1 Formal definition of DP Recursion

The backward recursive equation where t is stage, i is current state, j is next state, and action is x, the more useful form of the deterministic recursive equation is

$$V_t(i) = \min_x [c_{i,x,j} + V_{t+1}(j)], \quad \forall i, \tag{1}$$

To solve any sequential decision making problem with DP one must identify and define the following elements. The main elements of the DP recursive equation are

1. Stage t: For finite horizon problems it is usually time t (but not always true). For infinite horizon problems its always time.

2. State i or j and in general S: Usually its the AVAILABLE resource that need to be allocated at stage t. However, there are problems where the state is the inventory at hand, or the price of an asset in an asset acquisition problem (e.g. stock market, oil prices), and so on. The next state of a system depends on the current state and the action taken in the current state.

3. Action or decision x or a or k: The action (x_t or a_t) taken in (state i at stage t or S_t) that moves the system to state (j at stage $t+1$ or S_{t+1}) under the influence of a exogenous process W. This means S_{t+1} is a function of (S_t, x_t, W_{t+1}).

4. S_{t+1} is a function of (S_t, x_t, W_{t+1}). Define S_{t+1}, which in turn defines the state transition function.

5. Exogenous process W: Its either deterministic or stochastic (uncertainty). For example, in inventory control problems it is the demand.

6. Contribution function (reward or cost): C or c or r and depending on the type of problem you will use $C(i, a, j)$ or $C(S, x)$ or $c_{i,j}$ and so on. This is the immediate contribution of an action taken in a particular state. Also known as the one-step cost or reward function.

7. Value function of a state f or V : The long-run value $f_t(S)$ of being in a state S at stage t. This is used in making decisions in state S.

8. Transition probability p(i,a,j) or P(i,a,j): This is only for stochastic DP where it denotes the probability of transitioning from state i to state j under action a.

9. Objective function: This is the max or min operator that acts on the value function.

A myopic policy does not yield optimal solution. This is because its actions are dependent on the contribution function and not the value function.

In general, the formal recursive equation can be defined as follows,

$$f_t(S_t) = \min_{x_t} [C(S_t, x_t, S_{t+1}) + f_{t+1}(S_{t+1})], \tag{2}$$

where S_{t+1} is a function of (S_t, x_t, W_{t+1}). This is the Bellman's optimality equation for the deterministic and finite horizon case.

If you can define the elements given above and write the recursive formulation for a given problem then you have met the modeling GOAL for the course. Then you are left with finding the solution to the model, which is the second GOAL.

2 Examples in Deterministic DP- Finite Horizon

In this section, several examples are solved in which the recursive equation is adapted to the application [1]. The problems are formulated as shortest or longest path problems. Many of these fall under the umbrella of dynamic resource allocation.

2.1 Resource Allocation Problem

Resource-allocation problems, in with limited resources must be allocated among several activities, are often solved by dynamic programming. To use linear programming to do resource allocation, three assumptions must be made:

1. Assumption 1. The amount of a resource assigned to an activity may be any nonnegative number.
2. Assumption 2. The benefit obtained from each activity is proportional to the amount of the resource assigned to the activity.
3. Assumption 3. The benefit obtained from more than one activity is the sum of the benefits obtained from the individual activities.

Even if assumptions 1 and 2 do not hold, dynamic programming can be used to solve resource-allocation problems efficiently when assumption 3 is valid and when the amount of the resource allocated each activity is a member of a finite set.

2.1.1 Resource Allocation Problem: Investment

Finco has \$6,000 to invest, and three investments are available. If y_t dollars (in thousands) are invested in investment t, then a net present value (in thousands) of $r_t(y_t)$ is obtained, where $r_t(y_t)$'s are as follows:

$$r_1(y_1) = 7y_1 + 2 \quad (y_1 > 0) \tag{3}$$
$$r_2(y_2) = 3y_2 + 7 \quad (y_2 > 0) \tag{4}$$
$$r_3(y_3) = 4y_3 + 5 \quad (y_3 > 0) \tag{5}$$
$$r_1(0) = r_2(0) = r_3(0) = 0 \tag{6}$$

The amount placed in each investment must be an exact multiple of \$1,000. To maximize the net present value obtained from the investments, how should Finco allocate the \$6,000?

Solution:

The return on investment is not proportional to amount invested so LP cannot be used. Define $f_t(d_t)$ to be the maximum net present value that can be obtained by investing d_t in investment t. Since this is the maximum value, it means that d_t is the maximum available for investing in t. Let x_t be the amount out of d_t that was actually invested.

The recursion is now given as

$$f_t(d_t) = \max_{x_t}[r_t(x_t) + f_{t+1}(d_t - x_t)] \tag{7}$$

where $x_t = \{0, 1, \cdots, d_t\}$

Item	wt (lb)	Benefit
1	4	11
2	3	7
3	5	12

Figure 1: Knapsack problem.

2.1.2 Time value of money

1 dollar at time $t+1$ is worth $0 < \beta < 1$ dollar at time t.

The recursion is now given as

$$f_t(d_t) = \max_{x_t}[r_t(x_t) + \beta f_{t+1}(d_t - x_t)] \tag{8}$$

where $x_t = \{0, 1, \cdots, d_t\}$

Multiplying by β converts $f_{t+1}(d_t - x_t)$ to t dollars.

2.1.3 Knapsack Problem

Suppose a $d = 10$-lb knapsack is to be filled with the items listed in Figure 1. To maximize total benefit, how should the knapsack be filled?

Solution:

Define $f_t(d_t)$ be the maximum benefit from a d_t pound knapsack that is filled with items of type t. Let x_t be the quantity of item type t.

For item 3 or stage 3:

$$f_3(d_3) = \max_{x_3}[12x_3] \tag{9}$$

where $5x_3 \leq d_3$ and $0 \leq d_3 \leq d$.

In the space below write the equation for stage 2 and 1.

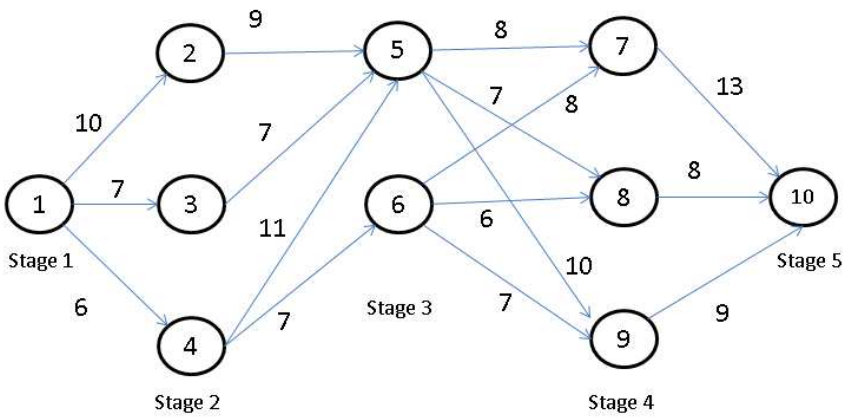

Figure 2: Minimax problem.

2.2 Minimax Problem: Non-additive recursion

Joe Cougar needs to drive from city 1 to city 10. He is no longer interested in minimizing the length of his trip, but he is interested in minimizing the maximum altitude above sea level that he will encounter during his drive. To get from city 1 to city 10, he must follow a path in the figure below. The length c_{ij} of the arc connecting city i and city j represents the maximum altitude (in thousands of feet above sea level) encountered when driving from city i to city j. Use dynamic programming to determine how Joe should proceed from city 1 to city 10. See Figure 2.

Solution:

Define $f_t(i)$ as the smallest maximum altitude that Joe can encounter from city i in stage t.

$$f_t(i) = \min_j [max\{c_{ij}, f_{t+1}(j)\}], i < j \tag{10}$$

2.3 Equipment replacement problem

An auto repair shop always needs to have an engine analyzer available. A new engine analyzer costs \$1,000. the cost m_i of maintaining an engine analyzer during its i^{th} year of operation is as follows:

$$m_1 = \$60 \tag{11}$$
$$m_2 = \$80 \tag{12}$$
$$m_3 = \$120 \tag{13}$$

An analyzer may be kept for 1,2, or 3 years; after i years of use (i=1,2,3), it may be traded in for a new one. If an i-year old engine analyzer is traded in, a salvage value s_i is obtained where:

$$s_1 = \$800 \tag{14}$$
$$s_2 = \$600 \tag{15}$$
$$s_3 = \$500 \tag{16}$$

Given that a new machine must be purchased now (time 0), the shop wants to determine a replacement and trade-in policy that minimizes net costs = (maintenance costs) + (replacement costs) - (salvage value received) during the next 5 years.

Solution:

Hint: Define $f_t(x)$ to be the minimum cost incurred from time t to 5 given that at time t the shop has an x year old analyzer.

Alternate solution:

Define i to be time of buying and j to be time of selling. Define $f(i)$ to be the minimum cost incurred from time i to 5 given that a new machine is purchased at time i

$$f(i) = \min_j [c_{ij} + f(j)],\ i < j \tag{17}$$

where

$$c_{ij} = 1000 + m_1 + \cdots + m_{j-i} - s_{j-i} \tag{18}$$

References

[1] W. L. Winston. *Introduction to Mathematical Programming, Vol 1.* Thompson, 2003.