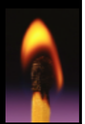
IE 454 Combinatorial Analysis
Fall 2010
Levent Kandiller
Department of Industrial Engineering

Dynamic Programming:
Shortest Path
Knapsack
Production Planning

Two Puzzles Example

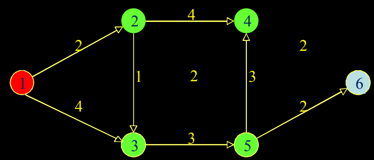
- We show how working backward can make a seemingly difficult problem almost trivial to solve.
- Suppose there are 20 matches on a table. I begin by picking up 1, 2, or 3 matches. Then my opponent must pick up 1, 2, or 3 matches. We continue in this fashion until the last match is picked up. The player who picks up the last match is the loser. How can I (the first player) be sure of winning the game?

Two Puzzles Example



- If I can ensure that it will be opponent’s turn when 1 match remains, I will certainly win.
- Working backward one step, if I can ensure that it will be my opponent's turn when 5 matches remain, I will win.
- If I can force my opponent to play when 5, 9, 13, 17, 21, 25, or 29 matches remain, I am sure of victory.
- Thus I cannot lose if I pick up 1 match on my first turn.

Shortest Path & DP

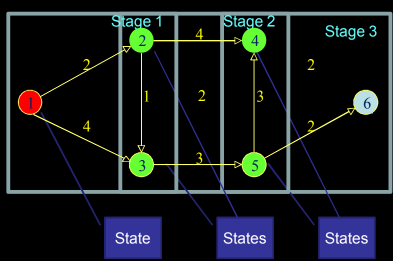


- Many applications of dynamic programming reduce to finding the shortest (or longest) path that joins two points in a given network.
- For larger networks dynamic programming is much more efficient for determining a shortest path than the explicit enumeration of all paths.

Characteristics of Dynamic Programming Applications

- Characteristic 1
  - The problem can be divided into stages with a decision required at each stage.
- Characteristic 2
  - Each stage has a number of states associated with it.
    - By a **state**, we mean the information that is needed at any stage to make an optimal decision.
- Characteristic 3
  - The decision chosen at any stage describes how the state at the current stage is transformed into the state at the next stage.

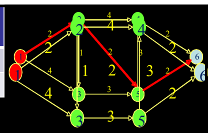
Shortest Path by DP



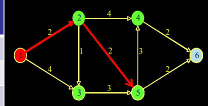
Characteristics of Dynamic Programming Applications

- Characteristic 4
  - Given the current state, the optimal decision for each of the remaining stages must not depend on previously reached states or previously chosen decisions.
    - This idea is known as the **principle of optimality**.
- Characteristic 5
  - If the states for the problem have been classified into on of T stages, there must be a recursion that related the cost or reward earned during stages *t, t*+1, …., *T* to the cost or reward earned from stages t+1, t+2, …. *T*.

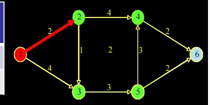
From (Node)	Via (Arc)	To (Node)	Distance from node 6
4	(4,6)	6	2
5	(5,6)	6	2



From (Node)	Via (Arc)	To (Node)	Distance from node 6
2	(2,4)	4	2+4
2	(2,5)	5	2+2=4
3	(3,5)	5	2+3



From (Node)	Via (Arc)	To (Node)	Distance from node 6
1	(1,2)	2	4+2=6
1	(1,3)	3	5+4



Formulating Dynamic Programming Recursions

- In many dynamic programming problems, a given stage simply consists of all the possible states that the system can occupy at that stage.
- If this is the case, then the dynamic programming recursion can often be written in the following form:

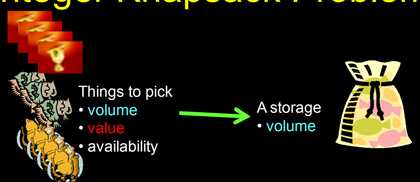
$$F_t(i) = \min\{(\text{cost during stage } t) + f_{t+1}\ (\text{new stage at stage } t+1)\}$$

where the minimum in the above equation is over all decisions that are allowable, or feasible, when the state at state *t* is *i*.

Formulating Dynamic Programming Recursions

- Correct formulation of a recursion of the form requires that we identify three important aspects of the problem:
  - Aspect 1: The set of decisions that is allowable, or feasible, for the given state and stage.
  - Aspect 2: We must specify how the cost during the current time periods (stage *t*) depends on the value of t, the current state, and the decision chosen at stage t.
  - Aspect 3: We must specify how the state at stage t+1 depends on the value of t, the states at stage t, and the decision chosen at stage t.

Integer Knapsack Problem



$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

s.t.

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$$0 \leq x_j \leq a_j \text{ for each } j = 1 \text{ to } 6$$

Knapsack by DP: Job Shop

- Use dynamic programming to solve the following knapsack problem. Three categories of jobs are available, with quantities, times, and values shown in the table.

Job Id	Possible #	Time per job	Value per job
1	3	3	30
2	4	2	20
3	7	1	15

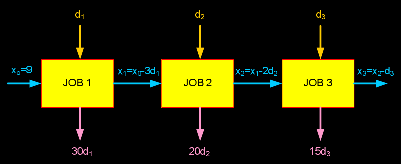
- We have 9 days available.

Knapsack by DP: Job Shop

Job Id	Possible #	Time per job	Value per job
1	3	3	30
2	4	2	20
3	7	1	15

Let

- *d_n: # jobs in category* n *selected,*
- *x_n: # days remaining when we reach* n.



Levent KANDILLER
IE 454 DP: Production Planning

Job Shop Example

x_2	d_3^*	$f_3(x_2)$
0	0	0
1	1	15
2	2	30
3	3	45
4	4	60
5	5	75

x_2	d_3^*	$f_3(x_2)$	x_3
7	7	105	0
8	7	105	
9	7	105	

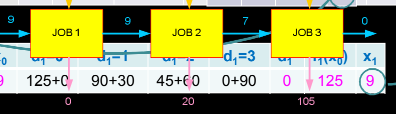
x_1	$d_2=0$	$d_2=1$	d
0	0+0		
1	15+0		
2	30+0	0+20	
3	45+0	15+20	
4	60+0	30+20	0
5	75+0	45+20	15
	90+0	60+20	30
	105+0	75+20	45
8	105+0	90+20	60
9	105+0	105+20	75

x_1	d_2^*	$f_2(x_1)$	x_2
0	0	0	0
1	0	15	1
2	0	30	2
3	0	45	3
4	0	60	4
5	0	75	5
6	0	90	6
7	0	105	7

x_1	d_2^*	$f_2(x_1)$	x_2
9	1	125	7

d_2^*	$f_2(x_1)$	x_2
0	0	0
0	15	1
0	30	2
0	45	3
0	60	4
0	75	5
0	90	6
0	105	7
1	110	6
1	125	7

x_0	$d_1=0$	$d_1=1$	$d_1=2$	$d_1=3$	d_1	$f_1(x_0)$	x_1
9	125+0	90+30	45+60	0+90	0	125	9



Levent KANDILLER
IE 454 DP: Production Planning

What if 8 days available?

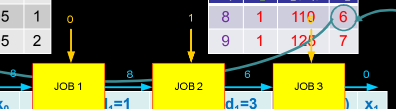
x_2	d_3^*	$f_3(x_2)$	x_3
0	0	0	0
1	1	15	0
2	2	30	0
3	3	45	0
4	4	60	0

x_2	d_3^*	$f_3(x_2)$	x_3
6	6	90	0
7	7	105	0
8	7	105	1
9	7	105	2

x_1	d_2^*	$f_2(x_1)$	x_2
0	0	0	0
1	0	15	1
2	0	30	2
3	0	45	3
4	0	60	4
5	0	75	5
6	0	90	6

x_1	d_2^*	$f_2(x_1)$	x_2
8	1	110	6
9	1	125	7

x_0		$d_1=1$		$d_1=3$			x_1
9	125+0	90+30	45+60	0+90	0	125	9
8	110+0	75+30	30+60	X	0	110	8



Levent KANDILLER
IE 454 DP: Production Planning

DP & Inventory Problem

- Dynamic programming can be used to solve an inventory problem with the following characteristics:
  1. Time is broken up into periods, the present period being period 1, the next period 2, and the final period *T*. At the beginning of period 1, the demand during each period is known.
  2. At the beginning of each period, the firm must determine how many units should be produced. Production capacity during each period is limited.

Levent KANDILLER
IE 454 DP: Production Planning

DP & Inventory Problem

3. Each period's demand must be met on time from inventory or current production. During any period in which production takes place, a fixed cost of production as well as a variable per-unit cost is incurred.
4. The firm has limited storage capacity. This is reflected by a limit on end-of-period inventory. A per-unit holding cost is incurred on each period's ending inventory.
5. The firms goal is to minimize the total cost of meeting on time the demands for periods 1,2,…,T.

Levent KANDILLER
IE 454 DP: Production Planning

DP & Inventory Problem

- In this model, the firm's inventory position is reviewed at the end of each period, and then the production decision is made.
- Such a model is called a **periodic review model**.
- This model is in contrast to the continuous review model in which the firm knows its inventory position at all times and may place an order or begin production at any time.

Levent KANDILLER
IE 454 DP: Production Planning

An Example

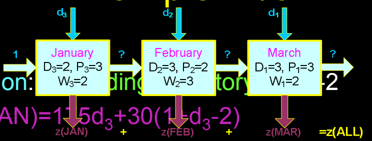
- N: number of periods
- D_n: demand during stage n=1,…,N
- P_n: production capacity in stage n
- W_n: storage capacity at the end of stage n
- C_n: unit production cost in stage n
- H_n: holding cost per unit of ending inventory for stage n
- Initial inventory level is 1.

An Example

Month	Demand	Production Capacity	Storage Capacity	Unit Production Cost	Unit Holding Cost
January	2	3	2	$175	$30
February	3	2	3	150	30
March	3	3	2	200	40

d_n: production quantity for stage n=1,…,N
x_n: a state variable representing the amount of inventory on hand at the begining of stage n

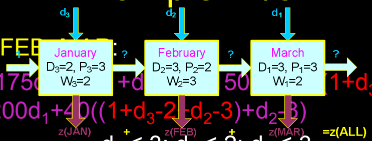
An Example: w/o DP



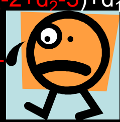
@JAN:
Production: d_3; Ending Inventory: $1+d_3-2$
Min z(JAN)=$175d_3+30(1+d_3-2)$
s.t.
$1+d_3-2 \le 2$
$1+d_3 \ge 2$
$d_3 \le 3$
$d_3 \ge 0$

@FEB:
Production: d_2; Ending Inventory: ? $+d_2-3$
Min z(FEB)=$150d_2+30(?+d_2-3)$
s.t.
$d_2 \le 2$
$?+d_2-3 \le 3$
$?+d_2 \ge 3$
$d_2 \ge 0$

An Example: w/o DP



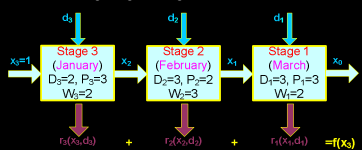
@JAN+FEB+MAR:
Min z=$175d_3+30(1+d_3-2)+150d_2+30((1+d_3-2)+d_2-3) + 200d_1+40((1+d_3-2+d_2-3)+d_1-3)$
s.t. $d_3 \le 3$; $d_2 \le 2$; $d_1 \le 3$
$1+d_3-2 \le 2$; $(1+d_3-2)+d_2-3 \le 3$; $(1+d_3-2+d_2-3)+d_1-3 \le 3$
$1+d_3 \ge 2$; $(1+d_3-2)+d_2 \ge 3$; $(1+d_3-2+d_2-3)+d_1 \ge 3$
$d_3, d_2, d_1 \ge 0$



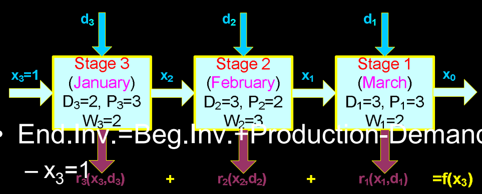
An Example: with DP

Month	Demand	Production Capacity	Storage Capacity	Unit Production Cost	Unit Holding Cost
January	2	3	2	$175	$30
February	3	2	3	150	30
March	3	3	2	200	40

d_n: production quantity for stage n=1,…,N
x_n: a state variable representing the amount of inventory on hand at the begining of stage n

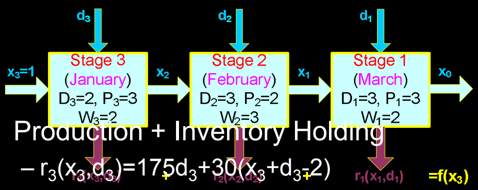


Transformation Functions



- End.Inv.=Beg.Inv.+Production-Demand
  - $x_3=1$
  - $x_2=x_3+d_3-2$
  - $x_1=x_2+d_2-3$
  - $x_0=x_1+d_1-3$

Return Functions



- Production + Inventory Holding
  - $r_3(x_3,d_3)=175d_3+30(x_3+d_3-2)$
  - $r_2(x_2,d_2)=150d_2+30(x_2+d_2-3)$
  - $r_1(x_1,d_1)=200d_1+40(x_1+d_1-3)$

Levent KANDILLER
IE 454 DP: Production Planning

An Example with DP (STAGE 1)

x_1	d_1^*	$f_1(x_1)=r_1(x_1,d_1^*)$
0	3	600
1	2	400
2	1	200
3	0	0

What are the possible values for x_1?

Min $r_1(x_1,d_1)=200d_1+40(x_1+d_1-3)$
s.t.

Month	Demand	Production Capacity	Storage Capacity	Unit Production Cost	Unit Holding Cost
January	2	3	2	$175	$30
February	3	2	3	150	30
March	3	3	2	200	40

Levent KANDILLER
IE 454 DP: Production Planning

(STAGE 2)

@FEB: Given x_2

x_1	d_1^*	$f_1(x_1)=r_1(x_1,d_1^*)$
0	3	600
1	2	400
2	1	200
3	0	0

Min $r_2(x_2,d_2)=150d_2+30(x_2+d_2-3)+f_1(x_1)$
s.t.
$d_2 \le 2$
$x_2+d_2-3 \le 3$
$x_2+d_2 \ge 3$
$d_2 \ge 0$

d_2	$r_2(x_2,d_2)+f_1(x_1)$					
x_2	0	1	2	d_2^*	$f_2(x_2)$	x_1
0				?	+M	?
1			900	2	900	0
2		750	730	2	730	1

Levent KANDILLER
IE 454 DP: Production Planning

(STAGE 3)

@JAN: Given x_3

Min $r_3(x_3,d_3)=175d_3+30(x_3+d_3-2)+f_2(x_2)$
s.t.
$d_3 \le 3$
$x_3+d_3-2 \le 2$
$x_3+d_3 \ge 2$
$d_3 \ge 0$

d_2	$r_2(x_2,d_2)+f_1(x_1)$					
x_2	0	1	2	d_2^*	$f_2(x_2)$	x_1
0				?	+M	?
1			900	2	900	0
2		750	730	2	730	1

d_3	$r_3(x_3,d_3)+f_2(x_2)$						
x_3	0	1	2	3	d_3^*	$f_3(x_3)$	x_2
1		M	1280	1315	2	1280	1

Levent KANDILLER
IE 454 DP: Production Planning

DP Solution of the Example

x_1	d_1^*	$f_1(x_1)=r_1(x_1,d_1^*)$
0	3	600
1	2	400
2	1	200
3	0	0

d_2	$r_2(x_2,d_2)+f_1(x_1)$					
x_2	0	1	2	d_2^*	$f_2(x_2)$	x_1
0				?	+M	?
1			900	2	900	0
2		750	730	2	730	1

Stage 3 (January) D3=2, P3=3 W3=2 — Stage 2 (February) D2=3, P2=2 W2=3 — Stage 1 (March) D1=3, P1=3 W1=2

d_3	$r_3(x_3,d_3)+f_2(x_2)$						
x_3	0	1	2	3	d_3^*	$f_3(x_3)$	X_2
1		M	1280	1315	2	1280	1

380 + 300 + 600 = 1280

Levent KANDILLER
IE 454 DP: Production Planning