INDIAN INSTITUTE OF SCIENCE

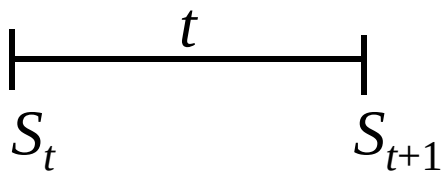
Water Resources Systems:
Modeling Techniques and Analysis

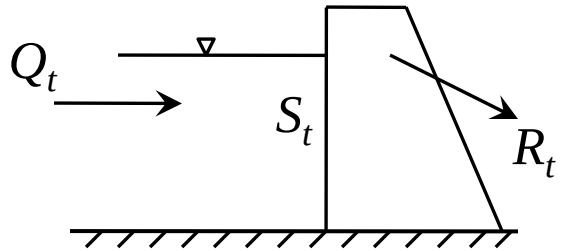
Lecture - 17

Course Instructor : Prof. P. P. MUJUMDAR
Department of Civil Engg., IISc.

Summary of the previous lecture

- Reservoir operation problem





Storage continuity (Mass balance)

$$S_{t+1} = S_t + Q_t - R_t \qquad \text{…... Neglecting losses}$$

where S_t : Storage at the beginning of period t

Q_t : Inflow during period t

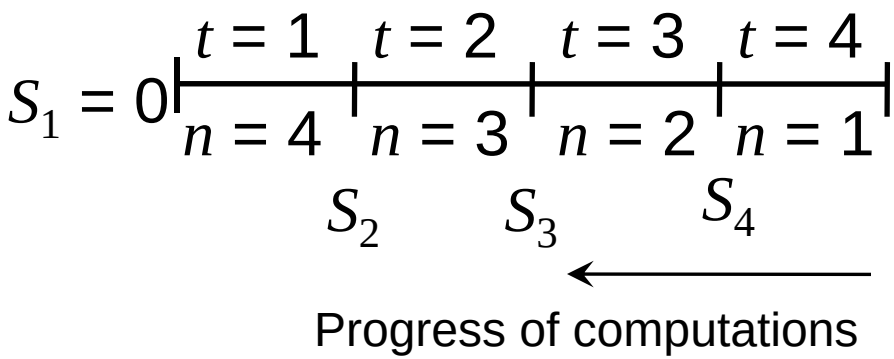
R_t : Release during period t

$$f_t^n(S_t) = \max_{\substack{0 \le R_t \le (S_t + Q_t) \\ S_t + Q_t - R_t \le S_{\max}}} \left[B_t(R_t) + f_{t+1}^{n-1}(S_t + Q_t - R_t) \right]$$

Dynamic Programming

Example:

- Inflows during four seasons to a reservoir with storage capacity of 4 units are 2, 1, 3 and 2 units respectively.



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

Dynamic Programming

Stage 1:

$Q_4 = 2$ $t = 4$ and $n = 1$

$$f_4^1(S_4) = \text{Max}\left[B_4(R_4)\right]$$

$$0 \leq R_4 \leq (S_4 + Q_4)$$

$$S_4 + Q_4 - R_4 \leq 4$$

Dynamic Programming

$Q_4 = 2$

S_4	R_4	$B_4(R_4)$	$f_4^1(S_4) = \text{Max}\left[B_4(R_4)\right]$	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		

Contd.

Dynamic Programming

$Q_4 = 2$
Contd.

S_4	R_4	$B_4(R_4)$	$f_4^1(S_4) = \text{Max}\left[B_4(R_4)\right]$	R_4^*
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		

Dynamic Programming

Stage 2:

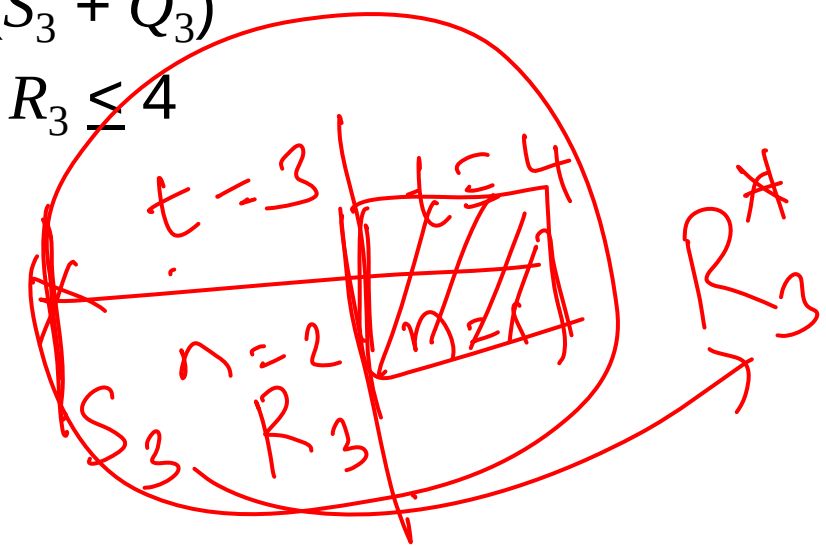
$Q_3 = 3$

$t = 3$ and $n = 2$

$$f_3^2(S_3) = \text{Max}\left[B_3(R_3) + f_4^1(S_3 + Q_3 - R_3) \right]$$

$$0 \le R_3 \le (S_3 + Q_3)$$

$$S_3 + Q_3 - R_3 \le 4$$



Dynamic Programming

$Q_3 = 3$

$$f_3^2(S_3) = \text{Max}\left[B_3(R_3) + f_4^1(S_3 + Q_3 - R_3)\right]$$

$$0 \leq R_3 \leq (S_3 + Q_3)\ ;\ S_3 + Q_3 - R_3 \leq 4$$

S_3	R_3	$B_3(R_3)$	$S_3 + Q_3 - R_3$	$f_4^1(S_3+Q_3-R_3)$	$B_3(R_3) + f_4^1(S_3+Q_3-R_3)$	$f_3^2(S_3)$	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		

Contd.

Contd. $Q_3 = 3$

S_3	R_3	$B_3(R_3)$	$S_3 + Q_3 - R_3$	$f_4^1(S_3+Q_3-R_3)$	$B_3(R_3) + f_4^1(S_3+Q_3-R_3)$	$f_3^2(S_3)$	R_3^*
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		
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		

Dynamic Programming

Stage 3:

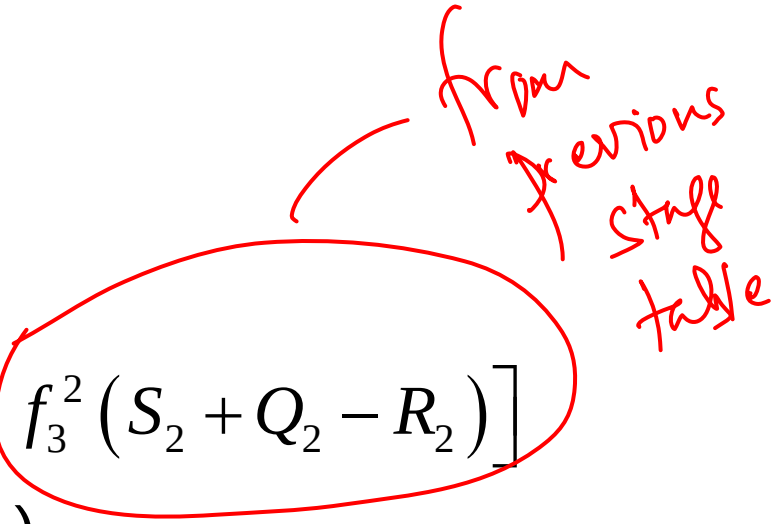
$Q_2 = 1$

$t = 2$ and $n = 3$

$$f_2^3(S_2) = \text{Max}\left[B_2(R_2) + f_3^2(S_2 + Q_2 - R_2) \right]$$

$$0 \le R_2 \le (S_2 + Q_2)$$

$$S_2 + Q_2 - R_2 \le 4$$



Dynamic Programming

$$f_2^3(S_2) = \text{Max}\left[B_2(R_2) + f_3^2(S_2 + Q_2 - R_2)\right]$$

$0 \leq R_2 \leq (S_2 + Q_2)$

$S_2 + Q_2 - R_2 \leq 4$

$Q_2 = 1$

S_2	R_2	$B_2(R_2)$	$S_2 + Q_2 - R_2$	$f_3^2(S_2+Q_2-R_2)$	$B_2(R_2) + f_3^2(S_2+Q_2-R_2)$	$f_2^3(S_2)$	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		

Contd.

Dynamic Programming

$Q_2 = 1$

Contd.

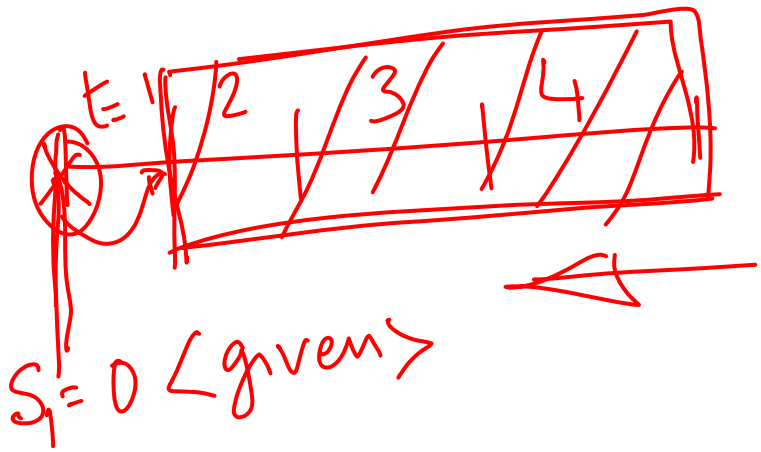
S_2	R_2	$B_2(R_2)$	$S_2 + Q_2 - R_2$	$f_3^2(S_2+Q_2-R_2)$	$B_2(R_2) + f_3^2(S_2+Q_2-R_2)$	$f_2^3(S_2)$	R_2^*
3	0	-100	4	1040	940	1440	3
	1	250	3	1040	1290		
	2	320	2	1000	1320		
	3	480	1	960	1440		
	4	520	0	800	1320		
4	1	250	4	1040	1290	1480	3, 4
	2	320	3	1040	1360		
	3	480	2	1000	1480		
	4	520	1	960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

Dynamic Programming

Stage 4:

$Q_1 = 2$

$t = 1$ and $n = 4$



$$f_1^4(S_1) = \text{Max}\left[B_1(R_1) + f_2^3(S_1 + Q_1 - R_1) \right]$$

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

$$S_1 + Q_1 - R_1 \le 4$$

Dynamic Programming

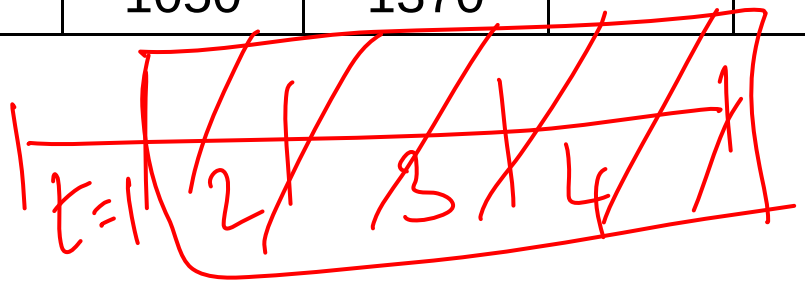
$$f_1^4(S_1) = \text{Max}\left[B_1(R_1) + f_2^3(S_1 + Q_1 - R_1) \right]$$

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

$$S_1 + Q_1 - R_1 \le 4$$

$Q_1 = 2$

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



Dynamic Programming

Trace back:

$R_1^* = 1$ ….. From 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, corresponding to $S_3 = 1$

Dynamic Programming

$$S_4 = S_3 + Q_3 - R_3^*$$

$$= 1 + 3 - 3 = 1$$

$R_4^* = 3$ ….. From stage-1 table, corresponding to $S_4 = 1$

- The optimal release sequence for the problem is {1, 1, 3, 3} during the four periods.
- The maximum net benefits that result from this release policy is 1460 units.

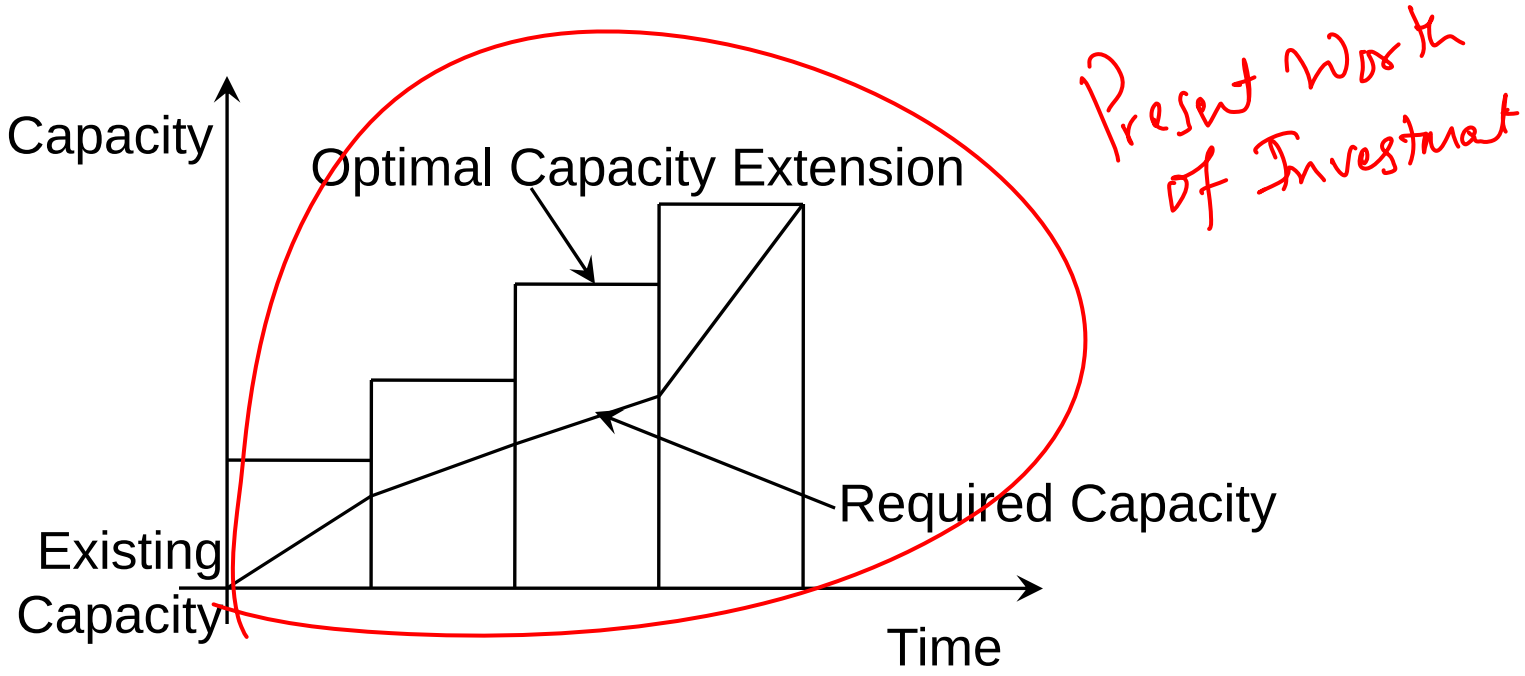
Dynamic Programming

Capacity expansion problem:

- Expansion of existing capacities of an entire system or individual components of a system is often required.
- The decisions deal with investments at different times, starting with the present time, on capacity expansion needed over years in future.
- A typical problem in this class of problems is to decide in what steps the expansion over the next n years should be carried out so that the present worth of the investment is a minimum.

Dynamic Programming

- For example we may need to decide on the investment on capacity expansion during next 5 years, 5 years after that and so on, if the planned capacity is to be achieved after, say, 25 years.



Dynamic Programming

Example:

- A city water supply project envisages expansion of the storage system from the existing capacity of 100 units to 200 units in the next 20 years.
- The additional capacity required at the end of each of the 5 years is as follows

Time	Required additional capacity (units)
End of 5^{th} year	20
End of 10^{th} year	40
End of 15^{th} year	60
End of 20^{th} year	100

Dynamic Programming

- The discounted present worth for additional capacities is as given below

t	Period (Years)	Additional capacity					
		0	20	40	60	80	100
		Discounted present worth of cost (units)					
1	1 – 5	0	120	150	200	250	280
2	6 – 10	0	80	110	130	150	
3	11 – 15	0	60	80	100		
4	15 – 20	0	40	50			

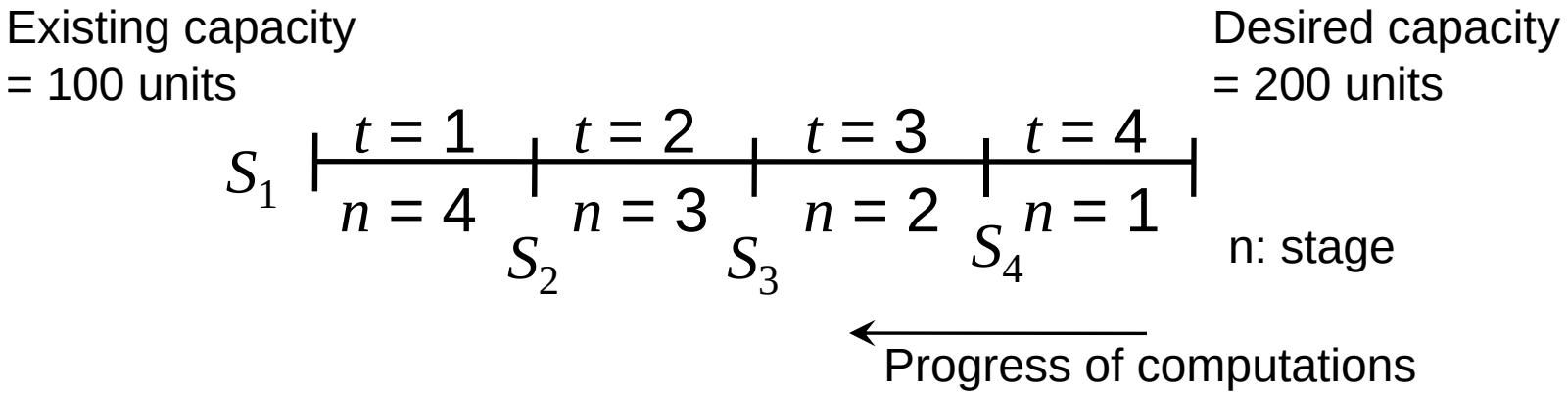
Dynamic Programming

- Since the minimum additional capacity created at the end of the first 5 years is 20, only a maximum of 80 additional units need to be created during the remaining periods.

- The cost of additional capacity is therefore shown only for a maximum of 80 units for years 6 – 10.

- Similarly, for the other years, costs are shown only up to the maximum capacity that still needs to be created till the end of the planning horizon.

Dynamic Programming



- Stage : period during which a decision on capacity expansion is required.
- State variable : capacity at the beginning of a period.
- Decision variable : capacity expansion during a period.

Dynamic Programming

Stage 1: $t = 4$

$$f_4(S_4) = \text{Min}\left[C_4(x_4)\right]$$

$$160 \leq S_4 \leq 200$$

$$S_4 + x_4 = 200$$

S_4	x_4	$C_4(x_4)$	$f_4(S_4) = \text{Min}\left[C_4(x_4)\right]$	x_4^*
160	40	50	50	40
180	20	40	40	20
200	0	0	0	0

Dynamic Programming

Stage 2: $t = 3$

$$f_3\left(S_3\right) = \text{Min}\left[C_3\left(x_3\right) + f_4\left(S_3 + x_3\right)\right]$$

$$140 \leq S_3 \leq 200$$

$$160 \leq S_3 + x_3 \leq 200$$

Dynamic Programming

$$f_3(S_3) = \text{Min}\left[C_3(x_3) + f_4(S_3 + x_3)\right]$$

S_3	x_3	$C_3(S_3)$	S_3+x_3	$f_4(C_3+S_3)$	$C_3(S_3)+$ $f_4(C_3+S_3)$	$f_3(S_3)$	x_3^*
140	20	60	160	50	110	100	60
	40	80	180	40	120		
	60	100	200	0	100		
160	0	0	160	50	50	50	0
	20	60	180	40	100		
	40	80	200	0	80		
180	0	0	180	40	40	40	0
	20	60	200	0	60		
200	0	0	200	0	0	0	0

Dynamic Programming

Stage 3: $t = 2$

$$f_2(S_2) = \text{Min}\left[C_2(x_2) + f_3(S_2 + x_2)\right]$$

$$120 \leq S_2 \leq 200$$

$$140 \leq S_2 + x_2 \leq 200$$

S_2	x_2	$C_2(S_2)$	S_2+x_2	$f_3(C_2+S_2)$	$C_2(S_2)+$ $f_3(C_2+S_2)$	$f_2(S_2)$	x_2^*
120	20	80	140	100	180	150	80
	40	110	160	50	160		
	60	130	180	40	170		
	80	150	200	0	150		

Dynamic Programming

$$f_2(S_2) = \text{Min}\left[C_2(x_2) + f_3(S_2 + x_2)\right]$$

S_2	x_2	$C_2(S_2)$	S_2+x_2	$f_3(C_2+S_2)$	$C_2(S_2)+$ $f_3(C_2+S_2)$	$f_2(S_2)$	x_2^*
140	0	0	140	100	100	100	0
	20	80	160	50	130		
	40	110	180	40	150		
	60	130	200	0	130		
160	0	0	160	50	50	50	0
	20	80	180	40	120		
	40	110	200	0	110		
180	0	0	180	40	40	40	0
	20	80	200	0	80		
200	0	0	200	0	0	0	0

Dynamic Programming

Stage 4: $t = 1$

$$f_1(S_1) = \text{Min}\left[C_1(x_1) + f_2(S_1 + x_1)\right]$$

$S_1 = 100$

$120 \leq S_1 + x_1 \leq 200$

S_1	x_1	$C_1(S_1)$	S_1+x_1	$f_2(C_1+S_1)$	$C_1(S_1)+$ $f_2(C_1+S_1)$	$f_1(S_1)$	x_1^*
100	20	120	120	150	270	250	40, 60
	40	150	140	100	250		
	60	200	160	50	250		
	80	250	180	40	290		
	100	280	200	0	280		

Dynamic Programming

Trace back of the solution:

x_1^*	x_2^*	x_3^*	x_4^*
40	0	60	0

Capacity: 100 140 140 200 200

Alternate solution is:

x_1^*	x_2^*	x_3^*	x_4^*
60	0	0	40

Capacity: 100 160 160 160 200

Optimum present worth of the investment = 250 units.

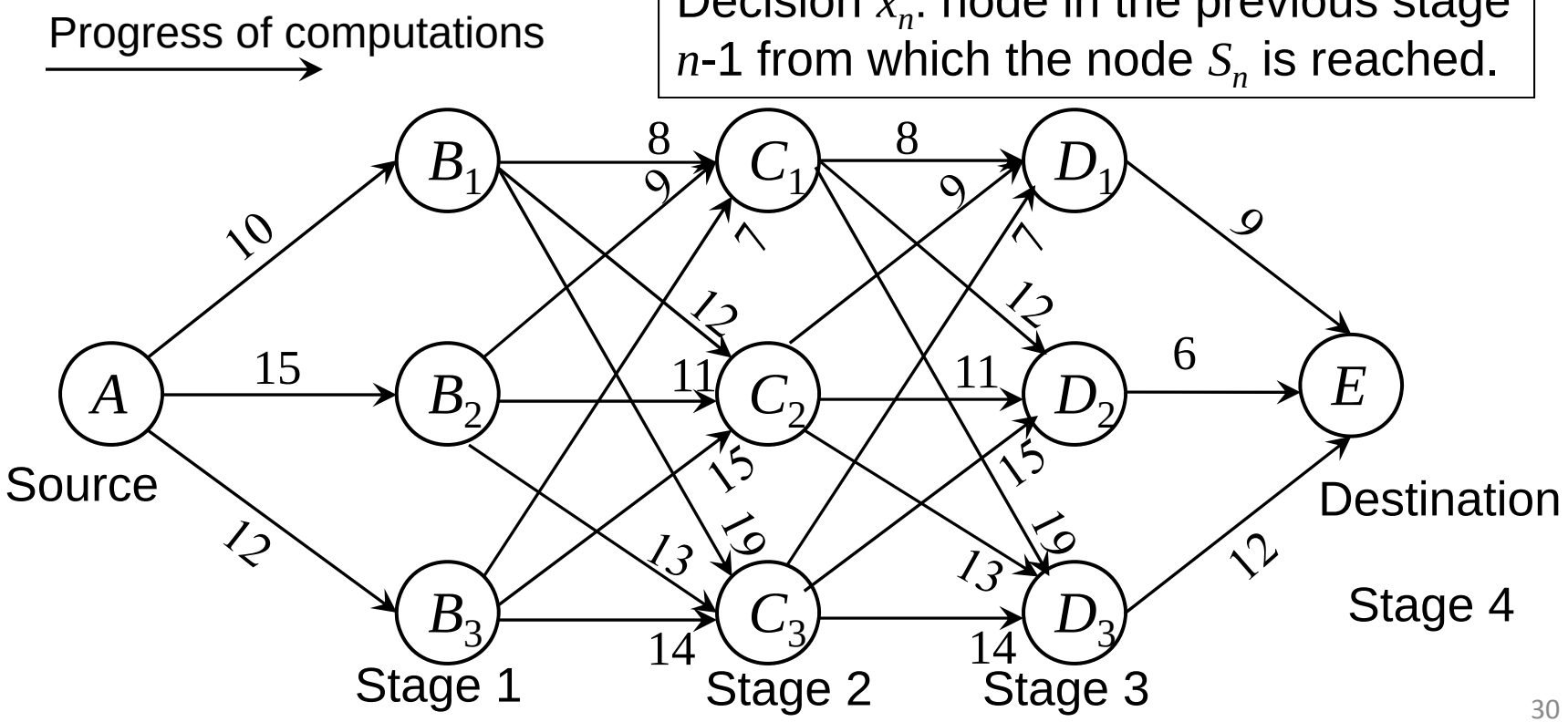
Dynamic Programming

Shortest route problem:

- Consider the problem of determining the shortest route for a pipeline from among various possible routes available from destination to source.

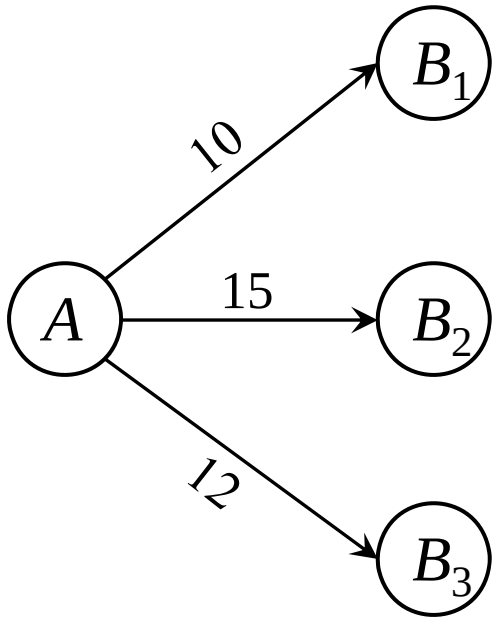
State S_n: node in stage n

Decision x_n: node in the previous stage n-1 from which the node S_n is reached.



Dynamic Programming

Stage 1:



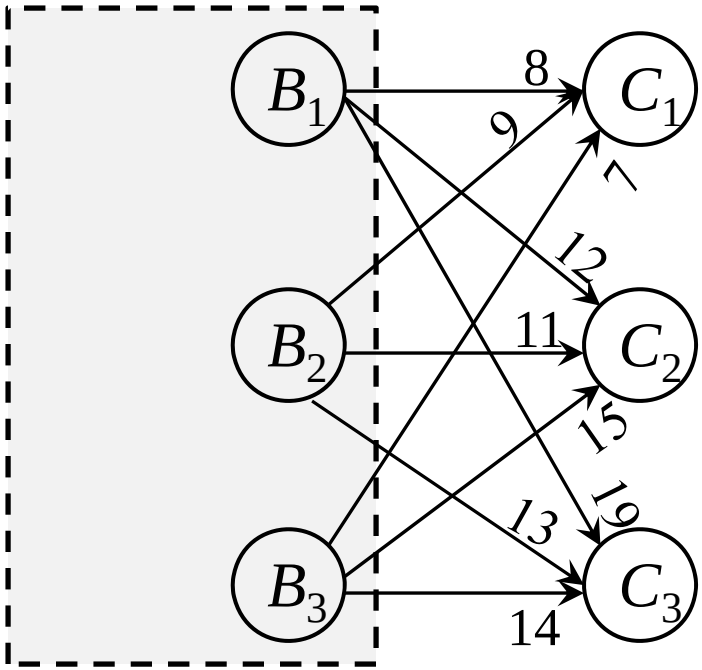
$$f_1^*(S_1) = \text{Min}\left[d(x_1, S_1) \right]$$

Dynamic Programming

S_1	x_1	$f_1^*(S_1) = \text{Min}\left[d(x_1, S_1)\right]$	x_1^*
B_1	10	10	A
B_2	15	15	A
B_3	12	12	A

Dynamic Programming

Stage 2:



$$f_2^*(S_2) = \text{Min}\left[d(x_2, S_2) + f_1^*(x_2) \right]$$

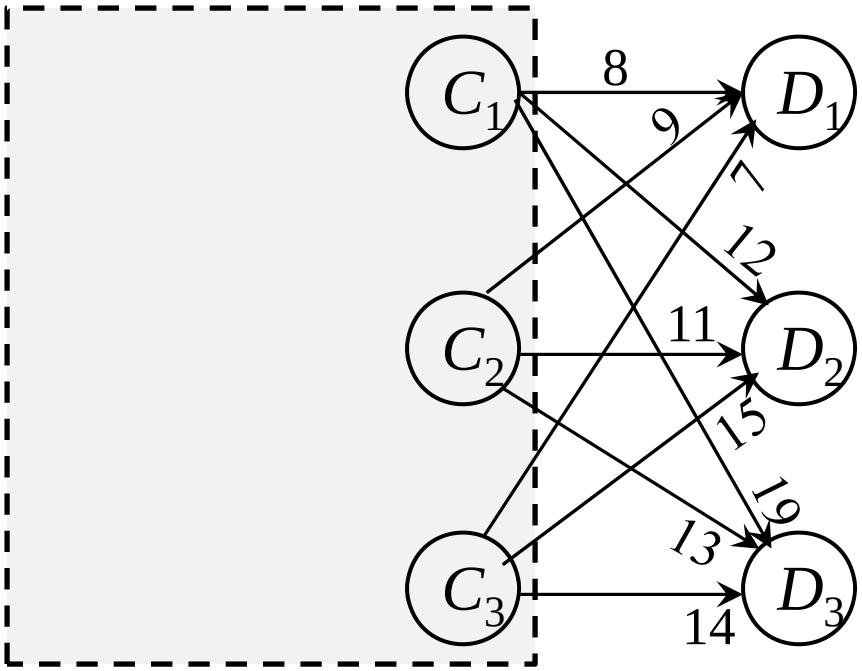
Dynamic Programming

$$f_2^*(S_2) = \text{Min}\left[d(x_2, S_2) + f_1^*(x_2) \right]$$

S_2	x_2	$d(x_2, S_2)$	$f_1^*(x_2)$	$d(x_2, S_2) + f_1^*(x_2)$	$f_2^*(S_2)$	x_2^*
C_1	B_1	8	10	18	18	B_1
	B_2	9	15	24		
	B_3	7	12	19		
C_2	B_1	12	10	22	22	B_1
	B_2	11	15	26		
	B_3	15	12	27		
C_3	B_1	19	10	29	26	B_3
	B_2	13	15	28		
	B_3	14	12	26		

Dynamic Programming

Stage 3:



$$f_3^*(S_3) = \text{Min}\left[d(x_3, S_3) + f_2^*(x_3) \right]$$

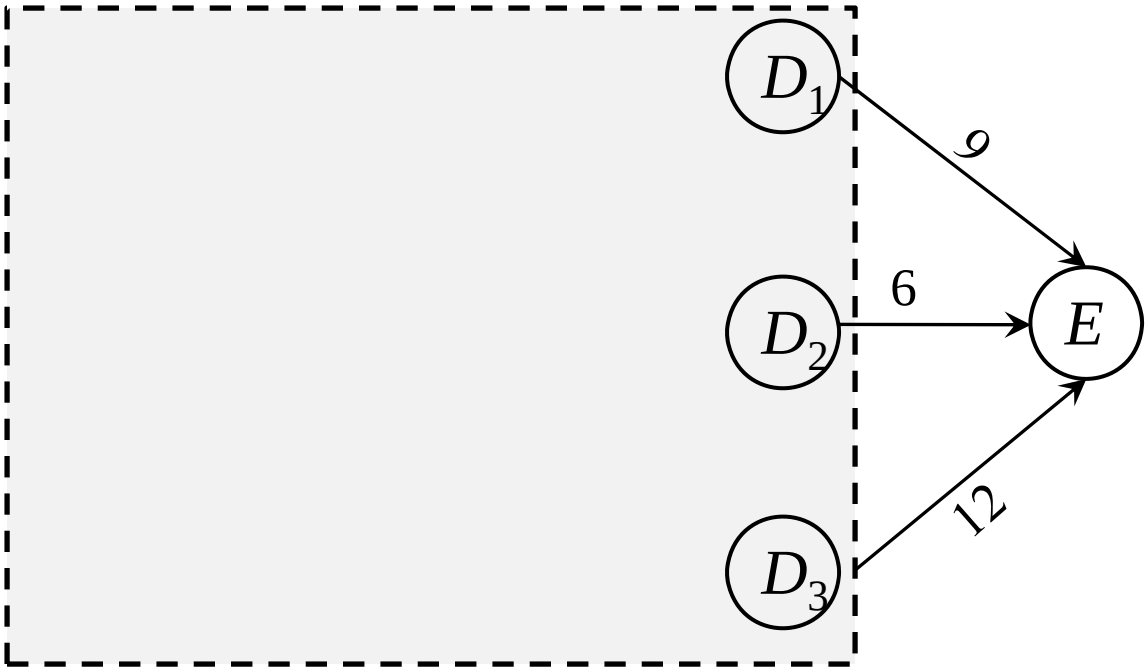
Dynamic Programming

$$f_3^*(S_3) = \text{Min}\left[d(x_3, S_3) + f_2^*(x_3) \right]$$

S_3	x_3	$d(x_3, S_3)$	$f_2^*(x_3)$	$d(x_3, S_3) + f_2^*(x_3)$	$f_3^*(S_3)$	x_3^*
D_1	C_1	8	18	26	26	C_1
	C_2	9	22	29		
	C_3	7	26	33		
D_2	C_1	12	18	30	30	C_1
	C_2	11	22	33		
	C_3	15	26	41		
D_3	C_1	19	18	37	35	C_2
	C_2	13	22	35		
	C_3	14	26	40		

Dynamic Programming

Stage 4:



$$f_4^*(S_4) = \text{Min}\left[d(x_4, S_4) + f_3^*(x_4) \right]$$

Dynamic Programming

$$f_4^*(S_4) = \text{Min}\left[d(x_4, S_4) + f_3^*(x_4) \right]$$

S_4	x_4	$d(x_4, S_4)$	$f_3^*(x_4)$	$d(x_4, S_4) + f_3^*(x_4)$	$f_4^*(S_4)$	x_4^*
E	D_1	9	26	35	35	D_1
	D_2	6	30	36		
	D_3	12	35	47		

Dynamic Programming

Trace back:

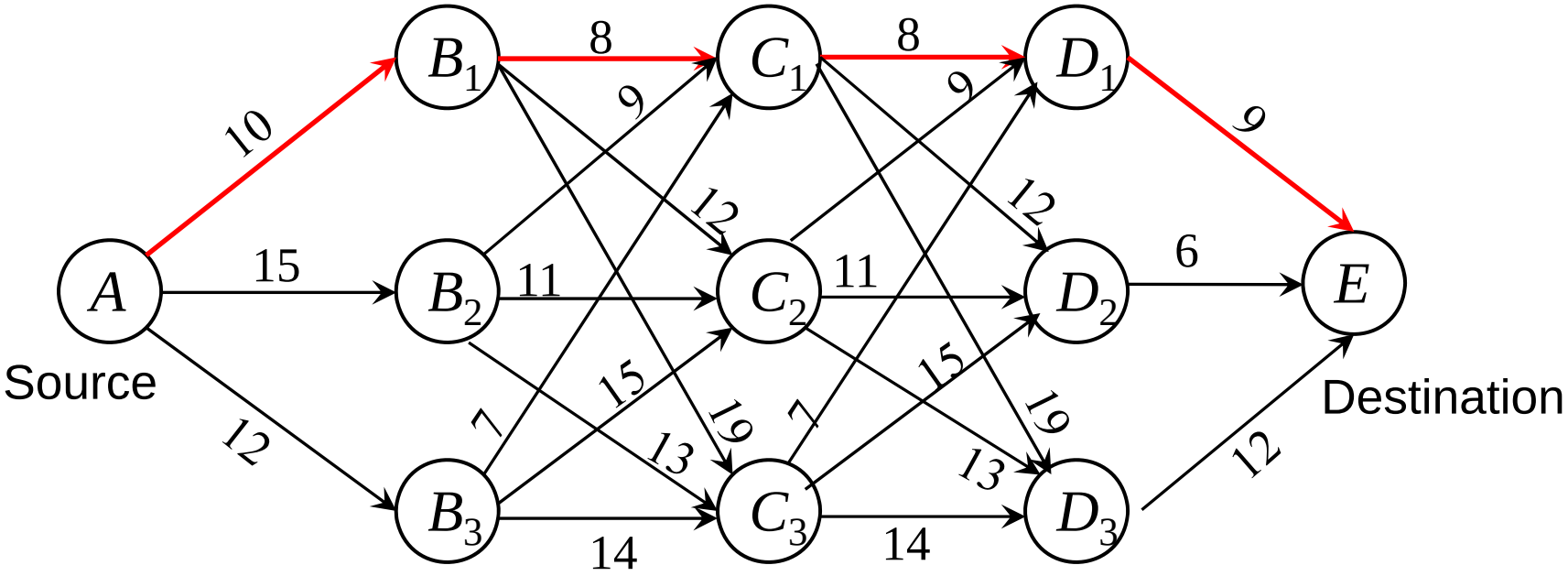
$x_4^* = D_1$ ….. From last table

$x_3^* = C_1$ ….. From stage-3 table, corresponding to $S_3 = D_1$

$x_2^* = B_1$ ….. From stage-2 table, corresponding to $S_2 = C_1$

$x_1^* = A$ ….. From stage-1 table, corresponding to $S_1 = B_1$

Dynamic Programming



Shortest distance = 35 units