# 13 Dynamic Programming

Dynamic programming is one of two modern approaches to the control problem.[1] It can be applied directly to the general control problem:[2]

$$
\begin{aligned}
&\max_{\{\mathbf{u}(t)\}} J = \int_{t_0}^{t_1} I(\mathbf{x}, \mathbf{u}, t)\, dt + F(\mathbf{x}_1, t_1) \\
&\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\
&\mathbf{x}(t_0) = \mathbf{x}_0 \qquad\qquad\qquad\qquad (13.0.1) \\
&\mathbf{x}(t_1) = \mathbf{x}_1 \\
&\{\mathbf{u}(t)\} \in U.
\end{aligned}
$$

The approach of dynamic programming is that of taking the particular control problem to be solved, embedding it in a wider class of problems characterized by certain parameters, and applying a basic principle, the "Principle of Optimality," to obtain a fundamental recurrence relation connecting members of this class of problems. With some additional smoothness assumptions the fundamental recurrence relation implies a basic partial differential equation, "Bellman's equation," which, when solved, yields the solution to the wider class of problems and hence, as a special case, the solution to the particular problem at hand.

## 13.1 The Principle of Optimality and Bellman's Equation

The *Principle of Optimality* states that:

> "An optimal policy has the property that, whatever the initial state and decision [i.e., control] are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."[3] **(13.1.1)**

This principle is illustrated for the case of a problem with a single state variable in Fig. 13.1. The curve $x^*(t)$ for $t_0 \leq t \leq t_1$ is the trajectory associated with the optimal control, where it is assumed that the initial and terminal states are given. This trajectory is divided into two parts: ① and ②

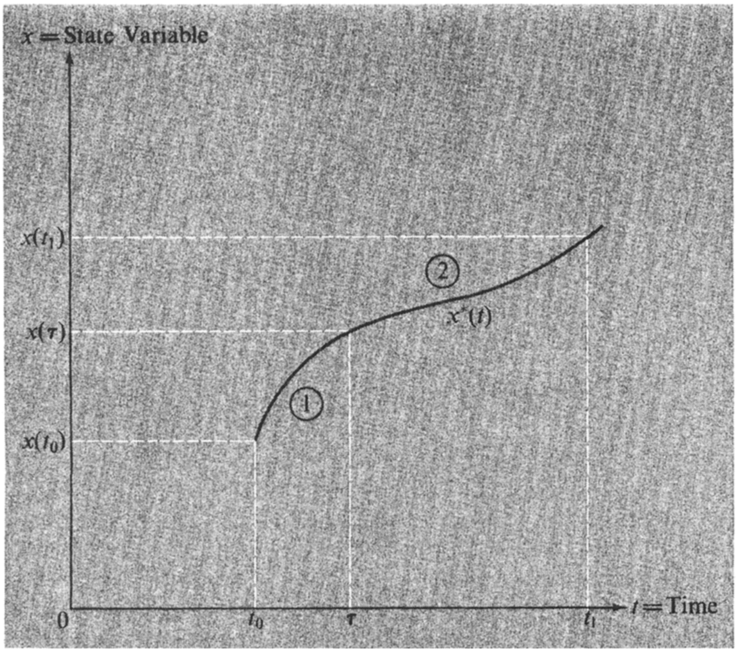


**Fig. 13.1**

According to the Principle of Optimality, ② Must in Its Own Right Represent an Optimal Trajectory

at time $\tau$. According to the Principle of Optimality, trajectory ②, defined for $\tau \leq t \leq t_1$, must, in its own right, represent an optimal trajectory with respect to the initial condition $x(\tau)$. Thus, the second portion of an optimal trajectory must be an optimal trajectory in its own right, independent of how the system arrived at the initial conditions for this second portion.

Assuming a solution exists for the general control problem (13.0.1) let:

$$J^*(\mathbf{x}, t) \tag{13.1.2}$$

be the *optimal performance function*, the maximized value of the objective functional for the problem starting at the initial state $\mathbf{x}$ at time $t$.[4] The problem is thereby embedded in a wider class of problems characterized by their $n + 1$ initial parameters. The optimal value of the objective function for the particular problem at hand, (13.0.1), is then:

$$J^* = J^*(\mathbf{x}_0, t_0). \tag{13.1.3}$$

According to the Principle of Optimality, if $J^*(\mathbf{x}, t)$ is the optimal performance function for the problem starting at state $\mathbf{x}$ and time $t$, then $J^*(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)$ is the optimal performance function for the second portion of the optimal trajectory, starting at state $\mathbf{x} + \Delta\mathbf{x}$ and time $t + \Delta t$. Over the interval of time between $t$ and $t + \Delta t$, however, the only increment to the optimal performance function could come from the intermediate function (integrand) which adds $I(\mathbf{x}, \mathbf{u}, t)\,\Delta t$. The optimal performance function over the entire time span starting at time $t$ should then equal the optimum sum of the contributions from the two portions of the time span. Thus:

$$J^*(\mathbf{x}, t) = \max_{\{\mathbf{u}(t)\}} [I(\mathbf{x}, \mathbf{u}, t)\,\Delta t + J^*(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)], \tag{13.1.4}$$

which is the *fundamental recurrence relation.*

A critical assumption of the dynamic programming approach is that the optimal performance function $J^*(\mathbf{x}, t)$ is a single-valued and continuously differentiable function of the $n + 1$ variables; that is, that solutions to the wider class of problems are single-valued and continuous with respect to variations in the initial parameters.[5] By this assumption a Taylor's series expansion can be employed to represent $J^*(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)$ at the point $(\mathbf{x}, t)$ as:

$$J^*(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) = J^*(\mathbf{x}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial J^*}{\partial t}\Delta t + \cdots \tag{13.1.5}$$

where $\partial J^*/\partial\mathbf{x}$ is the row vector:

$$\frac{\partial J^*}{\partial \mathbf{x}} = \left(\frac{\partial J^*}{\partial x_1}, \frac{\partial J^*}{\partial x_2}, \ldots, \frac{\partial J^*}{\partial x_n}\right). \tag{13.1.6}$$

Inserting (13.1.5) in (13.1.4) yields:

$$0 = \max_{\{\mathbf{u}(t)\}} \left[I(\mathbf{x}, \mathbf{u}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\frac{\Delta\mathbf{x}}{\Delta t} + \frac{\partial J^*}{\partial t} + \cdots\right]. \tag{13.1.7}$$

and taking the limit as $\Delta t \to 0$, where

$$\lim_{\Delta t \to 0} \frac{\Delta\mathbf{x}}{\Delta t} = \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{13.1.8}$$

yields:

$$-\frac{\partial J^*}{\partial t} = \max_{\{\mathbf{u}(t)\}} \left[I(\mathbf{x}, \mathbf{u}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}, t)\right]. \tag{13.1.9}$$

This equation is the basic partial differential equation of dynamic programming and is called *Bellman's equation.*[6] The second term in the bracket is the inner product of the row vector $\partial J^*/\partial \mathbf{x}$ and the column vector $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$, so Bellman's equation can also be written:

$$-\frac{\partial J^*}{\partial t} = \max_{\{\mathbf{u}(t)\}} \left[ I(\mathbf{x}, \mathbf{u}, t) + \sum_{j=1}^{n} \frac{\partial J^*}{\partial x_j} f_j(\mathbf{x}, \mathbf{u}, t) \right]. \tag{13.1.10}$$

The boundary condition associated with Bellman's equation is the terminal condition:

$$J^*(\mathbf{x}(t_1), t_1) = F(\mathbf{x}_1, t_1) \tag{13.1.11}$$

which states that the value of the optimal performance function for the problem starting at the terminal state and terminal time is simply the value of the final function $F(\cdot\ \cdot)$ evaluated at this state and time.

If Bellman's equation were solved, it would yield the optimal performance function and hence solve the problem as the particular value of this function for the specific initial conditions given. In general, however, this first order partial differential equation, which is typically nonlinear, has no analytic solution. Numerical methods, which solve discrete versions of Bellman's equation using high speed digital computers are possible in principle, but even modern high-speed computers have insufficient storage capacity to allow for a reasonable approximation to a solution when the dimensionality of the system, $n$, is even moderately large.[7] Bellman refers to this limitation as the "curse of dimensionality."

## 13.2 Dynamic Programming and the Calculus of Variations

The dynamic programming problem is more general than the classical calculus of variations problem, so if the dynamic programming problem is specialized into that of the classical calculus of variations, the necessary condition of dynamic programming, Bellman's equation, must imply the necessary conditions of the calculus of variations, including the Euler equation, the Legendre condition, the Weierstrass condition, and the Weierstrass-Edmann corner conditions.[8]

The classical calculus of variations problem is the special case of the dynamic programming problem (13.0.1) for which:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{u} \\ \Omega &= E^n, \end{aligned} \tag{13.2.1}$$

that is, the control variables are the time rates of change of state variables and the control variables are unconstrained. In this case Bellman's equation becomes:

$$-\frac{\partial J^*}{\partial t} = \max_{\{\dot{\mathbf{x}}\}} \left[ I(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial J^*}{\partial \mathbf{x}} \dot{\mathbf{x}} \right]. \tag{13.2.2}$$

Assuming the expression in brackets has a maximum, a necessary condition for its maximization is:

$$\frac{\partial}{\partial \dot{\mathbf{x}}} \left[ I(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial J^*}{\partial \mathbf{x}} \dot{\mathbf{x}} \right] = \mathbf{0}, \tag{13.2.3}$$

or, since $\partial J^*/\partial \mathbf{x}$ is independent of $\dot{\mathbf{x}}$:

$$\frac{\partial I}{\partial \dot{\mathbf{x}}} = -\frac{\partial J^*}{\partial \mathbf{x}}. \tag{13.2.4}$$

Taking a total time derivative:

$$\frac{d}{dt}\left(\frac{\partial I}{\partial \dot{\mathbf{x}}}\right) = -\frac{d}{dt}\left(\frac{\partial J^*}{\partial \mathbf{x}}\right) = -\frac{\partial^2 J^*}{\partial t\, \partial \mathbf{x}} - (\dot{\mathbf{x}})' \frac{\partial^2 J^*}{\partial \mathbf{x}^2}, \tag{13.2.5}$$

where use has been made of the fact that $\partial J^*/\partial \mathbf{x}$ depends on $\mathbf{x}$ and $t$ and where:

$$\frac{\partial^2 J^*}{\partial t\, \partial \mathbf{x}} = \left(\frac{\partial^2 J^*}{\partial t\, \partial x_1}, \ldots, \frac{\partial^2 J^*}{\partial t\, \partial x_n}\right)$$

$$\frac{\partial^2 J^*}{\partial \mathbf{x}^2} = \begin{pmatrix} \frac{\partial^2 J^*}{\partial x_1^2} & \cdots & \frac{\partial^2 J^*}{\partial x_1\, \partial x_n} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \frac{\partial^2 J^*}{\partial x_n\, \partial x_1} & \cdots & \frac{\partial^2 J^*}{\partial x_n^2} \end{pmatrix} \tag{13.2.6}$$

But from Bellman's equation:

$$-\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial J^*}{\partial t}\right) = \frac{\partial I}{\partial \mathbf{x}} + (\dot{\mathbf{x}})' \frac{\partial^2 J^*}{\partial \mathbf{x}^2}. \tag{13.2.7}$$

Combining (13.2.5) and (13.2.7) and using the equality of the mixed partials yields the Euler equation of the calculus of variations:

$$\frac{\partial I}{\partial \mathbf{x}} - \frac{d}{dt}\left(\frac{\partial I}{\partial \dot{\mathbf{x}}}\right) = \mathbf{0}. \tag{13.2.8}$$

The Legendre condition is obtained immediately from the second order necessary conditions for the above maximization:

$$\frac{\partial^2}{\partial \dot{\mathbf{x}}^2}\left[I(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\dot{\mathbf{x}}\right] \text{ negative semidefinite or negative definite.} \tag{13.2.9}$$

Since $\partial J^*/\partial \mathbf{x}$ is independent of $\dot{\mathbf{x}}$, the condition is:

$$\frac{\partial^2 I}{\partial \dot{\mathbf{x}}^2} \text{ negative semidefinite or negative definite,} \tag{13.2.10}$$

which is the Legendre condition.

The Weierstrass condition is also obtained from the maximization within Bellman's equation, which states that if $\{\dot{\mathbf{x}}(t)\}$ is a solution:

$$I(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\dot{\mathbf{x}} \geq I(\mathbf{x}, \dot{\mathbf{z}}, t) + \frac{\partial J^*}{\partial \mathbf{x}}\dot{\mathbf{z}} \tag{13.2.11}$$

for any column vector $\dot{\mathbf{z}}$. Rearranging terms and using (13.2.4):

$$I(\mathbf{x}, \dot{\mathbf{z}}, t) - I(\mathbf{x}, \dot{\mathbf{x}}, t) - \frac{\partial I}{\partial \dot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}, t)(\dot{\mathbf{z}} - \dot{\mathbf{x}}) \leq 0, \tag{13.2.12}$$

which is the Weierstrass condition.

Finally, the Weierstrass-Erdmann corner conditions are obtained from the equations:

$$\begin{gathered} \frac{\partial I}{\partial \dot{\mathbf{x}}} = -\frac{\partial J^*}{\partial \mathbf{x}} \\ I - \frac{\partial I}{\partial \dot{\mathbf{x}}}\dot{\mathbf{x}} = I + \frac{\partial J^*}{\partial \mathbf{x}}\dot{\mathbf{x}} = -\frac{\partial J^*}{\partial t}. \end{gathered} \tag{13.2.13}$$

Since $\partial J^*/\partial \mathbf{x}$ and $\partial J^*/\partial t$ are continuous, it follows that:

$$\frac{\partial I}{\partial \dot{\mathbf{x}}} \text{ and } \left(I - \frac{\partial I}{\partial \dot{\mathbf{x}}}\dot{\mathbf{x}}\right) \tag{13.2.14}$$

are continuous across corners, which are the Weierstrass-Erdmann corner conditions.

The dynamic programming approach, therefore, yields the necessary conditions for the classical calculus of variations problems. Dynamic programming can also be used to treat the constrained calculus of variations problems as discussed in Sec. 12.4. For example, for the isoperimetric problem, where the constraint is:

$$\int_{t_0}^{t_1} G(\mathbf{x}, \dot{\mathbf{x}}, t) = c, \tag{13.2.15}$$

Bellman's equation takes the form:

$$-\frac{\partial J^*}{\partial t} = \max_{\{\dot{\mathbf{x}}\}} \left[ I(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial J^*}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial J^*}{\partial c} G(\mathbf{x}, \dot{\mathbf{x}}, t) \right] \tag{13.2.16}$$

which yields the same conditions as the calculus of variations formulation since the Lagrange multiplier is:

$$y = \frac{\partial J^*}{\partial c} \tag{13.2.17}$$

i.e., the variation in the optimal value of the functional with respect to the constant $c$ of the constraint. In general, the partials of the optimal performance function can be interpreted as Lagrange multipliers, measuring the sensitivity of the solution.

## 13.3 Dynamic Programming Solution of Multistage Optimization Problems

In many dynamic problems time enters as a discrete rather than a continuous variable and such problems, referred to as *multistage optimization problems*, can be solved by dynamic programming.[9]

In multistage optimization problems the time variable takes the discrete values:

$$t_0, t_0 + 1, t_0 + 2, \dots, t_1. \tag{13.3.1}$$

The *state* of the system at time $t$ is given by the vector $\mathbf{x}_t$ and the *control* at time $t$ is given by the vector $\mathbf{u}_t$. The state at time $t + 1$ is then given by:

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t), \qquad t = t_0, t_0 + 1, \dots, t_1 - 1 \tag{13.3.2}$$

where $\mathbf{f}_t(\cdot\ \cdot)$ is a vector of continuously differentiable functions of the contemporary state and control variables. The initial state is:

$$\mathbf{x}_0 \tag{13.3.3}$$

which is assumed given. The objective function is

$$J = \sum_{t=t_0}^{t_1-1} I_t(\mathbf{x}_t, \mathbf{u}_t) + F(\mathbf{x}_{t_1}, t_1), \tag{13.3.4}$$

which is to be maximized by choice of a sequence of control vectors:

$$\{\mathbf{u}_{t_0}, \mathbf{u}_{t_0+1}, \ldots, \mathbf{u}_{t_1-1}\} \tag{13.3.5}$$

subject to the condition that these controls belong to a given control set:

$$\mathbf{u}_t \in \Omega, \qquad t = t_0, t_0 + 1, \ldots, t_1 - 1. \tag{13.3.6}$$

The analogies to the (continuous time) control problem should be evident.

The approach of dynamic programming here, as before, is to embed the problem to be solved in a wider class of problems characterized by certain parameters and then to use the Principle of Optimality to obtain a fundamental recurrence relation. Taking as parameters of the multistage optimization problem above the initial state and initial time, the optimal performance function is:

$$J^*(\mathbf{x}, t), \tag{13.3.7}$$

which is the optimal value of the objective function for a problem starting at state $\mathbf{x}$ at time $t$, the solution to the problem at hand being:

$$J^*(\mathbf{x}_0, t_0). \tag{13.3.8}$$

By the Principle of Optimality it follows that:

$$J^*(\mathbf{x}, t) = \max_{\mathbf{u}_t} [I_t(\mathbf{x}_t, \mathbf{u}_t) + J^*(\mathbf{x}_{t+1}, t + 1)], \tag{13.3.9}$$

which states that the optimal value of the objective function starting at state $\mathbf{x}$ at time $t$ consists of the optimal sum of the amount added at time $t$, $I_t(\mathbf{x}_t, \mathbf{u}_t)$, and the remaining optimal value, $J^*(\mathbf{x}_{t+1}, t + 1)$. Using equation (13.3.2) the recurrence relation is:

$$J^*(\mathbf{x}, t) = \max_{\mathbf{u}_t} [I_t(\mathbf{x}_t, \mathbf{u}_t) + J^*(\mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t), t + 1)]. \tag{13.3.10}$$

The boundary condition is:

$$J^*(\mathbf{x}_1, t_1) = F(\mathbf{x}_{t_1}, t_1), \tag{13.3.11}$$

which states that the optimal value of the objective function starting at $\mathbf{x}_1$ and $t_1$ is simply the value of the final function evaluated at this state and time. The analogies to the continuous time problem should be evident.

Another approach to multistage optimization problem is to characterize the problem not by the initial state and initial time, but by the initial state and the amount of time *left to go* in the problem. The optimal performance function is then:

$$J_\tau^*(\mathbf{x}_{t_1-\tau}) \tag{13.3.12}$$

which is the optimal value of the objective function for a problem of length $\tau$ starting from the state $\mathbf{x}_{t_1-\tau}$. The solution to the problem at hand is therefore that for $\tau = t_1$: $J_{t_1}^*(\mathbf{x}_0)$. In this case the method of dynamic programming solves the problem by working *back* from terminal time $t_1$ via a sequence of solutions. The first member of this sequence is $J_0^*(\mathbf{x}_{t_1})$, which is the optimal value of the objective function for a problem of zero length starting (and staying) at $\mathbf{x}_{t_1}$. But the optimal value for this problem is simply the value of the final objective function:

$$J_0^*(\mathbf{x}_{t_1}) = F(\mathbf{x}_{t_1}, t_1). \tag{13.3.13}$$

Now consider $J_1^*(\mathbf{x}_{t_1-1})$, which is the optimal value of the objective function for the problem of length one, starting at $\mathbf{x}_{t_1-1}$, called the first *stage*. This problem of length one, involving the choice of the control vector $\mathbf{u}_{t_1-1}$, is optimized by maximizing the particular part of the objective function relating to this time, $I_{t_1-1}(\mathbf{x}_{t_1-1}, \mathbf{u}_{t_1-1})$ *plus* the optimal value for the problem starting at $t_1$:

$$J_1^*(\mathbf{x}_{t_1-1}) = \max_{\mathbf{u}_{t_1-1}} [I_{t_1-1}(\mathbf{x}_{t_1-1}, \mathbf{u}_{t_1-1}) + J_0^*(\mathbf{x}_{t_1})] \tag{13.3.14}$$

or, using (13.3.2):

$$J_1^*(\mathbf{x}_{t_1-1}) = \max_{\mathbf{u}_{t_1-1}} [I_{t_1-1}(\mathbf{x}_{t_1-1}, \mathbf{u}_{t_1-1}) + J_0^*(\mathbf{f}_{t_1-1}(\mathbf{x}_{t_1-1}, \mathbf{u}_{t_1-1}))]. \tag{13.3.15}$$

This choice of control at stage one is consistent with the Principle of Optimality since the control $\mathbf{u}_{t_1-1}$ is optimal with respect to the state $\mathbf{x}_{t_1-1}$ resulting from the first $t_1 - 1$ choices of control vectors $\mathbf{u}_{t_0}, \mathbf{u}_{t_0+1}, \ldots, \mathbf{u}_{t_1-2}$. Similarly for the second stage, with two time units to go, for which:

$$J_2^*(\mathbf{x}_{t_1-2}) = \max_{\mathbf{u}_{t_1-2}} [I_{t_1-2}(\mathbf{x}_{t_1-2}, \mathbf{u}_{t_1-2}) + J_1^*(\mathbf{f}_{t_1-2}(\mathbf{x}_{t_1-2}, \mathbf{u}_{t_1-2}))]. \tag{13.3.16}$$

The general recurrence relation, for stage $\tau$, is:

$$J_\tau^*(\mathbf{x}_{t_1-\tau}) = \max_{\mathbf{u}_{t_1-\tau}} [I_{t_1-\tau}(\mathbf{x}_{t_1-\tau}, \mathbf{u}_{t_1-\tau}) + J_{\tau-1}^*(\mathbf{f}_{t_1-\tau}(\mathbf{x}_{t_1-\tau}, \mathbf{u}_{t_1-\tau}))]. \tag{13.3.17}$$

The problem is then solved as $J_{t_1}^*(\mathbf{x}_0)$, the last optimal value found in the sequence of single stage optimizing problems described by the functional equations (13.3.17) for $\tau = 1, 2, \ldots, t_1$, with the boundary condition

(13.3.13). The multistage optimization problem is thereby reduced, via dynamic programming, to a sequence of single stage optimization problems.[10]

As an example of the dynamic programming approach to multistage optimization problems, consider the problem of choosing a set of nonnegative numbers $u_{t_0}, u_{t_0+1}, \ldots, u_{t_1}$ summing to a given number $c$ so as to maximize a separable objective function.[11]

$$\begin{gathered}\max J = \sum_{t=t_0}^{t_1} I_t(u_t) \\ u_t \geq 0, \qquad t = t_0, t_0+1, \ldots, t_1 \\ \sum_{t=t_0}^{t_1} u_t = c.\end{gathered} \tag{13.3.18}$$

The constant $c$ can be interpreted as the total available level of resources, and can be regarded as a parameter of the problem. The optimal performance function is:

$$J_\tau^*(c) \tag{13.3.19}$$

for a process of length $\tau$ ending at $t_1$ where total resources equal $c$. For a process of length zero ending at $t = t_1$:

$$J_0^*(c) = \max_{u_{t_1}=c} I_{t_1}(u_{t_1}) = I_{t_1}(c). \tag{13.3.20}$$

For the one stage process ending at $t_1$, the resource has to be divided between $u_{t_1}$ and $u_{t_1-1}$. By the Principle of Optimality:

$$J_1^*(c) = \max_{0 \leq u_{t_1}-1 \leq c} [I_{t_1-1}(u_{t_1-1}) + J_0^*(c - u_{t_1-1})], \tag{13.3.21}$$

so, from (13.3.20):

$$J_1^*(c) = \max_{0 \leq u_{t_1}-1 \leq c} [I_{t_1}(u_{t_1-1}) + I_{t_1}(c - u_{t_1-1})]. \tag{13.3.22}$$

The general recurrence relation is then:

$$J_\tau^*(c) = \max_{0 \leq u_{t_1}-\tau \leq c} [I_{t_1-\tau}(u_{t_1-\tau}) + J_{\tau-1}^*(c - u_{t_1-\tau})] \tag{13.3.23}$$

showing how the total resources are optimally divided between $u_{t_1-\tau}$ applied to $I_{t_1-\tau}(u_{t_1-\tau})$ and $c - u_{t_1-\tau}$ applied to the remaining portion of the process, $J_{\tau-1}^*(c - u_{t_1-\tau})$. The problem is solved sequentially from the boundary condition, (13.3.20), using the general recurrence relation, (13.3.23), until the $t_1$ stage problem is solved with $J_{t_1}^*(c)$.

Consider the specific problem of minimizing the sum of squares of nonnegative variables subject to the constraint that they total to a given number:

$$\max J = -\sum_{t=t_0}^{t_1} u_t^2$$

$$u_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1 \tag{13.3.24}$$

$$\sum_{t=t_0}^{t_1} u_t = c.$$

Using the method of dynamic programming, the solution to the problem of length zero is:

$$J_0^*(c) = \max_{u_{t_1}=c} \; -u_{t_1}^2 = -c^2. \tag{13.3.25}$$

The first functional equation, for a process of length one, is, from (13.3.21):

$$J_1^*(c) = \max_{0\leq u_{t_1-1}\leq c} [-u_{t_1-1}^2 + J_0^*(c - u_{t_1-1})], \tag{13.3.26}$$

so, using (13.3.25):

$$J_1^*(c) = \max_{0\leq u_{t_1-1}\leq c} [-u_{t_1-1}^2 - (c - u_{t_1-1})^2]. \tag{13.3.27}$$

For a maximum the partial derivative of the bracket term must vanish, requiring:

$$u_{t_1-1} = \tfrac{1}{2}c, \tag{13.3.28}$$

which is consistent with the constraint that $0 \leq u_{t_1-1} \leq c$. Thus, half the resources should be applied at time $t_1$ and half at time $t_1 - 1$. The next functional equation is:

$$J_2^*(c) = \max_{0\leq u_{t_1-2}\leq c} [-u_{t_1-2}^2 + J_1^*(c - u_{t_1-2})], \tag{13.3.29}$$

but $J_1^*(c)$ equalled, at the optimum point, $-\frac{1}{2}c^2$, so:

$$J_2^*(c) = \max_{0\leq u_{t_1-2}\leq c} [-u_{t_1-2}^2 - \tfrac{1}{2}(c - u_{t_1-2})^2]. \tag{13.3.30}$$

For a maximum:

$$u_{t_1-2} = \tfrac{1}{3}c, \tag{13.3.31}$$

so one-third of the available resources are applied at time $t_1 - 2$, with the remaining two-thirds divided equally at $t_1 - 1$ and $t_1$. In general, the solution is:

$$u_{t_0} = u_{t_0+1} = \cdots = u_{t_1} = \frac{c}{(t_1 - t_0) + 1} \tag{13.3.32}$$

that is, equal amount of the resource are applied at each point in time in order to minimize the sum of squares.

## PROBLEMS

**13-A.** A classical control problem is the *brachistochrone* problem of determining a curve between two points $P$ and $Q$ such that a particle moving frictionlessly along the curve under the influence of gravity starting with zero velocity at $P$ reaches $Q$ in minimum time. Suppose the point $P'$ lies on the solution curve between $P$ and $Q$. Is the curve between $P'$ and $Q$ optimal and, if so, in what sense? What about the curve between $P$ and $P'$?

**13-B.** Find Bellman's equation for the problem:

$$\max_{\{u(t)\}} J = -\int_{t_0}^{t_1} [(x-c)^2 + u^2]\, dt$$

$$\dot{x} = ax + bu$$

$$x(t_0) = x_0$$

$$x(t_1) = x_1.$$

**13-C.** Find Bellman's equation for the problem:

$$\max_{\{u(t)\}} J = \int_{t_0}^{t_1} (x - u)\, dt$$

$$\dot{x} = \sqrt{u}$$

$$x(t_0) = x_0$$

$$x(t_1) = x_1$$

$$0 \le u \le x.$$

**13-D.** Using dynamic programming solve the control problem for a minimum effort servomechanism subject to linear equations of motion:

$$\max_{\{\mathbf{u}(t)\}} J = \int_{t_0}^{t_1} (\mathbf{x}'\,\mathbf{D}\mathbf{x} + \mathbf{u}'\mathbf{E}\mathbf{u})\,dt$$
$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{x}(t_0) = \mathbf{x}_0$$
$$\mathbf{x}(t_1) = \mathbf{x}_1,$$

where **D** and **E** are negative definite matrices and **A** and **B** are given matrices.

**13-E.** Find Bellman's equation for the problem of moving from a given initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ to the origin $\mathbf{x}(t_1) = \mathbf{0}$ in minimum time by choice of a control trajectory $\{\mathbf{u}(t)\} \in U$, where $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$.

**13-F.** Apply the results of the last problem to solve the special case of moving from $(x_1, x_2)'$ to $(0, 0)'$ in minimum time where:

$$\dot{x}_1 = V \cos x_3$$
$$\dot{x}_2 = V \sin x_3$$
$$\dot{x}_3 = u,$$

and where $V$, the magnitude of the velocity, is given as:

$$V = V_0\sqrt{1 + \left(\frac{x_2}{a}\right)^2}.$$

**13-G.** Suppose the optimal performance function for the control problem of the Lagrange type were taken to be a function of the initial state **x** and the *duration* of the process $\tau$:

$$J^*(x, \tau) = \max_{\{\mathbf{u}(t)\}} \int_{t_0}^{t_0+\tau} I(\mathbf{x}, \mathbf{u}, t)\,dt$$
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$
$$\mathbf{x}(t_0) = \mathbf{x}_0$$
$$\{\mathbf{u}(t)\} \in U,$$

where $J^*(\mathbf{x}_0, t_1, -t_0)$ is the solution to the given problem. Find the partial differential equation implied by the dynamic programming approach and compare to (13.1.9).

**13-H.** Using dynamic programming, show that in the calculus of variations problem for which $I(\cdots)$ is independent of time $t$ a necessary condition for a maximum is:

$$I - \frac{\partial I}{\partial \dot{\mathbf{x}}}\dot{\mathbf{x}} = \text{constant}.$$

**13-I.** Obtain the transversality condition of the calculus of variations using dynamic programming.

**13-J.** Consider the generalization of the example of section 13.3:

$$\max J = -\sum_{t=t_0}^{t_1} w_t u_t^2$$

$$u_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\sum_{t=t_0}^{t_1} u_t = c$$

where $w_{t_0}, w_{t_0+1}, \ldots, w_{t_1}$ are given nonnegative weights.

1. Solve the problem by dynamic programming.
2. Show that the dynamic programming solution is consistent with the nonlinear programming solution.
3. Solve the specific problem in which $t_0 = 0$, $t_1 = 2$, $w_{t_0} = 2$, $w_{t_0+1} = 3$, $w_{t_1} = 6$ and $c = 100$.

**13-K.** Another generalization of the example of section 13.3 is:

$$\max J = -\sum_{t=t_0}^{t_1} u_t^{p_t},$$

$$u_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\sum_{t=t_0}^{t_1} u_t = c,$$

where $p_{t_0}, p_{t_0+1}, \ldots, p_{t_1}$ are given positive constants.

1. Solve by dynamic programming.
2. Show that the dynamic programming solution is consistent with the nonlinear programming solution.
3. Solve the specific problem in which $t_0 = 0$, $t_1 = 2$, $p_{t_0} = 1$, $p_{t_0+1} = 2$, $p_{t_1} = 3$ and $c = 100$.
4. Solve the general problem if the conditions on the control variables are:

$$u_t \geq 1, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\prod_{t=t_0}^{t_1} u_t = c.$$

**13-L.** Solve the problem

$$\max J = \sum_{t=t_0}^{t_1} \frac{p_t s_t}{(s_t + u_t)}$$

where $p_t$ and $s_t$ are parameters such that:

$$p_t \geq 0, \qquad s_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\sum_{t=t_0}^{t_1} p_t = 1$$

and the control variables satisfy:

$$u_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\sum_{t=t_0}^{t_1} u_t = c.$$

**13-M.** In the problem:

$$\max J = \sum_{t=t_0}^{t_1} I(u_t)$$

$$u_t \geq 0, \qquad t = t_0, t_0 + 1, \ldots, t_1$$

$$\sum_{t=t_0}^{t_1} u_t = c,$$

show that if $F(\cdot)$ is a convex function, then the maximum is $F(c)$.

**13-N.** Solve the nonlinear programming problem:

$$\max F(\mathbf{x}) = x_1 x_2, \ldots, x_n = \prod_{j=1}^{n} x_j$$

$$x_j \geq 0, \qquad j = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} x_j = a,$$

by dynamic programming.

**13-O.** Solve by dynamic programming the problem of finding a path between entries in the matrix $\mathbf{A} = (a_{ij})$ starting at $a_{11}$ and ending at $a_{mn}$ which moves only to the right or down and which minimizes the sum of the entries $a_{ij}$ encountered.

**13-P.** The linear programming problem:

$$\max_{\mathbf{x}} F(\mathbf{x}) = \mathbf{cx} \quad \text{subject to} \quad \mathbf{Ax} \leq \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}$$

can be treated as a discrete multistage optimization problem and solved using the Principle of Optimality by letting $F_k^*(b_1, b_2, \ldots, b_m)$ be the optimal performance function, defined as the solution to the problem subject to the added constraints:

$$x_{k+1} = x_{k+2} = \cdots = x_n = 0.$$

Find the recurrence relation and boundary condition for the optimal performance function. Is this method a reasonable alternative to the simplex method?

## FOOTNOTES

[1] The basic references for dynamic programming are Bellman (1957) (1961), Bellman and Dreyfus (1962), Feldbaum (1965), Nemhauser (1966), Kaufmann and Cruon (1967), and White (1969).

[2] For a more complete discussion of the general control problem see Chapter 11.

[3] See Bellman (1957). The proof of the necessity of the Principle of Optimality follows immediately by contradiction. Aris (1964) expresses the principle as, "If you don't do the best with what you happen to have got, you'll never do the best you might have done with what you should have had."

[4] Note that, whereas $J$ is a *functional*, dependent on the control trajectory $\{\mathbf{u}(t)\}$, $J^*$ is a *function*, dependent on the $n + 1$ parameters $\mathbf{x}$ and $t$.

[5] In many problems these smoothness assumptions are *not* satisfied, and it is generally not known in advance whether they hold for any particular problem. See Pontryagin et al. (1962). As an example of a solution which does not vary smoothly with respect to the parameters, consider the problem of finding geodesics (shortest distances between points) on a sphere. The solution is a great circle. Thus, as a special case, the shortest distance between two points on the Earth's equator is along the equator itself. Now suppose the initial point is moved along the Equator but away from the terminal point. Eventually a point is reached where the shortest distance would be found by moving in a direction opposite to that first used. At this point, the derivative of the shortest distance with respect to the initial point (measured, for example, by the longitude of that point) would be discontinuous.

[6] If $\{\mathbf{u}^*(t)\}$ solves the maximization problem on the right hand side of Bellman's equation and the function $H(\mathbf{x}, \partial J^*/\partial \mathbf{x}, t)$ is defined as:

$$H\left(\mathbf{x}, \frac{\partial J^*}{\partial \mathbf{x}}, t\right) = I(\mathbf{x}, \mathbf{u}^*, t) + \frac{\partial J^*}{\partial \mathbf{x}}\, \mathbf{f}(\mathbf{x}, \mathbf{u}^*, t),$$

then the resulting partial differential equation:

$$H\left(\mathbf{x}, \frac{\partial J^*}{\partial \mathbf{x}}, t\right) + \frac{\partial J^*}{\partial t} = 0$$

is called the *Hamilton-Jacobi equation*. See, in addition to the basic references of Footnote 1, Gelfand and Fomin (1963) and Hestenes (1966).

[7] The temporary storage requirement in the dynamic programming approach requires $Q^n$ computer memory locations, where $Q$ is the size of the grid; i.e., the number of discrete points taken by each of the state variables. If, for example, each state variable is divided into 100 discrete points and $n = 4$, then the memory requirement is 100 million locations. Since the high speed (core) memory of most modern computers is less than 100 million locations, dynamic programming routines must rely extensively on low speed (disk or tape) memory. There are, however, several ways of reducing the problems of dimensionality. See Bellman and Dreyfus (1962).

[8] See Bellman (1957) (1961), Dreyfus (1965), and Berkovitz and Dreyfus (1966).

[9] See Bellman (1957), Aris (1961) (1964), Blackwell (1962), and Roberts (1964).

[10] As in the continuous case, the numerical solution of multistage optimization problems via dynamic programming using a computer can rapidly run into the problem of insufficient storage. For such a solution it is necessary to find and store the entire sequence of functions $J_\tau^*(\mathbf{x}_{t_1-\tau})$ and solutions are typically obtained only with the help of certain approximations. See Bellman and Dreyfus (1962).

[11] See Bellman (1957) and Bellman and Dreyfus (1962). This problem is formally similar to a nonlinear programming problem with a separable objective function. For a discussion of the use of dynamic programming to solve certain nonlinear programming problems see Hadley (1964).

## BIBLIOGRAPHY


Aris, R., *The Optimal Design of Chemical Reactors*. New York: Academic Press Inc., 1961.

———, *Discrete Dynamic Programming*. Waltham, Mass.: Blaisdell Publishing Co., 1964.

Bellman, R., *Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1957.

———, *Adaptive Control Processes: A Guided Tour*. Princeton, N.J.: Princeton University Press, 1961.

Bellman, R., and S. Dreyfus, *Applied Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1962.

Berkovitz, L., and S. Dreyfus, "A Dynamic Programming Approach to the Nonparametric Problem in the Calculus of Variations," *J. Math. and Mech.* 15 (1966):83–100.

Blackwell, D., "Discrete Dynamic Programming," *Annals of Mathematical Statistics*, 33 (1962):719–26.

Dreyfus, S., *Dynamic Programming and the Calculus of Variations*. New York: Academic Press Inc., 1965.

Feldbaum, A. A., *Optimal Control Systems*, trans. from Russian by A. Kraiman. New York: Academic Press Inc., 1965.

Gelfand, I. M., and S. V. Fomin, *Calculus of Variations*, trans. by R. A. Silverman. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.

Hadley, G., *Nonlinear and Dynamic Programming*. Reading, Mass.: Addison-Wesley Publishing Co., Inc., 1964.

Hestenes, M., *Calculus of Variations and Optimal Control Theory*. New York: John Wiley & Sons, Inc., 1966.

Kaufmann, A., and R. Cruon, *Dynamic Programming*, trans. from French by H. C. Sneyd. New York: Academic Press Inc., 1967.

Nemhauser, G., *An Introduction to Dynamic Programming*. New York: John Wiley & Sons, Inc., 1966.

Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, *The Mathematical Theory of Optimal Processes*, trans. by K. N. Trirogoff. New York: Interscience Publishers, John Wiley & Sons, Inc., 1962.

Roberts, S. M., *Dynamic Programming in Chemical Engineering and Process Control*. New York: Academic Press Inc., 1964.

White, D. J., *Dynamic Programming*. San Francisco, Calif.: Holden-Day, Inc., 1969.