Dynamic Programming

Recursive Equations

D Nagesh Kumar, IISc

Introduction and Objectives

Introduction

- Recursive equations are used to solve a problem in sequence
- These equations are fundamental to the dynamic programming

Objectives

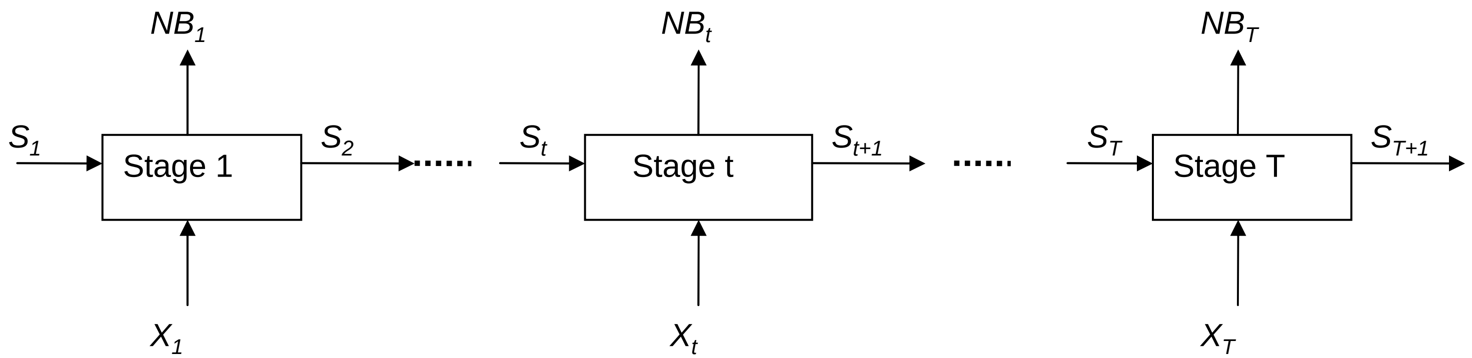
- To formulate recursive equations for a multistage decision process
  - In a backward manner and
  - In a forward manner

Recursive Equations

- Recursive equations are used to structure a multistage decision problem as a sequential process
- Each recursive equation represents a stage at which a decision is required
- A series of equations are successively solved, each equation depending on the output values of the previous equations
- A multistage problem is solved by breaking into a number of single stage problems through recursion
- Approached can be done in a backward manner or in a forward manner

Backward Recursion

- A problem is solved by writing equations first for the final stage and then proceeding backwards to the first stage
- Consider a serial multistage problem



- Let the objective function for this problem is

$$f = \sum_{t=1}^{T} NB_t = \sum_{t=1}^{T} h_t(X_t, S_t)$$

$$= h_1(X_1, S_1) + h_2(X_2, S_2) + \ldots + h_t(X_t, S_t) + \ldots + h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T) \qquad \ldots(1$$

Backward Recursion …contd.

- The relation between the stage variables and decision variables are

  $S_{t+1} = g(X_t, S_t), \quad t = 1,2,\ldots, T.$
- Consider the final stage as the first sub-problem. The input variable to this stage is S_T.
- Principle of optimality: X_T should be selected such that $h_T(X_T, S_T)$ is optimum for the input S_T
- The objective function f_T^* for this stage is

$$f_T^*(S_T) = \underset{X_T}{opt}\left[h_T(X_T, S_T)\right]$$

- Next, group the last two stages together as the second sub-problem. The objective function is

$$f_{T-1}^*(S_{T-1}) = \underset{X_{T-1}, X_T}{opt}\left[h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T)\right]$$

Backward Recursion …contd.

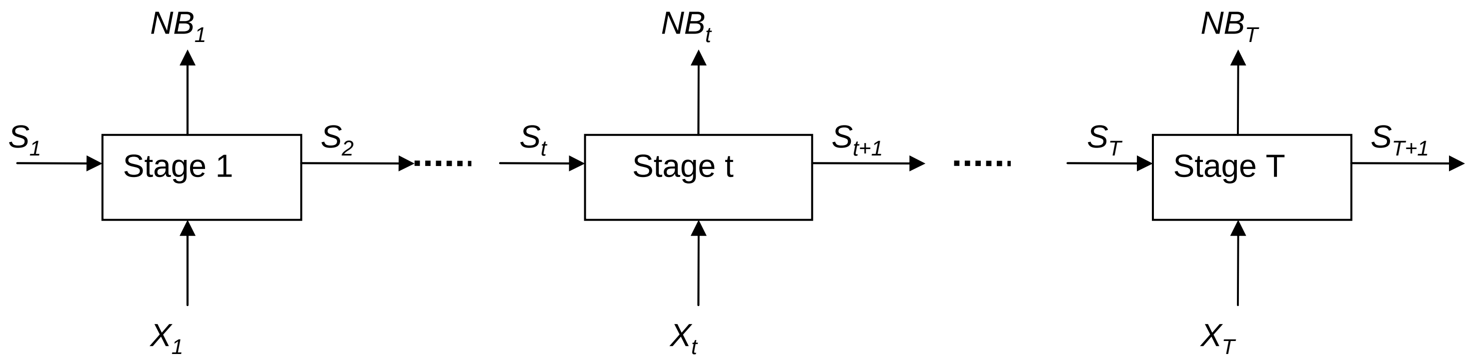
- By using the stage transformation equation, $f_{T-1}^{*}(S_{T-1})$ can be rewritten as
$$f_{T-1}^{*}(S_{T-1}) = \underset{X_{T-1}}{opt}\left[h_{T-1}\left(X_{T-1}, S_{T-1}\right) + f_{T}^{*}\left(g_{T-1}\left(X_{T-1}, S_{T-1}\right)\right)\right]$$
- Thus, a multivariate problem is divided into two single variable problems as shown
- In general, the *i+1*th sub-problem can be expressed as
$$f_{T-i}^{*}(S_{T-i}) = \underset{X_{T-i},\ldots,X_{T-1},X_{T}}{opt}\left[h_{T-i}\left(X_{T-i}, S_{T-i}\right) + \ldots + h_{T-1}\left(X_{T-1}, S_{T-1}\right) + h_{T}\left(X_{T}, S_{T}\right)\right]$$
- Converting this to a single variable problem
$$f_{T-i}^{*}(S_{T-i}) = \underset{X_{T-i}}{opt}\left[h_{T-i}\left(X_{T-i}, S_{T-i}\right) + f_{T-(i-1)}^{*}\left(g_{T-i}\left(X_{T-i}, S_{T-i}\right)\right)\right]$$

Backward Recursion …contd.

- $f^*_{T-(i-1)}$ denotes the optimal value of the objective function for the last *i* stages
- Principle of optimality for backward recursion can be stated as,
  - *No matter in what state of stage one may be, in order for a policy to be optimal, one must proceed from that state and stage in an optimal manner sing the stage transformation equation*

Forward Recursion

- The problem is solved by starting from the stage 1 and proceeding towards the last stage
- Consider a serial multistage problem



- Let the objective function for this problem is

$$f = \sum_{t=1}^{T} NB_t = \sum_{t=1}^{T} h_t(X_t, S_t)$$

$$= h_1(X_1, S_1) + h_2(X_2, S_2) + ... + h_t(X_t, S_t) + ... + h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T) \qquad ...(1$$

Forward Recursion …contd.

- The relation between the stage variables and decision variables are

$$S_t = g'(X_{t+1}, S_{t+1}) \qquad t = 1,2,....,T$$

where S_t is the input available to the stages *1* to *t*

- Consider the stage 1 as the first sub-problem. The input variable to this stage is S_1
- Principle of optimality: X_1 should be selected such that $h_1(X_1, S_1)$ is optimum for the input S_1
- The objective function f_1^* for this stage is

$$f_1^*(S_1) = \underset{X_1}{opt}\left[h_1(X_1, S_1)\right]$$

Backward Recursion …contd.

- Group the first and second stages together as the second sub-problem. The objective function is

$$f_2^*(S_2) = \underset{X_2, X_1}{opt} \left[h_2(X_2, S_2) + h_1(X_1, S_1) \right]$$

- By using the stage transformation equation, $f_2^*(S_2)$ can be rewritten as

$$f_2^*(S_2) = \underset{X_2}{opt} \left[h_2(X_2, S_2) + f_1^*(g_2'(X_2, S_2)) \right]$$

- In general, the *i^th^* sub-problem can be expressed as

$$f_i^*(S_i) = \underset{X_1, X_2, \ldots, X_i}{opt} \left[h_i(X_i, S_i) + \ldots + h_2(X_2, S_2) + h_1(X_1, S_1) \right]$$

Backward Recursion …contd.

- Converting this to a single variable problem

$$f_i^*(S_i) = \underset{X_i}{opt}\left[h_i(X_i, S_i) + f_{(i-1)}^*(g_i'(X_i, S_i))\right]$$

- f_i^* denotes the optimal value of the objective function for the last *i* stages
- Principle of optimality for forward recursion can be stated as,
  - *No matter in what state of stage one may be, in order for a policy to be optimal, one had to get to that state and stage in an optimal manner*

Thank You