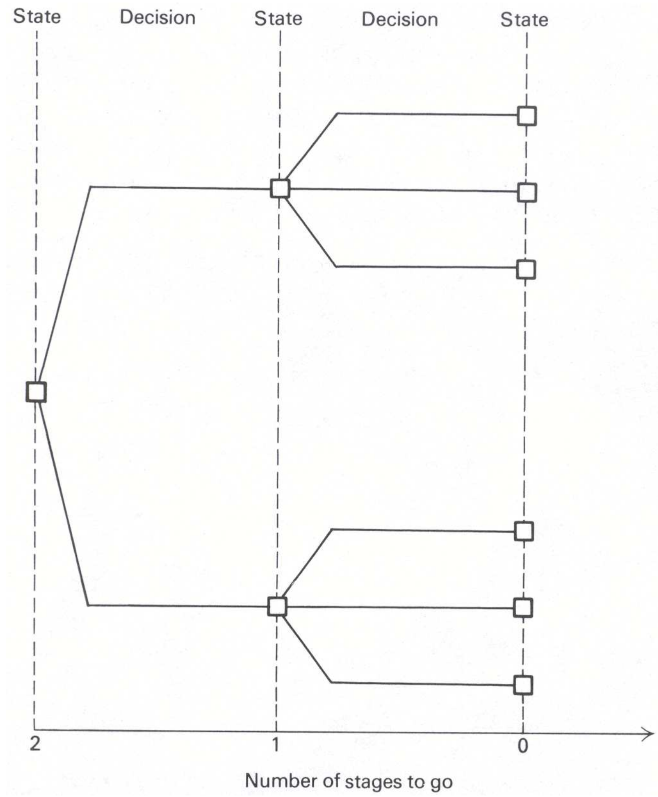

Figure 11.14 Decision tree for deterministic dynamic programming.

the system at the next stage, given by $s_{n-1} = t_n(d_n, s_n)$, are both known with certainty. This deterministic process can be represented by means of the decision tree in Fig. 11.14. As one can observe, given the current state, a specific decision leads with complete certainty to a particular state at the next stage. The stage returns are also known with certainty and are associated with the branches of the tree.

When uncertainty is present in a dynamic-programming problem, a specific decision for a given state and stage of the process does not, by itself, determine the state of the system at the next stage; this decision may not even determine the return for the current stage. Rather, in dynamic programming under uncertainty, given the state of the system s_n with n stages to go and the current decision d_n, an uncertain event occurs which is determined by a random variable $\tilde{e}_n$ whose outcome e_n is *not* under the control of the decision maker. The stage return function may depend on this random variable, that is,

$$f_n(d_n, s_n, \tilde{e}_n),$$

while the state of the system s_{n-1} with $(n-1)$ stages to go invariably will depend on the random variable by

$$\tilde{s}_{n-1} = t_n(d_n, s_n, \tilde{e}_n).$$

The outcomes of the random variable are governed by a probability distribution, $p_n(e_n|d_n, s_n)$, which may be the same for every stage or may be conditional on the stage, the state at the current stage, and even the decision at the current stage.

Figure 11.15 depicts dynamic programming under uncertainty as a *decision tree,* where squares represent states where decisions have to be made and circles represent uncertain events whose outcomes are not under the control of the decision maker. These diagrams can be quite useful in analyzing decisions under uncertainty if the number of possible states is not too large. The decision tree provides a pictorial representation of the sequence of decisions, outcomes, and resulting states, *in the order in which* the decisions must be made and the outcomes become known to the decision maker. Unlike deterministic dynamic programming wherein the optimal decisions at each stage can be specified at the outset, in dynamic programming under uncertainty, the optimal decision at each stage can be selected only after we know the outcome of the uncertain event at the previous stage. At the outset, all that can be specified is a set of decisions that would be made *contingent* on the outcome of a sequence of uncertain events.

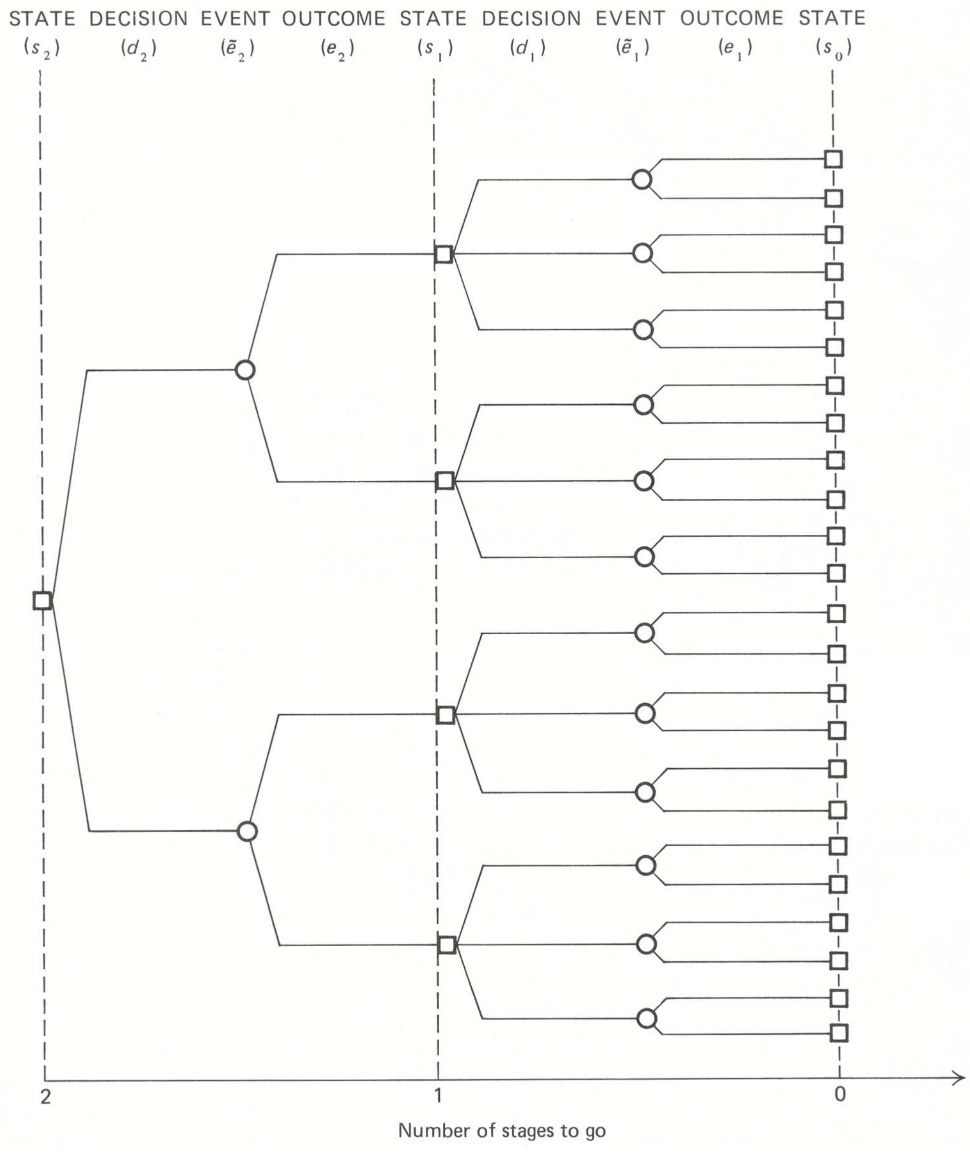

Figure 11.15 Decision tree for dynamic programming under uncertainty.

In dynamic programming under uncertainty, since the stage returns and resulting stage may both be uncertain at each stage, we cannot simply optimize the sum of the stage-return functions. Rather, we must optimize the *expected return* over the stages of the problem, taking into account the sequence in which decisions can be made and the outcomes of uncertain events become known to the decision maker. In this situation, backward induction can be applied to determine the optimal strategy, but forward induction cannot. The difficulty with forward induction is that it is impossible to assign values to states at the next stage that are independent of the uncertain evolution of the process from that future state on. With backward induction, on the other hand, no such difficulties arise since the states with zero stages to go are evaluated first, and then the states with one stage to go are evaluated by computing the expected value of any decision and choosing optimally.

We start the backward induction process by computing the optimal-value function at stage 0. This amounts to determining the value of ending in each possible stage with 0 stages to go. This determination may involve an optimization problem or the value of the assets held at the horizon. Next, we compute the optimal-value function at the previous stage. To do this, we first compute the expected value of each uncertain event, weighting the stage return plus the value of the resulting state for each outcome by the probability of each outcome. Then, for each state at the previous stage, we select the decision that has the maximum (or minimum) expected value. Once the optimal-value function for stage 1 has been determined, we continue in a similar manner to determine the optimal-value functions at prior stages by backward induction.

The optimal-value function for dynamic programming under uncertainty is then defined in the following recursive form:

$$v_n(s_n) = \text{Max } E\left[f_n(d_n, s_n, \tilde{e}_n) + v_{n-1}(\tilde{s}_{n-1}) \right], \tag{17}$$

subject to:

$$\tilde{s}_{n-1} = t_n(d_n, s_n, \tilde{e}_n),$$
$$d_n \in D_n,$$

where $E[\cdot]$ denotes the expected value of the quantity in brackets. To initiate the recursive calculations we need to determine the optimal-value function with zero stages to go, which is given by:

$$v_0(s_0) = \text{Max } E\left[f_0(d_0, s_0, \tilde{e}_0) \right],$$

subject to:

$$d_0 \in D_0.$$

The optimization problems that determine the optimal-value function with zero stages to go are not determined recursively, and therefore may be solved in a straight-forward manner. If the objective function is to maximize the expected discounted costs, then Eq. (17) is modified as in Section 11.4 by multiplying the term $v_{n-1}(\tilde{s}_{n-1})$ by β_n, the discount factor for period n.

We can make these ideas more concrete by considering a simple example. A manager is in charge of the replenishment decisions during the next two months for the inventory of a fairly expensive item. The production cost of the item is \$1000/unit, and its selling price is \$2000/unit. There is an inventory-carrying cost of \$100/unit per month on each unit left over at the end of the month. We assume there is no setup cost associated with running a production order, and further that the production process has a short lead time; therefore any amount produced during a given month is available to satisfy the demand during that month. At the present time, there is no inventory on hand. Any inventory left at the end of the next two months has to be disposed of at a salvage value of \$500/unit.

The demand for the item is uncertain, but its probability distribution is identical for each of the coming two months. The probability distribution of the demand is as follows:

Demand	*Probability*
0	0.25
1	0.40
2	0.20
3	0.15

The issue to be resolved is how many units to produce during the first month and, *depending on the actual demand in the first month*, how many units to produce during the second month. Since demand is uncertain, the inventory at the end of each month is also uncertain. In fact, demand could exceed the available units on hand in any month, in which case all excess demand results in lost sales. Consequently, our production decision must find the proper balance between production costs, lost sales, and final inventory salvage value.

The states for this type of problem are usually represented by the inventory level I_n at the beginning of each month. Moreover, the problem is characterized as a two-stage problem, since there are two months involved in the inventory-replenishment decision. To determine the optimal-value function, let

$v_n(I_n) =$ Maximum contribution, given that we have I_n units of inventory with n stages to go.

We initiate the backward induction procedure by determining the optimal-value function with 0 stages to go. Since the salvagevalue is \$500/unit, we have:

I_0	$v_0(I_0)$
0	0
1	500
2	1000
3	1500

To compute the optimal-value function with one stage to go, we need to determine, for each inventory level (state), the corresponding contribution associated with each possible production amount (decision) and level of sales (outcome). For each inventory level, we select the production amount that maximizes the expected contribution.

Table 11.3 provides all the necessary detailed computations to determine the optimal-value function with one stage to go. Column 1 gives the state (inventory level) of the process with one stage to go. Column 2 gives the possible decisions (amount to produce) for each state, and, since demand cannot be greater than three, the amount produced is at most three. Column 3 gives the possible outcomes for the uncertain level of sales for each decision and current state, and column 4 gives the probability of each of these possible outcomes. Note that, in any period, it is impossible to sell more than the supply, which is the sum of the inventory currently on hand plus the amount produced. Hence, the probability distribution of sales differs from that of demand since, whenever demand exceeds supply, the entire supply is sold and the excess demand is lost. Column 5 is the resulting state, given that we currently have I_1 on hand, produce d_1, and sell s_1. The transition function in general is just:

$$\tilde{I}_{n-1} = I_n + d_n - \tilde{s}_n,$$

where the tildes ($\sim$) indicate that the level of sales is uncertain and, hence, the resulting state is also uncertain. Columns 6, 7, and 8 reflect the revenue and costs for each state, decision, and sales level, and column 9 reflects the value of being in the resulting state at the next stage. Column 10 merely weights the sum of columns 6 through 9 by the probability of their occurring, which is an intermediate calculation in determining the expected value of making a particular decision, given the current state. Column 11 is then just this expected value; and the asterisk indicates the optimal decision for each possible state.

Table 11.3 Computation of optimal-value function with one stage to go.

(1) State I_1	*(2)* Produce d_1	*(3)* Sell s_1	*(4)* Probability $(\tilde{s}_1 = s_1)$	*(5)* Resulting state $\tilde{I}_0$	*(6)* Production cost	*(7)* Sales revenue	*(8)* Inventory cost	*(9)* $v_0(I_0)$	*(10)* Probability × $	*(11)* Expected contribution
0	0	0	1.	0	0	0	0	0	0	0
	1	0	.25	1	−1000	0	−100	500	−150	600*
		1	.75	0	−1000	2000	0	0	750	
	2	0	.25	2	−2000	0	−200	1000	−300	560
		1	.40	1	−2000	2000	−100	500	160	
		2	.35	0	−2000	4000	0	0	700	
	3	0	.25	3	−3000	0	−300	1500	−450	200
		1	.40	2	−3000	2000	−200	1000	−80	
		2	.20	1	−3000	4000	−100	500	280	
		3	.15	0	−3000	6000	0	0	450	
1	0	0	.25	1	0	0	−100	500	100	1600*
		1	.75	0	0	2000	0	0	1500	
	1	0	.25	2	−1000	0	−200	1000	−50	1560
		1	.40	1	−1000	2000	−100	500	560	
		2	.35	0	−1000	4000	0	0	1050	
	2	0	.25	3	−2000	0	−300	1500	−200	1200
		1	.40	2	−2000	2000	−200	1000	320	
		2	.20	1	−2000	4000	−100	500	480	
		3	.15	0	−2000	6000	0	0	600	
2	0	0	.25	2	0	0	−200	1000	200	2560*
		1	.40	1	0	2000	−100	500	960	
		2	.35	0	0	4000	0	0	1400	
	1	0	.25	3	−1000	0	−300	1500	50	2200
		1	.40	2	−1000	2000	−200	1000	720	
		2	.20	1	−1000	4000	−100	500	680	
		3	.15	0	−1000	6000	0	0	750	
3	0	0	.25	3	0	0	−300	1500	300	3200*
		1	.40	2	0	2000	−200	1000	1120	
		2	.20	1	0	4000	−100	500	880	
		3	.15	0	0	6000	0	0	900	

The resulting optimal-value function and the corresponding optimal-decision function are determined directly from Table 11.3 and are the following:

I_1	$v_1(I_1)$	$d_1^*(I_1)$
0	600	1
1	1600	0
2	2560	0
3	3200	0

Next we need to compute the optimal-value function with two stagesto go. However, since we have assumed that there is no initial inventory on hand, it is not necessary to describe the optimal-value function for every possible state, but only for $I_2 = 0$. Table 11.4 is similar to Table 11.3 and gives the detailed computations required to evaluate the optimal-value function for this case.

Table 11.4 Computation of optimal-value function with two stages to go, $I_2 = 0$ only.

(1) State I_2	(2) Produce d_2	(3) Sell s_2	(4) Probability $(\tilde{s}_2 = s_2)$	(5) Resulting state $\tilde{I}_1$	(6) Production cost	(7) Sales revenue	(8) Inventory cost	(9) $v_1(I_1)$	(10) Probability × $	(11) Expected contribution
0	0	0	1.	0	0	0	0	650.	650	650
	1	0	.25	1	−1000	0	0	1600.	150	1350
		1	.75	0	−1000	2000	0	600.	1200	
	2	0	.25	2	−2000	0	−200	2560.	90	1600.*
		1	.40	1	−2000	2000	−100	1600.	600	
		2	.35	0	−2000	4000	0	600	910	
	3	0	.25	3	−3000	0	−300	3200.	−25	1559
		1	.40	2	−3000	2000	−200	2560.	544	
		2	.20	1	−3000	4000	−100	1600.	500	
		3	.15	0	−3000	6000	0	600.	540	

The optimal-value function and the corresponding decision function for $I_2 = 0$ are taken directly from Table 11.4 and are the following:

I_2	$v_2(I_2)$	$d_2^*(I_2)$
0	1600	2

The optimal strategy can be summarized by the decision tree given in Fig. 11.16. The expected contribution determined by the dynamic-programming solution corresponds to weighting the contribution of every path in this tree by the *probability* that this path occurs. The decision tree in Fig. 11.16 emphasizes the contingent nature of the optimal strategy determined by dynamic programming under uncertainty.