Dynamic Programming

Additional Applications: Capacity Expansion & Reservoir Operation

Capacity Expansion Problem

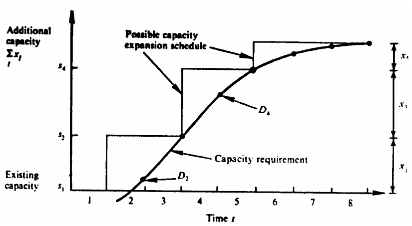
Consider a municipality that must plan for the future expansion of its water supply system or some component of that system, such as a reservoir, aqueduct or treatment plant.

The capacity needed at the end of each period t has been estimated to be D_t.

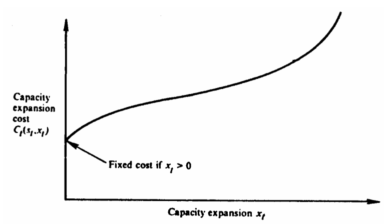
The cost, $C_t(s_t,x_t)$, of adding capacity x_t in each period t is a function of that added capacity as well as of the existing capacity s_t at the beginning of the period.

The planning problem is to find, that time sequence of capacity expansions that minimizes the present value of total future costs and meets the projected requirements.

Capacity Expansion Problem



Capacity Expansion – Cost Function



Capacity Expansion Problem

The planning problem is to find, that time sequence of capacity expansions that minimizes the present value of total future costs and meets the projected requirements.

$$\text{Minimize} \sum_{t=1}^{T} \alpha_t C_t(s_t, x_t)$$

where α_t is the discount factor $(1+r)^{-(t-1)}$.
This discount factor assumes an interest rate of r in each period and that the construction costs are incurred at the beginning of each period.

Capacity Expansion Problem – Contd.

The constraints define each period's final capacity, or equivalently the next period's initial capacity, s_{t+1} as a function of the known existing capacity s_t and each expansion x_t up to period t.

$$s_{t+1} = s_1 + \sum_{\tau=1}^{t} x_\tau \quad \textit{for } t = 1,2,\ldots\ldots,T$$

This may be simply expressed by a series of continuity relationships

$$s_{t+1} = s_t + x_t \quad \textit{for } t = 1,2,\ldots\ldots,T$$

In this problem, the constraints must also ensure that the actual capacity s_{t+1} at the end of each future period t is no less than the capacity required D_t at the end of that period.

$$s_{t+1} \geq D_t \quad \textit{for } t = 1,2,3,\ldots..,T$$

Capacity Expansion Problem – Contd.

There may also be constraints on the possible expansions in each period defined by a set Ω_t, of feasible capacity additions in each period t.

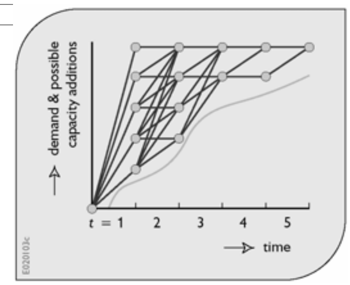
$$x_t \, \varepsilon \, \Omega_t$$

The constrained optimization model stated, can be restructured as a multistage decision-making process and solved using either a forward- or a backward-moving dynamic programming solution procedure.

Stages of model will be the periods in which capacity expansion decisions are made.

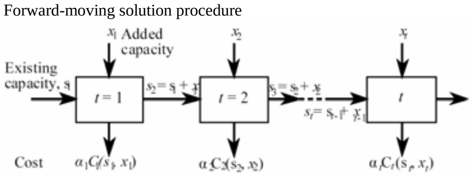
The states will be either the capacity s_{t+1} at the end of stage of period t if a forward-moving solution procedure is adopted, or the capacity s_t at the beginning of a stage or period t if a backward-moving solution procedure is used.

Network of discrete capacity-expansion decisions (links) that meet the projected demand



Solution

Forward-moving solution procedure



To write the recursive equations required for each stage of the forward-moving decision-making process, define the function $f_t(s_{t+1})$, as the minimum present value of the total cost of capacity expansion from the present up to and including period t given a capacity of s_{t+1} at the end of period t.

Recursive Equations

Since at the beginning of the first period t=1, the accumulated least cost is 0, $f_0(s_1)$=0.

For each final discrete state s_2 in stage t=1 ranging from D_1 to the maximum demand D_T, define

$$f_1(s_2) = \alpha_1 C_1(s_1, \; s_2 - s_1)$$

for $D_1 \le s_2 \le D_T$, where D_T is the maximum capacity that should be considered.
Note that the term x_1 in the cost function $C_1(s_1, x_1)$ is expressed in terms of the assumed known state variable s_1 and the specified state variable s_2.
In addition, the final capacity s_2 in period 1 must be no less than the required demand D_1 at the end of period 1. Thus x_1=s_2-s_1.

Above equation must be solved for discrete values of s_2 ranging from the capacity demand D_1 in period 1 to the maximum capacity to be considered D^{max} or D_T.

Recursive Equations – Contd.

For period t=2 the function $f_2(s_3)$ is defined as

$$f_2(s_3) = \underset{x_2 \,\&\, x_2 \varepsilon \Omega_2}{minimum} \left[\alpha_2 C_2(s_3 - x_2, \; x_2) + f_1(s_3 - x_2)\right]$$

Above equation must be solved for discrete values of s_3 ranging from the capacity D_2 to the maximum capacity to be considered D_T.

In general for any period t >1, the recursive equation is

$$f_t(s_{t+1}) = \underset{x_t \,\&\, x_t \varepsilon \Omega_t}{minimum} \left[\alpha_t C_t(s_{t+1} - x_t, \; x_t) + f_{t-1}(s_{t+1} - x_t)\right]$$

This equation must be solved for all $D_t \le s_{t+1} \le D_T$

Recursive Equations – Contd.

For the last period t=T, should be solved for the value of s_{T+1} equal to D_T which minimizes the total cost

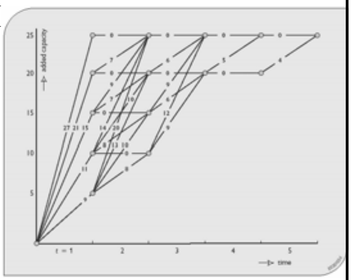
$$f_T(D_T) = \underset{x_T \,\&\, x_T \varepsilon \Omega_T}{minimum} \left[\alpha_T C_T(D_T - x_T, \; x_T) + f_{T-1}(D_T - x_T)\right]$$

A backward moving solution procedure can as well be formulated with similar notation

Capacity Expansion Problem – Example

Consider the five-period capacity-expansion problem

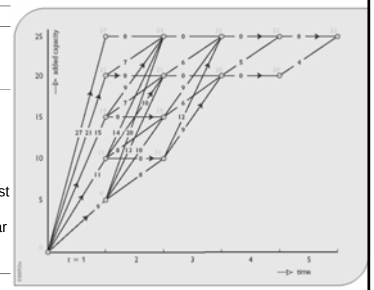
A discrete capacity expansion network showing the present value of the expansion costs associated with each feasible expansion decision. Finding the best path through the network can be done using forward or backward-moving discrete dynamic programming.



Solution - Forward-moving Algorithm

Results of a forward-moving dynamic programming algorithm.

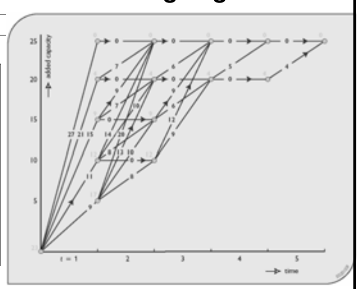
The numbers next to the nodes are the minimum cost to have reached that particular state at the end of the particular time period *t*.



Solution - Backward-moving Algorithm

Results of a backward-moving dynamic programming algorithm.

The numbers next to the nodes are the minimum remaining cost to have the particular capacity required at the end of the planning horizon given the existing capacity of the state.



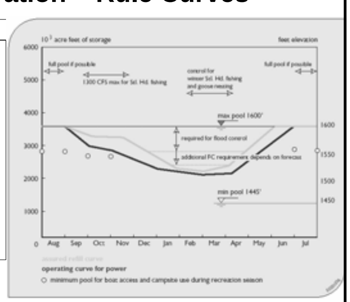
Reservoir Operation

- Reservoir operators need to know how much water to release and when.
- Reservoirs are designed to meet demands for water supplies, recreation, hydropower, the environment and/or flood control. They need to be operated in ways that meet those demands in the most reliable and effective manner.
- Since future inflows or storage volumes are uncertain, the challenge is to determine the best reservoir release or discharge for a variety of possible inflows and storage conditions.
- Reservoir release policies are often defined in the form of what are called 'rule curves'.

Reservoir Operation – Rule Curves

An example reservoir rule curve specifying the storage targets and some of the release constraints, given the particular current storage volume and time of year.

The release constraints also include the minimum and maximum release rates and the maximum downstream channel rate of flow and depth changes that can occur in each month.



Reservoir Operation Problem

Consider a single reservoir having inflows i_t and making releases r_t in each period *t*.

In deterministic problems such as this one, the sequence of inflows i_t is assumed known, and the sequence of releases r_t is to be determined.

Given a known reservoir storage capacity of *K*, the reservoir operating problem involves finding the sequence of releases r_t that maximizes total net benefits. These net benefits may be a function of the storage volume as well as of the release.

Reservoir Operation – Benefit Function

Let s_t be the initial storage volume in period t.

Assume that the net benefits in each period t can be defined as functions of the initial and final storage volumes (s_t and s_{t+1}) and release (r_t), and can be denoted by $NB_t(s_t, s_{t+1}, r_t)$.

Also assume that these net benefit functions for each period t will be the same from one year to the next, at least for the foreseeable future.

Assume that there are T periods in a year.

Benefits associated with storage might stem from hydropower, flood control, lake recreation and the protection of various species of wildlife and their habitats.

Release benefits could result from navigation, water supplies (irrigation, drinking water, industrial use etc.) and waste water dilution.

Reservoir Operation – Optimization

A management objective might be to maximize the total annual benefits.

$$Maximise \sum_{t=1}^{T} NB_t(s_t, s_{t+1}, r_t)$$

The constraints include a mass balance of inflows and outflows or releases in each period t. There are many ways to express this mass balance.

Assuming no significant evaporation or seepage losses, one approach is to equate the final storage volume s_{t+1} in period t (which is same as the initial storage volume in period t+1) to the initial storage volumes s_t plus inflow i_t minus release r_t.

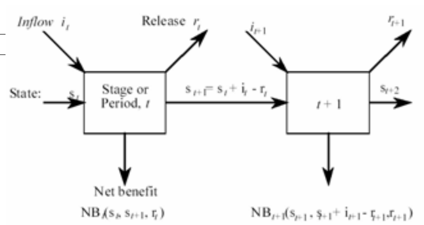
$$s_{t+1} = s_t + i_t - r_t \qquad for \;\; each \;\; period \;\; t$$

Note that if period t is T, then T + 1 is equal to 1, the initial period in the year.

The constraints must also include the capacity restriction K on each storage volume s_t.

$$s_t \le K \qquad for \;\; each \;\; period \;\; t$$

Sequential Reservoir Operation Process



The stages are the time periods, and the states are the storage volumes.

Again, either a forward- or backward-moving sequence of recursive equations can be formulated.

Backward recursion DP is used here.

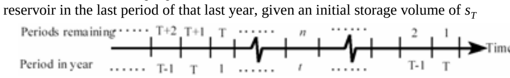
Sequential Reservoir Operation Process

Proceeding backward, a particular period is selected after which it is assumed the reservoir will no longer be operated.

This can be any period in any year, because the eventual "steady-state" optimal policy $\{r_t\}$ derived from the model will be independent of this arbitrary assumption provided that both the average inflows i_t and net benefit functions $NB_t(.)$ in each period t do not change from one year to the next.

Let the arbitrary terminal period be period T. Only one period of operation remains, which is the period on the far right of the time shown in Figure.

Next define a function $f_T^{\,1}(s_T)$ that is the maximum net benefit derived from operating the reservoir in the last period of that last year, given an initial storage volume of s_T



Sequential Reservoir Operation Process

$$f_T^1(s_T) = \underset{r_T \ge 0 \;\&\; r_T \le s_T + i_T \;\&\; r_T \ge s_T + i_T - K}{maximum} \left[NB_T(s_T,\; s_T + i_T - r_T,\; r_T) \right]$$

The constraints on the release r_T limit it to the water available, and force a spill if the available water exceeds the reservoir capacity K.

Above equation must be solved for discrete values of s_T from 0 (or some minimum allowable storage volume in that period) to the maximum possible storage volume K.

These values of $f_T^{\,1}(s_T)$ will be needed to solve the next recursive equation.

Sequential Reservoir Operation Process

Moving backward in time (from right to left in Figure), the next stage is the previous period, T-1. There are now two periods remaining for reservoir operation.

In this case, the function $f^{\,2}_{T-1}(s_{T-1})$ represents the maximum total net benefit with two periods to go, given an initial storage of s_{T-1} in period T-1.

Since $s_T = s_{T-1} + i_{T-1} - r_{T-1}$, $f^{\,1}_{T}(s_T)$ can be expressed in terms of the state variable s_{T-1}, the decision variable r_{T-1}, and the known average inflow i_{T-1}.

$$f_{T-1}^2(s_{T-1}) = \underset{\substack{r_{T-1} \ge 0 \\ r_{T-1} \le s_{T-1} + i_{T-1} \\ r_{T-1} \ge s_{T-1} + i_{T-1} - K}}{Maximum} \left[NB_{T-1}(s_{T-1}, s_{T-1} + i_{T-1} - r_{T-1}, r_{T-1}) + f_T^1(s_{T-1} + i_{T-1} - r_{T-1}) \right]$$

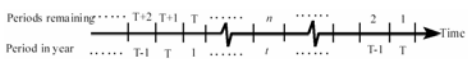
Again, this must be solved for all discrete values of s_{T-1} between 0 and K.

Sequential Reservoir Operation Process

Continuing backward in time, the general recursive equation for each period t with n (n>1) periods remaining can be written.

$$f_t^{\,n}(s_t) = \underset{\substack{r_t \geq 0 \\ r_t \leq s_t + i_t \\ r_t \geq s_t + i_t - K}}{Maximum}\left[NB_t(s_t, s_t + i_t - r_t, r_t) + f_{t+1}^{\,n-1}(s_t + i_t - r_t)\right]$$

The index n proceeds from 2 and increases at each successive stage and the index t cycles backward from period T to 1 and then to period T again.
The relationship between periods t and the index n can be seen in Figure.



Sequential Reservoir Operation Process

Now the question remains, how many recursive equations must be solved to obtain the optimal release policy r_t for each period t associated with each discrete value of the initial storage volumes s_t ?

Usually after proceeding through only three to four years, the optimal release r_t associated with each initial storage volume s_t will be the same as the corresponding r_t and s_t in the previous year.
This is called a **stationary solution**.

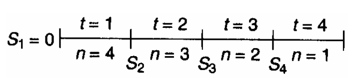
The maximum annual net benefit resulting from this policy will equal $\{f_t^{n+T}(s_t) - f_t^{n}(s_t)\}$ for any value of s_t and t.

One can recognize that indeed the stationary policy has been identified when the values $\{f_t^{n+T}(s_t) - f_t^{n}(s_t)\}$ are independent of s_t and t.

Reservoir Operation using DP Example-1

Inflows during four seasons to a reservoir with storage capacity of 4 units are, respectively, 2, 1, 3, and 2 units. Only discrete values, 0, 1, 2,..., are considered for storage and release. Overflows from the reservoir are also included in the release. Reservoir storage at the beginning of the year is 0 units. Release from the reservoir during a season results in the following benefits which are same for all the four seasons. Obtain the release policy using backward recursion.

Release	*Benefits*
0	-100
1	250
2	320
3	480
4	520
5	520
6	410
7	120



Example-1: Solution

Stage 1: t=4, n=1, Q_4=2

$$f_4^{\,1}(S_4) = \max\,[B_4(R_4)]$$
$$0 \leq R_4 \leq S_4 + Q_4$$
$$S_4 + Q_4 - R_4 \leq 4$$

Release	*Benefits*
0	-100
1	250
2	320
3	480
4	520
5	520
6	410
7	120

S_4	R_4	$B_4(R_4)$	$f_4^{\,1}(S_4)$	R_4^*
0	0	−100	320	2
	1	250		
	2	320		
1	0	−100	480	3
	1	250		
	2	320		
	3	480		
2	0	−100	520	4
	1	250		
	2	320		
	3	480		
	4	520		
3	1	250	520	4, 5
	2	320		
	3	480		
	4	520		
	5	520		
4	2	320	520	4, 5
	3	480		
	4	520		
	5	520		
	6	410		

Example-1: Solution Contd.

Release	*Benefits*
0	-100
1	250
2	320
3	480
4	520
5	520
6	410
7	120

Stage 2: t=3, n=2, Q_3=3

$$f_3^{\,2}(S_3) = \max\,[B_3(R_3) + f_4^{\,1}(S_3 + Q_3 - R_3)]$$
$$0 \leq R_3 \leq S_3 + Q_3$$
$$S_3 + Q_3 - R_3 \leq 4$$

S_3 (1)	R_3 (2)	$B_3(R_3)$ (3)	$S_3+Q_3-R_3$ (4)	$f_4^{\,1}(S_3+Q_3-R_3)$ (5)	(3)+(5)	$f_3^{\,2}(S_3)$	R_3^*
0	0	−100	3	520	420	800	2, 3
	1	250	2	520	770		
	2	320	1	480	800		
	3	480	0	320	800		
1	0	−100	4	520	420	960	3
	1	250	3	520	770		
	2	320	2	520	840		
	3	480	1	480	960		
	4	520	0	320	840		
2	1	250	4	520	770	1000	3, 4
	2	320	3	520	840		
	3	480	2	520	1000		
	4	520	1	480	1000		
	5	520	0	320	840		

(Contd)

Example-1: Solution Contd.

Stage 2: *contd.*

$$f_3^{\,2}(S_3) = \max\,[B_3(R_3) + f_4^{\,1}(S_3 + Q_3 - R_3)]$$
$$0 \leq R_3 \leq S_3 + Q_3$$
$$S_3 + Q_3 - R_3 \leq 4$$

(Contd)

S_3 (1)	R_3 (2)	$B_3(R_3)$ (3)	$S_3+Q_3-R_3$ (4)	$f_4^{\,1}(S_3+Q_3-R_3)$ (5)	(3)+(5)	$f_3^{\,2}(S_3)$	R_3^*
3	2	320	4	520	840	1040	4
	3	480	3	520	1000		
	4	520	2	520	1040		
	5	520	1	480	1000		
	6	410	0	320	730		
4	3	480	4	520	1000	1040	4, 5
	4	520	3	520	1040		
	5	520	2	520	1040		
	6	410	1	480	890		
	7	120	0	320	440		

Example-1: Solution Contd.

Stage 3: t=2, n=3, Q_2=1

$$f_2^3(S_2) = \max\,[B_2(R_2) + f_3^2(S_2+Q_2-R_2)]$$
$$0 \le R_2 \le S_2 + Q_2$$
$$S_2 + Q_2 - R_2 \le 4$$

S_2 (1)	R_2 (2)	$B_2(R_2)$ (3)	$S_2+Q_2-R_2$ (4)	$f_3^2(S_2+Q_2-R_2)$ (5)	(3)+(5)	$f_2^3(S_2)$	R_2^*
0	0	–100	1	960	860	1050	1
	1	250	0	800	1050		
1	0	–100	2	1000	900	1210	1
	1	250	1	960	1210		
	2	320	0	800	1120		
2	0	–100	3	1040	940	1280	2, 3
	1	250	2	1000	1250		
	2	320	1	960	1280		
	3	480	0	800	1280		
3	0	–100	4	1040	940	1440	3
	1	250	3	1040	1290		
	2	320	2	1000	1320		
	3	480	1	960	1440		
	4	520	0	800	1320		
4	1	250	4	1040	1290	1480	3, 4
	2	320	3	1040	1360		
	3	480	2	1000	1480		
	4	520	1	960	1480		
	5	520	0	800	1320		

Example-1: Solution Contd.

Stage 4: t=1, n=4, Q_1=2

$Q_1 = 2$ $f_1^4(S_1) = \max\,[B_1(R_1) + f_2^3(S_1 + Q_1 - R_1)]$

$S_1 = 0$ (given) $0 \le R_1 \le S_1 + Q_1$

$S_1 + Q_1 - R_1 \le 4$

S_1 (1)	R_1 (2)	$B_1(R_1)$ (3)	$S_1+Q_1-R_1$ (4)	$f_2^3(S_1+Q_1-R_1)$ (5)	(3)+(5)	$f_1^4(S_1)$	R_1^*
	0	–100	2	1280	1180		
0	1	250	1	1210	1460	1460	1
	2	320	0	1050	1370		

Trace back:

$R_1^* = 1$, from the last table. $S_2' = S_1 + Q_1 - R_1^* = 0 + 2 - 1 = 1$

$R_2^* = 1$ from Stage 3 table, corresponding to $S_2' = 1$

$S_3' = S_2' + Q_2 - R_2^* = 1 + 1 - 1 = 1$ $R_3^* = 3$ from Stage 2 table.

$S_4' = S_3' + Q_3 - R_3^* = 1 + 3 - 3 = 1$ $R_4^* = 3$ from Stage 1 table.

The optimal release sequence for this problem, thus, is (1,1,3,3) during the four periods. The maximum net benefits that result from this release policy is the optimized objective function value, $f_1^4(S_1) = 1460$ units.

Reservoir Operation using DP Example-2

Consider a reservoir site at which it is desirable to maintain a constant storage volume of 20 and a constant release of 25. Assume that the capacity K of the reservoir is 30 and that the inflows i_t are 10, 50, and 20 in three distinct seasons (t = 1,2,3), respectively. Desired is an operating policy that minimizes the annual sum of squared deviations from these desired storage volume and release values or targets.

Hence $NB_t(s_t, s_{t+1}, r_t)$ will be equal to $[(20 - s_t)^2 + (25 - r_t)^2]$.
In the above form, NB is not net benefits but the squared deviations from the targets. Therefore NB should be minimized (modeled losses).
Or consider negative of sum of squared deviations for maximization.

Let s_t take on the discrete values 0, 10, 20, and 30,
and r_t the discrete values of 10, 20, 30, and 40.

Solution

$$f_T^1(s_T) = \underset{r_T \ge 0\ \&\ r_T \le s_T + i_T\ \&\ r_T \ge s_T + i_T - K}{\text{Minimize}} [NB_T(s_T, s_T + i_T - r_T, r_T)]$$

Initial Storage, s_t	n=1 $f_3(s_3)$	n=1 r_3^*	n=2 $f_2(s_2)$	n=2 r_2^*	n=3 $f_1(s_1)$	n=3 r_1^*
0	425	20	450	30	1075	10
10	125	20, 30	250	30	575	10, 20
20	25	20, 30	350	40	275	20
30	125	20, 30	-	a	375	30

$$f_{T-1}^2(s_{T-1}) = \underset{\substack{r_{T-1} \ge 0 \\ r_{T-1} \le s_{T-1} + i_{T-1} \\ r_{T-1} \ge s_{T-1} + i_{T-1} - K}}{\text{Minimize}} \left[NB_{T-1}(s_{T-1}, s_{T-1} + i_{T-1} - r_{T-1}, r_{T-1}) + f_T^1(s_{T-1} + i_{T-1} - r_{T-1})\right]$$

* Requires $r_t \ge 50$, hence infeasible.

s_t	n=4 $f_3(s_3)$	n=4 r_3^*	n=5 $f_2(s_2)$	n=5 r_2^*	n=6 $f_1(s_1)$	n=6 r_1^*
0	1200	10	725	30	1350	10
10	600	10	525	30	850	10, 20
20	300	20	625	40	550	20
30	400	30	-	*	650	30

Solution Contd.

s_t	n=7 $f_3(s_3)$	n=7 r_3^*
0	1475	10
10	875	10
20	575	20
30	675	30

STATIONARY POLICY

s_t	OPTIMAL RELEASES r_1^*	r_2^*	r_3^*
0	10	30	10
10	10, 20	30	10
20	20	40	20
30	30	-	30

Stationary Solution
At this stage, not only the releases r_t^* are the same in each succeeding year but also the difference $\{f_t^{n+3}(s_t) - f_t^n(s_t)\}$ is a constant, 275 for all s_t and t.

This value (275) is the minimum annual sum of squared deviations that can be obtained by following the derived sequential operating policy.

The corresponding stationary storage volumes are 20, 10 and 30 for periods 1, 2, and 3, respectively.

Characteristics of DP Problem

- A single n-variable problem is divided into n number of single variable problems. This requires that the objective function of the optimization problem be separable with respect to stages. For example, the function, $R_1(X_1) + R_2(X_2) + ... + R_n(X_n)$ is separable, because we can identify exactly one variable in the objective function associated with each stage. Similarly the function, $X_1.X_2.X_3...X_n$ is also separable. However, the function, $X_1 X_2 + X_2 X_3 + X_3 X_4...$ is not separable and problems with such objective functions cannot be formulated as DP problems.
- Each stage has a number of possible states associated with it. In the water allocation problem, the amount of water available for allocation at a stage defines the state at that stage.
- The policy decision transforms the current state into a state associated with the next stage.
- A recursive relationship identifies the optimal decision at a given stage for a specific state, given the optimal decision for each state at the previous stage.
- A solution moves backward (or forward), stage by stage, till optimal decision for the last stage is found. From this solution, the optimal decisions for other stages are traced back.

Characteristics of DP Problem Contd.

- It is important to examine a major assumption made in each of the problems presented to be solved by DP.
- The net benefits or costs resulting from each decision at each stage of the problem are dependent only on the state variables and are otherwise independent of decisions made at other stages.
- If the returns at any stage are dependent on the decisions made at other stages in a way not captured by the state variables, then dynamic programming is not an appropriate solution technique, except perhaps as a rough approximation.
- For example, DP is not suited for determining the optimal capacity of a reservoir or the optimal target release or storage volumes along with its operating policy because capacity and target decisions affect the constraints on system operation and the net benefit function is not just one, but every time period or stage. For such planning problems other methods, such as linear optimization models, are more successful.

Thank You