Dynamic Programming: An overview

Russell Cooper

February 14, 2001

1 Overview

The mathematical theory of dynamic programming as a means of solving dynamic optimization problems dates to the early contributions of Bellman [1957] and Bertsekas [1976]. For economists, the contributions of Sargent [1987] and Stokey-Lucas [1989] provide a valuable bridge to this literature.

2 Indirect Utility

Intuitively, the approach of dynamic programming can be understood by recalling the theme of indirect utility from basic static consumer theory or a reduced form profit function generated by the optimization of a firm. This reduced form representations of payoffs summarizes information about the optimized value of the choice problems faced by households and firms. As we shall see, the theory of dynamic programming uses this insight in a dynamic context.

2.1 Consumers

Consumer choice theory focuses on households who solve:

$$V(I,p) = \max_c u(c) \text{ subject to: } pc = I$$

where c is a vector of consumption goods, p is a vector of prices and I is income.[1] The first order condition is given by

$$u_j(c)/p_j = \lambda \text{ for } j = 1, 2...J.$$

where λ is the multiplier on the budget constraint and $u_j(c)$ is the marginal utility from good j.

Here $V(I,p)$ is an indirect utility function. It is the maximized level of utility from the current state (I,p). So if someone is in this state, you can predict that they will attain this level of utility. You do not need to know what they will do with their income; it is enough to know that they will act optimally. This is very powerful logic and underlies the idea behind

[1] Assume that there are J commodities in this economy. This presentation assumes that you understand the conditions under which this optimization problem has a solution and when that solution can be characterized by first-order conditions.

the dynamic programming models studied below.

To illustrate, what happens if we give the consumer a bit more income? Welfare goes up by $V_I(I,p) > 0$. Can I predict what will happen with a little more income? Not really since the optimizing consumer is indifferent with respect to how this is spent:

$$u_j(c)/p_j = V_I(I,p) \text{ for all } j.$$

It is in this sense that the indirect utility function summarizes the value of the households optimization problem and allows us to determine the marginal value of income without knowing further details about consumption functions.

Is this all we need to know about household behavior? No, this theory is static and thus ignores savings, spending on durable goods as well as uncertainty over the future. We will return to these in later chapters on the dynamic behavior of households. The point here was simply to recall a key object from optimization theory: the indirect utility function.

2.2 Firms

Suppose that a firm chooses how many workers to hire at a wage of w given its stock of capital, k. Thus the firm solves:

$$\Pi(w,k) = \max_l pf(l,k) - wl.$$

This will yield a labor demand function which depends on (w,k). As with $V(I,p)$, $\Pi(w,k)$ summarizes the value of the firm given factor prices and the stock of capital, k. Both the flexible and fixed factors could be vectors. Think of $\Pi(w,k)$ as an indirect profit function.

As with the households problem, given $\Pi(w,k)$,we can directly compute the marginal value of giving the firm some additional capital as $\Pi_k(w,k)$=$pf_k(l,k)$ without knowing how the firm will adjust its labor input in response to the additional capital.

But, is this all there is to know about the firm's behavior? Surely not as we have not specified where k comes from. So the firm's problem is essentially dynamic though the demand for some of its inputs can be taken as a static optimization problem. These are important themes in the theory of factor demand and we will return to them in our firm applications.

3 Dynamic Optimization: A Cake Eating Example

Here we will look at a very simple dynamic optimization problem. We begin with a finite horizon and then discuss extensions to the infinite horizon.[2]

Suppose that you have a cake of size W_1. At each point of time, $t = 1, 2, 3, ...T$ you can consume some of the cake and thus save the remainder. Let c_t be your consumption in period t and let $u(c_t)$ represent the flow of utility from this consumption. The utility function is not indexed by time: preferences are stationary. Assume $u(\cdot)$ is real-valued, differentiable, strictly increasing and strictly concave. Assume that $\lim_{c \longrightarrow 0} u'(c) \rightarrow \infty$. Represent lifetime utility by

[2] For a very complete treatment of the finite horizon problem with uncertainty, see Bertsekas [1976, Ch. 2].

$$\sum_{t=1}^{T} \beta^{(t-1)} u(c_t)$$

where $0 \leq \beta \leq 1$ and β is called the **discount factor**. A related concept is the **discount rate** defined as β^{-1}.

For now, assume that the cake does not depreciate (melt) or grow. Hence, the evolution of the cake over time is governed by:

$$W_{t+1} = W_t - c_t \tag{1}$$

for $t = 1, 2, ..T$. How would you find the optimal path of consumption, $\{c_t\}_1^T$?

3.1 Direct Attack

One approach is to solve the constrained optimization problem directly. This is called the **sequence problem** by Stokey-Lucas [1989]. Consider the problem of:

$$\max_{\{c_t\}_1^T, \{W_t\}_2^{T+1}} \sum_{t=1}^{T} \beta^{(t-1)} u(c_t) \tag{2}$$

subject to the transition equation (1) which holds for $t = 1, 2, 3, ...T$. Also, there are non-negativity constraints on consumption and the cake given by: $c_t \geq 0$ and $W_t \geq 0$. For this problem, W_1 is given.

Alternatively, the flow constraints imposed by (1) for each t could be combined yielding:

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1. \tag{3}$$

The non-negativity constraints are simpler: $c_t \geq 0$ for $t = 1, 2, ..T$ and $W_{T+1} \geq 0$. For now, we will work with the single resource constraint. This is a well-behaved problem as the objective is concave and continuous and the constraint set is compact. So there is a solution to this problem.[3]

Letting λ be the multiplier on (3), the first order conditions are given by:

$$\beta^{t-1} u'(c_t) = \lambda$$

for $t = 1, 2, ..., T$ and

$$\lambda = \phi$$

where ϕ is the non-negativity constraint on W_{T+1}. The non-negativity constraints on $c_t \geq 0$ are ignored as we assume that the marginal utility of consumption becomes infinite as consumption approaches zero within any period.

Combining equations, we obtain an equation that links consumption across any two periods:

[3] This comes from the Weierstrass theorem. See Bertsekas, [1976, Appendix B] or Stokey-Lucas [1989, Ch. 3] for a discussion.

$$u'(c_t) = \beta u'(c_{t+1}). \tag{4}$$

This is a necessary condition for optimality for **any** t: if it was violated, the agent could do better by adjusting c_t and c_{t+1}. Frequently (4) is referred to as an **Euler equation**.

To understand this condition, suppose that you have a proposed (candidate) solution for this problem given by $\{c_t^*\}_1^T, \{W_t^*\}_2^{T+1}$. Essentially, the Euler equation says that the marginal utility cost of reducing consumption by ε in period t equals the marginal utility gain from consuming the extra ε of cake in the next period, which is discounted by β. If the Euler equation holds, then it is impossible to increase utility by moving consumption across adjacent periods given a candidate solution.

It should be clear though that this condition may not be sufficient: it does not cover deviations that last more than one period. For example, could utility be increased by reducing consumption by ε in period t saving the "cake" for two periods and then increasing consumption in period t+2? Clearly this is not covered by a single Euler equation. However, by combining the Euler equation that hold across period t and $t+1$ with that which holds for periods $t+1$ and $t+2$, we can see that such a deviation will not increase utility. This is simply because the combination of Euler equations implies:

$$u'(c_t) = \beta^2 u'(c_{t+2})$$

so that the two-period deviation from the candidate solution will not increase utility.

As long as the problem is finite, the fact that the Euler equation holds across all adjacent periods implies that any finite deviations from a candidate solution that satisfies the Euler equations will not increase utility. Is this enough? Not quite. Imagine a candidate solution that satisfies all of the Euler equations but has the property that $W_T > c_T$ so that there is cake left over. This is clearly an inefficient plan: having the Euler equations holding is necessary but not sufficient. Hence the optimal solution will satisfy the Euler equation for each period and the agent will consume the entire cake!

Formally, this involves showing the non-negativity constraint on W_{T+1} must bind. In fact, this constraint is binding in the above solution: $\lambda = \phi > 0$. This non-negativity constraint serves two important purposes. First, in the absence of a constraint that $W_{T+1} \geq 0$, the agent would clearly want to set $W_{T+1} = -\infty$ and thus die with outstanding obligations. This is clearly not feasible. Second, the fact that the constraint is binding in the optimal solution guarantees that cake is not being thrown away after period T.

So, in effect, the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T-1)$ Euler equations and (3), then determine the time path of consumption.

Let the solution to this problem be denoted by $V_T(W_1)$ where T is the horizon of the problem and W_1 is the initial size of the cake. $V_T(W_1)$ represents the maximal utility flow from a T period problem given a size W_1 cake. From now on, we call this a **value function**. This is completely analogous to the indirect utility functions expressed for the household and the firm.

As in those problems, a slight increase in the size of the cake leads to an increase in lifetime utility equal to the marginal utility in any period. That is,

$$V_T'(W_1) = \lambda = \beta^{t-1} u'(c_t), t = 1, 2, ...T.$$

It doesn't matter when the extra cake is eaten given that the consumer is acting optimally. This is analogous to the point raised above about the effect on utility of an increase in income in the consumer choice problem with multiple goods.

3.2 Dynamic Programming Approach

Suppose that we change the above problem slightly: we add a period 0 and give an initial cake of size W_0. One approach to determining the optimal solution of this augmented problem is to go back to the sequence problem and resolve it using this longer horizon and new constraint. But, having done all of the hard work with the T period problem, it would be nice not to have to do it again!

3.2.1 Finite Horizon Problem

The dynamic programming approach provides a means of doing so. It essentially converts a (arbitrary) T period problem into a 2 period problem with the appropriate rewriting of the objective function. In doing so, it uses the value function obtained from solving a shorter horizon problem.

So, when we consider adding a period 0 to our original problem, we can take advantage of the information provided in $V_T(W_1)$, the solution of the T period problem given W_1 from (2). Given W_0, consider the problem of

$$\max_{c_0} u(c_0) + \beta V_T(W_1) \tag{5}$$

where

$$W_1 = W_0 - c_0; W_0 \text{ given.}$$

In this formulation, the choice of consumption in period 0 determines the size of the cake that will available starting in period 1, W_1. So instead of choosing a sequence of consumption levels, we are just choosing c_0. Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$. This function completely summarizes optimal behavior from period 1 onwards. For the purposes of the dynamic programming problem, it doesn't matter how the cake will be consumed after the initial period. All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$. This is the **principle of optimality**, due to Richard Bellman, at work. With this knowledge, an optimal decision can be made regarding consumption in period 0.

Note that the first order condition (assuming that $V_T(W_1)$ is differentiable) is given by:

$$u'(c_0) = \beta V_T'(W_1)$$

so that the marginal gain from reducing consumption a little in period 0 is summarized by the derivative of the value function. As noted in the earlier discussion of the T period sequence problem,

$$V_T'(W_1) = u'(c_1) = \beta^t u'(c_{t+1})$$

for $t = 1, 2, ..T - 1$. Using these two conditions together yields

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

for $t = 0, 1, 2, ...T - 1$, a familiar necessary condition for an optimal solution.

Since the Euler conditions for the other periods underlie the creation of the value function, one might suspect that the solution to the $T + 1$ problem using this dynamic programming approach is identical to that from using the sequence approach.[4] This is clearly true for this problem: the set of first order conditions for the two problems are identical and thus, given the strict concavity of the $u(c)$ functions, the solutions will be identical as well.

The apparent ease of this approach though is a bit misleading. We were able to make the problem look simple by pretending that we actually knew $V_T(W_1)$. Of course, we had to solve for this either by tackling a sequence problem directly or by building it recursively starting from an initial single period problem.

On this latter approach, we could start with the single period problem implying $V_1(W_1)$. We could then solve (5) to build $V_2(W_1)$ for any T. Given this function, we could move to a solution of the $T + 3$ problem and proceed iteratively.

3.2.2 Example

We illustrate the construction of the value function in a specific example. Assume $u(c) = ln(c)$. Suppose that $T = 1$. Then $V_1(W_1) = \ln(W_1)$.

For $T = 2$, the first order condition from (2) is

$$1/c_1 = \beta/c_2$$

and the resource constraint is

$$W_1 = c_1 + c_2.$$

Working with these two conditions:

$$c_1 = W_1/(1 + \beta) \text{ and } c_2 = \beta W_1/(1 + \beta).$$

From this, we can solve for the value of the 2-period problem:

$$V_2(W_1) = \ln(c_1) + \beta \ln(c_2) = A_2 + B_2 \ln(W_1) \tag{6}$$

where A_2 and B_2 are constants associated with the two period problem. These constants are given by:

$$A_2 = \ln(1/(1 + \beta) + \beta \ln(\beta/(1 + \beta)) \qquad B_2 = (1 + \beta)$$

Importantly, (6) does not include the max operator as we are substituting the maximized values into that value function.

Using this function, the $T = 3$ problem can then be written as:

[4] By the sequence approach, we mean solving the problem using the direct approach outlined in the previous section.

$$V_3(W_1) = \max_{W_2} \ln(W_1 - W_2) + \beta V_2(W_2)$$

where the choice variable is the state in the subsequent period. The first order condition is:

$$\frac{1}{c_1} = \beta V_2'(W_2).$$

Using (6) evaluated at a cake of size W_2, we can solve for $V_2'(W_2)$ implying:

$$\frac{1}{c_1} = \beta \frac{B_2}{W_2} = \frac{\beta}{c_2}.$$ [5]

Further, we know from the 2-period problem that

$$1/c_2 = \beta/c_3.$$

This plus the resource constraint allows us to construct the solution of the 3-period problem:

$$c_1 = W_1/(1+\beta+\beta^2), \quad c_2 = \beta W_1/(1+\beta+\beta^2), \quad c_3 = \beta^2 W_1/(1+\beta+\beta^2).$$

Substituting into $V_3(W_1)$ yields

$$V_3(W_1) = A_3 + B_3 \ln(W_1)$$

where

$$A_3 = \ln(1/(1+\beta+\beta^2) + \beta \ln(\beta/(1+\beta+\beta^2)) + \beta^2 \ln(\beta^2/(1+\beta+\beta^2)), \qquad B_3 = (1+\beta+\beta^2)$$

This solution can be verified from a direct attack on the 3 period problem using (2) and (3).

4 Some Extensions of the Cake Eating Problem

Here we go beyond the T period problem to illustrate some ways to use the dynamic programming framework. This is intended as an overview and the details of the assertions and so forth will be provided below.

4.1 Infinite Horizon

4.1.1 Basic Structure

Suppose that we consider the above problem and allow the horizon to go to infinity. As before, one can consider solving the infinite horizon sequence problem given by:

$$\max_{\{c_t\}_1^\infty, \{W_t\}_2^\infty} \sum_{t=1}^{\infty} \beta^t u(c_t)$$

along with the transition equation of

$$W_{t+1} = W_t - c_t$$

[5] Here c_2 the consumption level in the second period of the three-period problem and thus is the same as the level of consumption in the first period of the two-period problem.

for t=1,2,......

Specifying this as a dynamic programming problem,

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c)$$

for all W. Here $u(c)$ is again the utility from consuming c units in the current period. $V(W)$ is the value of the infinite horizon problem starting with a cake of size W. So in the given period, the agent chooses current consumption and thus reduces the size of the cake to $W' = W - c$, as in the transition equation. We use variables with primes to denote future values. The value of starting the next period with a cake of that size in then given by $V(W - c)$ which is discounted at rate $\beta < 1$.

For this problem, the **state variable** is the size of the cake (W) that is given at the start of any period. The state completely summarizes all information from the past that is needed for the forward looking optimization problem. The **control variable** is the variable that is being chosen. In this case, it is the level of consumption in the current period, c. Note that c lies in a compact set. The dependence of the state tomorrow on the state today and the control today, given by

$$W' = W - c$$

is called the **transition equation**.

Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state.

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{7}$$

for all W. Either specification yields the same result. But choosing tomorrow's state often makes the algebra a bit easier so we will work with (7).

This expression is known as a **functional equation** and is often called a Bellman equation after Richard Bellman, one of the originators of dynamic programming. Note that the unknown in the Bellman equation is the value function itself: the idea is to find a function $V(W)$ that satisfies this condition for all W. Unlike the finite horizon problem, there is no terminal period to use to derive the value function. In effect, the fixed point restriction of having $V(W)$ on both sides of (7) will provide us with a means of solving the functional equation.

Note too that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.[6] In fact, we will ultimately use the stationarity of the problem to make arguments about the existence of a value function satisfying the functional equation.

A final very important property of this problem is that all information about the past that bears on current and future decisions is summarized by W, the size of the cake at the start of the period. Whether the cake is of this size because we initially had a large cake and ate a lot or a small cake and were frugal is not relevant. All that matters is that we have a cake of a given size. This property partly reflects the fact that the preferences of the agent do not

[6] As you may already know, stationarity is vital in econometrics as well. Thus making assumptions of stationarity in economic theory have a natural counterpart in empirical studies. In some cases, we will have to modify optimization problems to ensure stationarity.

depend on past consumption. But, in fact, if this was the case, we could amend the problem to allow this possibility.

The next part of this chapter addresses the question of whether there exists a value function that satisfies (7). For now, we assume that a solution exists and explore its properties.

The first order condition for the optimization problem in (7) can be written as

$$u'(c) = \beta V'(W').$$

This looks simple but what is the derivative of the value function? This seems particularly hard to answer since we do not know $V(W)$. However, we take use the fact that $V(W)$satisfies (7) for all W. Assuming that this value function is differentiable,

$$V'(W) = u'(c),$$

a result we have seen before. Since this holds for all W, it will hold in the following period yielding:

$$V'(W') = u'(c').$$

Substitution leads to the usual Euler equation:

$$u'(c) = \beta u'(c').$$

Thus the solution to the cake eating problem will satisfy this necessary condition for all W.

The link from the level of consumption and next period's cake (the controls from the different formulations) to the size of the cake (the state) is given by the **policy function:**

$$c = \phi(W), \quad W' = \varphi(W) \equiv W - \phi(W).$$

Using these in the Euler equation reduces the problem to these policy functions alone:

$$u'(\phi(W)) = \beta u'(\phi(W - \phi(W)))$$

for all W.

4.1.2 An Example

In general, actually finding closed form solutions for the value function and the resulting policy functions is not possible. In those cases, we try to characterize certain properties of the solution and, for some exercises, we solve these problems numerically.

However, as suggested by the analysis of the finite horizon examples, there are some versions of the problem we can solve completely. Suppose then, as above, that $u(c) = ln(c)$. Given the results for the T-period problem, we might conjecture that the solution to the functional equation takes the form of:

$$V(w) = A + B\ln(W)$$

for all W. With this guess we have reduced the dimensionality of the unknown function $V(W)$ to two parameters, A and B. But can we find values for A and B such that $V(W)$ will satisfy the functional equation?

Taking this guess as given and using the special preferences, the functional equation becomes:

$$A + B\ln(W) = \max_{W'} \ln(W - W') + \beta(A + B\ln(W')) \tag{8}$$

for all W. After some algebra, the first-order condition implies:

$$W' = \varphi(W) = \frac{\beta B}{(1+\beta B)}W.$$

Using this in (8) implies:

$$A + B\ln(W) = \ln\frac{W}{(1+\beta B)} + \beta(A + B\ln(\frac{\beta BW}{(1+\beta B)}))$$

for all W. Collecting terms into a constant and terms that multiply $ln(W)$ and then imposing the requirement that the functional equation must hold for all W, we find that

$$B = 1/(1-\beta)$$

is required for a solution. Given this, there is a complicated expression that can be used to find A. To be clear then we have indeed guessed a solution to the functional equation. We know that because we can solve for (A, B) such that the functional equation holds for all W using the optimal consumption and savings decision rules.

With this solution, we know that

$$c = W(1-\beta), W' = \beta W.$$

Evidently, the optimal policy is to save a constant fraction of the cake and eat the remaining fraction.

Interestingly, the solution to B could be guessed from the solution to the T-horizon problems where

$$B_T = \sum_{t=1}^{T} \beta^{t-1}.$$

Evidently, $B = lim_{T\to\infty} B_T$. In fact, we will be exploiting the theme that the value function which solves the infinite horizon problem is related to the limit of the finite solutions in much of our numerical analysis.

4.1.3 Exercises

Here are some exercises that add some interesting elements to this basic structure.

Exercise 1: Suppose that utility in period t was given by $u(c_t, c_{t-1})$. How would you solve the T period problem with these preferences? Interpret the first order conditions. How would you formulate the Bellman equation for the infinite horizon version of this problem?

Exercise 2: Suppose that the transition equation was modified so that

$$W_{t+1} = \rho W_t - c_t$$

where $\rho > 0$ represents a return from the holding of cake inventories. How would you solve the T period problem with this storage technology? Interpret the first order conditions. How would you formulate the Bellman equation for the infinite horizon version of this problem? Does the size of ρ matter in this discussion? Explain.

4.2 Taste Shocks

One of the convenient features of the dynamic programming problem is the simplicity with which one can introduce uncertainty.[7] For the cake eating problem, the natural source of uncertainty has to do with the agent's tastes. In other settings we will focus on other sources of uncertainty having to do with the productivity of labor or the endowment of households.

To allow for variations in tastes, suppose that utility over consumption is given by:

$$\varepsilon u(c)$$

where ε is a random variable whose properties we will describe below. The function $u(c)$ is strictly increasing and strictly concave. Otherwise, the problem is the original cake eating problem with an initial cake of size W.

In problems with stochastic elements, it is critical to be precise about the timing of events. Does the optimizing agent know the current shocks when making a decision? For this analysis, assume that the agent knows the value of the taste shock when making current decisions but does not know future values. Thus the agent must use expectations of future values of ε when deciding how much cake to eat today: it may be optimal to consume less today (save more) in anticipation of a high realization of ε in the future.

For simplicity, assume that the taste shock takes on only two values: $\varepsilon \in \{\varepsilon_h, \varepsilon_l\}$ with $\varepsilon_h > \varepsilon_l > 0$. Further, we assume that the taste shock follows a first -order Markov process which means that the probability a particular realization of ε occurs in the current period depends **only** the value of ε attained in the previous period.[8] For notation, let π_{ij} denote the probability that the value of ε goes from state i in the current period to state j in the next period. For example, π_{lh} is given by:

$$\pi_{lh} \equiv \text{Prob}(\varepsilon' = \varepsilon_h | \varepsilon = \varepsilon_l)$$

where ε' refers to the future value of ε. Clearly $\pi_{ih} + \pi_{il} = 1$ for $i = h, l$. Let Π be a $2x2$ matrix with a typical element π_{ij} which summarizes the information about the probability of moving across states. This matrix is naturally called a **transition matrix**.

Given this notation and structure, we can turn to the cake eating problem. It is critical to carefully define the state of the system for the optimizing agent. In the nonstochastic problem, the state was simply the size of the cake. This provided all the information the agent needed to make a choice. When taste shocks are introduced, the agent needs to take this into account as well. In fact, the taste shocks provide information about current payoffs and, through the Π matrix, are informative about the future value of the taste shock as well.

Formally, Bellman equation is:

---

[7] To be careful, here we are adding shocks that take values in a finite and thus countable set. See the discussion in Bertsekas [1976, 2.1] for an introduction to the complexities of the problem with more general statements of uncertainty.

[8] The evolution can also depend on the control of the previous period. Note too that by appropriate rewriting of the state space, richer specifications of uncertainty can be encompassed.

$$V(W,\varepsilon) = \max_{W'} \varepsilon u(W - W') + \beta E_{\varepsilon'|\varepsilon} V(W', \varepsilon')$$

for all (W, ε) where $W' = W - c$ as usual. Note that the conditional expectation is denoted here by $E_{\varepsilon'|\varepsilon} V(W', \varepsilon')$ which, given Π, is something we can compute.

The first order condition for this problem is given by:

$$\varepsilon u'(W - W') = \beta E_{\varepsilon'|\varepsilon} V_1(W', \varepsilon')$$

for all (W, ε). Using the functional equation to solve for the marginal value of cake, we find:

$$\varepsilon u'(W - W') = \beta E_{\varepsilon'|\varepsilon} [\varepsilon' u'(W' - W'')] \tag{9}$$

which, of course, is the stochastic Euler equation for this problem.

The optimal policy function is given by

$$W' = \varphi(W, \varepsilon)$$

The Euler equation can be rewritten in these terms as:

$$\varepsilon u'(W - \varphi(W,\varepsilon)) = \beta E_{\varepsilon'|\varepsilon} [\varepsilon' u'(\varphi(W,\varepsilon) - \varphi(\varphi(W,\varepsilon), \varepsilon')))]$$

The properties of this policy function can then be deduced from this condition. Clearly both ε' and c' depend on the realized value of ε' so that the expectation on the right side of (9) cannot be split into two separate pieces.

4.3 Discrete Choice

To illustrate some of the flexibility of the dynamic programming approach, we build on this stochastic problem and suppose the cake must be eaten in one period. Perhaps we should think of this as the wine drinking problem recognizing that once a good bottle of wine is opened, it should be consumed! Further, we modify the transition equation to allow the cake to grow (depreciate) at rate ρ.

The problem is then an example of a dynamic, stochastic discrete choice problem. This is an example of a family of problems called **optimal stopping problems**. The common element in all of these problems is the emphasis on timing of a single event: when to eat the cake; when to take a job; when to stop school, when to stop revising a chapter, etc. In fact, for many of these problems, these choices are not once in a lifetime events and so we will be looking at problems even richer than the optimal stopping variety.

Let $V^E(W, \varepsilon)$ and $V^N(W, \varepsilon)$ be the value of eating the size W cake now (E) and waiting (N) respectively given the current taste shock, $\varepsilon \in \{\varepsilon_h, \varepsilon_l\}$. Then,

$$V^E(W, \varepsilon) = \varepsilon u(W)$$

and

$$V^N(W) = \beta E_{\varepsilon'|\varepsilon} V(\rho W, \varepsilon').$$

where

$$V(W, \varepsilon) = \max(V^E(W, \varepsilon), V^N(W, \varepsilon)).$$

To understand these terms, $\varepsilon u(W)$ is the direct utility flow from eating the cake. Once the cake is eaten the problem has ended. So $V^E(W,\varepsilon)$ is just a one-period return. If the agent waits, then there is no cake consumption in the current period and next period the cake is of size (ρW). As tastes are stochastic, then the agent choosing to wait must take expectations of the future taste shock, ε'. The agent has an option next period of eating the cake or waiting further. Hence the value of having the cake in any state is given by $V(W,\varepsilon)$, which is the value attained by maximizing over the two options of eating or waiting. The cost of delaying the choice is determined by the discount factor β while the gains to delay are associated with the growth of the cake, parameterize by ρ. Further, the realized value of ε will surely influence the relative value of consuming the cake immediately.

If $\rho \leq 1$, then the cake doesn't grow. In this case, there is no gain from delay when $\varepsilon = \varepsilon_h$. If the agent delays, then utility in the next period will have to be lower due to discounting and, with probability π_{hl}, the taste shock will switch from low to high. So, waiting to eat the cake in the future will not be desirable. Hence,

$$V(W,\varepsilon_h) = V^E(W,\varepsilon_h) = \varepsilon_h u(W)$$

for all W.

In the low ε state, matters are more complex. If β and ρ are sufficiently close to 1 then there is not a large cost to delay. Further, if π_{lh} is sufficiently close to 1, then it is likely that tastes will switch from low to high. Thus it will be optimal not to eat the cake in state (W,ε_l).[9]

Here are some additional exercises.

Exercise 3: Suppose that $\rho = 1$. For a given β, show that there exists a critical level of π_{lh},denoted by $\bar{\pi}_{lh}$ such that if $\pi_{lh} > \bar{\pi}_{lh}$,then the optimal solution is for the agent to wait when $\varepsilon = \varepsilon_l$ and to eat the cake when ε_h is realized.

When $\rho > 1$,the problem is more difficult. Suppose that there are no variations in tastes: $\varepsilon_h = \varepsilon_l = 1$. In this case, there is a trade-off between the value of waiting (as the cake grows) and the cost of delay from discounting.

Exercise 4: Suppose that $\rho > 1$ and $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$. What is the solution to the optimal stopping problem when $\beta\rho^{1-\gamma} < 1$? What happens if $\beta\rho^{1-\gamma} > 1$? What happens when uncertainty is added.

5 General Formulation

Building on the intuition gained from this discussion of the cake eating problem, we now consider a more formal abstract treatment of the dynamic programming approach.[10]

[9] In the following chapter on the numerical approach to dynamic programming, we study this case in considerable detail.

[10] This section is intended to be self-contained and thus repeats some of the material from the earlier examples. Our presentation is by design not as formal as say that provided in Bertsekas [1976] or Stokey-Lucas [1989]. The reader interested in more mathematical rigor is urged to review those texts and their many references.

We begin with a discussion of the non-stochastic problem and then add uncertainty to the formulation.

5.1 Non-Stochastic Case

Consider the infinite horizon optimization problem of an agent with a payoff function for period t given by $\tilde{\sigma}(s_t, c_t)$. The first argument of the payoff function is termed the **state vector**, (s_t). As noted above, this represents a set of variables that influences the agent's return within the period but, by assumption, these variables are outside of the agent's control within period t. The state variables evolve over time in a manner that may be influenced by the **control vector** (c_t), the second argument of the payoff function. The connection between the state variables over time is given by the transition equation:

$$s_{t+1} = \tau(s_t, c_t).$$

So, given the current state and the current control, the state vector for the subsequent period is determined.

Note that the state vector has a very important property: it completely summarizes all of the information from the past that is needed to make a forward-looking decision. While preferences and the transition equation are certainly dependent on the past, this dependence is represented by s_t : other variables from the past do not affect current payoffs or constraints and thus cannot influence current decisions. This may seem restrictive but it is not: the vector s_t may include many variables so that the dependence of current choices on the past can be quite rich.

While the state vector is effectively determined by preferences and the transition equation, the researcher has some latitude in choosing the control vector. That is, there may be multiple ways of representing the same problem with alternative specifications of the control variables.

We assume that $c \in C$ and $s \in S$. In some cases, the control is restricted to be in subset of C which depends on the state vector: $c \in C(s)$. Finally assume that $\tilde{\sigma}(s, c)$ is bounded for $(s, c) \in SxC$.[11]

For the cake eating problem described above, the state of the system was the size of the current cake (W_t) and the control variable was the level of consumption in period t, (c_t). The transition equation describing the evolution of the cake was given by

$$W_{t+1} = W_t - c_t.$$

Clearly the evolution of the cake is governed by the amount of current consumption. An equivalent representation, as expressed in (7), is to consider the future size of the cake as the control variable and then to simply write current consumption as $W_{t+1} - W_t$.

There are two final properties of the agent's dynamic optimization problem worth specifying: **stationarity** and **discounting**. Note that neither the payoff nor the transition equations depend explicitly on time. True the problem is dynamic but time *per se* is not of the essence. The optimal choice of the agent will be the same regardless of when he

[11] Ensuring that the problem is bounded is an issue in some economic applications, such as the growth model. Often these problems are dealt with by bounding the sets C and S.

optimizes. Stationarity is important both for the analysis of the optimization problem and for empirical implementation. In fact, because of stationarity we can dispense with time subscripts as the problem is completely summarized by the current values of the state variables.

The agent's preferences are also dependent on the rate at which the future is discounted. Let β denote the discount factor and assume that $0 < \beta < 1$. Then we can represent the agent's payoffs over the infinite horizon as

$$\sum_{t=0}^{t=\infty} \beta^t \tilde{\sigma}(s_t, c_t) \tag{10}$$

One approach to optimization is then to maximize (10) through the choice of $\{c_t\}$ for $t = 0, 1, 2, ...$ given s_0 and subject to the transition equation. Let $V(s_0)$ be the optimized value of this problem given the initial state.

Alternatively, one can adopt the dynamic program approach and consider the following equation, called Bellman's equation:

$$V(s) = \max_{c \in C(s)} \tilde{\sigma}(s, c) + \beta V(s') \tag{11}$$

for all $s \in S$, where $s' = \tau(s, c)$. Here time subscripts are eliminated, reflecting the stationarity of the problem. Instead, current variables are unprimed while future ones are denoted by a prime ($'$).

Alternatively, as in Stokey-Lucas [1989], the problem can be formulated as

$$V(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta V(s') \tag{12}$$

This is a more compact formulation and we will use it for our presentation.[12] Nonetheless, the presentations in Bertsekas [1976] and Sargent [1987] follow (11). Assume that S is a convex subset of $\Re^k$.

Let the policy function that determines the optimal value of the control (the future state) given the state be given by $s' = \phi(s)$. Our interest is ultimately in the policy function since we generally observe the actions of agents rather than their levels of utility. Still, to determine $\phi(s)$ we need to "solve" (12). That is, we need to find the value function that satisfies (12). It is important to realize that while the payoff and transition equations are primitive objects that models specify *a priori,* the value function is derived as the solution of the functional equation, (12)

There are many results in the lengthy literature on dynamic programming problems on the existence of a solution to the functional equation. Here, we present one set of sufficient conditions. The reader is referred to Bertsekas [1976], Sargent [1987] and Stokey-Lucas [1989] for additional theorems under alternative assumptions about the payoff and transition functions.[13]

[12] Essentially, this formulation inverts the transition equation and substitutes for c in the objective function. This substitution is reflected in the alternative notation for the return function.

[13] Some of the applications explored in this book will not exactly fit these conditions either. In those cases, we will alert the reader and discuss the conditions under which there exists a solution to the functional equation.

Theorem 1 *Assume $\sigma(s,c)$ is real-valued, continuous and bounded, $0 < \beta < 1$ and the constraint set is non-empty, compact-valued and continuous, then there exists a unique value function $V(s)$ that solves (12)*

Proof: See Stokey-Lucas [1989, Theorem 4.6].

Instead of a formal proof, we give an intuitive sketch. The key component in the analysis is the definition of an operator, commonly denoted as T, defined by:

$$T(W)(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta W(s') \text{ for all } s \in S.$$ [14]

So, this mapping takes a guess on the value function and, working through the maximization for all s, produces another value function, $T(W)(s)$. Clear, any $V(s)$ such that $V(s) = T(V)(s)$ for all $s \in S$ is a solution to (12). So, we can reduce the analysis to determining the fixed points of $T(W)$.

The fixed point argument proceeds by showing the $T(W)$ is a contraction using a pair of sufficient conditions from Blackwell [1965]. This implies that: (i) there is a unique fixed point and (ii) this fixed point can be reached by an iteration process using an arbitrary initial condition. The first property is reflected in the theorem given above and the second is used extensively as a means of finding the solution to (12).[15]

Blackwell's conditions are: (i) monotonicity and (ii) discounting of the mapping $T(V)$. **Monotonicity** means that if $W(s) \geq Q(s)$ for all $s \in S$, then $T(W)(s) \geq T(Q)(s)$ for all $s \in S$. This property can be directly verified from the fact that $T(V)$ is generated by a maximization problem. So that if one adopts the choice of $c_Q(s)$ obtained from

$$\max_{s' \in \Gamma(s)} \sigma(s, s') + \beta Q(s') \text{ for all } s \in S.$$

When the proposed value function is $W(s)$ then:

$$\begin{aligned} T(W)(s) &= \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta W(s') \geq \sigma(s, c_Q(s)) + \beta W(c_Q(s)) \\ &\geq \sigma(s, c_Q(s)) + \beta Q(c_Q(s)) \equiv T(Q)(s) \end{aligned}$$

for all $s \in S$.

Discounting means that adding a constant to W leads $T(W)$ to increase by less than this constant. That is, for any constant k, $T(W + k)(s) \leq T(W)(s) + \beta k$ for all $s \in S$ where $\beta \in [0, 1)$. The term discounting reflects the fact that β must be less than 1. This property is easy to verify in the dynamic programming problem:

$$T(W + k) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta[W(s') + k] = T(W) + \beta k, \text{ for all } s \in S$$

since we assume that the discount factor is less than 1.

Besides these properties, there are two other points of note concerning this application of the contraction mapping theorem. First, this theorem not only implies the existence of a unique solution to (12) but also provides a way to find such a solution. Let $V_0(s)$ for all for all $s \in S$ be an initial guess of the solution to (12). Consider $V_1 = T(V_0)$. If $V_1 = V_0$ for all

[14] The notation dates back at least to Bertsekas [1976].

[15] Bertsekas proves a version of this theorem directly though his proof uses monotonicity and discounting.

$s \in S$,then we have the solution. Else, consider $V_2 = T(V_1)$ and continue until $T(V) = V$ so that the functional equation is satisfied. Of course, in general, there is no reason to think that this iterative process will converge. However, if $T(V)$ is a contraction, as it is for our dynamic programming framework, then the $V(s)$ that satisfies (12) can be found from the iteration of $T(V_0(s))$ for any initial guess, $V_0(s)$. This procedure is called **value function iteration** and will be a valuable tool for applied analysis of dynamic programming problems.

The second property of immense value is that the value function that satisfies (12) may inherit some properties from the more primitive functions that are the inputs into the dynamic programming problem: the payoff and transition equations. As we shall see, the property of strict concavity is useful for various applications.[16] The result is given formally by:

Theorem 2 *Assume $\sigma(s, s')$ is real-valued, continuous, concave and bounded, $0 < \beta < 1$, S is a convex subset of $\Re^k$ and the constraint set is non-empty, compact-valued, convex and continuous, then the unique solution to (12) is strictly concave. Further, $\phi(s)$ is a continuous, single-valued function.*

Proof: See Theorem 4.8 in Stokey-Lucas [1989].

The proof of the theorem relies on showing that strict concavity is preserved by $T(V)$: i.e. if $V(s)$ is strictly concave, then so is $T(V(s))$. Given that $\sigma(s, c)$ is concave, then we can let our initial guess of the value function be the solution to the one-period problem:

$$V_0(s) \equiv \max_{s' \in \Gamma(s)} \sigma(s, s')$$

which will be strictly concave. Since $T(V)$ preserves this property then the solution to (12) will be strictly concave.

As noted earlier, our interest is in the policy function. Note that from this theorem, there is a stationary policy **function** which depends only on the state vector. This result is important for econometric application since stationarity is often assumed in characterizing the properties of various estimators.

The cake eating example relied on the Euler equation to determine some properties of the optimal solution. However, the first-order condition from (12) combined with the strict concavity of the value function is useful in determining properties of the policy function. Benveniste and Schienkman [1979] provide conditions such that $V(s)$ is differentiable (Stokey-Lucas [1989], Theorem 4.11). In our discussion of applications, we will see arguments that use the concavity of the value function to characterize the policy function.

5.2 Stochastic Dynamic Programming

While the nonstochastic problem is perhaps a natural starting point, in terms of applications it is necessary to consider stochastic elements. Clearly the stochastic growth model, consumption/savings decisions by households, factor demand by firms, pricing decisions by sellers, search decisions all involve the specification of dynamic stochastic environments.

[16] Define $\sigma(s, s')$ as concave if $\sigma(\lambda(s_1, s_1') + (1-\lambda)(s_2, s_2')) \geq \lambda\sigma(s_1, s_1') + (1-\lambda)\sigma(s_2, s_2')$ for all $0 < \lambda < 1$ where the inequality is strict if $s_1 \neq s_2$.

While stochastic elements can be added in many ways to dynamic programming problems, we consider the following formulation which is used in our applications. Letting ε represent the current value of a vector of "shocks"; i.e. random variables that are partially determined by nature. Let $\varepsilon \in \Psi$ which is assumed to be a finite set.[17] Then using the notation developed above, the functional equation becomes:

$$V(s,\varepsilon) = \max_{s' \in \Gamma(s,\varepsilon)} \sigma(s,s',\varepsilon) + \beta E_{\varepsilon'|\varepsilon} V(s',\varepsilon') \tag{13}$$

for all (s,ε).

Further, we have assumed that the stochastic process itself is purely exogenous as the distribution of ε' depends on ε but is independent of the current state and control. Note too that the distribution of ε' depends on only the realized value of ε : i.e. ε follows a first-order Markov process. This is not restrictive in the sense that if values of shocks from previous periods were relevant for the distribution of ε', then they could simply be added to the state vector.

Finally, note that the distribution of $\varepsilon'|\varepsilon$ is time invariant. This is analogous to the stationarity properties of the payoff and transition equations.. In this case, the conditional probability of $\varepsilon'|\varepsilon$ are characterized by a transition matrix, Π. The element π_{ij} of this matrix is defined as:

$$\pi_{ij} \equiv \text{Prob}(\varepsilon' = \varepsilon_j | \varepsilon = \varepsilon_i)$$

which is just the likelihood that ε_j occurs in the next period, given that ε_i occurs today. Thus this transition matrix is used to compute the transition probabilities in (13). Throughout we assume that $\pi_{ij} \in (0,1)$ and $\sum_j \pi_{ij} = 1$ for each i. With this structure:

Theorem 3 *If $\sigma(s,s',\varepsilon)$ is real-valued, continuous, concave and bounded, $0 < \beta < 1$ and the constraint set is compact and convex, then:*

1. there exists a unique value function $V(s,\varepsilon)$ that solves (13)
2. there exists a stationary policy function, $\phi(s,\varepsilon)$.

Proof: As in the proof of Theorem 2, this is a direct application of Blackwell's Theorem. That is, with $\beta < 1$, discounting holds. Likewise, monotonicity is immediate as in the discussion above. See also the proof of Proposition 2 in Bertsekas [1976, Chp. 6].

6 References

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

Bertsekas, D. *Dynamic Programming and Stochastic Control*, New York: Academic Press, 1976.

[17] As noted earlier, this structure is stronger than necessary but accords with the approach we will take in our empirical implementation. The results reported in Bertsekas [1976] require that Ψ is countable.

Ljungqvist, L. and T. Sargent, *Recursive Macroeconomic Theory*, Cambridge, Mass: MIT Press, 2000.

Sargent, T. *Dynamic Macroeconomic Theory*, Cambridge, Mass.:Harvard University Press: 1987.

Stokey, N. and R. Lucas, *Recursive Methods in Economic Dynamics*, Cambridge, Mass.: Harvard University Press, 1989.