Dynamic Programming:
Theory and Algorithms

Sirong Luo

Faculty of Statistics and Data Sciences
Shanghai University of Finance and Economics

Introduction

Dynamic programming is an approach to model and solve multi-period decision problems. The fundamental principle of dynamic programming is the Bellman equation, a certain kind of optimality condition. As we detail in this chapter, the central idea of the Bellman equation is to break down a multi-stage problem into multiple two-stage problems. Under suitable conditions, the Bellman equation yields a recursion that helps in characterizing the solution and in computing it. Before embarking on a formal description, we illustrate the dynamic programming approach via some examples.

Example 13.1 (Matches puzzle)

Suppose there are 30 matches on a table and I play the following game with a clever opponent: I begin by picking up 1, 2, or 3 matches. Then my opponent must pick 1,2 , or 3 matches. We continue alternating until the last match is picked up. The player who picks up the last match loses. How can I (the first player) be sure of winning?

Solution. If I can ensure that it will be my opponent's turn when 1 match remains, I certainly win. Let us work backwards one step: If I can ensure that it will be my opponent's turn when 5 matches remain, I will also win. The reason for this is that no matter what he does when there are 5 matches left, I can make sure that when he has his next turn, only 1 match will remain. Hence it is clear that I win if I can force my opponent to play when 5 matches remain. We can continue working backwards and conclude that I will ensure victory if I can force my opponent to play when $5, 9, 13, 17, 21, 25$, or 29 matches remain. Since the game starts with 30 matches on the table, I can ensure victory by picking 1 match at the beginning, bringing the number down to 29 .

Example 13.2 (Knapsack problem)

Given a set of items, each with a certain weight and value, select the collection of items with total maximum value such that their total weight does not exceed some fixed weight limit W.

Solution. Let $w_t > 0$ and $v_t > 0$ be the weight and value respectively of item t for $t = 1, \ldots, n$. The knapsack problem can be formulated as an integer program and solved via the technique covered in Chapter 8.2. We next illustrate an alternative approach via dynamic programming. Consider the problem as a sequence of binary decisions $x_t \in \{0, 1\}$ corresponding to "include" or "do not include" item t for $t = 1, \ldots, n$. To find the optimal selection of items, we can work "backwards" as we did in the matches puzzle. Let W_t be the remaining amount of weight available at stage $t = 1, \ldots, n$ with $W_1 = W$, and let $J_t(W_t)$ denote the value of an optimal collection of items if we started selecting items at stage t with remaining weight limit W_t.

The value function $J_t(W_t)$ satisfies the following backward recursion for $t = 1, 2, \ldots, n-1$:

$$J_t(W_t) = \begin{cases} J_{t+1}(W_t) & \text{if} \quad w_t > W_t \\ \max\{J_{t+1}(W_t), J_{t+1}(W_t - w_t) + v_t\} & \text{if} \quad w_t \leq W_t \end{cases} \tag{13.1}$$

Our goal is to obtain the value $J_1(W)$ and the corresponding optimal collection of items. The steps that lead to $J_1(W)$ in the above recursion are tied to the optimal decisions x_t^* for $t = 1, \ldots, n-1$. On the one hand, $x_t^* = 0$ corresponds to $J_t(W_t) = J_{t+1}(W_t)$; that is, do not select item t. On the other hand, $x_t^* = 1$ corresponds to $J_t(W_t) = J_{t+1}(W_t - w_t) + v_t$. Observe that for $0 \leq W_n \leq W$ the last-stage value function satisfies

$$J_n(W_n) = \begin{cases} 0 & \text{if} \quad w_n > W_n \\ v_n & \text{if} \quad w_n \leq W_n \end{cases}$$

Example 13.3 (Optimal consumption problem)

Assume that now (beginning of year 0) you have an initial amount of wealth $W_0 > 0$. At the beginning of year t you choose to consume C_t dollars and invest the rest of your wealth in one-year treasury bills. You can consume at most the wealth available in year t. Consuming C_t in year t provides a utility $U(C_t)$. On the other hand, each dollar invested in one-year treasury bill yields $1 + r$ dollars cash at the beginning of the next year. Suppose you want to maximize your total utility of consumption over the next T years:

$$\max_{C_0,\ldots,C_T} \sum_{t=0}^{T} U(C_t).$$

How much should you consume each year?

Solution. The key to solving the optimal consumption problem is again to work "backwards" in time just like we did in the matches puzzle and knapsack problem. Let W_t denote the amount of wealth available at the beginning of year t and let $J_t(W_t)$ be the total utility of consumption from year t to year T if we start at year t with wealth W_t. The value function $J_t(W_t)$ satisfies the following backwards recursion for $t = 0, 1, 2, \ldots, T-1$:

$$J_t(W_t) = \max_{0 \le C_t \le W_t} \{J_{t+1}((W_t - C_t) \cdot (1+r)) + U(C_t)\} \tag{13.2}$$

and the maximizer C_t^* is the optimal consumption level at year t. Observe that for $W_T \ge 0$ the last-stage value function satisfies

$$J_T(W_T) = U(W_T)$$

attained at the optimal consumption level $C_T^* = W_T$.

Model of a Sequential System (Deterministic Case)

We next introduce the formal notation and terminology of dynamic programming. The presentation follows the approach popularized in the classical book of Bertsekas (2005). For ease of exposition, we first consider the deterministic case. That is, the context without random components.

A sequential system is defined by the following elements.

Stages: These are the points in time when decisions are made. We will normally consider $t = 0, 1, \ldots, T$ or $t = 1, 2, \ldots, T$.

States: The state of the system at a particular stage is the information that is relevant for subsequent decisions. We will generally denote the state at stage t as $\mathbf{s}_t$, for $t = 0, 1, \ldots, T$. Sometimes it is convenient to include also a "final state" $\mathbf{s}_{T+1}$.

Decisions: These are also called controls or actions that we can make at each stage and that affect the behavior of the system. We will generally denote the decisions as $\mathbf{x}_t$, for $t = 0, 1, \ldots, T$.

Law of motion: This defines how the state of the system evolves. A general law of motion has the form

$$\mathbf{s}_{t+1} = f_t\left(\mathbf{s}_t, \mathbf{x}_t\right), \quad t = 0, 1, \ldots, T$$

Assume we are interested in optimizing some overall objective function

$$\sum_{t=0}^{T} g_t\left(\mathbf{s}_t, \mathbf{x}_t\right) + g_{T+1}\left(\mathbf{s}_{T+1}\right), \tag{13.3}$$

where each $g_t\left(\mathbf{s}_t, \mathbf{x}_t\right)$, for $t = 0, 1, \ldots, T$, and $g_{T+1}\left(\mathbf{s}_{T+1}\right)$ is some cost or reward per stage. This defines a sequential decision problem: find $\mathbf{x}_t$, for $t = 0, 1, \ldots, T$, to minimize the total cost or maximize the reward (13.3).
Both Examples 13.2 and 13.3 can be readily stated in this framework.

Dynamic programming formulation for the knapsack problem

Stages: $t = 1, 2, \ldots, n$.

State at stage t : remaining weight capacity W_t.

Decision at stage t : binary variable $x_t \in \{0, 1\}$ indicating whether to include item t or not. This decision is constrained to be $x_t = 0$ if $w_t > W_t$ as in this case the weight of item t exceeds the remaining weight capacity.

Law of motion: the remaining weight capacity at stage $t + 1$ is the one from stage t reduced by w_t if item t is included. Otherwise they are the same. More precisely,

$$W_{t+1} = W_t - w_t x_t, t = 1, 2, \ldots, n - 1$$

Objective: maximize the total value of the selected items

$$\max_{t=1,\ldots,n} \sum_{t=1}^{n} v_t x_t$$

Dynamic programming formulation for the optimal consumption problem

Stages: $t = 0, 1, 2, \ldots, T$.

State at stage t : available wealth W_t. It is also convenient to assume that terminal wealth $W_{T+1} = 0$.

Decision at stage t : consumption $C_t \in [0, W_t]$. Law of motion: the wealth at stage $t+1$ is the portion of wealth from stage t that was not consumed increased by a factor $1 + r$. More precisely,

$$W_{t+1} = (W_t - C_t)(1 + r), t = 0, 1, 2, \ldots, T.$$

Objective: maximize the total utility of consumption

$$\max_{C_0, \ldots, C_T} \sum_{t=0}^{T} U(C_t).$$

Bellman's Principle of Optimality

The heart of dynamic programming is a principle of optimality due to Bellman. Its flavor was suggested by the solutions to Examples 13.1, 13.2, and 13.3. To state the principle precisely, we need a bit of notation. Suppose we are maximizing total reward

$$J(\mathbf{s}_0) := \max_{\mathbf{x}_0,\ldots,\mathbf{x}_T} \left\{ \sum_{t=0}^{T} g_t(\mathbf{s}_t, \mathbf{x}_t) + g_{T+1}(\mathbf{s}_{T+1}) \right\}.$$

Consider the "tail problem" that starts at stage t :

$$J_t(\mathbf{s}_t) := \max_{\mathbf{x}_t,\ldots,\mathbf{x}_T} \left\{ \sum_{\tau=t}^{T} g_\tau(\mathbf{s}_\tau, \mathbf{x}_\tau) + g_{T+1}(\mathbf{s}_{T+1}) \right\}.$$

Bellman's optimality principle can be stated as follows. The value-to-go functions $J_t(s_t)$ satisfy the recursive relationship

$$J_t(\mathbf{s}_t) = \max_{\mathbf{x}_t} \{ g_t(\mathbf{s}_t, \mathbf{x}_t) + J_{t+1}(f_t(\mathbf{s}_t, \mathbf{x}_t)) \}. \tag{13.4}$$

The recursive relationship (13.4) is called the Bellman equation. Observe that the recursive relationships (13.1) and (13.2) are exactly the Bellman equation (13.4) in the particular context of Examples 13.2 and 13.3 respectively.
There is a certain jargon associated with the solution to a sequential decision problem and Bellman's optimality principle. The function $J_t(\mathbf{s}_t)$ is called the value-to-go function at stage t. If the objective is to minimize a total cost, Sometimes it is called the *cost-to-go function*. The solution $\mathbf{x}_t^*(\mathbf{s}_t)$ of Bellman's equation (13.4) at stage t is called an *optimal decision rule at stage* t. Notice that this solution depends on the state $\mathbf{s}_t$ at stage t. The vector of optimal decision rules $\left(\mathbf{x}_0^*(\cdot), \ldots, \mathbf{x}_T^*(\cdot)\right)$ is called the *optimal policy*.
Bellman's optimality principle can be phrased as:

> If $\left(\mathbf{x}_0^*(\cdot), \ldots, \mathbf{x}_T^*(\cdot)\right)$ is an optimal policy for the entire problem, then $\left(\mathbf{x}_t^*(\cdot), \ldots, \mathbf{x}_T^*(\cdot)\right)$ is an optimal policy for the tail problem beginning at stage t.

Linear-Quadratic Regulator

We next illustrate Bellman's optimality principle with a popular model from control engineering called the linear-quadratic regulator. It provides the foundation for a model of dynamic investment with transaction costs and predictable returns that we will discuss in the next chapter. The linear-quadratic regulator is a model for the problem of steering the location $\mathbf{s}_t$ of an object towards the origin via a control input $\mathbf{u}_t$. Instead of a constraint on the location of the object, the linear-quadratic regulator imposes a penalty for deviating from the origin. Assume the states and controls evolve according to the following linear law of motion:

$$\mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t, t = 0, 1, \ldots, N-1$$

Assume we have a quadratic cost function

$$\sum_{t=0}^{N-1} \left(\mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t\right) + \mathbf{s}_N^\top \mathbf{Q}\mathbf{s}_N$$

where $\mathbf{Q}, \mathbf{R}$ are symmetric positive definite matrices of appropriate sizes.

The goal is to determine the optimal sequence of controls $\mathbf{u}_t, t = 0, 1, \ldots, N-1$, that minimize the above cost when the initial position of the object is s_0 :

$$J(\mathbf{s}_0) := \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} \left\{ \sum_{t=0}^{N-1} \left(\mathbf{s}_t^\top \mathbf{Q} \mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \right) + \mathbf{s}_N^\top \mathbf{Q} \mathbf{s}_N \right\}$$

We next apply the backwards dynamic programming principle. For the last stage N we evidently have

$$J_N(\mathbf{s}_N) = \mathbf{s}_N^\top \mathbf{Q} \mathbf{s}_N$$

For stage $N-1$ we have the Bellman equation

$$\begin{aligned}
J_{N-1}(\mathbf{s}_{N-1}) &= \min_{\mathbf{u}_{N-1}} \left\{ \mathbf{s}_{N-1}^\top \mathbf{Q} \mathbf{s}_{N-1} + \mathbf{u}_{N-1}^\top \mathbf{R} \mathbf{u}_{N-1} + J_N(\mathbf{s}_N) \right\} \\
&= \min_{\mathbf{u}_{N-1}} \Big\{ \mathbf{s}_{N-1}^\top \mathbf{Q} \mathbf{s}_{N-1} + \mathbf{u}_{N-1}^\top \mathbf{R} \mathbf{u}_{N-1} \\
&\qquad + (\mathbf{A}\mathbf{s}_{N-1} + \mathbf{B}\mathbf{u}_{N-1})^\top \mathbf{Q} (\mathbf{A}\mathbf{s}_{N-1} + \mathbf{B}\mathbf{u}_{N-1}) \Big\} \\
&= \min_{\mathbf{u}_{N-1}} \Big\{ \mathbf{s}_{N-1}^\top \mathbf{Q} \mathbf{s}_{N-1} + \mathbf{s}_{N-1}^\top \mathbf{A}^\top \mathbf{Q} \mathbf{A} \mathbf{s}_{N-1} + 2\mathbf{s}_{N-1}^\top \mathbf{A}^\top \mathbf{Q} \mathbf{B} \mathbf{u}_{N-1} \\
&\qquad + \mathbf{u}_{N-1}^\top \left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q} \mathbf{B} \right) \mathbf{u}_{N-1} \Big\} .
\end{aligned}$$

The latter is a convex quadratic function of $\mathbf{u}_{N-1}$. To find its minimum, we compute its gradient and equate it to zero to obtain:

$$2\mathbf{B}^\top \mathbf{Q}\mathbf{A}\mathbf{s}_{N-1} + 2\left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q}\mathbf{B}\right)\mathbf{u}_{N-1} = \mathbf{0}.$$

Thus, the optimal control at stage $N-1$ is

$$\mathbf{u}^*_{N-1} = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{Q}\mathbf{A}\mathbf{s}_{N-1} = \mathbf{L}_{N-1}\mathbf{s}_{N-1},$$

where

$$\mathbf{L}_{N-1} = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{Q}\mathbf{A}.$$

Plugging this value of $\mathbf{u}^*_{N-1}$ in the above expression for $J_{N-1}(\mathbf{s}_{N-1})$ we get

$$\begin{aligned} J_{N-1}(\mathbf{s}_{N-1}) =& \mathbf{s}_{N-1}^\top \mathbf{Q}\mathbf{s}_{N-1} + \mathbf{s}_{N-1}^\top \mathbf{A}^\top \mathbf{Q}\mathbf{A}\mathbf{s}_{N-1} \\ & - \mathbf{s}_{N-1}^\top \mathbf{A}^\top \mathbf{Q}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{Q}\mathbf{A}\mathbf{s}_{N-1} \\ =& \mathbf{s}_{N-1}^\top \mathbf{K}_{N-1}\mathbf{s}_{N-1} \end{aligned}$$

where

$$\mathbf{K}_{N-1} = \mathbf{Q} + \mathbf{A}^\top \left(\mathbf{Q} - \mathbf{Q}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{Q}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{Q}\right)\mathbf{A}$$

Next we will prove by induction that

$$J_t(\mathbf{s}_t) = \mathbf{s}_t^\top \mathbf{K}_t \mathbf{s}_t, \mathbf{u}_t^* = \mathbf{L}_t \mathbf{s}_t,$$

where

$$\mathbf{K}_N = \mathbf{Q}$$

$$\mathbf{K}_t = \mathbf{Q} + \mathbf{A}^\top \left(\mathbf{K}_{t+1} - \mathbf{K}_{t+1}\mathbf{B} \left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1} \mathbf{B} \right)^{-1} \mathbf{B}^\top \mathbf{K}_{t+1} \right) \mathbf{A}, t = N-1, \ldots, 0$$

and

$$\mathbf{L}_t = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1} \mathbf{B} \right)^{-1} \mathbf{B}^\top \mathbf{K}_{t+1} \mathbf{A}, t = N-1, \ldots, 0.$$

We already showed that the above holds for $t = N - 1$. Assume that it holds for $t + 1$. At stage t we have the Bellman equation

$$
\begin{aligned}
& J_t(\mathbf{s}_t) \\
&= \min_{\mathbf{u}_t} \left\{ \mathbf{s}_t^\top \mathbf{Q} \mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + J_{t+1}(\mathbf{s}_{t+1}) \right\} \\
&= \min_{\mathbf{u}_t} \left\{ \mathbf{s}_t^\top \mathbf{Q} \mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + (\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t)^\top \mathbf{K}_{t+1} (\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t) \right\} \\
&= \min_{\mathbf{u}_t} \left\{ \mathbf{s}_t^\top \mathbf{Q} \mathbf{s}_t + \mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}_{t+1} \mathbf{A} \mathbf{s}_t + 2\mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}_{t+1} \mathbf{B} \mathbf{u}_t + \mathbf{u}_t^\top \left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1} \mathbf{B} \right) \mathbf{u}_t \right\}.
\end{aligned}
$$

The latter is a convex quadratic function of $\mathbf{u}_t$. To find its minimum, we compute its gradient and equate it to zero to obtain:

$$2\mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{A}\mathbf{s}_t + 2\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{B}\right)\mathbf{u}_t = \mathbf{0}.$$

Thus, the optimal control at stage t is

$$\mathbf{u}_t^* = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{A}\mathbf{s}_t = \mathbf{L}_t\mathbf{s}_t,$$

where

$$\mathbf{L}_t = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{A}.$$

Plugging this value of $\mathbf{u}_t^*$ in the above expression for $J_t(\mathbf{s}_t)$ we get

$$\begin{aligned} J_t(\mathbf{s}_t) &= \mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}_{t+1}\mathbf{A}\mathbf{s}_t - \mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}_{t+1}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{A}\mathbf{s}_t \\ &= \mathbf{s}_t^\top \mathbf{K}_t\mathbf{s}_t, \end{aligned}$$

where

$$\mathbf{K}_t = \mathbf{Q} + \mathbf{A}^\top\left(\mathbf{K}_{t+1} - \mathbf{K}_{t+1}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}^\top \mathbf{K}_{t+1}\right)\mathbf{A}.$$

Sequential Decision Problem with Infinite Horizon

Infinite horizon problems are often appropriate models for problems where there is no terminal stage, such as investments for an endowment or a foundation. They are also often appropriate to model problems with very long time horizons. The infinite horizon setting tends to simplify some issues since the dependence of the value function on t can be eliminated.
Consider an infinite horizon problem whose law of motion is of the form

$$\mathbf{s}_{t+1} = f\left(\mathbf{x}_t, \mathbf{s}_t\right)$$

and whose objective function is

$$\max_{\mathbf{x}_0, \mathbf{x}_1, \ldots} \sum_{t=0}^{\infty} \theta^t \cdot g\left(\mathbf{x}_t, \mathbf{s}_t\right),$$

where $\theta \in (0, 1)$ is a given discount factor.

Define the value-to-go function $V(\cdot)$ as

$$V(\mathbf{s}_0) := \max_{\mathbf{x}_0, \mathbf{x}_1, \ldots} \sum_{t=0}^{\infty} \theta^t \cdot g(\mathbf{x}_t, \mathbf{s}_t).$$

Observe that at any intermediate stage t we have

$$V(\mathbf{s}_t) := \max_{\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots} \sum_{\tau=t}^{\infty} \theta^{\tau-t} \cdot g(\mathbf{x}_\tau, \mathbf{s}_\tau)$$

Thus, in this case the Bellman equation can be written as

$$V(\mathbf{s}_t) = \max_{\mathbf{x}_t} g(\mathbf{x}_t, \mathbf{s}_t) + \theta \cdot V(\mathbf{s}_{t+1}).$$

Linear-Quadratic Regulator with Infinite Horizon

Consider now the infinite horizon version of the linear-quadratic regulator that we discussed in Section 13.4. The goal now is to determine the optimal sequence of controls $\mathbf{u}_t, t = 0, 1, \ldots$, that minimizes the following cost:

$$V(\mathbf{s}_0) := \min_{\mathbf{u}_0, \mathbf{u}_1, \ldots} \left\{ \sum_{t=0}^{\infty} \left(\mathbf{s}_t^\top \mathbf{Q} \mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \right) \right\}$$

A common technique to solve the Bellman equation (and similar differential equations) is "ansatz", which can be loosely described as "make an educated guess and later verify". In this problem, we try the following quadratic ansatz for the form of the value function:

$$V(\mathbf{s}_t) = \mathbf{s}_t^\top \mathbf{K} \mathbf{s}_t$$

for some symmetric positive definite matrix $\mathbf{K}$.

With this educated guess we now apply the Bellman equation (infinite horizon case):

$$\begin{aligned} V(\mathbf{s}_t) &= \min_{\mathbf{u}_t} \left[\mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t + V(\mathbf{s}_{t+1})\right] \\ &= \min_{\mathbf{u}_t} \left[\mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t + (\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t)^\top \mathbf{K}(\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t)\right] \\ &= \min_{\mathbf{u}_t} \left[\mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}\mathbf{A}\mathbf{s}_t + 2\mathbf{s}_t^\top \mathbf{A}^\top \mathbf{K}\mathbf{B}\mathbf{u}_t + \mathbf{u}_t^\top \left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)\mathbf{u}_t\right] \end{aligned}$$

The latter is a convex quadratic function of $\mathbf{u}_t$. To find its minimum, we compute its gradient and equate it to zero to obtain:

$$2\mathbf{B}^\top \mathbf{K}\mathbf{A}\mathbf{s}_t + 2\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)\mathbf{u}_t = \mathbf{0}.$$

Thus, the optimal control at stage t is

$$\mathbf{u}_t^* = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{K}\mathbf{A}\mathbf{s}_t = \mathbf{L}\mathbf{s}_t$$

where

$$\mathbf{L} = -\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{K}\mathbf{A}.$$

Plugging this value of $\mathbf{u}_t^*$ in the above Bellman equation we get

$$V(\mathbf{s}_t) = \mathbf{s}_t^\top \mathbf{Q}\mathbf{s}_t + \mathbf{s}_t^\top \mathbf{A}^\top \left(\mathbf{K} - \mathbf{K}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{K}\right) \mathbf{A}\mathbf{s}_t.$$

Hence for the above guess to be correct, we must have:

$$\mathbf{K} = \mathbf{Q} + \mathbf{A}^\top \left(\mathbf{K} - \mathbf{K}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^\top \mathbf{K}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{K}\right) \mathbf{A}.$$

This is the so-called Ricatti equation. Under suitable assumptions on $\mathbf{Q}, \mathbf{R}, \mathbf{A}, \mathbf{B}$, this equation is known to have a unique symmetric positive definite solution $\mathbf{K}$. Consider the following special case: $\mathbf{A} = \mathbf{B} = \mathbf{I}$ and $\mathbf{R} = \lambda \mathbf{Q}$ with $\lambda > 0$. In this case the law of motion is

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{u}_t$$

and the Ricatti equation is

$$\mathbf{K} = \mathbf{Q} + \mathbf{K} - \mathbf{K}(\lambda \mathbf{Q} + \mathbf{K})^{-1}\mathbf{K}.$$

We thus obtain

$$\mathbf{Q} = \mathbf{K}(\lambda \mathbf{Q} + \mathbf{K})^{-1}\mathbf{K}$$

To solve for $\mathbf{K}$, try to find a solution of the form $\mathbf{K} = a\mathbf{Q}$. Plugging this in the above equation yields

$$1 = \frac{a^2}{\lambda + a}.$$

This is a quadratic equation in a with two roots, but only one that is positive, namely

$$a = \frac{1 + \sqrt{1 + 4\lambda}}{2}.$$

Therefore we get

$$\mathbf{K} = a\mathbf{Q} = \frac{1 + \sqrt{1 + 4\lambda}}{2}\mathbf{Q}$$

and consequently

$$\mathbf{L} = -\frac{1 + \sqrt{1 + 4\lambda}}{2\lambda + 1 + \sqrt{1 + 4\lambda}}\mathbf{I}.$$

In particular, the optimal control at time t is

$$\mathbf{u}_t = -\frac{1 + \sqrt{1 + 4\lambda}}{2\lambda + 1 + \sqrt{1 + 4\lambda}}\mathbf{s}_t.$$

Notes

Note that when $\lambda = 0$, there is no direct cost associated with the control variable $\mathbf{u}_t$ and therefore it is optimal to select $\mathbf{u}_t$ to minimize the cost of $\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{u}_t$, which is given by $\mathbf{s}_{t+1}^\top \mathbf{Q}\mathbf{s}_{t+1}$. Clearly, this is minimized when $\mathbf{s}_{t+1} = 0$, or equivalently, when $\mathbf{u}_t = -\mathbf{s}_t$. On the other hand, for $\lambda > 0$, the cost $\lambda \mathbf{u}_t^\top \mathbf{Q}\mathbf{u}_t$ keeps $\mathbf{u}_t$ from reaching all the way to $-\mathbf{s}_t$. Instead, $\mathbf{u}_t$ is a scalar multiple of $-\mathbf{s}_t$, where the scalar multiple is less than 1 . In addition, the larger λ, the higher the cost of the control variable $\mathbf{u}_t$, and therefore the smaller this scalar multiple.

Model of Sequential System (Stochastic Case)

The above dynamic programming machinery has a straightforward extension to a more general context that includes a stochastic component in the law of motion. A stochastic sequential system is an extension of the deterministic case. Like a deterministic sequential system, the main components of a stochastic sequential system are stages, states, decisions, and law of motion. The first three are exactly as before. On the other hand, the law of motion of a stochastic sequential system is of the more general form

$$\mathbf{s}_{t+1} = f_t\left(\mathbf{s}_t, \mathbf{x}_t, \omega_t\right), \quad t = 0, 1, \ldots, T$$

As before, $\mathbf{s}_t, \mathbf{x}_t$ are the state and action at stage t and $\mathbf{s}_{t+1}$ is the state at stage $t+1$. In addition, ω_t is some random disturbance that occurs at stage t.

Assume we are interested in optimizing some overall objective function

$$\mathbb{E}\left[\sum_{t=0}^{T} g_t\left(\mathbf{s}_t, \mathbf{x}_t, \omega_t\right) + g_{T+1}\left(\mathbf{s}_{T+1}\right)\right], \tag{13.5}$$

where each $g_t\left(\mathbf{s}_t, \mathbf{x}_t, \omega_t\right), t = 0, 1, \ldots, T$, and $g_{T+1}\left(\mathbf{s}_{T+1}\right)$ is a cost or a reward per stage. This defines a stochastic sequential decision problem: find $\mathbf{x}_t, t = 0, 1, \ldots, T$, to minimize or maximize the expected total cost or reward (13.5). Bellman's optimality principle also extends in a natural fashion. Suppose we are maximizing the expected reward

$$J\left(\mathbf{s}_0\right) := \max_{\mathbf{x}_0, \ldots, \mathbf{x}_T} \mathbb{E}\left[\sum_{t=0}^{T} g_t\left(\mathbf{s}_t, \mathbf{x}_t, \omega_t\right) + g_{T+1}\left(\mathbf{s}_{T+1}\right)\right].$$

Consider the "tail problem" that starts at stage t :

$$J_t\left(\mathbf{s}_t\right) := \max_{\mathbf{x}_t, \ldots, \mathbf{x}_T} \mathbb{E}\left[\sum_{\tau=t}^{T} g_\tau\left(\mathbf{s}_\tau, \mathbf{x}_\tau, \omega_\tau\right) + g_{T+1}\left(\mathbf{s}_{T+1}\right)\right].$$

Bellman's optimality principle

Bellman's optimality principle can be stated as follows. The value-to-go functions $J_t(s_t)$ satisfy the following Bellman equation:

$$J_t(s_t) = \max_{x_t} \mathbb{E}_t \left[g_t(s_t, x_t, \omega_t) + J_{t+1}(f_t(s_t, x_t, \omega_t)) \right] \tag{13.6}$$

The stochastic case also has an infinite horizon version. Consider an infinite horizon problem with a law of motion of the form

$$\mathbf{s}_{t+1} = f(\mathbf{x}_t, \mathbf{s}_t, \omega_t)$$

and objective function

$$\max_{\mathbf{x}_0, \mathbf{x}_1, \ldots} \mathbb{E} \left[\sum_{t=0}^{\infty} \theta^t \cdot g(\mathbf{x}_t, \mathbf{s}_t, \omega_t) \right],$$

where $\theta \in (0, 1)$ is a given discount factor.

Define the value-to-go function $V(\cdot)$ as

$$V(\mathbf{s}_0) := \max_{\mathbf{x}_0, \mathbf{x}_1, \ldots} \mathbb{E}\left[\sum_{t=0}^{\infty} \theta^t \cdot g(\mathbf{x}_t, \mathbf{s}_t, \omega_t)\right].$$

Observe that at any intermediate stage t we have

$$V(\mathbf{s}_t) := \max_{\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots} \mathbb{E}\left[\sum_{\tau=t}^{\infty} \theta^{\tau-t} \cdot g(\mathbf{x}_\tau, \mathbf{s}_\tau, \omega_\tau)\right].$$

In this case the Bellman equation can be written as

$$V(\mathbf{s}_t) = \max_{\mathbf{x}_t} \mathbb{E}_t\left[g(\mathbf{x}_t, \mathbf{s}_t, \omega_t) + \theta \cdot V(\mathbf{s}_{t+1})\right]$$