The Equipment-Replacement Problem

Consider a piece of equipment, say a car, that has to be operated throughout a planning horizon of N periods. Suppose that each period corresponds to one year; and that we are required to make a decision as to whether or not to replace the car at the beginning of every year. The problem of interest is to determine an optimal replacement policy under the following set of assumptions.

- The annual operating cost of an i-year-old car is $c(i)$, where $i = 1, 2, \ldots, N$.
- The price of a new car is p.
- The trade-in value of an i-year-old car is $t(i)$, for $i = 1, 2, \ldots, N$.
- The salvage value of an i-year-old car at the end of year N is $s(i)$, for $i = 1, 2, \ldots, N$.

To illustrate these assumptions, let us consider a simple numerical example. Suppose a car is needed for three years; that is, suppose $N = 3$. At the beginning of the first year, we have a 2-year-old car. The annual cost of operating a car is a function of its age; and this cost function is given by: $c(0) = 10$, $c(1) = 20$, $c(2) = 40$, $c(3) = 60$, and $c(4) = 70$. The fact that these costs are increasing is a reflection of aging. Since the operating costs are increasing, it may become more cost effective to replace a car after it has been in operation for some periods. The price of a new car is 60, i.e., $p = 60$. (Our solution below can be easily adapted to reflect price variations over time.) The trade-in value of a used car is a function of its age at the time of trade in; and this function is given by: $t(1) = 30$, $t(2) = 20$, $t(3) = 15$, and $t(4) = 10$. These trade-in values are decreasing, as a reflection of the decline in desirability of a car over time. Finally, a car will no longer be needed at the end of year N; therefore, the car in service at that time will be salvaged. The salvage value of a used car is again a function of its age; and this function is given by: $s(1) = 20$, $s(2) = 15$, $s(3) = 10$, $s(4) = 0$, and $s(5) = 0$. Like the trade-in values, the salvage values are also decreasing. Moreover, notice that the salvage value of a car, at any given age, is less than the trade-in value of a car with the same age.

A replacement *policy* is a specification of a sequence of "keep" or "replace" actions, one for each period. Two simple examples are the policy of replacing the car every year and the policy of keeping the first car until the end of period N. An optimal policy is a policy that achieves the smallest total net cost of ownership over the entire planning horizon.

To illustrate the calculation of total net cost, consider the policy of replacing the car at the beginning of every year. Recall that our initial condition is to start with a 2-year-old car. If this car is traded in, then we will pay p for a new car, receive $t(2)$ from the trade-in, and incur $c(0)$ for operating the new car (at age 0). It follows that the net cost for the first year is given by $p - t(2) + c(0)$. Similarly, for both the second and the third year, the annual net cost is given by $p - t(1) + c(0)$. Finally, since the car in service is salvaged, at age 1, at the end of year 3 (or at the beginning of year 4), we will receive a terminal payment of $s(1)$. Hence, the total net cost over the entire planning horizon is:

$$\begin{aligned} & [p - t(2) + c(0)] + [p - t(1) + c(0)] + [p - t(1) + c(0)] - s(1) \\ = \; & [60 - 20 + 10] + [60 - 30 + 10] + [60 - 30 + 10] - 20 \\ = \; & 110. \end{aligned}$$

As a second example, the total net cost for the policy of never replacing the car can be easily calculated as:

$$c(2) + c(3) + c(4) - s(5) = 40 + 60 + 70 - 0 = 170.$$

It follows that this policy is worse than the previous one. Now, with two available actions for each year, the total number of possible policies is finite, and it is equal to $2^3 = 8$. Therefore, continuation of similar calculations for the remaining 6 policies will eventually lead to the identification of the optimal policy. However, for problems with a longer planning horizon, brutal enumeration will be too time consuming.

We now describe how to derive the optimal policy for this problem using dynamic programming. The solution procedure will be organized into four steps:

1. Definition of appropriate stages and states.
2. Definition of the optimal-value function.
3. Construction of a recurrence relation.
4. Recursive Computation.

Stages and States

Since there is one decision per year, it is natural to consider each year a stage.

We shall refer to the year count (or index) as the *stage variable*.

The definition of states requires a little bit more thought. As noted in our discussion of the elementary path problem, the state information corresponds to a specification of "where we are" within a given stage. So, what is the appropriate definition of states in this problem?

An extremely helpful notion in this regard is to ask the so-called *consultant question*. What this means is explained as follows. Imagine yourself as a consultant who is hired at the beginning of, say, year k, where $1 \leq k \leq N$. Suppose you are charged with making all remaining decisions from year k to the end, regardless of what has been done prior to your hiring. Then, as the consultant, it is important to ask yourself: What is the minimal amount of information that will enable me to make these remaining decisions? The conventional wisdom is that the "correct" answer to this consultant question invariably motivates the appropriate state definition.

A little bit of reflection should convince you that the answer to the consultant question in the context of the equipment-replacement problem is the age of the car in service at the end of year $k-1$, or equivalently, at the beginning of year k. It follows that we should define the age as the state.

We shall refer to the age of the car in service at the beginning of a year as the *state variable.*

Optimal-Value Function

Recall that the optimal-value function is a function that returns, for any given pair of stage and state, the best possible total cost from that point to the end. With the stage and state variables appropriately defined, the definition of the optimal-value function is a simple matter of adapting this statement to the particular context of a given problem. That is, we will define

$V_k(i)$ = the minimal total net cost from year k to the end of year N, starting with an i-year-old car in year k.

Our goal, in the particular numerical example above, is to determine $V_1(2)$ via a stage-by-stage recursive computation.

Recurrence Relation

Imagine being at the beginning of year k with an i-year-old car. There are two available actions: keep or replace (the car).

Suppose the action chosen is to keep the i-year-old car. Then, the immediate one-stage cost is simply $c(i)$. Since the next stage and state as a result of this action is $k+1$ and $i+1$, the minimal total future net cost from that point to the end is, by definition, $V_{k+1}(i+1)$. It follows that the best possible total net cost associated with the keep action is given by $c(i) + V_{k+1}(i+1)$.

Suppose, on the other hand, the action chosen is to replace the i-year-old car. Then, the immediate one-stage cost is the sum of: p (the price of a new car), $-t(i)$ (the negative of the revenue from trading in the i-year-old car), and $c(0)$ (the operating cost of a new car).

Since the next stage and state as a result of this action is $k+1$ and 1, the minimal total future net cost from that point to the end is, by definition, $V_{k+1}(1)$. It follows that the best possible total net cost associated with the replace action is given by $p-t(i)+c(0)+V_{k+1}(1)$.

Since our goal is to minimize the total net cost, the recurrence relation is:

$$V_k(i) = \min\left[c(i) + V_{k+1}(i+1),\, p - t(i) + c(0) + V_{k+1}(1)\right].$$

With the recurrence relation in place, the final step of the solution procedure consists of the recursive computation of the $V_k(i)$'s.

Computation

We begin with the specification of the boundary condition. For this purpose, it is convenient to view the end of year 3 as the beginning of a final stage 4, where the only available action is to salvage the car in service. Since the revenue received from salvaging a car can be interpreted as a negative cost, this yields the boundary condition specified in the table below.

Stage 4:

i	$V_4(i)$
1	-20
2	-15
3	-10
4	0
5	0

Note that the highest possible state is 5. This is a consequence of the fact that we begin year 1 with a 2-year-old car and the planning horizon is 3 years.

We now consider stage 3, where the highest possible state is 4. For state 1, the one-stage costs associated with the keep and replace actions are $c(1) = 20$ and $p - t(1) + c(0) = 60 - 30 + 10 = 40$, respectively. For state 2, the one-stage costs associated with the keep and replace actions are $c(2) = 40$ and $p - t(2) + c(0) = 60 - 20 + 10 = 50$, respectively. For state 3, the one-stage costs associated with the keep and replace actions are $c(3) = 60$ and $p - t(3) + c(0) = 60 - 15 + 10 = 55$, respectively. Finally, for state 4, the one-stage costs associated with the keep and replace actions are $c(4) = 70$ and $p - t(4) + c(0) = 60 - 10 + 10 = 60$, respectively. Substitution of these one-stage costs and the relevant $V_4(i)$'s from the stage-4 table above into the recurrence relation

$$V_3(i) = \min\left[c(i) + V_4(i+1),\, p - t(i) + c(0) + V_4(1)\right]$$

now yields the table below.

Stage 3:

i	Actions: Keep	Actions: Replace	$V_3(i)$	Optimal Action
1	$20+(-15)=5$	$40+(-20)=20$	5	Keep
2	$40+(-10)=30$	$50+(-20)=30$	30	Keep or Replace
3	$60+0=60$	$55+(-20)=35$	35	Replace
4	$70+0=70$	$60+(-20)=40$	40	Replace

Note that for state 2, the costs associated with the keep and replace actions are tied at 30; therefore, both actions are optimal.

Next, we move back one more stage to stage 2, where the highest possible state is 3. For all three states, the one-stage costs associated with the keep and replace actions are identical to the ones computed earlier in stage 3. Substitution of these one-stage costs and the relevant $V_3(i)$'s from the stage-3 table above into the recurrence relation

$$V_2(i) = \min\left[c(i) + V_3(i+1),\ p - t(i) + c(0) + V_3(1)\right]$$

yields the table below.

Stage 2:

i	Actions: Keep	Actions: Replace	$V_2(i)$	Optimal Action
1	$20+30=50$	$40+5=45$	45	Replace
2	$40+35=75$	$50+5=55$	55	Replace
3	$60+40=100$	$55+5=60$	60	Replace

It follows that we should replace the car in service regardless which state we happen to be in within this stage.

Finally, in stage 1, the only state is 2. Substitution of $c(2) = 40$, $p - t(2) + c(0) = 60 - 20 + 10 = 50$, $V_2(1) = 45$, and $V_2(3) = 60$ into the recurrence relation

$$V_1(2) = \min\left[c(2) + V_2(3),\ p - t(2) + c(0) + V_2(1)\right]$$

yields the table below.

Stage 1:

i	Actions: Keep	Actions: Replace	$V_1(i)$	Optimal Action
2	$40+60=100$	$50+45=95$	95	Replace

Since $V_1(2) = 95$, we conclude that the minimal total net cost from year 1 to the end of year 3, starting with a 2-year-old car in year 1, is 95.

The sequence of optimal actions can be read from the above tables sequentially as follows. An inspection of the stage-1 table shows that we should immediately replace the original

2-year-old car. This implies that the age of the car in service at the start of year 2 will be 1. Next, an inspection of the first row of the stage-2 table shows that we should replace again in year 2. Finally, from the first row of the stage-3 table, we see that we should keep the 1-year-old car at the start of year 3. Thus, the optimal policy prescribes the following sequence of actions: replace, replace, and keep. This completes the solution of our problem.

Discussion

A careful review of our calculations shows that $V_2(2)$, $V_3(3)$, and $V_4(4)$ are never invoked in the recursive computation. Therefore, if one strives for absolute economy in computation, then the calculations associated with these three optimal values can be avoided. In practice, however, it may not be desirable to do so, because a characterization of the precise set of states that are actually needed in every stage can be rather cumbersome.

The recursive computation above can also be done visually via a network representation of the problem. The idea is to represent each stage and state combination as a node embedded in the two-dimensional coordinate system, and to represent each action as an arc. Thus, state 2 in stage 1, for example, will be represented by a node at $(1, 2)$; and the keep and replace actions at state 2 in stage 1 will be represented by two arcs that connect $(1, 2)$ with $(2, 3)$ and $(2, 1)$, respectively. Moreover, the one-stage costs associated with these two actions, namely 40 and 50, can be thought of as the travel distances from $(1, 2)$ to $(2, 3)$ and $(2, 1)$, respectively. It follows that our problem is equivalent to that of finding the shortest path between node $(1, 2)$ and any of the nodes $(4, 5)$, $(4, 3)$, $(4, 2)$, and $(4, 1)$ (which have "terminal distances" 0, -10, -15, and -20, respectively) in the resulting network. Hence, the optimal values can be computed directly on the network, in a manner similar to what was done in Figure DP-2. The results are shown in Figure DP-5. You should verify that the optimal values in Figure DP-5 are identical to those presented in the series of tables above.

A little bit of reflection should convince you that, in fact, every dynamic program is conceptually equivalent to an elementary path problem.

For a problem with a large N, it is not difficult to write a computer program, or to use a spreadsheet program such as Excel, to carry out the required calculations on the basis of the recurrence relation.

For additional realism, one may argue that the price of a new car should depend on the time period. In other words, it may be desirable to replace p by a set of p_k's, where p_k is the price of a new car in year k. Such a scenario can be easily accommodated in our solution procedure by revising the recurrence relation to:

$$V_k(i) = \min\left[c(i) + V_{k+1}(i+1),\ p_k - t(i) + c(0) + V_{k+1}(1)\right].$$