Dynamic Programming Applications

Water Allocation – Numerical Example

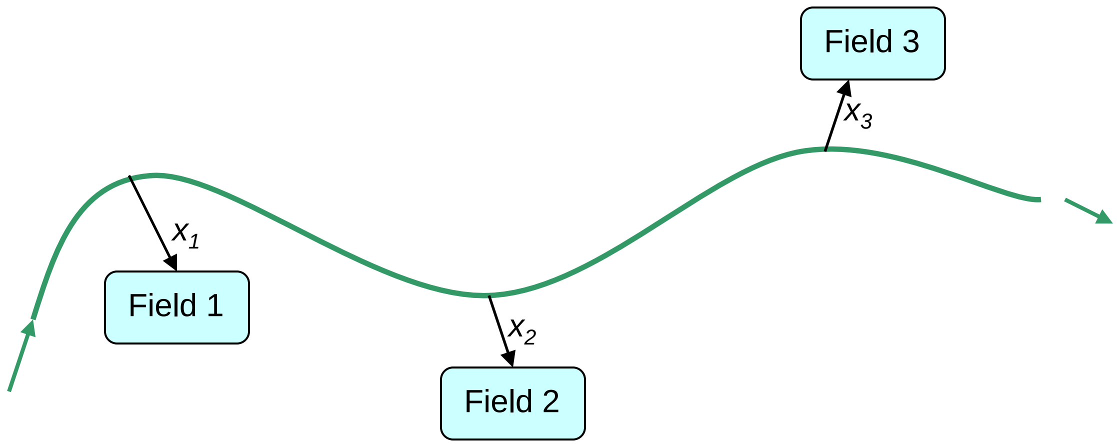
D Nagesh Kumar, IISc

Objectives

- To demonstrate the water allocation problem through a numerical example using
  - Backward approach
  - Forward approach

Numerical Problem

- Consider a canal supplying water for three different crops
- Maximum capacity of the canal is 4 units of water.



- Optimization Problem: Determine the optimal allocations x_i to each crop that maximizes the total net benefits from all the three crops

Numerical Problem …contd.

Net benefits from producing the crops can be expressed as a function of the water allotted. $NB_1(x_1) = 5x_1 - 0.5x_1^{\,2}$

$$NB_2(x_2) = 8x_2 - 1.5x_2^{\,2}$$

$$NB_3(x_3) = 7x_3 - x_3^{\,2}$$

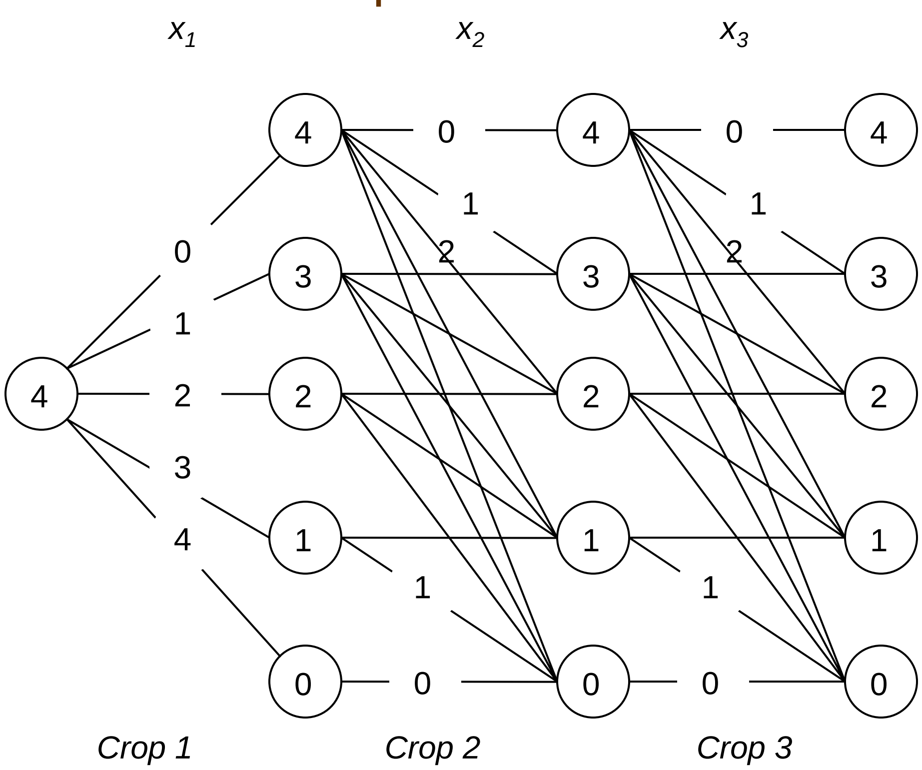
The net benefit values are calculated for each crop and are as shown

Table 1

x_i	$NB_1(x_1)$	$NB_2(x_2)$	$NB_3(x_3)$
0	0.0	0.0	0.0
1	4.5	6.5	6.0
2	8.0	10.0	10.0
3	10.5	10.5	12.0
4	12.0	8.0	12.0

Numerical Problem …contd.

- Representation of the problem as a set of nodes and links



Numerical Problem …contd.

- The values inside the node show the value of state variable at each stage
- Number of nodes for any stage corresponds to the number of discrete states possible for each stage.
- The values over the links show the different values taken by decision variables corresponding to the value taken by state variables

Numerical Problem: Solution by Backward recursion

- Sub-optimization function for the 3rd crop:

$$f_3(S_3) = \max_{\substack{x_3 \\ 0 \le x_3 \le S_3}} NB_3(x_3)$$ with the range of S_3 from 0 to 4.

Table 2

State S_3	$NB_3(x_3)$ x_3 :	0	1	2	3	4	$f_3(S_3)$	x_3^*
0		0					0	0
1		0	6				6	1
2		0	6	10			10	2
3		0	6	10	12		12	3
4		0	6	10	12	12	12	3,4

Solution by Backward recursion … contd.

- Next, by considering last two stages together, the sub-optimization function is $$f_2(S_2) = \max_{\substack{x_2 \\ x_2 \le S_2}} \left[NB_2(x_2) + f_1(S_2 - x_2) \right]$$
- The calculations are shown below

Table 3

State S_2	x_2	$NB_2(x_2)$	$(S_2 - x_2)$	$f_3(S_2 - x_2)$	$f_2(S_2) = NB_2(x_2) + f_3(S_2 - x_2)$	$f_2^*(S_2)$	x_2^*
0	0	0	0	0	0	0	0
1	0	0	1	6	6	6.5	1
	1	6.5	0	0	6.5		

Table contd. on next slide

Solution by Backward recursion … contd.

2	0	0	2	10	10	12.5	1
	1	6.5	1	6	12.5		
	2	10	0	0	10		
3	0	0	3	12	12	16.5	1
	1	6.5	2	10	16.5		
	2	10	1	6	16		
	3	10.5	0	0	10.5		
4	0	0	4	12	12	20	2
	1	6.5	3	12	18.5		
	2	10	2	10	20		
	3	10.5	1	6	16.5		
	4	8	0	0	8		

Solution by Backward recursion … contd.

- Considering all the three stages together

$$f_1(Q) = \max_{\substack{x_1 \\ 0 \le x_1 \le Q}} \left[NB_1(x_1) + f_2(Q - x_1) \right] \quad \text{with} \quad S_1 = Q = 4$$

Table 4

State $S_1 = Q$	x_1	$NB_1(x_1)$	$(Q - x_1)$	$f_2(Q - x_1)$	$f_1(S_1) = NB_1(x_1) + f_2(Q - x_1)$	$f_1^*(S_1)$	x_1^*
4	0	0	4	20	20	21	1
	1	4.5	3	16.5	21		
	2	8	2	12.5	20.5		
	3	10.5	1	6.5	17		
	4	12	0	0	12		

Solution by Backward recursion … contd.

- Backtrack through each table,
- Optimal allocation for crop 1, x_1^* = 1 and S_1 = 4
- Thus, $S_2 = S_1 - x_1 = 3$
- From 2nd stage, the optimal allocation for crop 2, x_2 = 1.
- Now, $S_3 = S_2 - x_2 = 2$
- From 3rd stage calculations, x_3* = 2
- Maximum total net benefit from all the crops = 21

Numerical Problem: Solution by Forward recursion

- Start from the first stage and proceed towards the final stage
- Suboptimization function for the first stage

$$f_1(S_1) = \max_{\substack{x_1 \\ x_1 \le S_1}} NB_1(x_1)$$

- Range of values for S_1 is from *0* to *4*
- The calculations are shown in the table

Solution by Forward recursion … contd.

Table 5

State S_1	x_1	$NB_1(x_1)$	$f_2^*(S_2)$	x_1^*
0	0	0	0	0
1	0	0	4.5	1
	1	4.5		
2	0	0	8	2
	1	4.5		
	2	8		
3	0	0	10.5	3
	1	4.5		
	2	8		
	3	10.5		
4	0	0	12	4
	1	4.5		
	2	8		
	3	10.5		
	4	12		

Solution by Forward recursion … contd.

$$f_2(S_2) = \max_{\substack{x_2 \\ x_2 \le S_2}} \begin{bmatrix} NB_2(x_2) + \\ f_1(S_2 - x_2) \end{bmatrix}$$

Table 6

State S_2	x_2	$NB_2(x_2)$	(S_2-x_2)	$f_1(S_2-x_2)$	$f_2(S_2) = NB_2(x_2) + f_2(S_2-x_2)$	$f_2^*(S_2)$	x_2^*
0	0	0	0	0	0	0	0
1	0	0	1	4.5	4.5	6.5	1
	1	6.5	0	0	6.5		
2	0	0	2	8	8	11	1
	1	6.5	1	4.5	11		
	2	10	0	0	10		
3	0	0	3	10.5	10.5	14.5	1,2
	1	6.5	2	8	14.5		
	2	10	1	4.5	14.5		
	3	10.5	0	0	10.5		
4	0	0	4	12	12	18	2
	1	6.5	3	10.5	17		
	2	10	2	8	18		
	3	10.5	1	4.5	15		
	4	8	0	0	8		

Solution by Forward recursion … contd.

- $$f_3(S_3) = \max_{\substack{x_3 \\ x_3 \le S_3 = Q}} \left[NB_3(x_3) + f_2(S_3 - x_3) \right] \quad \text{with} \quad S_3 = 4$$

Table 7

State S_3	x_3	$NB_3(x_3)$	$S_3 - x_3$	$f_2(S_3 - x_3)$	$f_3(S_3) = NB_3(x_3) + f_2(S_3 - x_3)$	$f_3^*(S_3)$	x_3^*
4	0	0	4	18	18	21	2
	1	6	3	14.5	20.5		
	2	10	2	11	21		
	3	12	1	6.5	18.5		
	4	12	0	0	12		

Solution by Forward recursion … contd.

- Backtrack through each table,
- Optimal allocation for crop 3, x_3* = 2 and S_3 = 4
- Then, $S_2 = S_3 - x_3 = 2$
- The optimal allocation for crop 2, $x_2^* = 1$
- Now, $S_1 = S_2 - x_2 = 1$
- From 1st stage calculations, $x_1^* = 1$
- Maximum total net benefit from all the crops = 21
- These solutions are the same as those we got from backward recursion method.

Thank You