Chapter 3
Dynamic Programming

This chapter introduces basic ideas and methods of dynamic programming.[1] It sets out the basic elements of a recursive optimization problem, describes a key functional equation called the Bellman equation, presents three methods for solving the Bellman equation, and gives the Benveniste-Scheinkman formula for the derivative of the optimal value function. Let's dive in.

3.1. Sequential problems

Let $\beta \in (0,1)$ be a discount factor. We want to choose an infinite sequence of "controls" $\{u_t\}_{t=0}^{\infty}$ to maximize

$$\sum_{t=0}^{\infty} \beta^t r\left(x_t, u_t\right), \tag{3.1.1}$$

subject to $x_{t+1} = g(x_t, u_t)$, with $x_0 \in I\!R^n$ given. We assume that $r(x_t, u_t)$ is a concave function and that the set $\{(x_{t+1}, x_t) : x_{t+1} \leq g(x_t, u_t), u_t \in I\!R^k\}$ is convex and compact. Dynamic programming seeks a time-invariant *policy function* h mapping the *state* x_t into the control u_t, such that the sequence $\{u_s\}_{s=0}^{\infty}$ generated by iterating the two functions

$$\begin{aligned} u_t &= h\left(x_t\right) \\ x_{t+1} &= g\left(x_t, u_t\right), \end{aligned} \tag{3.1.2}$$

starting from initial condition x_0 at $t = 0$, solves the original problem. A solution in the form of equations (3.1.2) is said to be *recursive*. To find the policy function h we need to know another function $V(x)$ that expresses the optimal value of the original problem, starting from an arbitrary initial condition $x \in X$. This is called the *value function*. In particular, define

$$V\left(x_0\right) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r\left(x_t, u_t\right), \tag{3.1.3}$$

[1] This chapter aims to the reader to start using the methods quickly. We hope to promote demand for further and more rigorous study of the subject. In particular see Bertsekas (1976), Bertsekas and Shreve (1978), Stokey and Lucas (with Prescott) (1989), Bellman (1957), and Chow (1981). This chapter covers much of the same material as Sargent (1987b, chapter 1).

where again the maximization is subject to $x_{t+1} = g(x_t, u_t)$, with x_0 given. Of course, we cannot possibly expect to know $V(x_0)$ until after we have solved the problem, but let's proceed on faith. If we knew $V(x_0)$, then the policy function h could be computed by solving for each $x \in X$ the problem

$$\max_u \{r(x,u) + \beta V(\tilde{x})\}, \tag{3.1.4}$$

where the maximization is subject to $\tilde{x} = g(x,u)$ with x given, and $\tilde{x}$ denotes the state next period. Thus, we have exchanged the original problem of finding an infinite *sequence* of controls that maximizes expression (3.1.1) for the problem of finding the optimal value function $V(x)$ and a function h that solves the continuum of maximum problems (3.1.4)—one maximum problem for each value of x. This exchange doesn't look like progress, but we shall see that it often is.

Our task has become jointly to solve for $V(x), h(x)$, which are linked by the *Bellman equation*

$$V(x) = \max_u \{r(x,u) + \beta V[g(x,u)]\}. \tag{3.1.5}$$

The maximizer of the right side of equation (3.1.5) is a *policy function* $h(x)$ that satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[x, h(x)]\}. \tag{3.1.6}$$

Equation (3.1.5) or (3.1.6) is a *functional equation* to be solved for the pair of unknown functions $V(x), h(x)$.

Methods for solving the Bellman equation are based on mathematical structures that vary in their details depending on the precise nature of the functions r and g.[2] All of these structures contain versions of the following four findings. Under various particular assumptions about r and g, it turns out that

[2] There are alternative sets of conditions that make the maximization (3.1.4) well behaved. One set of conditions is as follows: (1) r is concave and bounded, and (2) the constraint set generated by g is convex and compact, that is, the set of $\{(x_{t+1}, x_t) : x_{t+1} \leq g(x_t, u_t)\}$ for admissible u_t is convex and compact. See Stokey, Lucas, and Prescott (1989) and Bertsekas (1976) for further details of convergence results. See Benveniste and Scheinkman (1979) and Stokey, Lucas, and Prescott (1989) for the results on differentiability of the value function. In Appendix A (see Technical Appendixes), we describe the mathematics for one standard set of assumptions about (r, g). In chapter 5, we describe it for another set of assumptions about (r, g).

1. The functional equation (3.1.5) has a unique strictly concave solution.
2. This solution is approached in the limit as $j \to \infty$ by iterations on

$$V_{j+1}(x) = \max_u \{r(x,u) + \beta V_j(\tilde{x})\},$$

   subject to $\tilde{x} = g(x,u), x$ given, starting from any bounded and continuous initial V_0.
3. There is a unique and time-invariant optimal policy of the form $u_t = h(x_t)$, where h is chosen to maximize the right side of (3.1.5).
4. Off corners, the limiting value function V is differentiable.

Since the value function is differentiable, the first-order necessary condition for problem (3.1.4) becomes[3]

$$r_2(x,u) + \beta V'\{g(x,u)\}\, g_2(x,u) = 0. \tag{3.1.7}$$

If we also assume that the policy function $h(x)$ is differentiable, differentiation of expression (3.1.6) yields[4]

$$\begin{aligned} V'(x) = {} & r_1[x, h(x)] + r_2[x, h(x)]\ h'(x) \\ & + \beta V'\{g[x, h(x)]\} \left\{ g_1[x, h(x)] + g_2[x, h(x)]\ h'(x) \right\}. \end{aligned} \tag{3.1.8}$$

When the states and controls can be defined in such a way that only u appears in the transition equation, i.e., $\tilde{x} = g(u)$: the derivative of the value function becomes, after substituting expression (3.1.7) with $u = h(x)$ into (3.1.8),

$$V'(x) = r_1[x, h(x)]. \tag{3.1.9}$$

This is a version of a formula of Benveniste and Scheinkman (1979).

At this point, we describe three broad computational strategies that apply in various contexts.

---

[3] Here and below, subscript 1 denotes the vector of derivatives with respect to the x components and subscript 2 denotes the derivatives with respect to the u components.

[4] Benveniste and Scheinkman (1979) proved differentiability of $V(x)$ under broad conditions that do not require that $h(x)$ be differentiable. For conditions under which $h(x)$ is differentiable, see Santos (1991,1993).

3.1.1. Three computational methods

There are three main types of computational methods for solving dynamic programs. All aim to solve the functional equation (3.1.4).

Value function iteration. The first method proceeds by constructing a sequence of value functions and associated policy functions. The sequence is created by iterating on the following equation, starting from $V_0 = 0$, and continuing until V_j has converged:

$$V_{j+1}(x) = \max_u \{r(x,u) + \beta V_j(\tilde{x})\}, \tag{3.1.10}$$

subject to $\tilde{x} = g(x,u), x$ given.[5] This method is called *value function iteration* or *iterating on the Bellman equation*.

Guess and verify. A second method involves guessing and verifying a solution V to equation (3.1.5). This method relies on the uniqueness of the solution to the equation, but because it relies on luck in making a good guess, it is not generally available.

Howard's improvement algorithm. A third method, known as *policy function iteration* or *Howard's improvement algorithm*, consists of the following steps:

1. Pick a feasible policy, $u = h_0(x)$, and compute the value associated with operating forever with that policy:

$$V_{h_j}(x) = \sum_{t=0}^{\infty} \beta^t r[x_t, h_j(x_t)],$$

   where $x_{t+1} = g[x_t, h_j(x_t)]$, with $j = 0$.
2. Generate a new policy $u = h_{j+1}(x)$ that solves the two-period problem

$$\max_u \{r(x,u) + \beta V_{h_j}[g(x,u)]\},$$

   for each x.

[5] See Appendix A on functional analysis (see Technical Appendixes) for what it means for a sequence of functions to converge. A proof of the uniform convergence of iterations on equation (3.1.10) is contained in that appendix.

3. Iterate over j to convergence on steps 1 and 2.

In Appendix A (see Technical Appendixes), we describe some conditions under which the policy improvement algorithm converges to the solution of the Bellman equation. The policy improvement algorithm often converges faster than does value function iteration (e.g., see exercise *3.1* at the end of this chapter).[6] The policy improvement algorithm is also a building block for methods used to study government policy in chapter 24.

Each of our three methods for solving dynamic programming problems has its uses. Each is easier said than done, because it is typically impossible analytically to compute even *one* iteration on equation (3.1.10). This fact thrusts us into the domain of computational methods for approximating solutions: pencil and paper are insufficient. Chapter 4 describes computational methods that can applied to problems that cannot be solved by hand. Here we shall describe the first of two special types of problems for which analytical solutions *can* be obtained. It involves Cobb-Douglas constraints and logarithmic preferences. Later, in chapter 5, we shall describe a specification with linear constraints and quadratic preferences. For that special case, many analytic results are available. These two classes have been important in economics as sources of examples and as inspirations for approximations.

3.1.2. Cobb-Douglas transition, logarithmic preferences

Brock and Mirman (1972) used the following optimal growth example.[7] A planner chooses sequences $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ to maximize

$$\sum_{t=0}^{\infty} \beta^t \ln(c_t)$$

subject to a given value for k_0 and a transition law

$$k_{t+1} + c_t = Ak_t^{\alpha}, \tag{3.1.11}$$

where $A > 0, \alpha \in (0,1), \beta \in (0,1)$.

[6] The speed of the policy improvement algorithm comes from its implementing Newton's method, which converges quadratically while iteration on the Bellman equation converges at a linear rate. See chapter 4 and Appendix A (see Technical Appendixes).

[7] See also Levhari and Srinivasan (1969).

This problem can be solved "by hand," using any of our three methods. We begin with iteration on the Bellman equation. Start with $v_0(k) = 0$, and solve the one-period problem: choose c to maximize $\ln(c)$ subject to $c + \tilde{k} = Ak^\alpha$. The solution is evidently to set $c = Ak^\alpha, \tilde{k} = 0$, which produces an optimized value $v_1(k) = \ln A + \alpha \ln k$. At the second step, we find $c = \frac{1}{1+\beta\alpha} Ak^\alpha, \tilde{k} = \frac{\beta\alpha}{1+\beta\alpha} Ak^\alpha, v_2(k) = \ln \frac{A}{1+\alpha\beta} + \beta \ln A + \alpha\beta \ln \frac{\alpha\beta A}{1+\alpha\beta} + \alpha(1+\alpha\beta) \ln k$. Continuing, and using the algebra of geometric series, gives the limiting policy functions $c = (1-\beta\alpha)Ak^\alpha, \tilde{k} = \beta\alpha Ak^\alpha$, and the value function $v(k) = (1-\beta)^{-1}\{\ln[A(1-\beta\alpha)] + \frac{\beta\alpha}{1-\beta\alpha} \ln(A\beta\alpha)\} + \frac{\alpha}{1-\beta\alpha} \ln k$.

Here is how the guess-and-verify method applies to this problem. Since we already know the answer, we'll guess a function of the correct form, but leave its coefficients undetermined.[8] Thus, we make the guess

$$v(k) = E + F \ln k, \tag{3.1.12}$$

where E and F are undetermined constants. The left and right sides of equation (3.1.12) must agree for all values of k. For this guess, the first-order necessary condition for the maximum problem on the right side of equation (3.1.10) implies the following formula for the optimal policy $\tilde{k} = h(k)$, where $\tilde{k}$ is next period's value and k is this period's value of the capital stock:

$$\tilde{k} = \frac{\beta F}{1 + \beta F} \, Ak^\alpha. \tag{3.1.13}$$

Substitute equation (3.1.13) into the Bellman equation and equate the result to the right side of equation (3.1.12). Solving the resulting equation for E and F gives $F = \alpha/(1-\alpha\beta)$ and $E = (1-\beta)^{-1}[\ln A(1-\alpha\beta) + \frac{\beta\alpha}{1-\alpha\beta} \ln A\beta\alpha]$. It follows that

$$\tilde{k} = \beta\alpha Ak^\alpha. \tag{3.1.14}$$

Note that the term $F = \alpha/(1-\alpha\beta)$ can be interpreted as a geometric sum $\alpha[1 + \alpha\beta + (\alpha\beta)^2 + \ldots]$.

Equation (3.1.14) shows that the optimal policy is to have capital move according to the difference equation $k_{t+1} = A\beta\alpha k_t^\alpha$, or $\ln k_{t+1} = \ln A\beta\alpha + \alpha \ln k_t$. That α is less than 1 implies that k_t converges as t approaches infinity for any positive initial value k_0. The stationary point is given by the solution of $k_\infty = A\beta\alpha k_\infty^\alpha$, or $k_\infty^{\alpha-1} = (A\beta\alpha)^{-1}$.

[8] This is called the *method of undetermined coefficients*.

3.1.3. Euler equations

In many problems, there is no unique way of defining states and controls, and several alternative definitions lead to the same solution of the problem. When the states and controls can be defined in such a way that only u appears in the transition equation, i.e., $\tilde{x} = g(u)$: the first-order condition for the problem on the right side of the Bellman equation (expression (3.1.7)) in conjunction with the Benveniste-Scheinkman formula (expression (3.1.9)) implies

$$r_2(x_t, u_t) + \beta\, r_1(x_{t+1}, u_{t+1})\; g'(u_t) = 0, \qquad x_{t+1} = g\,(u_t)\,.$$

The first equation is called an *Euler equation*. Under circumstances in which the second equation can be inverted to yield u_t as a function of x_{t+1}, using the second equation to eliminate u_t from the first equation produces a second-order difference equation in x_t, since eliminating u_{t+1} brings in x_{t+2}.

3.1.4. A sample Euler equation

As an example of an Euler equation, consider the Ramsey problem of choosing $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ to maximize $\sum_{t=0}^{\infty} \beta^t u(c_t)$ subject to $c_t + k_{t+1} = f(k_t)$, where k_0 is given and the one-period utility function satisfies $u'(c) > 0, u''(c) < 0, \lim_{c_t \searrow 0} u'(c_t) = \infty$, and where $f'(k) > 0, f''(k) < 0$. Let the state be k and the control be $\tilde{k}$, where $\tilde{k}$ denotes next period's value of k. Substitute $c = f(k) - \tilde{k}$ into the utility function and express the Bellman equation as

$$v\,(k) = \max_{\tilde{k}} \left\{ u\left[f\,(k) - \tilde{k}\right] + \beta v\left(\tilde{k}\right) \right\}. \tag{3.1.15}$$

Application of the Benveniste-Scheinkman formula gives

$$v'\,(k) = u'\left[f\,(k) - \tilde{k}\right] f'\,(k)\,. \tag{3.1.16}$$

Notice that the first-order condition for the maximum problem on the right side of equation (3.1.15) is $-u'[f(k) - \tilde{k}] + \beta v'(\tilde{k}) = 0$, which, using equation (3.1.16), gives

$$u'\left[f\,(k) - \tilde{k}\right] = \beta u'\left[f\left(\tilde{k}\right) - \hat{k}\right] f'\left(\tilde{k}\right), \tag{3.1.17}$$

where $\hat{k}$ denotes the two-period-ahead value of k. Equation (3.1.17) can be expressed as

$$1 = \beta \frac{u'(c_{t+1})}{u'(c_t)} f'(k_{t+1}),$$

an Euler equation that is exploited extensively in the theories of finance, growth, and real business cycles.

3.2. Stochastic control problems

We now consider a modification of problem (3.1.1) to permit uncertainty. Essentially, we add some well-placed shocks to the previous nonstochastic problem. So long as the shocks are either independently and identically distributed or Markov, straightforward modifications of the method for handling the nonstochastic problem will work.

Thus, we modify the transition equation and consider the problem of maximizing

$$E_0 \sum_{t=0}^{\infty} \beta^t r(x_t, u_t), \qquad 0 < \beta < 1, \tag{3.2.1}$$

subject to

$$x_{t+1} = g(x_t, u_t, \epsilon_{t+1}), \tag{3.2.2}$$

with x_0 known and given at $t = 0$, where ϵ_t is a sequence of independently and identically distributed random variables with cumulative probability distribution function $\text{prob}\{\epsilon_t \leq e\} = F(e)$ for all t; $E_t(y)$ denotes the mathematical expectation of a random variable y, given information known at t. At time t, x_t is assumed to be known, but $x_{t+j}, j \geq 1$ is not known at t. That is, ϵ_{t+1} is realized at $(t+1)$, after u_t has been chosen at t. In problem (3.2.1)–(3.2.2), uncertainty is injected by assuming that x_t follows a random difference equation.

Problem (3.2.1)–(3.2.2) continues to have a recursive structure, stemming jointly from the additive separability of the objective function (3.2.1) in pairs (x_t, u_t) and from the difference equation characterization of the transition law (3.2.2). In particular, controls dated t affect returns $r(x_s, u_s)$ for $s \geq t$ but not earlier. This feature implies that dynamic programming methods remain appropriate.