Dynamic Programming

- Dynamic programming (DP) is ideally suited for sequential decision problems.
- DP is a mathematical technique well suited for the optimization of multistage decision problems.
- Developed by Richard Bellman in the early 1950's.
- Applications :
  - Reservoir operation, water allocation, capacity expansion, irrigation scheduling, water quality control, shortest route problems etc.

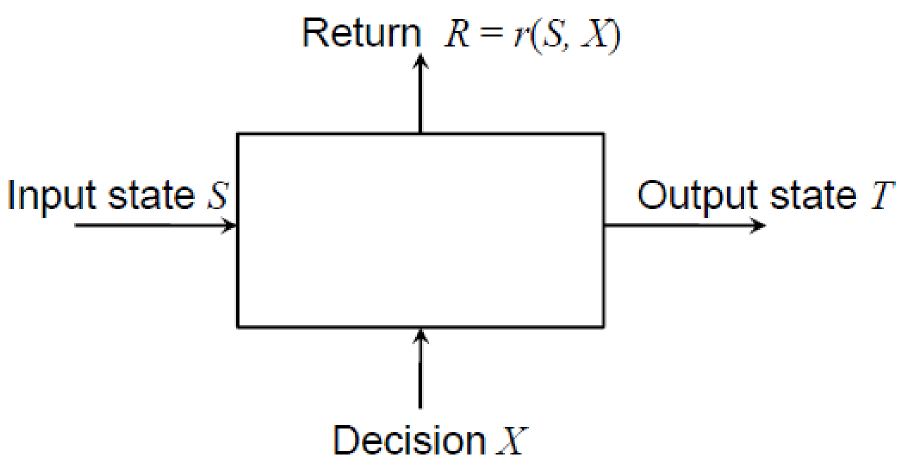
- State of the system
  - Current position, speed and orientation of Missile and Target.

- Decision: Speed and orientation for the Missile during next time interval.

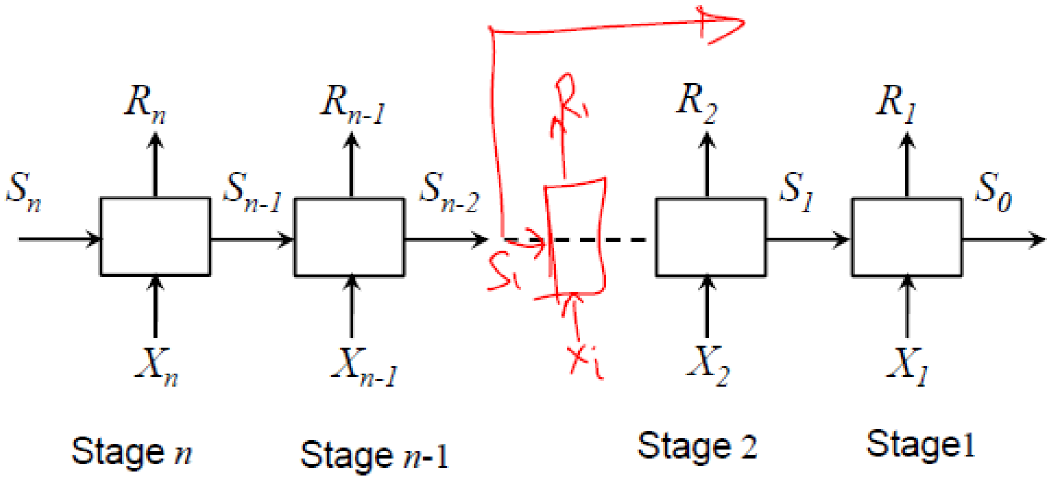
- Objective: To hit the Target in minimum time.

Representation of DP problem:

- Single stage decision problem



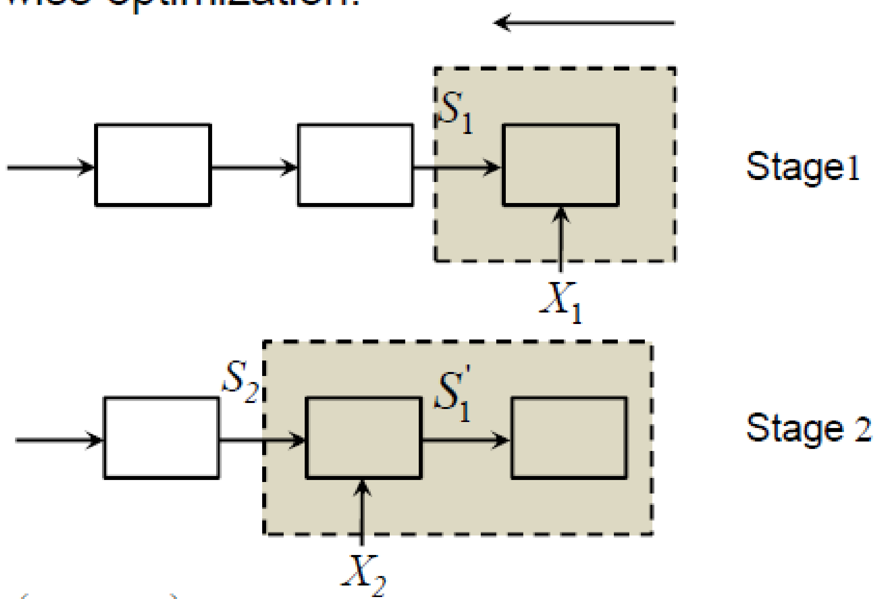
- Serial multi-stage decision problem
  - Output from one state is input to the next



Bellman's principle of optimality:

- "Given the current state of system, the optimal policy (sequence of decisions) for the remaining stages is independent of the policy adopted in the previous stages".
- The principle implies that, given the state S_i of the system at a stage i, one must proceed optimally till the last stage, irrespective of how one arrived at the state S_i.

Stage-wise optimization:

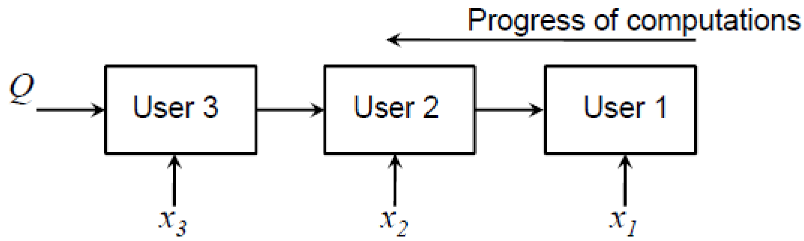


$S_1' = T(S_2, X_2)$ …. State transformation: A function of state variable S_2 and decision X_2 .

Water allocation problem:

- A total of 6 units of water is to be allocated optimally to three users, User 1, User 2 and User 3.

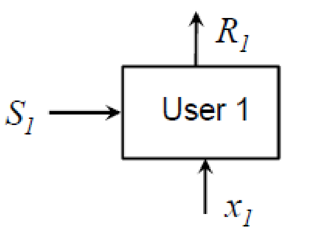
- The allocation is made in discrete steps of one unit ranging from 0 to 6.



- The returns obtained from the users for a given allocation are as follows

Amount of water allocated x	Return from		
	User 3 $R_3(x)$	User 2 $R_2(x)$	User 1 $R_1(x)$
0	0	0	0
1	5	5	7
2	8	6	12
3	9	3	15
4	8	-4	16
5	5	-15	15
6	0	-30	12

Stage 1:



S_i : Amount of Water available to be allocated to all users included in that stage.

$$f_1^*(S_1) = \text{Max}\left[R_1(x_1)\right]$$

$$0 \le x_1 \le S_1$$

$$0 \le S_1 \le Q$$

Total amount available (=6)

S_1 : Amount of water available for allocation to User 1

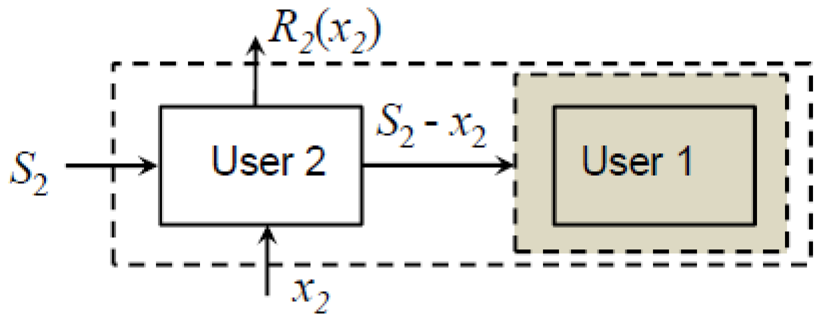
x_1 : Amount of water allocated to User 1

x_1^* : Allocation to User 1, that results in $f_1^*(S_1)$

$f_1^*(S_1)$: Maximum return due to allocation ~~of~~ S_1

S_1	x_1	$R_1(x_1)$	$f_1^*(S_1) = \text{Max}\left[R_1(x_1)\right]$	x_1^*
0	0	0	0	0
1	0	0	7	1
	1	7		
2	0	0	Max 12	2
	1	7		
	2	12		
3	0	0	Max 15	3
	1	7		
	2	12		
	3	15		
4	0	0	16	4
	1	7		
	2	12		
	3	15		
	4	16		
5	0	0	16	4
	1	7		
	2	12		
	3	15		
	4	16		
	5	15		
6	0	0	16	4
	1	7		
	2	12		
	3	15		
	4	16		
	5	15		
	6	12		

Stage 2:



$$f_2^*(S_2) = \text{Max}\left[R_2(x_2) + \underbrace{f_1^*(S_2 - x_2)}\right]$$

$$0 \le x_2 \le S_2$$

$$0 \le S_2 \le Q$$

the optimized value from previous stage.

S_2 : Amount of water available for allocation to User 2 and User 1 together

x_2 : Amount of water allocated to User 2

$S_2 - x_2$: Amount of water available for allocation at stage 1 (to User 1)

$f_2^*(S_2)$: Maximum return due to allocation of S_2

x_2^* : Allocation to User 2, that results in $f_2^*(S_2)$

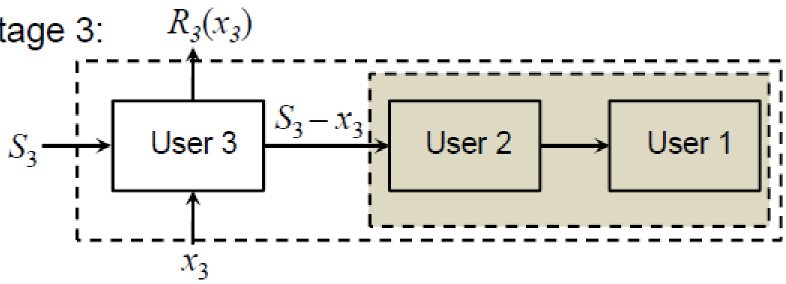
Stage 2

$$f_2^*(S_2) = \text{Max}\left[R_2(x_2) + f_1^*(S_2 - x_2)\right]$$

$$0 \le x_2 \le S_2 \; ; \; 0 \le S_2 \le Q$$

S_2	x_2	$R_2(x_2)$	$S_2 - x_2$	$f_1^*(S_2 - x_2)$	$R_2(x_2) + f_1^*(S_2 - x_2)$	$f_2^*(S_2)$ =MAX	x_2^*
0	0	0	0	0	0	0	0
1	0	0	1	7	7	Max 7	0
	1	5	0	0	5		
2	0	0	2	12	12	12	0, 1
	1	5	1	7	12		
	2	6	0	0	6		
3	0	0	3	15	15	17	1
	1	5	2	12	17		
	2	6	1	7	13		
	3	3	0	0	3		
4	0	0	4	16	16	20	1
	1	5	3	15	20		
	2	6	2	12	19		
	3	3	1	7	10		
	4	-4	0	0	-4		
5	0	0	5	16	16	21	1, 2
	1	5	4	16	21		
	2	6	3	15	21		
	3	3	2	12	15		
	4	-4	1	7	3		
	5	-15	0	0	-15		
6	0	0	6	16	16	22	2
	1	5	5	16	21		
	2	6	4	16	22		
	3	3	3	15	18		
	4	-4	2	12	8		
	5	-15	1	7	-8		
	6	-30	0	0	-30		

Stage 3:



$$f_3^*(S_3) = \text{Max}\left[R_3(x_3) + f_2^*(S_3 - x_3)\right]$$

$$0 \le x_3 \le S_3$$
$$S_3 = Q$$

S_3 : Amount of water available for allocation to User 1, User 2 and User 3 together = 6 units

x_3 : Amount of water allocated to User 3

$S_3 - x_3$: Amount of water available for allocation at stage 2 (to User 1 and User 2 together)

$f_3^*(S_3)$: Maximum return due to allocation of S_3

x_3^* : Allocation to User 3, that results in $f_3^*(S_3)$

Stage 3

$$f_3^*(S_3) = \text{Max}\left[R_3(x_3) + f_2^*(S_3 - x_3)\right]$$

$$0 \le x_3 \le S_3$$
$$S_3 = Q$$

S_3	x_3	$R_3(x_3)$	$S_3 - x_3$	$f_2^*(S_3 - x_3)$	$R_3(x_3) + f_2^*(S_3 - x_3)$	$f_3^*(S_3)$	x_3^*
	0	0	6	22	22		
	1	5	5	21	26		
	2	8	4	20	28	Max	
6	3	9	3	17	26	28	2
	4	8	2	12	20		
	5	5	1	5	10		
	6	0	0	0	0		

- When the third stage is solved, all the three users are considered for allocation; thus the total maximum return is

$$f_3^*(6) = 28$$

- The allocations to individual users are traced back.

From the table for stage 3,

$$x_3^* = 2$$

From this the water available for stage 2 is obtained as,

$$S_2 = Q - x_3^* = 6 - 2 = 4$$

From the table for stage 2, with the value of S_2 = 4 ,

$$x_2^* = 1$$

From this the amount of water available for allocation at stage 1 is obtained as,

$$S_1 = S_2 - x_2^* = 4 - 1 = 3$$

From the table for stage 1, with the value of S_1 = 3 ,

$$x_1^* = 3$$

Thus the optimal allocations are

x_1^* = Allocation to User 1 = 3 units

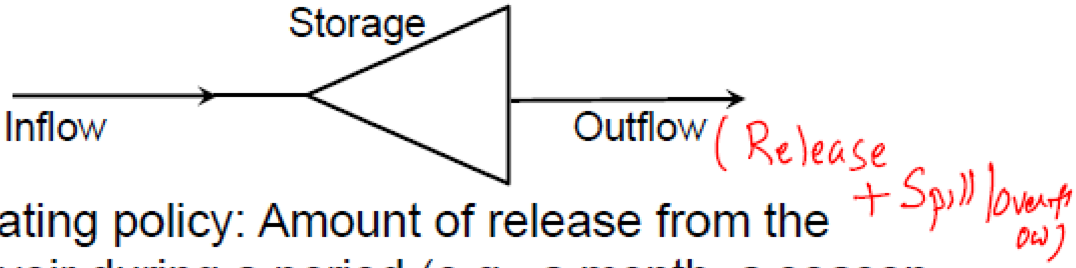
x_2^* = Allocation to User 2 = 1 units

x_3^* = Allocation to User 3 = 2 units

Maximum return resulting from the allocations = 28

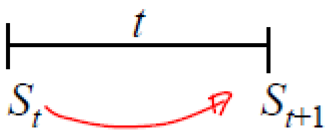
Dynamic Programming

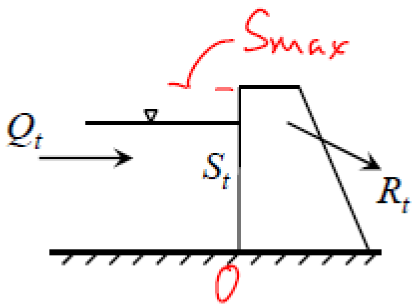
Reservoir operation problem:



- Operating policy: Amount of release from the reservoir during a period (e.g., a month, a season etc.), for a given storage level at the beginning of that period.
- Stage: Time period (e.g., month) for which decisions are required.
- State variable: Storage at the beginning of a stage.
- Decision variable: Release from the reservoir during a period.

State transformation:





Storage continuity (Mass balance)

$$S_{t+1} = S_t + Q_t - R_t$$ …… Neglecting losses

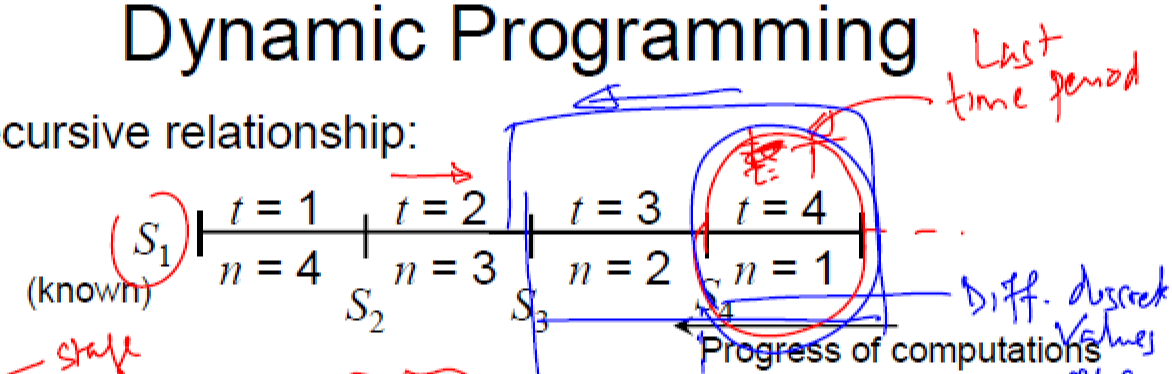
where S_t : Storage at the beginning of period t

Q_t : Inflow during period t

R_t : Release during period t

Dynamic Programming

Recursive relationship:



state

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

time period — max storage — Total water available in period t — S_{t+1}

$B_t\,(R_t)$: Benefits associated with release R_t in period t

$0 \le S_t \le S_{max} \qquad \forall \quad t$

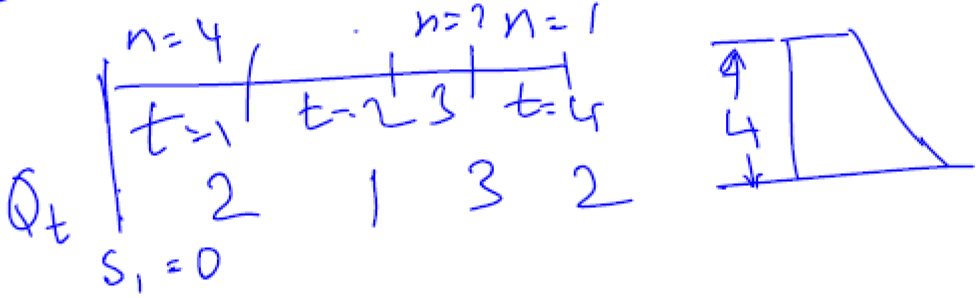
S_{max} : Reservoir capacity

$\{ Q_t \}$ .... known

Inflow during time period t

Example:

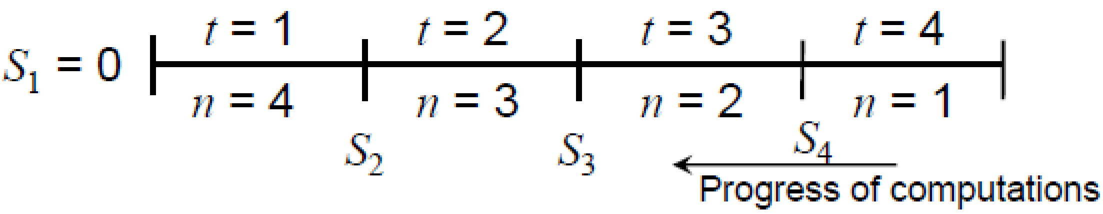
- Inflows during four seasons to a reservoir with storage capacity of 4 units are 2, 1, 3 and 2 units respectively.
- Overflows from the reservoir are also included in the release.
- Reservoir storage at the beginning of the year is 0 units.



- Release from the reservoir during the season results in the following benefits which are same for all the four seasons.

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

- To obtain the release policy backward recursive equation is used, starting with the last stage.



Stage 1:

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

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

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

Total water available for release.

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

Max. storage

$Q_4 = 2$

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	Max: 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		

Stage 2:

$Q_3 = 3$

Stage

$t = 3$ and $n = 2$

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

time

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

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

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

$Q_3 = 3$

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

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

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		

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		

Stage 3:

$Q_2 = 1$

$t = 2$ and $n = 3$

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

from previous stage table

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

$S_2 + Q_2 - R_2 \le 4$

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

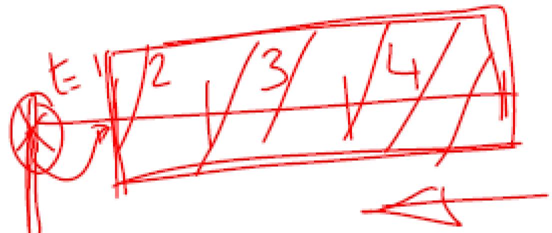
$0 \le R_2 \le (S_2 + Q_2)$

$S_2 + Q_2 - R_2 \le 4$

$Q_2 = 1$

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		
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		

Stage 4:



$S_1 = 0$ <given>

$Q_1 = 2$

$t = 1$ and $n = 4$

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

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

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

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

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

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

$Q_1 = 2$

S_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	-100	2	1280	1180		
0	1	250	1	1210	1460	1460	1
	2	320	0	1050	1370		

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 $S_3 = 1$

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

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

$R_4^* = 3$ ….. 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.

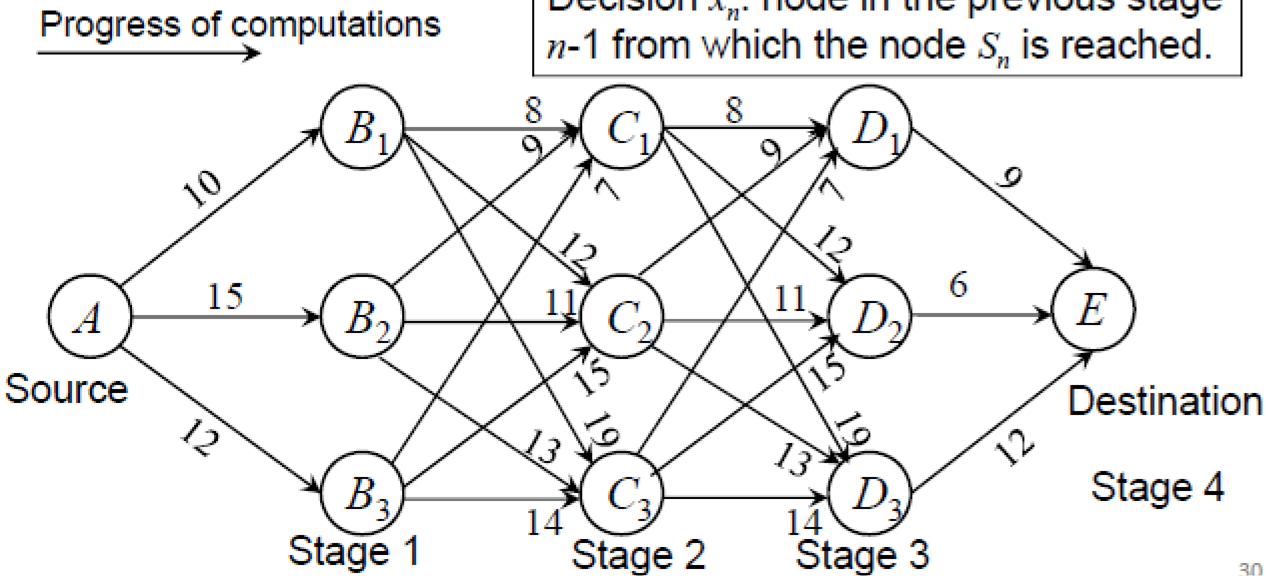
Dynamic Programming

Shortest route problem:

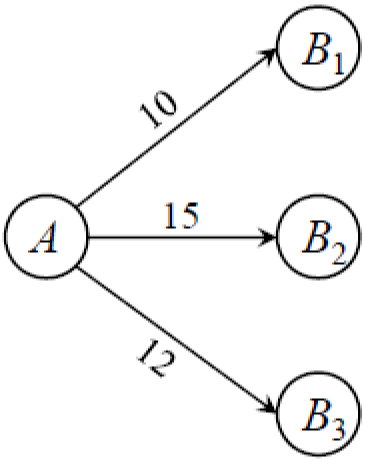
- Consider the problem of determining the shortest route for a pipeline from among various possible routes available from destination to source.

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.



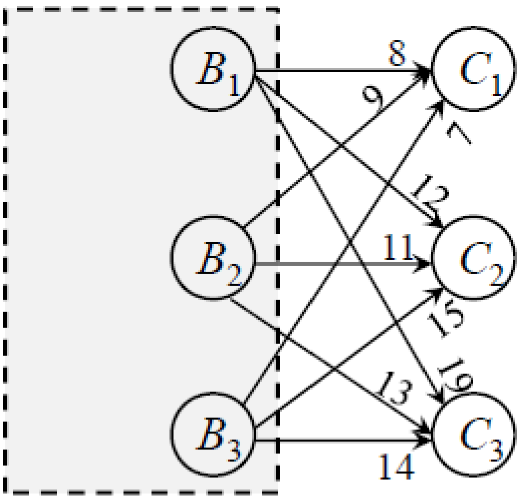
Stage 1:



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

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

Stage 2:

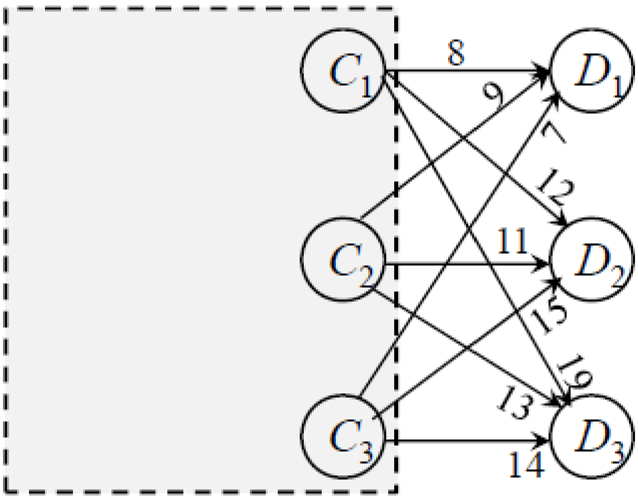


$$f_2^*(S_2) = \text{Min}\left[d(x_2, S_2) + f_1^*(x_2)\right]$$

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

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		

Stage 3:

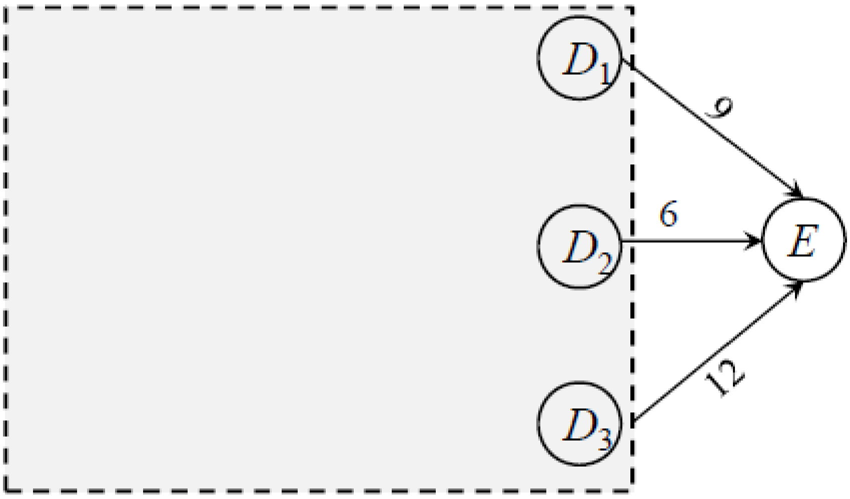


$$f_3^*(S_3) = \text{Min}\left[d(x_3, S_3) + f_2^*(x_3)\right]$$

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

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		

Stage 4:



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

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

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^*
	D_1	9	26	35		
E	D_2	6	30	36	35	D_1
	D_3	12	35	47		

Trace back:

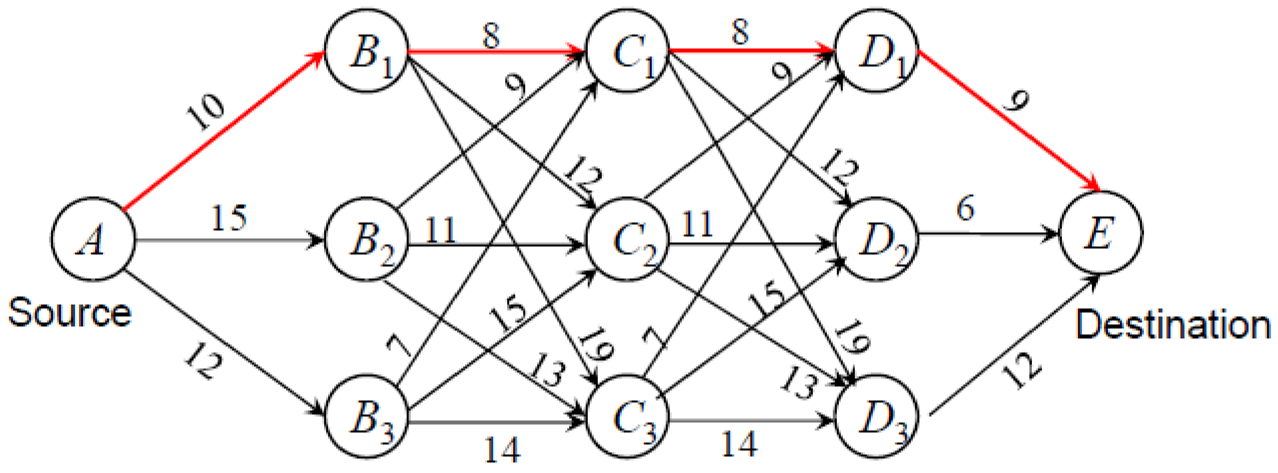
$x_4^* = D_1$ ….. From last table

$x_3^* = C_1$ ….. 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 $S_1 = B_1$

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



Shortest distance = 35 units