Lesson 14.

Formulating DP recursions, cont.

Example 1. The Dijkstra Brewing Company is planning production of its new limited run beer, Primal Pilsner. The company must supply 1 batch next month, then 2 and 4 in successive months. Each month in which the company produces the beer requires a factory setup cost of \$5,000. Each batch of beer costs \$2,000 to produce. Batches can be held in inventory at a cost of \$1,000 per batch per month. Capacity limitations allow a maximum of 3 batches to be produced during each month. In addition, the size of the company's warehouse restricts the ending inventory for each month to at most 3 batches. The company has no initial inventory.

The company wants to find a production plan that will meet all demands on time and minimizes its total production and holding costs over the next 3 months. Formulate this problem as a dynamic program by giving its recursive representation. Solve the dynamic program.

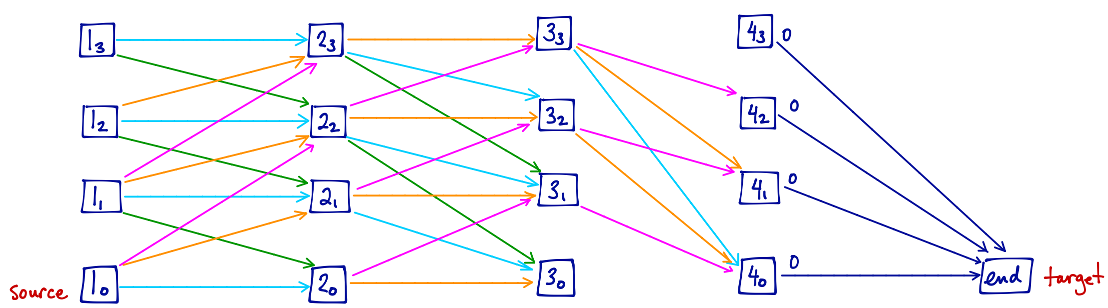
Formulating the DP

- Recall that in Lesson 9, we formulated this problem as a dynamic program with the following shortest path representation:

Stage t: beginning of month t

Node t_n: n batches in inventory at stage t (with months $t, t+1, \dots$ remaining)

Find the shortest path from source to target:



Month	Prod. amt.	Edges	Edge lengths
1	0	$(1_n, 2_{n-1})$ for $n = 1,2,3$	$1(n-1)$
1	1	$(1_n, 2_n)$ for $n = 0,1,2,3$	$5 + 2(1) + 1(n)$
1	2	$(1_n, 2_{n+1})$ for $n = 0,1,2$	$5 + 2(2) + 1(n+1)$
1	3	$(1_n, 2_{n+2})$ for $n = 0,1$	$5 + 2(3) + 1(n+2)$
2	0	$(2_n, 3_{n-2})$ for $n = 2,3$	$1(n-2)$
2	1	$(2_n, 3_{n-1})$ for $n = 1,2,3$	$5 + 2(1) + 1(n-1)$
2	2	$(2_n, 3_n)$ for $n = 0,1,2,3$	$5 + 2(2) + 1(n)$
2	3	$(2_n, 3_{n+1})$ for $n = 0,1,2$	$5 + 2(3) + 1(n+1)$
3	0	not possible!	
3	1	$(3_n, 4_{n-3})$ for $n = 3$	$5 + 2(1) + 1(n-3)$
3	2	$(3_n, 4_{n-2})$ for $n = 2,3$	$5 + 2(2) + 1(n-2)$
3	3	$(3_n, 4_{n-1})$ for $n = 1,2,3$	$5 + 2(3) + 1(n-1)$

- Let d_t = number of batches required in month t, for $t = 1, 2, 3$
- Stages:
- States:
- Allowable decisions x_t at stage t and state n:
- Reward of decision x_t at stage t and state n:
- Reward-to go function $f_t(n)$ at stage t and state n:
- Boundary conditions:
- Recursion:
- Desired reward-to-go function value:

Solving the DP

- Stage 4 computations – boundary conditions:

- Stage 3 computations:

$f_3(3) =$

$f_3(2) =$

$f_3(1) =$

$f_3(0) =$

- Stage 2 computations:

$f_2(3) =$

$f_2(2) =$

$f_2(1) =$

$f_2(0) =$

- Stage 1 computations – desired cost-to-go function: