CAPACITY EXPANSION & RESERVOIR OPERATION

CAPACITY EXPANSION

INTRODUCTION

The most common applications of dynamic programming in water resources include water allocation, capacity expansion of infrastructure and reservoir operation. In this lecture, dynamic programming formulation for capacity expansion and a numerical example are discussed.

CAPACITY EXPANSION

Consider a municipality planning to increase the capacity of its infrastructure (ex: water treatment plant, water supply system etc) in future. The increments are to be made sequentially in specified time intervals. Let the capacity at the beginning of time period *t* be S_t (existing capacity) and the required capacity at the end of that time period be K_t. Let x_t be the added capacity in each time period. The cost of expansion at each time period can be expressed as a function of S_t and x_t, i.e. $C_t(S_t, x_t)$. The problem is to plan the time sequence of capacity expansions which minimizes the present value of the total future costs subjected to meet the capacity demand requirements at each time period. Hence, the objective function of the optimization model can be written as,

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

where $C_t(S_t, x_t)$ is the present value of the cost of adding an additional capacity x_t in the time period *t* with an initial capacity S_t. Each period's final capacity or next period's initial capacity should be equal to the sum of initial capacity and the added capacity. Also at the end of each time period, the required capacity is fixed. Thus, for a time period *t*, the constraints can be expressed as

$$S_{t+1} = S_t + x_t \qquad \text{for } t = 1,2,\ldots,T$$
$$S_{t+1} \geq K_t \qquad \text{for } t = 1,2,\ldots,T$$

In some problems, there may be constraints to the amount of capacity added x_t in each time period i.e., x_t can take only some feasible values. Thus, $x_t \in \Omega_t$.

The capacity expansion problem defined above can be solved in a sequential manner using dynamic programming. The solution procedure using forward recursion and backward recursion are explained below.

Forward Recursion

Consider the stages of the model to be the time periods in which capacity expansion is to be made and the state to be the capacity at the end of each time period *t*, S_{t+1}. Let S_1 be the present capacity before expansion and $f_t(S_{t+1})$ be the minimum present value of total cost of capacity expansion from present to the time *t*.

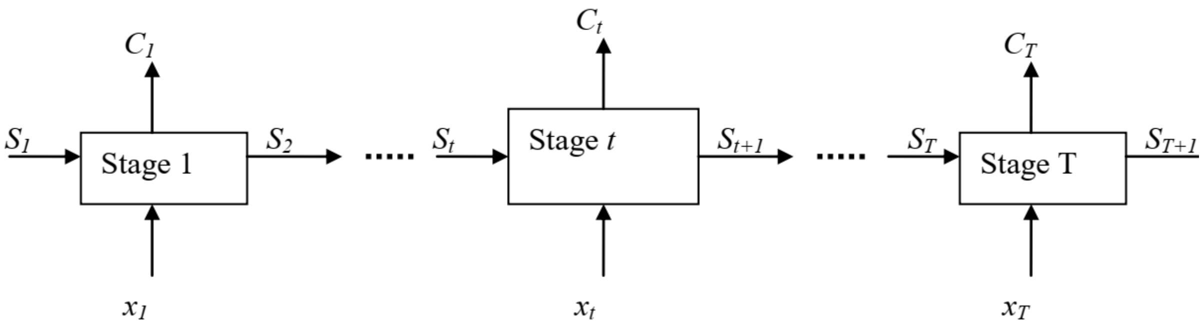

Fig. 1 Stages of model

Considering the first stage, the objective function can be written as,

$$f_1(S_2) = \min \; C_1(S_1, x_1)$$
$$= \min \; C_1(S_1, S_2 - S_1)$$

The values of S_2 can be between K_1 and K_T where K_1 is the required capacity at the end of time period 1 and K_T is the final capacity required. In other words, $f_1(S_2)$ should be solved for a range of S_2 values between K_1 and K_T. Then considering first two stages, the suboptimization function is

$$f_2(S_3) = \min_{\substack{x_2 \\ x_2 \in \Omega_2}} \left[C_2(S_2, x_2) + f_1(S_2) \right]$$
$$= \min_{\substack{x_2 \\ x_2 \in \Omega_2}} \left[C_2(S_3 - x_2, x_2) + f_1(S_3 - x_2) \right]$$

which should be solved for all values of S_3 ranging from K_2 to K_T. Hence, in general for a time period *t*, the suboptimization function can be represented as

$$f_t(S_{t+1}) = \min_{\substack{x_t \\ x_t \in \Omega_t}} \left[C_t\left(S_{t+1} - x_t, x_t\right) + f_{t-1}(S_{t+1} - x_t) \right]$$

with constraint as $K_t \leq S_{t+1} \leq K_T$. For the last stage, i.e. *t*=*T*, the function $f_T(S_{T+1})$ need to be solved only for $S_{T+1} = K_T$.

Backward Recursion

The expansion problem can also be solved using a backward recursion approach with some modifications. Consider the state S_t be the capacity at the beginning of each time period *t*. Let $f_T(S_T)$ be the minimum present value of total cost of capacity expansion in periods *t* through *T*.

For the last period *T*, the final capacity should reach K_T after doing the capacity expansions. Thus, the objective function can be written as,

$$f_T(S_T) = \min_{\substack{x_T \\ x_T \in \Omega_T}} \left[C_T\left(S_T, x_T\right) \right]$$

This is solved for all S_T values ranging from K_{T-1} to K_T.

In general, for a time period *t*, the function $f_t(S_t)$ can be expressed as

$$f_t(S_t) = \min_{\substack{x_t \\ x_t \in \Omega_t}} \left[C_t\left(S_t, x_t\right) + f_{t+1}(S_t + x_t) \right]$$

which should be solved for all discrete values of S_t ranging from K_{t-1} to K_T.

For period 1, the above equation must be solved only for the actual value of S_1 given.

Numerical example (Loucks et al., 1981)

Consider a five stage capacity expansion problem. The minimum capacity to be achieved at the end of each time period is given below.

Table 1

t	K_t
1	5
2	10
3	20
4	20
5	25

The expansion costs for each combination of expansion for each stage are shown in the corresponding links in the form of a figure 2.

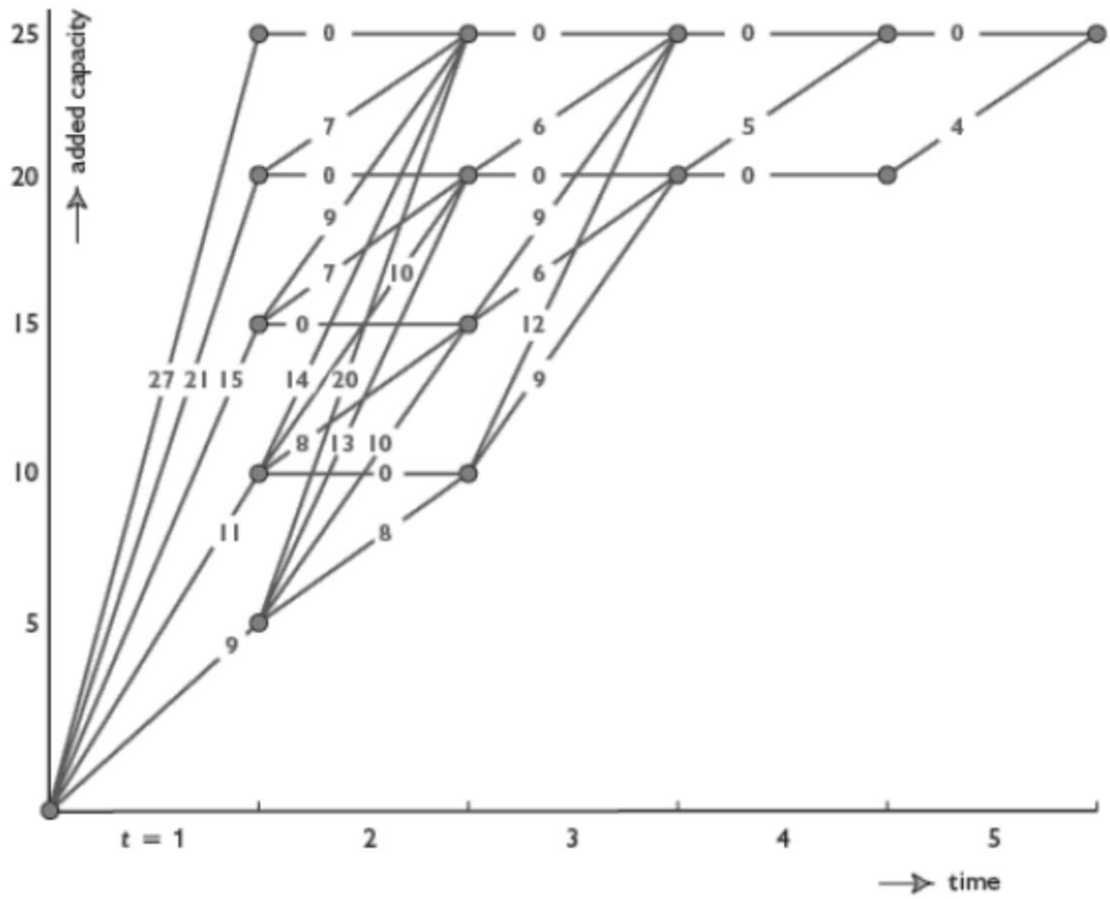

Fig. 2

<u>Solution Using Forward Recursion</u>

The capacity at the initial stage is given as $S_1 = 0$.

Consider the first stage, t =1. The final capacity for stage 1, S_2 can take values between K_1 to K_5. Let the state variable take discrete values of 5, 10, 15, 20 and 25. The objective function for 1st subproblem with state variable as S_2 can be expressed as

$$f_1(S_2) = \min \; C_1(S_1, x_1)$$
$$= \min \; C_1(S_1, S_2 - S_1)$$

The computations for stage 1 are given in the table 2.

Table 2

Stage 1

State Variable, S_2	Added Capacity, $x_1 = S_2 - S_1$	$C_1(S_2)$	$f_1^*(S_2)$
5	5	9	9
10	10	11	11
15	15	15	15
20	20	21	21
25	25	27	27

Considering the 1st and 2nd stages together, the state variable S_3 can take values from K_2 to K_5. Thus, the objective function for 2nd subproblem is

$$f_2(S_3) = \min_{\substack{x_2 \\ x_2 \in \Omega_2}} \left[C_2\left(S_2, x_2\right) + f_1(S_2) \right]$$
$$= \min_{\substack{x_2 \\ x_2 \in \Omega_2}} \left[C_2\left(S_3 - x_2, x_2\right) + + f_1(S_3 - x_2) \right]$$

The value of x_2 should be taken in such a way that the minimum capacity at the end of stage 2 should be 10, i.e. $S_3 \geq 10$.

The computations for stage 2 are given in the table 3.

Table 3

Stage 2

State Variable, S_3	Added Capacity, x_2	$C_2(S_3)$	$S_2 = S_3 - x_2$	$f_1^*(S_2)$	$f_2(S_3)=C_2(S_3)+f_1^*(S_2)$	$f_2^*(S_3)$
10	0	0	10	11	11	11
	5	8	5	9	17	
15	0	0	15	15	15	15
	5	8	10	11	19	
	10	10	5	9	19	
20	0	0	20	21	21	21
	5	7	15	15	22	
	10	10	10	11	21	
	15	13	5	9	22	
25	0	0	25	27	27	24
	5	7	20	21	28	
	10	9	15	15	24	
	15	14	10	11	25	
	20	20	5	9	29	

Like this, repeat this steps till t = 5. For the 5th subproblem, state variable $S_6 = K_5$.

The computations for stages 3 to 5 are shown in tables 4 to 6 below.

Table 4

Stage 3

State Variable, S_4	Added Capacity, x_3	$C_3(S_4)$	$S_3 = S_4 - x_3$	$f_2^*(S_3)$	$f_3(S_4)=C_3(S_4)+f_2^*(S_3)$	$f_3^*(S_4)$
20	0	0	20	21	21	20
	5	6	15	15	21	
	10	9	10	11	20	
25	0	0	25	24	24	23
	5	6	20	21	27	
	10	9	15	15	34	
	15	12	10	11	23	

Table 5

Stage 4

State Variable, S_5	Added Capacity, x_4	$C_4(S_5)$	$S_4 = S_5 - x_4$	$f_3^*(S_4)$	$f_4(S_5)=C_4(S_5)+f_3^*(S_4)$	$f_4^*(S_5)$
20	0	0	20	20	20	20
25	0	0	25	23	23	23
	5	5	20	20	25	

Table 6

Stage 5

State Variable, S_6	Added Capacity, x_5	$C_5(S_6)$	$S_5 = S_6 - x_5$	$f_4^*(S_5)$	$f_5(S_6)=C_5(S_6)+f_4^*(S_5)$	$f_5^*(S_6)$
25	0	0	25	23	23	23
	5	4	20	20	24	

Figure 3 shows the solutions with the cost of each addition along the links and the minimum total cost at each node.

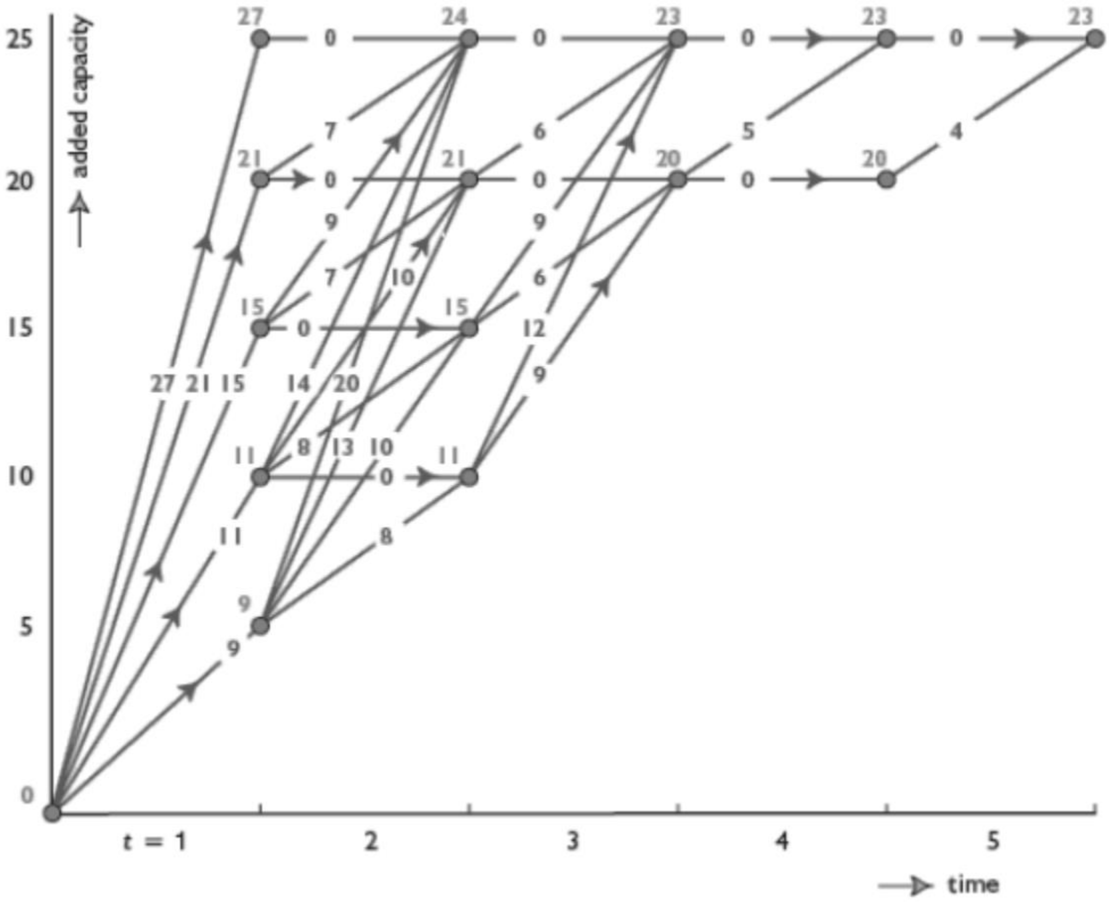

Fig. 3

From the figure, the optimal cost of expansion is 23 units. By doing backtracking from the last stage (farthest right node) to the initial stage, the optimal expansion to be done at 1st stage = 10 units, 3rd stage = 15 units and rest all stages = 0 units.

<u>Solution Using Backward Recursion</u>

The capacity at the final stage is given as S_6 = 25. Consider the last stage, t =5. The initial capacity for stage 5, S_5 can take values between K_4 to K_5. The objective function for 1st subproblem with state variable as S_5 can be expressed as

$$f_5(S_5) = \min_{\substack{x_T \\ x_T \in \Omega_T}} \left[f_5(S_5, x_5) \right]$$

The computations for stage 5 are given in the table 7.

Table 7

Stage 5

State Variable, S_5	Added Capacity, x_5	$C_5(S_5)$	$f_5^*(S_5)$
20	5	4	4
25	0	0	0

Following the same procedure for all the remaining stages, the optimal cost of expansion is achieved. The computations for all stages 4 to 1 are given in Tables 8 to 11 below.

Table 8

Stage 4

State Variable, S_4	Added Capacity, x_4	$C_4(S_4)$	$S_5 = S_4 + x_4$	$f_5^*(S_5)$	$f_4(S_4)=C_4(S_4)+f_5^*(S_5)$	$f_4^*(S_4)$
20	0	0	20	4	4	4
	5	5	25	0	5	
25	0	0	25	0	0	0

Table 9

Stage 3

State Variable, S_3	Added Capacity, x_3	$C_3(S_3)$	$S_4 = S_3 + x_3$	$f_4^*(S_4)$	$f_3(S_3)=C_3(S_3)+f_4^*(S_4)$	$f_3^*(S_3)$
10	10	9	20	4	13	12
	15	12	25	0	12	
15	5	6	20	4	10	10
	10	9	25	0	10	
20	0	0	20	4	4	4
	5	6	25	0	5	
25	0	0	25	0	0	0

Table 10

Stage 2

State Variable, S_2	Added Capacity, x_2	$C_2(S_2)$	$S_3 = S_2 + x_2$	$f_3^*(S_3)$	$f_2(S_2)=C_2(S_2)+f_3^*(S_3)$	$f_2^*(S_2)$
5	5	8	10	12	20	17
	10	10	15	10	20	
	15	13	20	4	17	
	20	20	25	0	20	
10	0	0	10	12	12	12
	5	8	15	10	18	
	10	10	20	4	14	
	15	14	25	0	14	
15	0	0	15	10	10	9
	5	7	20	4	11	
	10	9	25	0	9	
20	0	0	20	4	4	4
	5	7	25	0	7	
25	0	0	25	0	0	0

Table 11

Stage 1

State Variable, S_1	Added Capacity, x_1	$C_1(S_1)$	$S_2 = S_1 + x_1$	$f_2^*(S_2)$	$f_1(S_1)=C_1(S_1)+f_2^*(S_2)$	$f_1^*(S_2)$
0	5	9	5	17	26	23
	10	11	10	12	23	
	15	15	15	9	24	
	20	21	20	4	25	
	25	27	25	0	27	

The solution is given by figure 4 with the minimum total cost of expansion at the nodes.

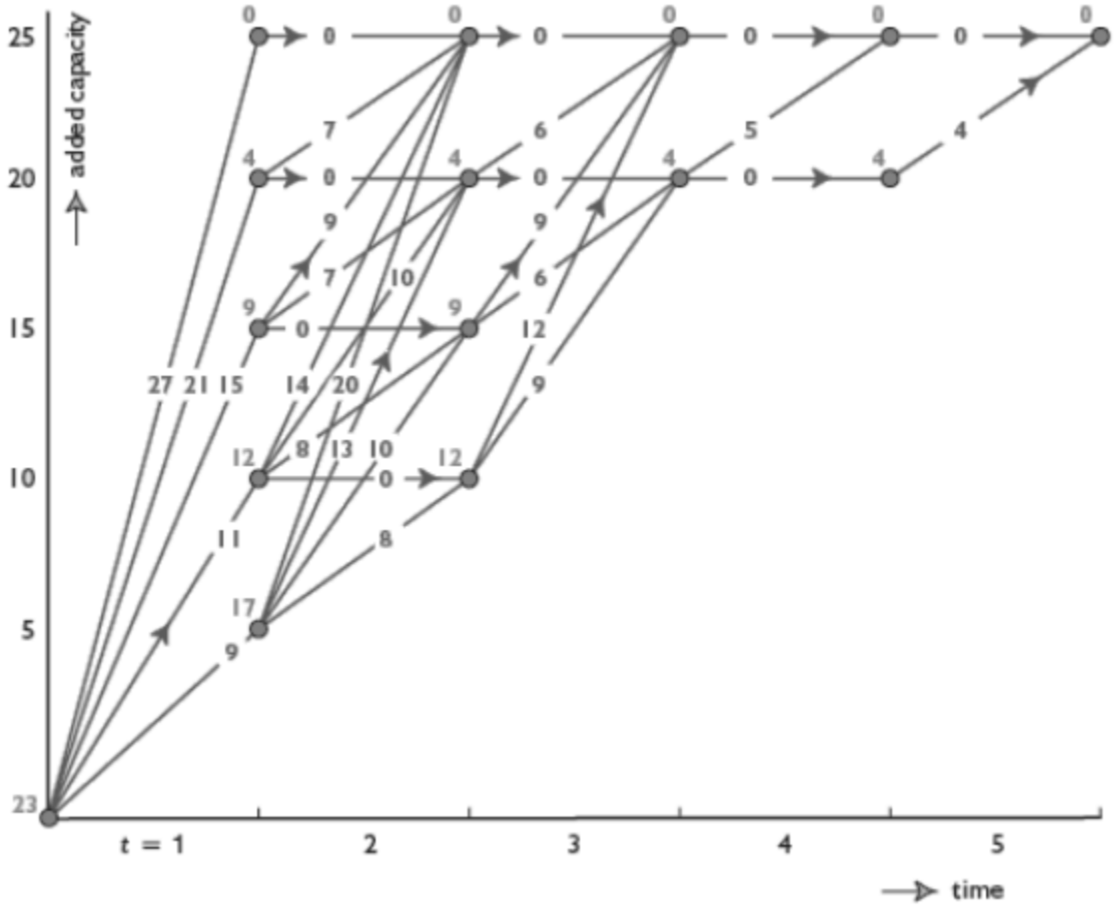

Fig. 4

The optimal cost of expansion is obtained from the node value at the first node i.e., 23 units which is the same as obtained from forward recursion. The optimal expansions at each time period can be obtained by moving forward from the first node to the last node. Thus, the optimal expansion to be made are 10 units at the first stage and 15 units at the last stage. Hence the final requirement of 25 units is achieved.

Although this type of expansion problem can be solved, the future demand and the future cost of expansion are highly uncertain. Hence, the solution obtained cannot be used for making expansions till the end period, *T*. It can be very well used to make decisions about the expansion to be done in the current period. For this to be done, the final period *T* should be selected far away from the current period, so that the uncertainty on current period decisions is much less.

It may be noted that, generally water supply projects are planned for a period of 25-30 years to avoid undue burden to the present generation. In addition, change of value of money in time (due to inflation and other aspects) is not considered in the above examples.

RESERVOIR OPERATION

INTRODUCTION

In the previous lectures, we discussed about the application of dynamic programming in water allocation and capacity expansion of infrastructure. Another major application is in the field of reservoir operation, which will be discussed in this lecture.

RESERVOIR OPERATION – STEADY STATE OPTIMAL POLICY

Consider a single reservoir receiving inflow i_t and making releases r_t for each time period *t*. The maximum capacity of the reservoir is *K*. The optimization problem is to find the sequence of releases to be made from the reservoir that maximizes the total net benefits. These benefits may be from hydropower generation, irrigation, recreation etc. Let S_t and S_{t+1} be the initial and final storages for time period *t*. Expressing net benefits as a function of S_t, S_{t+1} and r_t, the net benefit for period *t* is $NB_t(S_t, S_{t+1}, r_t)$.

If there are *T* periods in a year, then the objective function is to maximize the total net benefits from all periods.

$$\text{Maximize} \sum_{t=1}^{T} NB_t(S_t, S_{t+1}, r_t)$$

This is subject to continuity and also capacity constraints. Neglecting all minor losses like evaporation, seepage etc and assuming that there is no overflow, the continuity relation can be written as,

$$S_{t+1} = S_t + i_t - r_t \qquad for \quad t = 1,2,..T$$

The capacity constraint can be expressed as,

$$S_t \le K \qquad for \quad t = 1,2,..T$$

The above formulated problem can be solved as a sequential process either using forward or backward approach. Here the stages are the time periods and the states are the storage volumes.

Assume that there are *T* periods in a year. In order to find the steady state policy, select a period in a particular year in the near future (to get steady solution). Usually in almost all problems, the last period T is taken as the terminal period. At this stage, the optimal release r_t will be independent of the inflow i_t and also the net benefit NB_t.

Now, consider the terminal period as T of a particular year after which reservoir is no longer useful (figure 1). Solving this problem in a backward recursion method, let t represent the period in a year from T to 1 and n represents the periods remaining from t till end. Thus, t will take values starting from T, decreasing to 1 (which will complete one year) and then again taking a value of T and repeating the values. The value of n starts from 1 (while considering the T^{th} period of last year) and while moving backwards its value keeps on increasing i.e. at the beginning of the last year, the value of $n = T$ and at the beginning of second last year its value will be equal to $2T$ and so on.

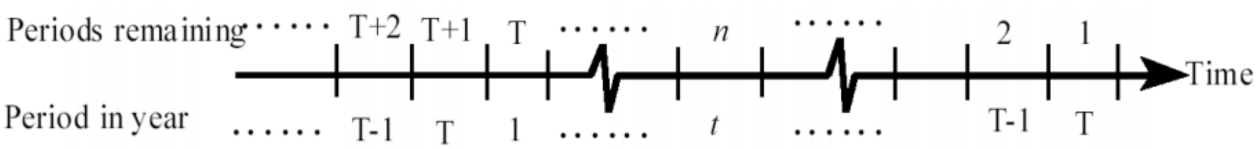

Fig. 1 Sequential process

Starting from T of last year, which is at the far right, there is only one period remaining. Thus, in this case $t=T$ and $n=1$. Let $f_T^1(S_T)$ be the maximum net benefit in the last period of the year considered. $f_T^1(S_T)$ can be expressed as

$$f_T^1(S_T) = \max_{\substack{r_T \geq 0 \\ r_T \leq S_T + i_T \\ r_T \geq S_T + i_T - K}} \left[NB_T\left\{S_T, (S_T + i_T - r_T), r_T\right\} \right]$$

which should be solved for all S_T values from 0 to K.

Considering the last two stages together for which $t=T-1$ and $n=2$, the objective function can be written as

$$f_{T-1}^2(S_{T-1}) = \max_{\substack{r_{T-1} \geq 0 \\ r_{T-1} \leq S_{T-1} + i_{T-1} \\ r_{T-1} \geq S_{T-1} + i_{T-1} - K}} \left[NB_{T-1}\left\{S_{T-1}, (S_{T-1} + i_{T-1} - r_{T-1}), r_{T-1}\right\} + f_T^1(S_{T-1} + i_{T-1} - r_{T-1}) \right]$$

This also is solved for all S_{T-1} values from 0 to K.

In general, for a period t of a particular year with n periods remaining, the function can be written as

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

where the index t decreases from T to 1 and then takes the value T again for the previous year and the cycle repeats while the index n starts from 1 and increases at each successive stage.

This cycle can be repeated till the optimum values of r_t for an initial storage S_t will be the same as the corresponding r_t and S_t of previous year. Such a solution is called stationary solution. The maximum net benefit can be obtained as the difference of $f_t^{n+T}(S_t)$ and $f_t^n(S_t)$ for any S_t and t.

Numerical example (Loucks et al., 1981)

Consider a reservoir for which the desirable constant storage is 20 units and the constant release is 25 units. The capacity of the reservoir is 30 units and the inflows for three seasons are given as 10, 50 and 20 units. The problem is to find the optimum S_t and r_t that minimizes the total squared deviation from the release and storage targets given. Hence, the objective function is $[20 - S_t)^2 + [25 - r_t]^2$. Let S_t take the discrete values of 0, 10, 20, 30 and r_t take the values of 10, 20, 30, 40.

Solution:

Consider a year after which the reservoir is no longer useful. The problem can be expressed as a sequential process as shown in the figure 2.

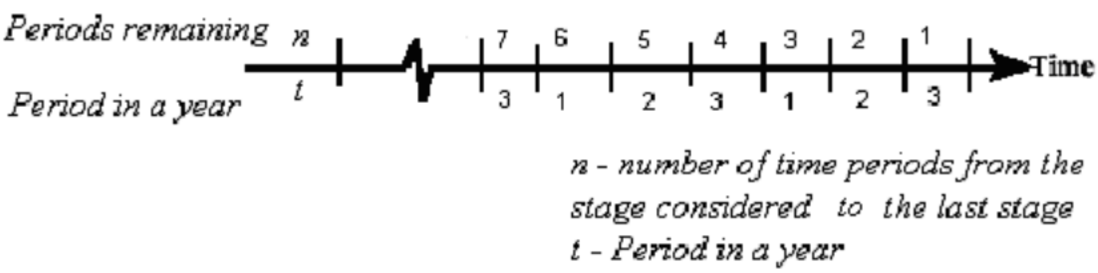

Fig. 2

Here number of seasons (periods), T = 3. Considering the last period for which $t = 3$ *and* $n = 1$, the optimization function is

$$\text{Minimize } f_3^1(S_3) = \left[(20 - S_3)^2 + (25 - r_3)^2\right]$$

Inflow for 3rd season, $I_3 = 20$ units and capacity of the reservoir, $K = 30$ units.

The release constraints can be expressed as

$$r_3 \le S_3 + I_3 \le S_3 + 20 \quad \text{and}$$

$$r_3 \ge S_3 + I_3 - K \ge S_3 + 20 - 30$$

The computation for the first subproblem (n = 1) is shown in the Table 1.

Table 1

State variable, S_3	Release, r_3	$(20 - S_3)^2 + (25 - r_3)$	$f_3^1(S_3)$	Optimal release, r_3^*
0	10	625	425	20
	20	425		
10	10	325	125	20, 30
	20	125		
	30	125		
20	10	225	25	20, 30
	20	25		
	30	25		
	40	225		
30	10	325	125	20, 30
	20	125		
	30	125		
	40	325		

Now considering the last two periods *(n =2)*, the optimization function is

$$\text{Minimize } f_2^2(S_2) = \left[(30 - S_2)^2 + (25 - r_2)^2\right] + f_3^1(S_2 + I_2 - r_2)$$

Inflow for 2nd season, $I_2 = 50$ units.

The release constraints can be expressed as

$$r_2 \le S_2 + 50 \quad \text{and}$$

$$r_2 \ge S_2 + 50 - 30$$

The computation for the second subproblem (n = 2) is shown in the table 2.

For $S_2=30$, $r_2 \ge S_2 + 50 - 30$ *i.e.* $r_2 \ge 50$ i.e. $r_2 \ge 50$. Since r_2 can take values only of 10, 20, 30 and 40 only, the release cannot be made for $S_2=30$.

Table 2

State variable, S_2	Release, r_2	$\left[\begin{matrix}(20-S_2)^2 \\ +(25-r_2)^2\end{matrix}\right]$	$S_2+I_2-r_2$	$f_3^1(S_2+I_2-r_2)$	(5)+(3)	$f_2^2(S_2)$	Optimal release, r_2^*
0	20	425	30	125	550	450	30
	30	425	20	25	450		
	40	625	10	125	750		
10	30	125	30	125	250	250	30
	40	325	20	25	350		
20	40	225	30	125	350	350	40
30	*na*	*na*	*na*	*na*	*na*	*na*	*na*

The same procedure is repeated for all stages till *n* = 7. The summarized solution for this problem is given in the tables 3 to 5.

Table 3

Initial Storage, S_t	*n = 1*		*n = 2*		*n =3*	
	$f_3^1(S_3)$	r_3^*	$f_2^2(S_2)$	r_2^*	$f_1^3(S_1)$	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	--	*na*	375	30

Table 4

Initial Storage, S_t	$n = 4$		$n = 5$		$n = 6$	
	$f_3^4(S_3)$	r_3^*	$f_2^5(S_2)$	r_2^*	$f_1^6(S_1)$	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	--	*na*	650	30

Table 5

Initial Storage, S_t	$n = 7$	
	$f_3^7(S_3)$	r_3^*
0	1475	10
10	875	10
20	575	20
30	675	30

At this stage, the value of r_3^* at $n = 7$ and $n = 4$ are exactly the same. Also the difference $f_3^7(S_3) - f_3^4(S_3) = 275$ is same for all S_t. This value is the minimum total squared deviations from the target release and storage.

Thus, the stationary policy obtained is given in Table 6.

Table 6

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

A main assumption made in dynamic programming is that the decisions made at one stage is dependent only on the state variable and is independent of the decisions taken in other stages. In cases where decisions made at one stage are dependent on the earlier decisions, then dynamic programming will not be an appropriate optimization technique.

BIBLIOGRAPHY / FURTHER READING:

1. Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, N.J, 1957.
2. Hillier F.S. and G.J. Lieberman, *Operations Research*, CBS Publishers & Distributors, New Delhi, 1987.
3. Loucks, D.P., J.R. Stedinger, and D.A. Haith, *Water Resources Systems Planning and Analysis*, Prentice-Hall, N.J., 1981.
4. Rao S.S., *Engineering Optimization – Theory and Practice*, Fourth Edition, John Wiley and Sons, 2009.
5. Taha H.A., *Operations Research – An Introduction*, 8th edition, Pearson Education India, 2008.
6. Vedula S., and P.P. Mujumdar, *Water Resources Systems: Modelling Techniques and Analysis*, Tata McGraw Hill, New Delhi, 2005.