4].

C. *Stochastic Control*

An important extension of the problem formulation of Part A is the stochastic control problem as defined by Bellman [2]. In this problem the state variables are assumed to be perfectly measurable, but the system equations are affected by stochastic inputs. Many multi-reservoir control problems fit this formulation; the perfectly measurable state variables are the levels in the reservoirs, and the stochastic inputs are those streamflows that cannot be predicted exactly.

The basic formulation is modified by including $w(k)$, an s-dimensional vector of stochastic variables. It is assumed that the probability density function of $w(k)$, $p[w(k)]$, is known and that samples of $w(k)$ at different values of k are uncorrelated.[1] The relation of the stochastic variables to the other system variables is made explicit by including them in the system equation

$$x(k+1) = \Phi[x(k), u(k), w(k), k]. \tag{12}$$

This equation shows that the state variables at time $k+1$ depend not only on $x(k)$, the set of state variables at time k, and $u(k)$, the control decision made at time k, but also on the stochastic variables, $w(k)$. The next state thus is not determined exactly, but has a probability density function resulting from that of $w(k)$.

Because of this uncertainty in the evolution of the system state variables, it is not possible to optimize a deterministic performance criterion. Instead, a probabilistic performance measure must be considered. Customarily, the expected value of a cost function is minimized, where the expectation is taken over the sequence of stochastic inputs. Formally,

$$J = \underset{w(0),w(1),\ldots w(K)}{\mathrm{E}} \left\{ \sum_{k=0}^{K} L[x(k), u(k), w(k), k] \right\}, \tag{13}$$

where E denotes the expectation operator. By modifying the form of the functional L, it is possible to consider other types of performance measures, such as the probability that performance is better than a certain specified value.

Constraints on state and control variables are imposed exactly as in Part A of this section:

$$x(k) \in X(k) \tag{14}$$

$$u(k) \in U[x(k), k]. \tag{15}$$

A recursive relation analogous to Eq. (7) can now be derived. The minimum expected cost function, $I''(x, k)$, is defined as

$$I''(x, k) = \min_{u(k),u(k+1),\ldots,u(K)} \left(\underset{w(k),w(k+1),\ldots,w(K)}{\mathrm{E}} \left\{ \sum_{j=k}^{K} L[x(j), u(j), w(j), j] \right\} \right), \tag{16}$$

[1] The latter assumption can be relaxed at the expense of defining additional state variables to account for correlation in k.

where

$$x(k) = x.$$

Under the assumption that $w(k)$ is uncorrelated in k, the recursive relation is found as

$$I''(x, k) = \min_{u(k)} \left[\operatorname*{E}_{w(k)} (L[x, u(k), w(k), k] + I''\{\Phi[x, u(k), w(k), k], k + 1\}) \right]. \tag{17}$$

As before, the optimal control for a given state x and stage k, $\hat{u}(x, k)$, is found as the value of $u(k)$ for which the minimum is attained.

D. *Iteration in Policy Space*

If the system equation functional, Φ, and the cost functional, L, have no explicit dependence on k, and if the total of number of stages, K, becomes infinite, the resulting optimization problem is called the "steady-state" optimization problem. This refers to the fact that the optimal control depends only on the state variables and not on the stage variable. This problem is of importance to water resource problems in long-term operations and planning. Because both the deterministic and stochastic version of this problem occur in practice, both will be analyzed.

The major change in the basic iteration equation (7) for the deterministic case is that the minimum cost function depends only on x and not on k. The iterative equation thus becomes

$$I(x) = \min_u \{L(x, u) + I[\Phi(x, u)]\}, \tag{18}$$

where the function I now appears on *both* sides of the equation. As before, the optimal control, $\hat{u}(x)$, is found as the value of u for which the minimum is attained.

A number of iterative techniques are available for solving these equations. The most straightforward approach is to treat the problem as a finite-stage problem and repeat the use of Eq. (7) until the differences between $I(x, k)$ and $I(x, k + 1)$ and/or $\hat{u}(x, k)$ and $\hat{u}(x, k + 1)$ are sufficiently small. If convergence occurs after a reasonable number of stages, then this technique is generally satisfactory.

An alternative approach is to *guess* the form of $I(x)$, say $I^{(0)}(x)$, and solve for a sequence of approximations to $I(x)$, $I^{(\ell)}(x)$, by using the equation

$$I^{(\ell+1)}(x) = \min_u \{L(x, u) + I^{(\ell)}[\Phi(x, u)]\}. \tag{19}$$

This technique, called by Bellman "approximation in function space" [1], is effective when a good initial approximation to $I(x)$ can be found.

Another approach, which generally has better convergence properties, is to guess an optimal control function, $\hat{u}^{(0)}(x)$, compute the corresponding minimum cost function, $I^{(0)}(x)$, and then obtain a sequence of optimal control functions, $\hat{u}^{(\ell)}(x)$, by using Eq. (19). The main difference between this approach, called by Bellman "approximation in policy space" [1], and the technique just discussed is that $I^{(\ell)}(x)$ is always computed as the true minimum cost function corresponding to $\hat{u}^{(\ell)}(x)$; generally, this is achieved by direct iteration using the following equation

$$I^{(\ell,r+1)}(x) = L[x, \hat{u}^{(\ell)}(x)] + I^{(\ell,r)}\{\Phi[x, \hat{u}^{(\ell)}(x)]\}, \tag{20}$$

where $I^{(j,r)}(x)$ denotes the value of $I^{(j)}(x)$ obtained after r iterations. Often, as an initial condition for these iterations,

$$I^{(j,0)}(x) = 0. \tag{21}$$

For the stochastic control case, the iterative equation becomes

$$I(x) = \min_{u} \left\{ \mathop{\mathsf{E}}_{w} [L(x, u, w)] + I[\Phi(x, u, w)] \right\}. \tag{22}$$

Computational procedures analogous to all three of those mentioned for the deterministic case can be developed.

The particular procedure that has received most attention in the literature is Howard's iteration in policy space [5]. In this procedure state variables are quantized to a set $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$, control variables are quantized to a set $u^{(1)}, u^{(2)}, \ldots, u^{(M)}$, and the stochastic variables are quantized to a set $w^{(1)}, w^{(2)}, \ldots, w^{(R)}$. It is assumed that the quantization levels and the state transformation equation are such that for a given quantized present state, quantized control, and quantized stochastic input, the next state is always a quantized state as well. It is then possible to define transition probabilities $p_{i\ell}^{m}$, where

$$p_{ij}^{m} = \text{prob}[x(k+1) = x^{(j)} \mid x(k) = x^{(i)}, u(k) = u^{(m)}]. \tag{23}$$

These probabilities are easily computable in terms of the functional $\Phi(x, u, w)$ and $p(w)$.

It can then be shown that the minimum cost function for any finite number of stages takes the form

$$I[x^{(i)}, k] = V_i + gk, \tag{24}$$

where V_i represents a transient cost that depends on the present state, and where g represents the steady-state cost per stage. For a given optimal control policy, $\hat{u}^{(\ell)}(x)$, the corresponding minimum cost function is computed by assuming the form of Eq. (24), substituting into Eq. (22), and utilizing the definition in Eq. (23). It is then found that

$$V_i + g = \sum_{j=1}^{N} p_{ij}^{m(\ell,i)}(L\{x^{(i)}, \hat{u}^{(\ell)}[x^{(i)}]\} + V_j), \tag{25}$$

where $m(\ell, i)$ is the index of the optimal control $\hat{u}^{(\ell)}[x^{(i)}]$ and where, for simplicity in notation and without loss of generality, it is assumed that the single-stage cost is not an explicit function of w. Equation (25) specifies N equations for the $(N + 1)$ unknowns V_i, $i = 1, 2, \ldots, N$, and g. However, it can easily be shown that one of the V_i—say, V_N—can be set to zero, and the resulting system of equations solved exactly. This operation Howard calls *value determination operation* (VDO) [5].

Having now obtained the $I^{(\ell)}[x^{(i)}]$ corresponding to $\hat{u}^{(\ell)}[x^{(i)}]$, a new optimal control function, $\hat{u}^{(\ell+1)}[x^{(i)}]$, is found by Howard's *policy improvement routine* (PIR) [5]. Basically, this procedure consists of minimizing g, where g is obtained as

$$g = \sum_{\ell=1}^{N} p_{ij}^{m}\{L[x^{(i)}, u^{(m)}] + V_j\} - V_i .$$

The minimization is over $u^{(m)}$, $m = 1, 2, \ldots, M$.

It can be shown [5] that iterative use of the VDO and PIR, starting from a given initial control function $\hat{u}^{(0)}[x^{(i)}]$, will eventually lead to the optimum control function $\hat{u}[x^{(i)}]$ and that the succesively determined values of g are monotonically nonincreasing. The strength of these results are the main reason this technique is used more than any of the iterative procedures for the stochastic case.

E. *Successive Approximations*

One of the most promising techniques for solving high-dimensional water resource problems is Bellman's successive approximations [3]. In this technique it is assumed that the optimal control sequence from a given initial state is desired, rather than optimal control at every admissible state. This is typically the case in a real-time control or operations problems. It is also convenient to assume that there are as many control variables as state variables; in multi-reservoir control problems, where the state variables are the water levels in the reservoirs and the control variables are the releases from each dam, this is the case.

The problem formulation can be as in any of the preceding sections. However, attention will be confined in this section to the deterministic, finite-stage problem of Part A of this section.

In order to begin the approach, a guess is made of an initial control sequence, $u^{(0)}(k)$, $k = 0, 1,..., K$. The corresponding trajectory is then determined as $x^{(0)}(k)$, $k = 1, 2,..., K$, from knowledge of $x(0)$, the given initial state, and the control sequence.

The basic idea behind the method is to solve a sequence of one-dimensional problems, rather than the complete n-dimensional problem. In one version of this technique that has been studied extensively, $(n - 1)$ of the state variables are fixed along the most recent sequence of states, $x^{(j)}(k)$. The remaining state variable is taken to be the state variable in a one-dimensional problem. The control for this problem is the complete control vector, $u(k)$, except that the restriction that $(n - 1)$ of these state variables follow the prescribed trajectory imposes $(n - 1)$ constraints on admissible controls. The performance criterion is that of the original problem, and other constraints are handled exactly as before. When a new optimal control sequence, $\hat{u}^{(j+1)}(k)$, has been found, the corresponding trajectory, $x^{(j+1)}(k)$, is computed and the process repeated for another state variable. Convergence is monotonic, but it is not always possible to attain the absolute optimum. Some variations of this basic technique that improve the likelihood of finding the absolute optimum are considered in Ref. [4]. The extension to stochastic control problems is also covered in Ref. [4].

Potentially, this technique has tremendously reduced computational requirements over other dynamic programming methods; these requirements grow linearly with the number of state variables rather than exponentially. In the example of Section III-B a four-reservoir optimization problem was solved in less than a minute on a moderate-speed computer.[2] Also, in Ref. [4] an airline scheduling problem with 70 state variables was solved in 3 minutes on the same computer.

F. *State Increment Dynamic Programming*

Another promising technique for high-dimensional problems is state increment dynamic programming [4]. This technique is particularly valuable for reducing the high-speed memory requirement over that of the method of Section II-A in problems where optimal control is desired at every admissible state. When only the optimal control sequence from a given initial state is desired, the technique can be modified to have reduced computing time as well. Finally, when there are as many control variables as state variables, a particularly efficient algorithm can be employed. The details of this technique are covered in Ref. [4].

In one application, where the high-speed memory requirement for the

[2] B5500.

normal dynamic programming approach was 10^6 locations, the high-speed memory requirement of state increment dynamic programming was on the order of 10^2 locations [4]. By using this technique, a computer program capable of solving general dynamic optimization problems having four or less state variables was recently written [4].

G. *Other Techniques with Reduced Computational Requirements*

A number of other dynamic programming techniques are of potential value in solving water resource problems but with reduced computational requirements over the method of Section II-A. These include quasilinearization [6], iterations about a nominal trajectory with increasingly finer grid sizes [7], use of Lagrange multipliers to replace state variables [7], and polynomial approximation of the minimum cost function [1]. These techniques and others are covered in a recent survey paper [8].

III. Examples

A. *Short-Term Optimization of a Pumped-Storage Two-Reservoir System*

1. *Problem Statement*

In Ref. [9] a pump-storage system is described. The basis for the problem is the San Luis Reservoir and its forebay, a joint facility of the State of California and Bureau of Reclamations in the State Water Project. However, simplifying assumptions were made to obtain a problem in which the basic principles were not obscured by details. The solution of this problem utilized forward dynamic programming.

The network configuration of the problem to be solved is shown in Fig. 1.

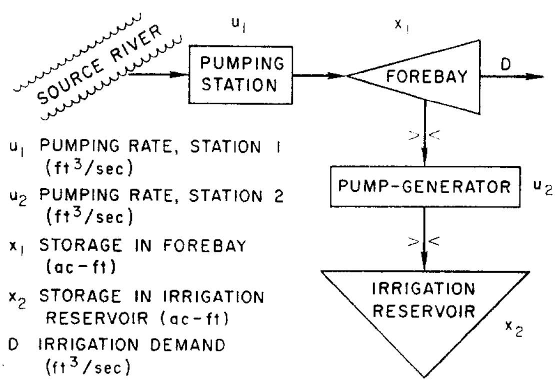

Fig. 1. Network configuration for two-reservoir example.

The water from the source river is pumped in the forebay x_1, from which it is either pumped into the storage reservoir x_2 or used to meet an irrigation demand. The pumps of the large reservoir can also function as generators of electrical power when water flows back to the forebay. The rate at which water can be removed from the source river has an upper limit, and the pumping plants have capacity limitations. Dollar values can be put on all costs and revenues. The problem is to operate within all constraints and to meet all demands on the system at minimum cost.

The quantized state variables are x_1 and x_2. The control variables u_1 and u_2 are allowed to vary continuously within certain upper and lower limits. Control u_1 varies from zero to an upper limit and control u_2 varies from some negative lower limit to some positive upper limit. A negative u_2 indicates that Pumping Station 2 is being used to generate electrical power. The irrigation demand D is limited to positive values and has the same units as the controls (a flow rate). Because of the dimensional differences in the u's and x's, a conversion factor is needed:

$$C = 12.3 \frac{\text{acre feet}}{\text{ft}^3/\text{sec}}.$$

The operating procedure is to be computed one day in advance and is reconsidered every hour; therefore, time is quantized into increments of one hour.

The state equations are the following:

$$x_1[(k+1)\Delta t] = x_1(k\Delta t) + C[u_1(k\Delta t)\Delta t - u_2(k\Delta t)\Delta t - D(k\Delta t)\Delta t]$$

$$x_2[(k+1)\Delta t] = x_2(k\Delta t) + C[u_2(k\Delta t)\Delta t], \tag{26}$$

where in this problem

$$\Delta t = 1 \text{ hour}.$$

Thus, the equations become

$$x_1(k+1) = x_1(k) + C[u_1(k) - u_2(k) - D(k)]$$

$$x_2(k+1) = x_2(k) + Cu_2(k). \tag{27}$$

However, operating the pumping stations for an hour incurs certain costs. The only pumping station operating cost considered in this problem is the cost of electrical power. This cost (K) is expressed as the cost of pumping at the rate of one ft^3/sec for one hour. The efficiencies of both pumping plants are the same, so the per-unit operating cost of each one is K if u_2 is positive. The efficiency of Station 2 changes when it is used as a generator; thus, there must be a different cost (benefit) K' when μ_2 is negative. It is assumed that the electrical power that Station 2 generates can be sold at the same price that power can be purchased and that the power cost varies during a day. Thus, the cost of producing additional power is greater than the cost of producing the base level of power. Intuitively, it appears that in order to meet the irrigation demand, and minimize cost, there are times during the day when it is most profitable to release water. The solution to the problem verifies this supposition and determines when each policy should be followed.

The cost accrued during the kth time increment is

$$L(k) = Ku_1(k) + K_2u_2(k)$$
$$K_2 = \begin{cases} K, & u_2 \geq 0 \\ K', & u_2 < 0. \end{cases} \tag{28}$$

The total cost from the initial time to time k is thus

$$I(x, k) = I(x, k-1) + L(k)$$
$$I(x, 0) = L(0) = 0 \qquad \text{for all} \qquad x. \tag{29}$$

The quantity $I(x, N)$ is the cost of operating the system from initial to final time ($t = t_f = N\Delta t$ or $k = N$) and terminating in state x. The problem becomes one of choosing the controls $u(k)$ of Eq. (27) for all values of k such that all constraints are satisfied and $I(x, N)$ is minimized for all x.

2. *A Typical Problem*

A FORTRAN program using forward dynamic programming has been implemented for the two-reservoir, two-pump station facility (Ref. [9]). The control is not quantized, but allowed to vary continuously between certain limits. The computed trajectories can therefore be forced to go from one quantized state to a quantized state at the next stage of the process. Thus, no interpolation is required and one has continuous, piecewise-linear trajectories in the state space.

Figure 2 shows the demand curve of irrigation water and the incremental power cost curve. The incremental power cost is the cost of the last megawatt-hour produced during each hour. Since the whole system analyzed here operates as an additional load or source to the electric power grid, it will either have to buy power at the incremental power cost or replace power which costs this much. This curve was derived from information given in Ref. [10]. The irrigation demand curve was arbitrarily assumed to be as shown. The initial value of the reservoir levels are the $k = 0$ values shown in Fig. 3.

The optimum cost for each terminal state varies considerably, and the one which is the overall optimum depends on the penalty assessed for arriving at each of these final states. If there is no penalty assessed for arriving at different terminal states, but a terminal constraint is imposed that the total amount of water in the two reservoirs must be 10 units, then

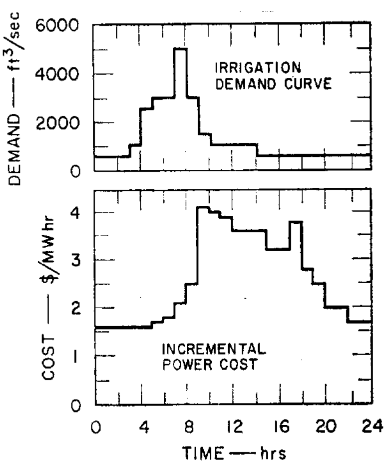

FIG. 2. Input quantities for the numerical example: Irrigation demand and incremental cost as a function of time.

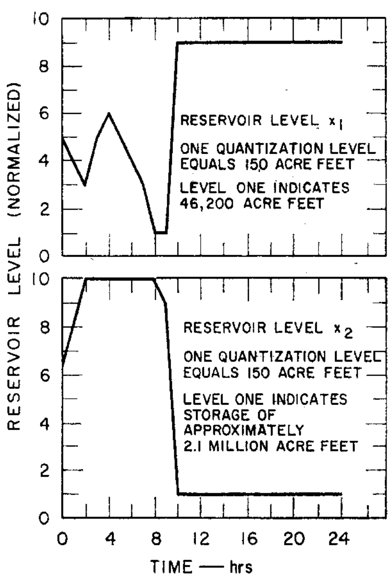

FIG. 3. State variables as a function of time.

the optimum terminal state is $x_1 = 9$, $x_2 = 1$. The minimum cost for this state is $I(x, N) = \$611.93$. The optimal policy corresponding to this state is shown in Figs. 3-5; the reservoir levels as function of time are shown in Fig. 3, the optimum controls are shown in Fig. 4, and the cumulative operating cost is shown in Fig. 5.

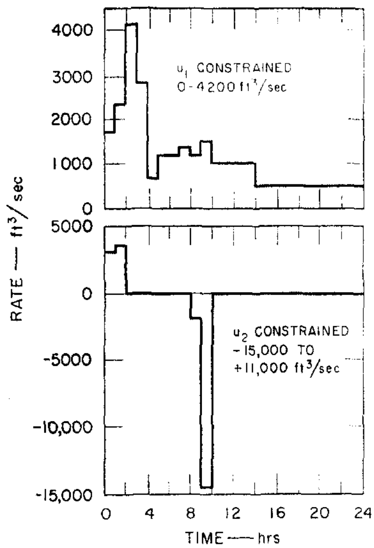

FIG. 4. Pumping rates of the numerical example as a function of time.

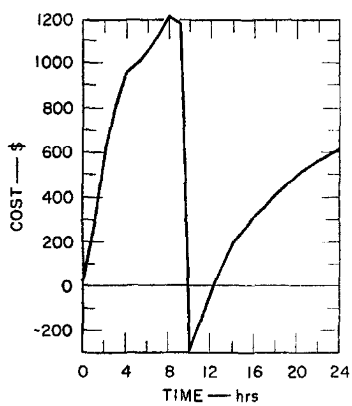

FIG. 5. Cumulative operating cost.

In this case, the best policy is to fill Reservoir 2 early in the day when power is least costly and then drain all that is possible during the period (the tenth hour) when the return is greatest. Reservoir 1 is first operated to ensure that it is at its lowest allowable level in the tenth hour and is thus able to receive the water released from Reservoir 2. In the tenth hour the reservoir rises to the ninth quantization level and remains there. No change in the level of the reservoir is possible because the irrigation demand requires less than one quantization level of water and the additional demand is met by u_1. The cumulative operating cost (Fig. 5) reflects the pumping policy shown in Fig. 4. It shows high cost as the pumps fill Reservoir 2 early in the day, but the return for this policy is high during the tenth hour.

3. *Extensions*

The real world is never as simple as the example described above. However, dynamic programming is able to handle a wide variety of constraints that do result from physical situations. Some of these which can be expressed in the context of this example are cited below. One constraint is a limitation on the amount of water than can be pumped from the source river during a 24-hour period. This is a very real problem in California: The Sacramento River Delta could be contaminated by salt water if the flow of the river were disturbed too much. As a result, the irrigation requirement often is also expressed as the amount to be delivered during a 24-hour period.

Many pump/generator stations have already been built and integrated into a power system. These stations provide a "spinning reserve" during certain hours of the day. This responsibility requires that u_2 be constrained to be less than some negative value during these hours. Other contractual requirements would be imposed on a realistic system. These include penalties for not exceeding minimum levels of irrigation or electric power demand. If too much water or electric power is produced, the return for the excess may be less than for the basic deliveries. Since the short-term control situation is imbedded in a longer-term operation, the final values of the two reservoir levels are confined to certain regions of the state space. A penalty cost is assessed for not reaching the desired final state and bonus given if this value is exceeded.

B. *Short-Term Optimization of a Multipurpose Four-Reservoir System*

In this section, the optimum operation over 24 hours of a multipurpose four-reservoir system is determined. The reservoir network, which contains both series and parallel connections, is shown in Fig. 6. In this optimization, use of water for power generation, irrigation, flood control and recreation is considered. Interaction of the short-term optimization with longer-term operating policies is also taken into account.

The amount of water in the ith reservoir is denoted as x_i, $i = 1, 2, 3, 4$, where each x_i is expressed in normalized units.

On the basis of potential use of the reservoir for recreation purposes, a minimum water level for each reservoir is specified; the amount of water needed to achieve this level is arbitrarily set as $x_i = 0$, and a constraint is

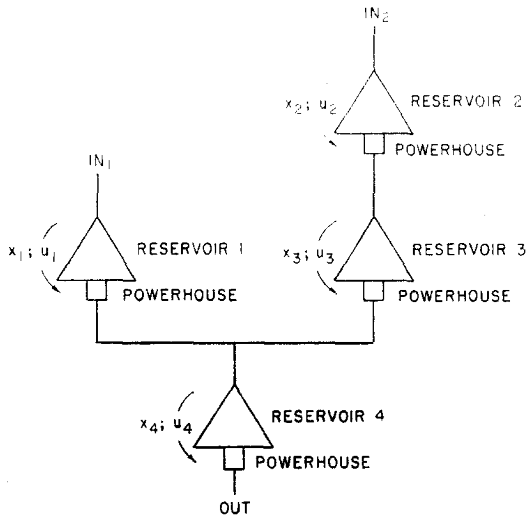

FIG. 6. Network configuration of four-reservoir problem.

imposed that the amount of water in each reservoir cannot drop below this value.

On the basis of flood control considerations, a maximum water level for each reservoir is established. The amount of water needed to raise the level from the minimum to the maximum value is then expressed in terms of the normalized units, and a constraint is imposed that each x_i cannot exceed this level.

The particular constraints considered in this example are expressed as:

$$\begin{aligned} 0 &\leqslant x_1 \leqslant 10 \\ 0 &\leqslant x_2 \leqslant 10 \\ 0 &\leqslant x_3 \leqslant 10 \\ 0 &\leqslant x_4 \leqslant 15. \end{aligned} \tag{30}$$

The flow of water between reservoirs is also expressed in the same normalized units; the control variables $u_i(k)$, $i = 1, 2,..., 4$, specify the amount of water released from the ith reservoir over the kth time interval. In this example each time interval is two hours. For each reservoir a maximum flow is determined by the capacity of the power generators, and a minimum flow is determined by considering the use of the downstream river beds for navigation, conservation, and municipal and industrial water supplies. For this example the constraints were

$$0 \leqslant u_1 \leqslant 3$$
$$0 \leqslant u_2 \leqslant 4$$
$$0 \leqslant u_3 \leqslant 4$$
$$0 \leqslant u_4 \leqslant 7. \tag{31}$$

The system equations express how the water flows between the reservoirs. They are:

$$x_1(k+1) = x_1(k) - u_1(k) + IN_1$$
$$x_2(k+1) = x_2(k) - u_2(k) + IN_2$$
$$x_3(k+1) = x_3(k) - u_3(k) + u_2(k)$$
$$x_4(k+1) = x_4(k) - u_4(k) + u_3(k) + u_1(k)$$
$$k = 0, 1, \ldots 11. \tag{32}$$

The inflows IN_1 and IN_2 are assumed constant over the day as

$$IN_1 = 2$$
$$IN_2 = 3. \tag{33}$$

The performance criterion considers the use of water for both power generation and irrigation. It is assumed that there is a power generation station at each reservoir outflow. The benefit from the flow over a given two-hour period is assumed to be a linear function of the flow, (i.e., the benefit from a flow out of reservoir at time k is $c_i(k)\ u_i(k)$. The function $c_i(k)$ is based on the power curve in Part A of this section. The values of $c_i(k)$ are shifted in k with respect to each other to account for the transport delay of water between reservoirs. This delay is four hours from Reservoir 1 to Reservoir 4, four hours from Reservoir 2 to Reservoir 3, and two hours from Reservoir 3 to Reservoir 4. The values of $c_i(k)$, $i = 1, 2, 3, 4$ are shown in Table I.

Irrigation benefits are considered only for the outflow from Reservoir 4. The benefit is again linear with flow—i.e., the benefit from flow $u_4(k)$ is $c_5(k)\ u_4(k)$. The function $c_5(k)$ is shown in Table I.

The benefit function also includes a terminal cost for failing to reach a specified level for each reservoir at the end of the day. This function accounts for the long-term policy of filling or emptying the reservoir during a particular season. This function assesses a heavy penalty for having less than the specified amount of water at the end of the day, but gives no credit for having more than this amount. The particular function used was

$$\psi_i[x_i(12), m_i] = \begin{Bmatrix} -40[x_i(12) - m_i]^2, & x_i(12) \leqslant m_i \\ 0, \text{ otherwise} & \end{Bmatrix}, \tag{34}$$

where m_i = desired level of reservoir i at end of the day ($k = 12$).

TABLE I

CONSTANTS IN PERFORMANCE CRITERION

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

This problem has been solved by successive approximations. The initial state was taken to be

$$\begin{aligned} x_1(0) &= 5 \\ x_2(0) &= 5 \\ x_3(0) &= 5 \\ x_4(0) &= 5. \end{aligned} \tag{35}$$

The desired final state was

$$\begin{aligned} m_1 &= 5 \\ m_2 &= 5 \\ m_3 &= 5 \\ m_4 &= 7. \end{aligned} \tag{36}$$

The system dynamic equations are as in Eqs. (32) and (33). The constraints are expressed in Eqs. (30) and (31). The performance criterion is

$$J = \sum_{k=0}^{11} \sum_{i=1}^{4} c_i(k)\, u_i(k) + \sum_{k=0}^{11} c_5(k)\, u_4(k) + \sum_{i=1}^{4} \psi_i[x_i(12), m_i], \tag{37}$$

where $c_i(k)$, $i = 1, 2,... 5$ is specified in Table I, $\psi_i[x_i(12), m_i]$ is as shown in Eq. (34) and m_i, $i = 1, 2, 3, 4$, are given in Eq. (36).

The initial policy chosen is shown in Table II. Basically, this policy consists of setting outflow equal to inflow at every time period, so that the water level in each reservoir remains constant. The only exception to this policy occurs at the end of the day, when the terminal cost function is taken into account.

TABLE II

INITIAL POLICY

k	$x_1(k)$	$x_2(k)$	$x_3(k)$	$x_4(k)$	$u_1(k)$	$u_2(k)$	$u_3(k)$	$u_4(k)$
0	5	5	5	5	2	3	3	5
1	5	5	5	5	2	3	3	5
2	5	5	5	5	2	3	3	5
3	5	5	5	5	2	3	3	5
4	5	5	5	5	2	3	3	5
5	5	5	5	5	2	3	3	5
6	5	5	5	5	2	3	3	5
7	5	5	5	5	2	3	3	5
8	5	5	5	5	2	3	3	5
9	5	5	5	5	2	3	3	5
10	5	5	5	5	2	3	3	5
11	5	5	5	5	2	3	3	3
12	5	5	5	7				

Total Benefit = 362.5

The optimum policy is shown in Table III. The improvement in benefit was from 362.5 units to 401.3 units. The amount of computer time required for convergence to the optimum policy was about 30 seconds in the B5500.

The extension of this approach to larger systems is clearly feasible. Time-varying constraints and more general types of performance criteria can easily be handled. Furthermore, the problem formulation can be modified to perform optimization over time periods other than 24 hours. The effects of stochastic variations can also be taken into account. At this time it appears that optimization of 20-reservoir systems is well within the capability of present-day computers.

C. *Optimization in the Presence of Stochastic Inflows*

1. *Problem Statement*

The following example (see Ref. [11]) shows how dynamic programming can be applied to an annual scheduling problem with stochastic inputs. The

TABLE III

OPTIMUM POLICY

k	$x_1(k)$	$x_2(k)$	$x_3(k)$	$x_4(k)$	$u_1(k)$	$u_2(k)$	$u_3(k)$	$u_4(k)$
0	5	5	5	5	1	4	0	0
1	6	4	8	7	0	1	0	2
2	8	5	10	5	0	2	4	7
3	10	7	8	1	2	0	4	7
4	10	10	4	0	3	3	4	7
5	9	10	3	0	3	4	4	7
6	8	9	3	0	3	4	4	7
7	7	8	3	0	3	4	4	7
8	6	7	3	0	3	4	4	7
9	5	6	3	0	3	4	4	7
10	4	5	3	0	3	4	4	0
11	3	4	3	7	0	2	0	0
12	5	5	5	7				

Total Benefit = 401.3

problem posed can be solved by means of iteration in policy space to yield a series of optimum policies for the management of one reservoir.

The problem is expressed in terms of a transaction between two businessmen—one the manager of a reservoir and one the owner of a hydroelectric plant fed by this reservoir. A similar problem could be posed even if both facilities were operated by the same group. The manager of a water storage reservoir wishes to maximize the average return from his reservoir over a long period of time. The reservoir has three sources of income.

(1) An annual payment from agricultural users of water which is released during the growing season—April through September.

(2) An annual return from recreational use which is a function of the reservoir level on 30 September—the end of the water year.

(3) A return for each acre-foot of water which is released during the winter months between 1 October and 31 March. This revenue comes from the owner of a hydroelectric power generator downstream.

During the winter, when much of the precipitation falls as snow and thus is not immediately available, this power facility is faced with a severe water shortage. Thus, the owner is willing to pay well for each acre-foot of water guaranteed to be delivered, less for each acre-foot delivered in excess of this guarantee, and invokes a penalty for each promised acre-foot that is not delivered.

Each October the manager is faced with the problem of deciding how much water should be promised to the hydroelectric company. He knows the distribution of annual inflows into the reservoir and that 65 percent of this inflow will occur during the duration of the contract. He must meet the terms of the contract unless there is no water in the reservoir. He will deliver no more water than is specified in the contract unless the reservoir will otherwise overflow.

The operating policy during the summer months is already specified. The reservoir is operated during these months to yield the best immediate return. The only trade-off is between use of the reservoir for recreational purposes and sale of water for irrigation. There is no return for release of irrigation water above a certain specified amount. In the spring the inflow for the rest of the water year is much better defined. The manager knows how much snow fell in the watershed of his reservoir and thus is able to specify the remaining inflow more closely, but not with certainty.

Under the assumptions outlined here the whole operating policy for a year is specified once the contract with the hydroelectric company is signed. The manager's problem thus becomes to decide how much water he should promise to deliver and how this amount should vary with the level of the reservoir in October. This example answers this question for the reservoir whose characteristics are described in the next section.

2. *The Source of Data*

The problem described in Part 1 is fictional, but it is strongly related to problems which are faced by water planners. The tables and graphs described below were adapted from the records of several government agencies concerned with water resources. A fictitious reservoir is assumed to have a storage capacity of 50,000 acre-feet, which is discretized into eleven values ranging from 0 to 50,000 acre-feet in 5,000-acre-foot increments.

Figure 7 relates the annual benefit to the annual delivery of irrigation water. Negative value is given to small deliveries of water because the agricultural investment of the users is not utilized. The graph approximates a smooth curve by a series of linear functions to simplify digital computer use of this data. Figure 8 relates the annual recreation benefit to the end of the water year (30 September) storage in the reservoir. A negative value is given to zero storage because the investment in recreation facilities is not utilized. The values of this benefit at the quantized values of reservoir storage are used in computations. An analysis of the history of water year inflows yielded the discrete probability distribution function which is shown in Table IV.

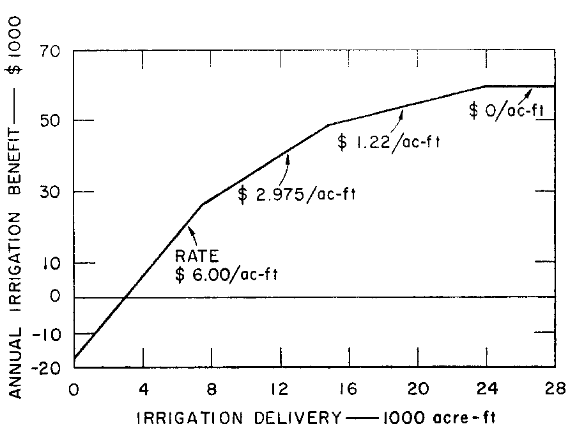

FIG. 7. Annual irrigation benefit.

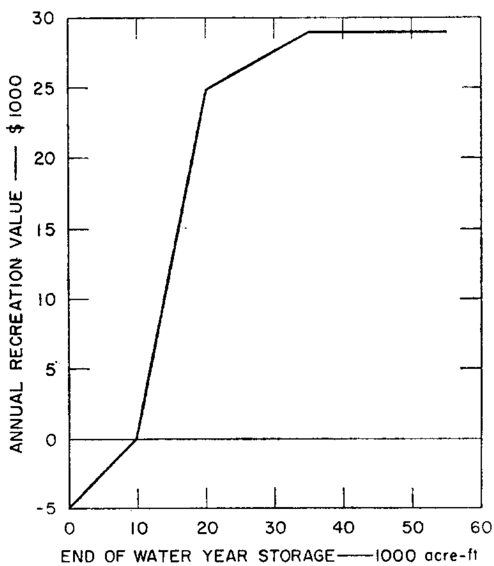

FIG. 8. Recreation benefit.

Figures 7 and 8 and Table IV describe the state of knowledge of the manager of the reservoir in the autumn when the contract with the electric power company is negotiated.

By comparing records of the predicted and actual inflows into a number of reservoirs during the summer months, it is possible to obtain a probability distribution function of the expected flow during the summer given the amount that flowed during the winter. Table V shows this result. To ease computations it is assumed that 65 percent of the annual inflow (Table IV) occurs during the winter months. However, once this period is past the rest of the inflow is described by a "more packed" probability distribution function than the one available at the beginning of the water year. The decrease in uncertainty results from knowledge of the amount of water stored in the mountain in the form of snow and from the shorter prediction interval necessary to predict to the end of the water year.

TABLE IV

PROBABILITY DISTRIBUTION OF INFLOW

Probability of Occurrence	Volume (acre-feet)
0.02	1,715
0.08	3,920
0.10	6,550
0.10	9,300
0.10	12,200
0.10	15,200
0.10	18,800
0.10	23,500
0.10	29,400
0.10	39,100
0.08	55,800
0.02	92,000

TABLE V

SUMMER PROBABILITY DISTRIBUTION OF INFLOW

Probability	Annual Predicted Rainfall which Occurs in the Summer (percentage)
0.10	20
0.20	30
0.40	35
0.20	40
0.10	50

3. *Problem Formulation and Indicated Solution*

Application of the technique of iteration in policy space (Section II-D) requires that quantized state variables, transition probabilities, and rewards be defined. The level of the reservoir at the beginning of each water year is a continuous variable which can be quantized into a relatively small number of discrete values. Neither the transition probabilities nor the rewards for these transitions are specified directly in Part C-2 of this section, but both can be computed from this data and the conditions specified in Part C-1.

Once P_{ij}^m and Q_i^m have been computed for all m, i, and j (in notation of Section II-D), the remainder of the solution to this problem becomes straightforward application iteration in policy space discussed in Section II-D. The first step is to apply PIR:

$$g + V_i = Q_i^m + \sum_{j=1}^{N} (P_{ij}^m)(V_j) \qquad \text{for all} \quad m \quad \text{and} \quad i, \tag{38}$$

where

g = gain from process each year

V_i = the relative value of being in state $x^{(i)}$ in the steady-state situation

N = number of quantitizations of x.

In the first iteration, all V_j are set equal to zero. The policy m_i^* for which $g + V_i$ is largest is chosen for each $x^{(i)}$. The resulting quantities are substituted into the VDO routine:

$$g + V_i = Q_i^{m_i^*} + \sum_{j=1}^{N} (P_{ij}^{m_i^*})(V_j) \qquad i = 1, 2, \ldots N$$

$$V_1 = 0. \tag{39}$$

The resulting values of V_j are substituted into Eq. (39). The process is repeated until two successive iterations yield the same set of m_i^* $(1 \leqslant i \leqslant N)$. The release levels corresponding to this set of m_i^* are the outflows that the manager should promise as a function of the reservoir level. These values are the answer to the problem.

4. *Results*

The technique described in Section III-C-4 above has been implemented in a FORTRAN computer program. This program has been machine translated to ALGOL and run on the Burroughs B5500 at Stanford Research Institute. It requires about one minute to complete the computation of one case and usually converges to a solution in three or four policy iterations.

A number of cases have been run (Ref. [11]) using the data outlined in part C-2 of this section. The only quantities that were allowed to vary were the outflows that could be promised the hydroelectric company and the charges associated with this contract (S1, S2, and S3). These charges are chosen so that the return per acre-foot is comparable to that obtained from the sale of irrigation water. The charge under contract (S1) is chosen in the range of the slopes shown in Fig. 7 so there is a conflict between various policies. Two cases and the results are shown in Table VI.

In these two cases, the decision options and the contractual penalties remain the same, but the contract payment (S1) changes. When the contract price drops from $10 to $6 per acre-foot, the expected annual income (the gain) declines from $110,284 to $72,839. The policy changes indicate that the reservoir management should not risk losing irrigation and recreation revenue at the lower contract price. The relative values show the long-term value of being in a given state $x^{(i)}$ at the present time compared to the value of being in state $x^{(1)}$ at the present time. Notice that the percentage loss in value of a full reservoir is even greater than the percentage decline in contract price.

D. *Optimum Planning of System Additions*

The benefit of a large natural resource project is dependent on the timing of its construction. If it is completed too early, years may pass before its benefits can be fully utilized. However, if it is completed too late, there will be a long period when system users are denied its benefits or forced to pay higher costs than necessary. Thus, the decision of when to commit capital to a large project becomes critical. Unused investment is waste, and so is underdevelopment.

TABLE VI

EXAMPLE FOR ITERATION IN POLICY SPACE

Alternative Polices	
0 acre-feet	15000 acre-feet
2000 acre-feet	20000 acre-feet
5000 acre-feet	27000 acre-feet
8000 acre-feet	35000 acre-feet
11000 acre-feet	45000 acre-feet

TABLE VI (*continued*)

Results

State	Case 1 S1 = 10 $/acre-feet S2 = 3 $/acre-feet S3 = 15 $/acre-feet Optimal Policy acre-feet	Relative Value $	Case 2 S1 = 6 $/acre-feet S2 = 3 $/acre-feet S3 = 15 $/acre-feet Optimal Policy acre-feet	Relative Value $
1	8000	0	5000	0
2	15000	51,108	11000	29,287
3	20000	101,108	15000	60,000
4	27000	148,312	20000	90,000
5	27000	202,875	27000	118,443
6	35000	251,108	27000	148,787
7	35000	297,460	35000	180,000
8	45000	351,108	35000	206,744
9	45000	397,460	45000	240,000
10	45000	428,952	45000	266,744
11	45000	461,914	45000	290,872
Gain	$110,284		$72,839	

S1—payment for each acre-foot delivered under contract
S2—payment for each acre-foot delivered in excess of the contract
S3—penalty for each acre-foot contracted for but not delivered

Dynamic programming is one way to optimally schedule when additional investment should be made when a long range solution is desired. A. Korsak of SRI[12] has worked out the example given below of planning expansion of a power facility 30 years into the future. The problem may be stated as follows.

A power system has a current hydro capacity of 200-MW and a current demand of 500-MW. The hydro-generated energy remains constant and no cost is associated with this type of generation. The power demand is assumed to grow at a rate of 7 percent per year. To simplify the model, it is assumed that the power demand makes discrete jumps of 7 percent at the beginning of a given year and remains constant until the beginning of the next year.

The difference between the power demand and the hydro capacity can be made up each year in one of two ways. One may use plants that were purchased in preceding years and buy the remaining power at the rate of 12 mills per kWh or one may buy and install either a 250-MW or a 500-MW nuclear plant (but not one of each) at the beginning of the year. The operating cost of a nuclear plant is 3 mills per kWh. A 250-MW plant costs \$3.45 $\times$ 10^7 dollars and a 500-MW plant cost \$5.60 $\times$ 10^7 dollars. Since buying a plant might not meet the power requirement in a given year, one might still need to buy some power.

An interest rate of 12 percent per annum on initial capital is used in determining the costs of any power plants.

The cost incurred during the kth year can thus be expressed as follows:

$$C(k) = \frac{V(k)}{1.12^{k-1}} + \frac{1.27 \times 10^7}{1.12^k} x(k) + 5.08 \times 10^7[1.07^k - h(k) - x(k)], \tag{40}$$

where

$C(k)$ = cost incurred during the kth year adjusted to beginning of first year with interest of 12 percent
$V(k)$ = cost of purchasing a plant in the kth year
$h(k)$ = hydro capacity in the kth year
$x(k)$ = total number of 500-MW units installed at time k (a 250-MW unit is considered half a 500-MW unit).

If no power is purchased,

$$C(k) = \frac{V(k)}{1.12^{k-1}} + \frac{1.27 \times 10^7}{1.12^k} x(k). \tag{41}$$

The optimization is to minimize $\sum_{i=0}^{29} C(k)$ subject to specified terminal constraints. The state variable $x(k)$ satisfies the equation

$$x(k) = x(k-1) + u(k),$$

where $u(k)$ is the control variable representing the decision to add 0, 250-MW, or 500-MW.

This problem was solved by forward dynamic programming. As indicated in Section II-B, the optimum policy for reaching any feasible final state is determined. If the final state is selected as the one for which least cost is incurred, the optimal policy is as shown in Fig. 9. In this case the system capacity at the end is less than the demand; however, it can also be seen that in early years the capacity exceeds demand. These results reflect the assumption that no consideration is made of system operation after the final year. If the final state is constrained so that system capacity at the end exceeds demand, the optimal policy is exactly the same as in Fig. 9, except that during the last year before termination a 500-MW unit is purchased.

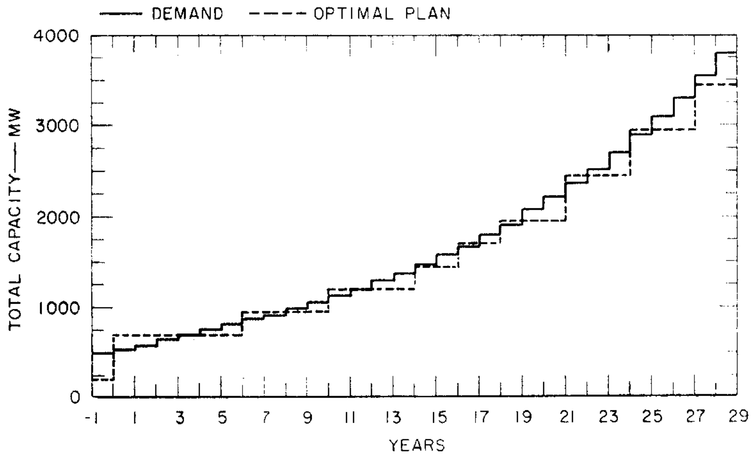

FIG. 9. Comparison of demand and capacity for 30 years.

IV. CONCLUSIONS

In this paper it has been shown that dynamic programming provides a powerful approach to many of the optimization problems that occur in water resource systems. Extremely general system equations and performance criteria can be handled, multiple constraints of a wide variety present no difficulties, an absolute optimum solution is obtained, the results are in a feedback control form, and stochastic variations can be explicitly taken into account. The major difficulty in applying it to practical problems has been the computational requirements associated with the standard computational algorithm.

This paper has discussed many extensions of the standard algorithm that reduce these computational difficulties. State increment dynamic programming can substantially reduce the number of high-speed memory locations that are required with little loss in computer execution time. Forward dynamic programming is a particularly efficient method for obtaining solutions when initial conditions are specified. The technique of successive approximations offers a very promising means of providing major reductions in both the computer memory and computer time required. By using a computational algorithm based on one or a combination of these techniques many additional problems can be solved.

The four problems discussed at length in Section III show the breadth of the water resource problems than can be solved. These range from hourly control of a system involving hydroelectric power, water storage, and irrigation to long-range optimum investment planning. The stochastic character of nature is considered in the example of Section III-C. None of these examples are the most difficult of their type that can now be solved, but they do demonstrate the principles and power of dynamic programming. Much more complicated problems are being solved now, and further research in computer technology and dynamic programming techniques will allow an even greater range of applications.

References

1. R. Bellman. "Dynamic Programming." Princeton Univ. Press, Princeton, New Jersey, 1957.
2. R. Bellman. "Adaptive Control Processes." Princeton Univ. Press, Princeton, New Jersey, 1961.
3. R. Bellman and S. Dreyfus. "Applied Dynamic Programming." Princeton Univ. Press, Princeton, New Jersey, 1962.
4. R. E. Larson. "State Increment Dynamic Programming." American Elsevier, New York, New York, 1967.
5. R. A. Howard. "Dynamic Programming and Markov Processes." Wiley, New York, New York, 1960.
6. R. Bellman and R. E. Kalaba. "Quasilinearization and Nonlinear Boundary-Value Problems." American Elsevier, New York, New York, 1965.
7. G. L. Nemhauser. "Introduction to Dynamic Programming." Wiley, New York, New York, 1966.
8. R. E. Larson. Computational aspects of dynamic programming. *In* "1967 IEEE International Convention Record," Part 3, March 1967.
9. W. G. Keckler. Optimum management of a combined pumped hydro and irrigation storage facility. Technical Memorandum Stanford Research Institute, Menlo Park, California, March 1966.
10. P. J. Bernard, J. F. Dopazo, and G. W. Stagg. A method for economic scheduling of a combined pumped hydro and steam generating system. *IEEE Trans. Power Apparat. Sys.* **PAS-83**, No. 1, pp. 23–29, 1964.
11. W. G. Keckler. Reservoir management and policy iteration. Technical Memorandum, Stanford Research Institute, Menlo Park (June 1966).
12. M. Cuénod and A. J. Korsak. Optimal planning of power systems. Technical Memorandum, Stanford Research Institute, Menlo Park, California, June 1966.