Lesson 13. Introduction to Stochastic Dynamic Programming

1 Motivation

- In the dynamic programs we have studied so far, the transitions from one state to the next are **deterministic**
- For example, the knapsack problem:
  - Suppose we are in stage t and state n (deciding whether to take metal t with n kg of space remaining)
  - If we decide to take metal t in stage t, we know exactly what state we will be in stage t+1: $n - \begin{pmatrix}\text{weight} \\ \text{of metal } t\end{pmatrix}$
- What if the transitions between states are subject to some randomness or **stochasticity**?

2 A production and inventory problem with stochastic demand

Example 1. The Dijkstra Brewing Company is planning production of its new limited run beer, Primal Pilsner, over the next 2 months. Based on some market analysis studies, the company has determined that the demand for the new beer in each month will be:

Demand (batches)	Probability
0	1/4
2	3/4

Each batch of beer costs \$3,000 to produce. Batches can be held in inventory at a cost of \$1,000 per batch per month. Each month, the company can produce either 0 or 1 batches, due to capacity limitations. In addition, the size of the company's warehouse restricts the ending inventory for each month to at most 2 batches. The company has 1 batch ready to go in inventory.

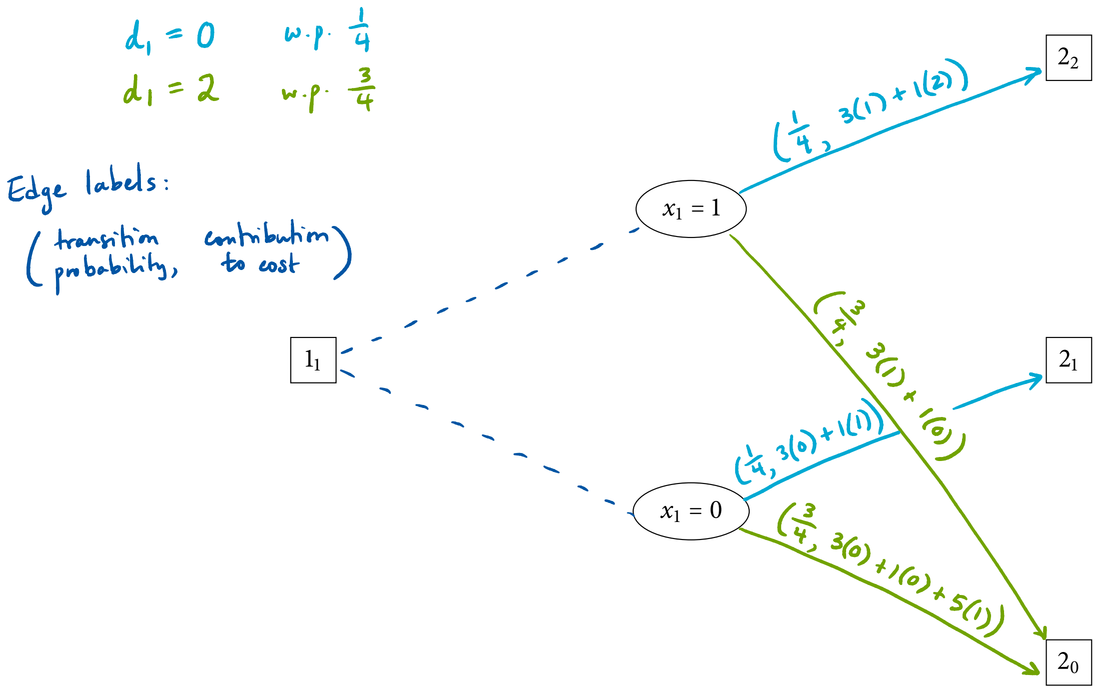
Due to contractual obligations, there is a penalty of \$5,000 for each batch of demand not met. Any batches produced that cannot be stored in the company's warehouse gets thrown away, and cannot be used to meet future demand.

The company wants to find a production plan that will minimizes its total production and holding costs over the next 3 months.

2.1 Modeling the problem visually

- Let's think about the decision-making process starting at month 1
- Let:
  - Node t_n represent month t with n batches in inventory
  - x_t represent the number of batches to produce in month t
  - d_t represent the number of batches in demand in month t

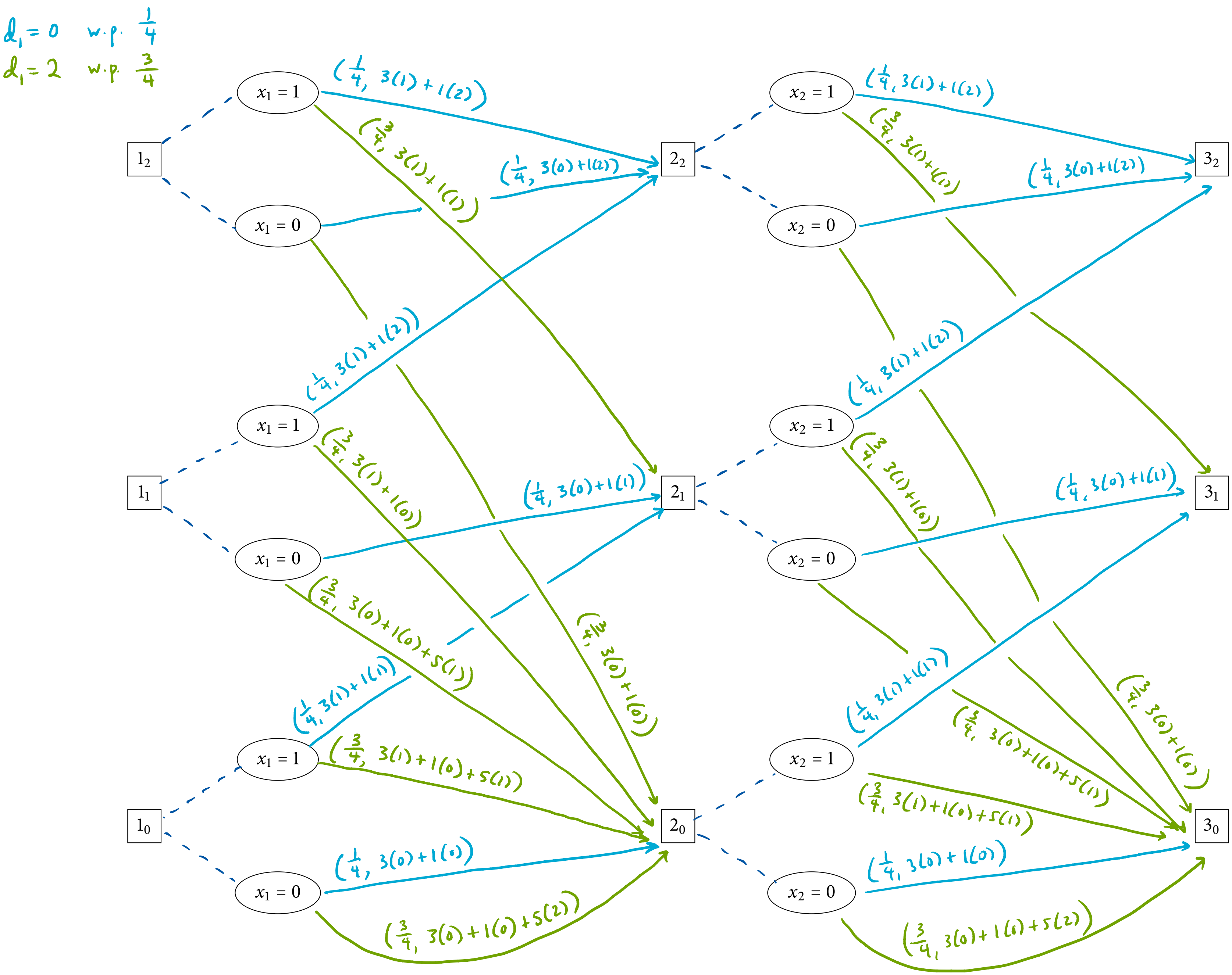
- We can draw the following diagram (that looks like a graph) that models the decision-making process



- We can diagram the entire 2-month process in a similar fashion:

$$f_t(n) = \min_{x_t \in \{0,1\}} \left\{ \begin{array}{l} \frac{1}{4}\left[3x_t + 1 \cdot \min\{n+x_t, 2\} + f_{t+1}\left(\min\{n+x_t,2\}\right)\right] \\ + \frac{3}{4}\left[\begin{array}{l} 3x_t + 1 \cdot \max\{n+x_t-2, 0\} + 5\max\{2-(n+x_t), 0\} \\ \quad + f_{t+1}\left(\max\{n+x_t-2,0\}\right) \end{array}\right] \end{array} \right\}$$

$$f_1(1) = \min\left\{ \begin{array}{l} \underbrace{\frac{1}{4}\left[3(0)+1(1)+f_2(1)\right] + \frac{3}{4}\left[3(0)+1(0)+5(1)+f_2(0)\right]}_{x_1=0}, \\ \underbrace{\frac{1}{4}\left[3(1)+1(2)+f_2(2)\right] + \frac{3}{4}\left[3(1)+1(0)+5(0)+f_2(0)\right]}_{x_1=1} \end{array} \right\}$$


d1 = 0 w.p. 1/4
d1 = 2 w.p. 3/4
1_2
1_1
1_0
x1 = 1
x1 = 0
2_2
2_1
2_0
x2 = 1
x2 = 0
3_2
3_1
3_0
(1/4, 3(1) + 1(2))
(3/4, 3(1) + 1(1))
(1/4, 3(0) + 1(2))
(1/4, 3(1) + 1(1))
(3/4, 3(1) + 1(0))
(1/4, 3(0) + 1(1))
(3/4, 3(0) + 1(0) + 5(1))
(3/4, 3(0) + 1(0))
(3/4, 3(1) + 1(0) + 5(1))
(1/4, 3(0) + 1(0))
(3/4, 3(0) + 1(0) + 5(2))

- Consider the following production policy:
  - In month 1, produce 1 batch
  - In month 2:
    - If there are 2 batches in inventory, produce 0 batches
    - If there are 0 batches in inventory, produce 1 batch
- What is the expected cost of this policy?
- Working backwards:
  - Expected cost in month 2 with 2 batches in inventory (node 2_2):

$$\frac{1}{4}\big(3(0)+1(2)\big) + \frac{3}{4}\big(3(0)+1(0)\big) = \frac{1}{2}$$

  - Expected cost in month 2 with 0 batches in inventory (node 2_0):

$$\frac{1}{4}\big(3(1)+1(1)\big) + \frac{3}{4}\big(3(1)+1(0)+5(1)\big) = 7$$

  - Expected cost in month 1 (node 1_1):

$$\frac{1}{4}\Big(3(1)+1(2)+\frac{1}{2}\Big) + \frac{3}{4}\big(3(1)+1(0)+7\big) = 8.875$$

Expected cost in month 2 w/ 2 batches in inventory (node 2_2)

Expected cost in month 2 w/ 0 batches in inventory (node 2_0)

2.2 Things to think about

- The policy above gives **contingency plans**
- The diagram we drew on page 3 sort of looks like a shortest path problem, but it's not!
- We cannot solve this example as a shortest path problem, since the edges "are random"
- We can, however, still write a recursion to represent this example problem, and others like it
- We'll explore this next...

2.3 Writing down the model

- What we really want: a production policy with minimum expected cost
- Let's write down the recursive representation of our model/diagram
- We can then solve this recursion by working backwards and determine the minimum expected cost and associated optimal policy
- Stages:

$$\text{Stage } t \longleftrightarrow \begin{cases} \text{month } t & (t=1,2) \\ \text{end of decision-making process} & (t=3) \end{cases}$$

- States:

$$\text{State } n \longleftrightarrow \text{having } n \text{ batches in inventory at the beginning of the month} \quad (n=0,1,2)$$

- Transition probability $p(m \mid n, t, x_t)$ of moving from state n to state m in stage t under decision x_t:

$$p(m \mid n,t,x_t) = \begin{cases} \frac{1}{4} & \text{if } m = \min\{n + x_t, \overset{\text{warehouse capacity}}{2}\} \qquad d_t = 0 \text{ w.p. } \frac{1}{4} \\ \frac{3}{4} & \text{if } m = \max\{n + x_t - 2, \overset{\text{unmet demand?}}{0}\} \qquad d_t = 2 \text{ w.p. } \frac{3}{4} \\ 0 & \text{o/w} \end{cases}$$

- Contribution $c(m \mid n, t, x_t)$ of moving from state n to state m in stage t under decision x_t:

$$c(m \mid n,t,x_t) = \begin{cases} 3x_t + 1 \cdot \min\{n+x_t, 2\} & \text{if } m = \min\{n+x_t, 2\} \qquad d_t = 0 \text{ w.p. } \frac{1}{4} \\ 3x_t + 1 \cdot \max\{n+x_t-2, 0\} + 5 \cdot \underline{\max\{2-(n+x_t), 0\}} & \text{if } m = \max\{n+x_t-2, 0\} \qquad d_t = 2 \text{ w.p. } \frac{3}{4} \\ \text{undefined} & \text{o/w} \end{cases}$$

$$\begin{cases} n + x_t \geq 2 \Rightarrow \text{we can meet demand} \\ n + x_t < 2 \Rightarrow \text{unmet demand} = 2 - (n + x_t) \end{cases}$$

- Allowable decisions x_t at stage t and state n:

Let x_t = number of batches to produce in month t

x_t must satisfy: $x_t \in \{0, 1\}$ for $t = 1, 2$
$n = 0, 1, 2$.

- In words, the value-to-go $f_t(n)$ at stage t and state n is:

$f_t(n)$ = minimum total expected production and inventory cost for months $t, \ldots, 2$ w/ n batches in inventory.

for $t = 1, 2, 3$
$n = 0, 1, 2$

- Boundary conditions:

$f_3(n) = 0$ for $n = 0, 1, 2$

- Value-to-go recursion:

$$f_t(n) = \min_{x_t \in \{0,1\}} \left\{ \begin{array}{l} \frac{1}{4}\left[3x_t + 1 \cdot \min\{n + x_t, 2\} + f_{t+1}(\min\{n+x_t, 2\})\right] \\ + \frac{3}{4}\left[3x_t + 1 \cdot \max\{n + x_t - 2, 0\} + 5\max\{2 - (n + x_t), 0\} \right. \\ \left. \qquad + f_{t+1}(\max\{n + x_t - 2, 0\})\right] \end{array} \right\}$$

for $t = 1, 2$
$n = 0, 1, 2$

- Desired value-to-go function value: $f_1(1)$

2.4 Interpreting the value-to-go function

- We can solve this recursion just like with a deterministic DP: start at the boundary conditions and work backwards
- For this problem, we get the following value-to-go function values $f_t(n)$ for $t = 1, 2$ and $n = 0, 1, 2$, as well as the decision x_t^* that attained each value:

t	n	$f_t(n)$	x_t^*
1	0	13.125	1
1	1	8.875	1
1	2	5.875	0
2	0	7	1
2	1	3.5	1
2	2	0.5	0

- Based on this, what should the company's policy be?

$t=1$ (month 1): $n = 1 \Rightarrow \left.\begin{array}{l} f_1(1) = 8.875 \\ x_1^* = 1 \end{array}\right\} \Rightarrow$ produce 1 batch.

$t=2$ (month 2): If $n=2 \Rightarrow \left.\begin{array}{l} f_2(2) = 0.5 \\ x_2^* = 0 \end{array}\right\} \Rightarrow$ produce 0 batches

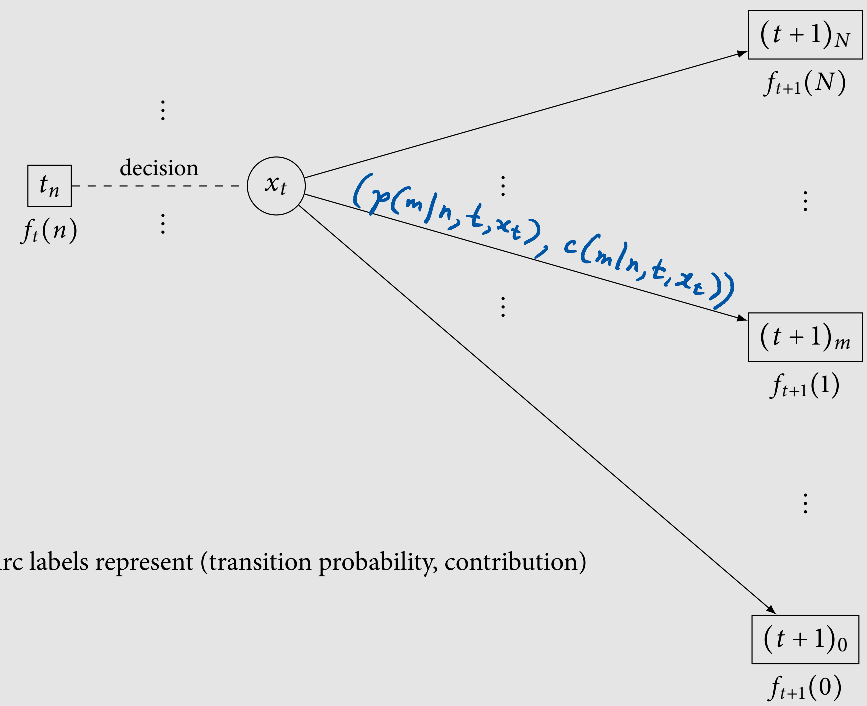
If $n=0 \Rightarrow \left.\begin{array}{l} f_2(0) = 7 \\ x_2^* = 1 \end{array}\right\} \Rightarrow$ produce 1 batch

- What is the company's total expected cost? $f_1(1) = 8.875$

3 Stochastic dynamic programs, more generally

Stochastic dynamic program

- **Stages** $t = 1, 2, \ldots, T$ and **states** $n = 0, 1, 2, \ldots, N$
- Allowable **decisions** x_t at each stage t and state n
- **Transition probability** $p(m \mid n, t, x_t)$ of moving from state n to state m in stage t under decision x_t
- **Contribution** $c(m \mid n, t, x_t)$ for moving from state n to state m in stage t under decision x_t



Arc labels represent (transition probability, contribution)

- **Value-to-go** function $f_t(n)$ at each stage t and state n
- **Boundary conditions** on $f_T(n)$ for each state n
- **Recursion** on $f_t(n)$ at stage t and state n

$$f_t(n) = \min/\max_{x_t \text{ allowable}} \left\{ \sum_{m=0}^{N} p(m \mid n, t, x_t) \Big[c(m \mid n, t, x_t) + f_{t+1}(m) \Big] \right\}$$

for $t = 1, 2, \ldots, T-1$ and $n = 0, 1, \ldots, N$

- **Desired value-to-go**, usually $f_1(m)$ for some state m