Characteristics of Dynamic Programming Problems

Problem can be divided into stages with a policy decision required at each stage.

1. Each stage has a set of states associated with the beginning of the stage.

2. The effect of the policy decision at each stage is to transform the current state to a state associated with the beginning of the next stage.

3. The solution procedure finds an optimal policy for the overall problem.
.
4. *The Principle of Optimality*: An optimal policy for the remaining stages is independent of the policy decisions adopted at previous stages.

5. Solution procedure begins by finding optimal policy for the last stage.

6. A recursive relationship that identifies the optimal policy for stage *n* given the optimal policy for *n*+1 is known.

$$f_n^*(s) = \min_{x_n}\left[C_{sx_n} + f_{n+1}^*(x_n) \right]$$

$$f_n^*(s) = \min_{x_n}\left[C_{sx_n} + f_{n+1}^*(x_n) \right]$$

N = number of stages n = label for current stage (n = 1,2,..,N) s_n = current state for stage n	x_n = decision variable for stage n x_n^* = optimal value for x_n given s_n

$f_n(s_n, x_n)$ = contribution of stages n, n+1,…,N to objective function if the system starts in state s_n at stage n, we make decision x_n, and we make optimal decision at all future stages.

7. Use the recursive relationship to work backward stage by stage. Construct a table at each stage:

s_n \ x_n	$f_n(s_n, x_n)$	$f_n^*(s_n)$	x_n^*