Plan for today

- Dynamic Programming
  - Employment Scheduling Problem
  - Reject Allowance Problem
  - Winning in Vegas Problem

DYNAMIC PROGRAMMING

… where we see the algorithmic technique of dynamic programming (through examples)

EMPLOYMENT SCHEDULING

… a problem where the number of states is infinity and the decision variables can take real values

Employment Scheduling Problem

- Consider a local shop that is subject to seasonal fluctuation
- At the end of Spring, the shop has 255 workers

Required number of workers in each season

Season	Summer	Fall	Winter	Spring
Requirements	220	240	200	255

- Requirement has to be met in all seasons,
  - i.e., cannot hire below the numbers given in the above table, but can hire more than the requirement
- The planning horizon ends next spring; should have exactly 255 workers for Spring
- Costs:
  - Hiring or firing workers costs $2x^2$, where x is the number of workers hired/fired
  - Each worker in excess of the requirement costs $20 per season
- Fractional employment levels are allowed (due to part-time employees)
  - cost also applies on a fractional basis

Question: determine the number of workers to hire/fire at the beginning of each season to minimize the total cost

Stages, States, Decision vars, Optimality criterion

- Stages: four stages, representing decision-making at the beginning of summer, fall, winter, and spring
- States: s_n represents the number of workers at the beginning of n'th stage

	beginning state	hire/fire	ending state	requirement
1. Summer	$s_1 = 255$	x_1	s_1+x_1	**220**
2. Fall	$s_2 = s_1 + x_1$	x_2	s_2+x_2	**240**
3. Winter	$s_3 = s_2 + x_2$	x_3	s_3+x_3	**200**
4. Spring	$s_4 = s_3 + x_3$	x_4	s_4+x_4 $= 255$	**255**

- Decision Variables/Policy: x_n = Number of workers hired/fired at stage n
- Optimality criterion: minimize total cost

Value function

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**
4. Spring	s_4 (=s_3+x_3)	x_4	s_4+x_4=255	**255**

$f_n(s_n, x_n)$ = Cost of starting from state s_n in stage n, hiring/firing x_n in stage n and making optimum hiring/firing decisions thereafter

$$f_n^*(s_n) = \min_{x_n : s_n + x_n \geq \text{ requirement at stage } n} f_n(s_n, x_n)$$

Need: $f_1^*(255)$

Value function: final stage

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 ($=s_1+x_1$)	x_2	s_2+x_2	**240**
3. Winter	s_3 ($=s_2+x_2$)	x_3	s_3+x_3	**200**
4. Spring	s_4 ($=s_3+x_3$)	x_4	$s_4+x_4=255$	**255**

$f_n(s_n, x_n)$ = Cost of starting from state s_n in stage n, hiring/firing x_n in stage n and making optimum hiring/firing decisions thereafter

$$f_n^*(s_n) = \min_{x_n : s_n + x_n \geq \text{ requirement at stage } n} f_n(s_n, x_n)$$

Need: $f_1^*(255)$

$$f_4^*(s_4) = 2x_4^2 = 2(255 - s_4)^2$$

Value function: recursive relationship

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**

$f_n(s_n, x_n)$ = Cost of starting from state s_n in stage n, hiring/firing x_n in stage n and making optimum hiring/firing decisions thereafter

$$f_n^*(s_n) = \min_{x_n : s_n + x_n \geq \text{ requirement at stage } n} f_n(s_n, x_n)$$

Need: $f_1^*(255)$

$$f_3(s_3, x_3) = \underbrace{2x_3^2 + 20(s_3 + x_3 - 200)} + f_4^*(s_3 + x_3)$$

$$f_3^*(s_3) = \min_{x_3 : s_3 + x_3 \geq 200} \{ f_3(s_3, x_3) \}$$

$f_3(s_3, x_3) =$
Immediate cost of starting at s_3 and hiring/firing x_3 in stage 3
+
the cost of optimal decision if we start stage 4 from state $s_3 + x_3$

Value function: recursive relationship

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**

$f_n(s_n, x_n) =$ Cost of starting from state s_n in stage n, hiring/firing x_n in stage n and making optimum hiring/firing decisions thereafter

$$f_n^*(s_n) = \min_{x_n : s_n + x_n \geq \text{ requirement at stage } n} f_n(s_n, x_n)$$

Need: $f_1^*(255)$

$$f_3(s_3, x_3) = 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)$$

$$f_3^*(s_3) = \min_{x_3 : s_3 + x_3 \geq 200} \{ f_3(s_3, x_3) \}$$

$$f_n(s_n, x_n) = 2x_n^2 + 20(s_n + x_n - \text{requirement at stage } n) + f_{n+1}^*(s_n + x_n)$$

$$f_n^*(s_n) = \min_{x_n : s_n + x_n \geq \text{ requirement at stage } n} \{ f_n(s_n, x_n) \}$$

Solution procedure: backward induction (stage 3)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**

Given that $f_4^*(s_4) = 2x_4^2 = 2(255 - s_4)^2$

$$f_3(s_3, x_3) = 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)$$

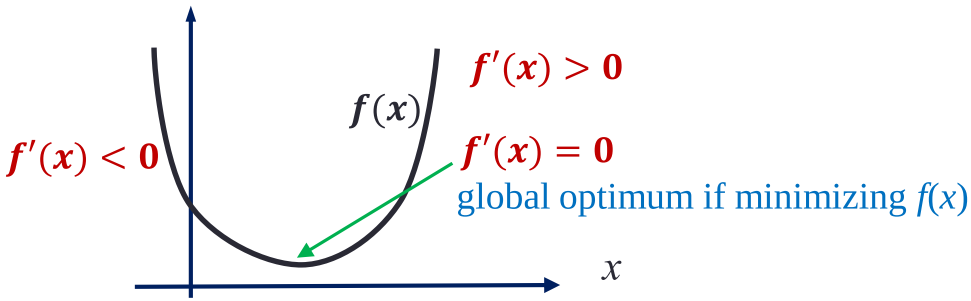
$$f_3^*(s_3) = \min_{x_3: s_3 + x_3 \geq 200} \{ 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)\}$$

$$= \min_{x_3: s_3 + x_3 \geq 200} \{2x_3^2 + 20(s_3 + x_3 - 200) + 2(255 - s_3 - x_3)^2\}$$

Ignoring the $s_3 + x_3 \geq 200$ constraint, the optimum value is attained at $x_3^* =$

Minimizing Single Variable Functions

- Continuous function $f(x)$



1. Find a point x^* where $f'(x^*) = 0$
2. Verify if x^* is a minimizer (or a maximizer)

   … by checking if $f''(x^*) > 0$ (or $f''(x^*) < 0$)

Example 3

Solution procedure: backward induction (stage 3)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**

Given that $f_4^*(s_4) = 2x_4^2 = 2(255 - s_4)^2$

$$f_3(s_3, x_3) = 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)$$

$$f_3^*(s_3) = \min_{x_3 : s_3 + x_3 \geq 200} \{ 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)\}$$

$$= \min_{x_3 : s_3 + x_3 \geq 200} \{2x_3^2 + 20(s_3 + x_3 - 200) + 2(255 - s_3 - x_3)^2\}$$

Ignoring the $s_3 + x_3 \geq 200$ constraint, the optimum value is attained at $x_3^* =$

$$g(x_3) \coloneqq f_3(s_3, x_3) = 2x_3^2 + 20(s_3 + x_3 - 200) + 2(255 - s_3 - x_3)^2$$

$$\frac{dg(x_3)}{dx_3} = 4x_3 + 20 - 4(255 - s_3 - x_3) = 8x_3 + 4s_3 - 1040$$

$$= 8\left(x_3 - 125 + \frac{1}{2}s_3\right)$$

$$\frac{d^2g(x_3)}{dx_3^2} = 8 > 0$$

$\Rightarrow g$ is minimized at $x_3 = 125 - \frac{1}{2}s_3$

Solution procedure: backward induction (stage 3)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**

Given that $f_4^*(s_4) = 2x_4^2 = 2(255 - s_4)^2$

$$f_3(s_3, x_3) = 2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)$$

$$f_3^*(s_3) = \min_{x_3: s_3 + x_3 \geq 200} \{2x_3^2 + 20(s_3 + x_3 - 200) + f_4^*(s_3 + x_3)\}$$

$$= \min_{x_3: s_3 + x_3 \geq 200} \{2x_3^2 + 20(s_3 + x_3 - 200) + 2(255 - s_3 - x_3)^2\}$$

Ignoring the $s_3 + x_3 \geq 200$ constraint, the optimum value is attained at $x_3^* = 125 - \frac{1}{2}s_3$

Constraint: $s_3 + x_3 \geq 200$,

Substituting $x_3^* = 125 - \frac{1}{2}s_3$, we need $125 + \frac{1}{2}s_3 \geq 200$, i.e., we need $s_3 \geq 150$.

Note that this is automatically satisfied since $s_3 \geq 240$!

So we plug the solution for x_3^* into the value function

$$f_3^*(s_3) = 2\left(125 - \frac{1}{2}s_3\right)^2 + 20\left(\frac{1}{2}s_3 - 75\right) + 2\left(130 - \frac{1}{2}s_3\right)^2$$

Solution procedure: backward induction (stage 2)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**

Given that $f_3^*(s_3) = 2\left(125 - \frac{1}{2}s_3\right)^2 + 20\left(\frac{1}{2}s_3 - 75\right) + 2\left(130 - \frac{1}{2}s_3\right)^2$

$$f_2(s_2, x_2) = 2x_2^2 + 20(s_2 + x_2 - 240) + f_3^*(s_2 + x_2)$$

$$= 2x_2^2 + 20(s_2 + x_2 - 240) + 2\left(125 - \frac{1}{2}(s_2 + x_2)\right)^2 + 20\left(\frac{1}{2}(s_2 + x_2) - 75\right) + 2\left(130 - \frac{1}{2}(s_2 + x_2)\right)^2$$

$$f_2^*(s_2) = \min_{x_2 : s_2 + x_2 \geq 240} \{f_2(s_2, x_2)\}$$

$$g(x_2) := f_2(s_2, x_2)$$

$$g(x_2) = 2x_2^2 + 20(s_2 + x_2 - 240) + 2\left(125 - \frac{1}{2}(s_2 + x_2)\right)^2 + 20\left(\frac{1}{2}(s_2 + x_2) - 75\right) + 2\left(130 - \frac{1}{2}(s_2 + x_2)\right)^2$$

$$\frac{dg(x_2)}{dx_2} = 4x_2 + 20 + (-2)\left(125 - \frac{1}{2}(s_2 + x_2)\right) + 10 + (-2)\left(130 - \frac{1}{2}(s_2 + x_2)\right)$$

$$\frac{dg(x_2)}{dx_2} = 6x_2 + 2s_2 - 480 \quad \Rightarrow \frac{dg(x_2)}{dx_2} = 0 \Rightarrow x_2 = 80 - \frac{1}{3}s_2$$

$$\frac{d^2g(x_2)}{dx_2^2} = 6 > 0$$

$\Rightarrow g(x_2)$ is minimized at $x_2 = 80 - \frac{1}{3}s_2$ $\quad \Rightarrow x_2^* = 80 - \frac{1}{3}s_2$ if $x_2^* + s_2 \geq 240$, i.e., if $s_2 \geq 240$

Solution procedure: backward induction (stage 2)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**

Given that $f_3^*(s_3) = 2\left(125 - \frac{1}{2}s_3\right)^2 + 20\left(\frac{1}{2}s_3 - 75\right) + 2\left(130 - \frac{1}{2}s_3\right)^2$

$$f_2(s_2, x_2) = 2x_2^2 + 20(s_2 + x_2 - 240) + f_3^*(s_2 + x_2)$$

$$= 2x_2^2 + 20(s_2 + x_2 - 240) + 2\left(125 - \frac{1}{2}(s_2 + x_2)\right)^2 + 20\left(\frac{1}{2}(s_2 + x_2) - 75\right) + 2\left(130 - \frac{1}{2}(s_2 + x_2)\right)^2$$

$$f_2^*(s_2) = \min_{x_2 : s_2 + x_2 \geq 240} \{f_2(s_2, x_2)\}$$

If $s_2 \leq 240$, (we also need $x_2 \geq 240 - s_2$)

$g(x_2) \coloneqq f_2(s_2, x_2)$

$$g(x_2) = 2x_2^2 + 20(s_2 + x_2 - 240) + 2\left(125 - \frac{1}{2}(s_2 + x_2)\right)^2 + 20\left(\frac{1}{2}(s_2 + x_2) - 75\right) + 2\left(130 - \frac{1}{2}(s_2 + x_2)\right)^2$$

$$\frac{dg(x_2)}{dx_2} = 4x_2 + 20 + (-2)\left(125 - \frac{1}{2}(s_2 + x_2)\right) + 10 + (-2)\left(130 - \frac{1}{2}(s_2 + x_2)\right)$$

$$\frac{dg(x_2)}{dx_2} = 6x_2 + 2s_2 - 480$$

$\geq 6(240 - s_2) + 2s_2 - 480$ (since $x_2 \geq 240 - s_2$)

$= 960 - 4s_2 > 0$ (since $s_2 \leq 240$)

$\Rightarrow g(x_2)$ is increasing if we increase x_2

$\Rightarrow g(x_2)$ is minimized at $x_2 = 240 - s_2$

Solution procedure: backward induction (stage 2)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**

Given that $f_3^*(s_3) = 2\left(125 - \frac{1}{2}s_3\right)^2 + 20\left(\frac{1}{2}s_3 - 75\right) + 2\left(130 - \frac{1}{2}s_3\right)^2$

$$f_2(s_2, x_2) = 2x_2^2 + 20(s_2 + x_2 - 240) + f_3^*(s_2 + x_2)$$

$$= 2x_2^2 + 20(s_2 + x_2 - 240) + 2\left(125 - \frac{1}{2}(s_2 + x_2)\right)^2 + 20\left(\frac{1}{2}(s_2 + x_2) - 75\right) + 2\left(130 - \frac{1}{2}(s_2 + x_2)\right)^2$$

$$f_2^*(s_2) = \min_{x_2: s_2 + x_2 \geq 240} \{f_2(s_2, x_2)\}$$

Solving the above minimization problem yields $x_2^* = \begin{cases} 80 - \frac{s_2}{3} \text{ if } s_2 \geq 240 \\ 240 - s_2 \text{if } s_2 < 240 \end{cases}$

Plugging x_2^* into $f_2^*(s_2)$, gives

$$f_2^*(s_2) = \begin{cases} 2\left(80 - \frac{1}{3}s_2\right)^2 + 20(s_2 - 195) + 2\left(85 - \frac{1}{3}s_2\right)^2 + 2\left(90 - \frac{1}{3}s_2\right)^2 & \text{if } s_2 \geq 240 \\ 2(240 - s_2)^2 + 1150 & \text{if } s_2 < 240 \end{cases}$$

Solution procedure: backward induction (stage 1)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**

$$f_1(s_1 = 255, x_1) = 2x_1^2 + 20(255 + x_1 - 220) + f_2^*(255 + x_1)$$

$$f_1^*(s_1) = \min_{x_1 : s_1 + x_1 \geq 220} f_1(s_1, x_1)$$

Recall that the value function $f_2^*(s_2)$ has two branches, depending on the value of s_2, so here we need to discuss two separate cases:
Case 1: $s_2 < 240$
Case 2: $s_2 \geq 240$

Case 1:

If $s_2 = s_1 + x_1 < 240$, i.e., $x_1 < 240 - s_1 = -15$, then

$$f_2^*(s_2) = f_2^*(s_1 + x_1) = 2(240 - 255 - x_1)^2 + 1150$$

Substituting, we get $f_1(s_1, x_1) = 2x_1^2 + 20(255 + x_1 - 220) + 2(15 + x_1)^2 + 1150$

$$f_1^*(s_1) = \min_{x_1 : s_1 + x_1 \geq 220} \{f_1(s_1, x_1)\}$$

$g(x_1) \coloneqq f_1(s_1, x_1)$

$g(x_1) = 2x_1^2 + 20(35 + x_1 - 220) + 2(15 + x_1)^2 + 1150$

$\frac{dg(x_1)}{dx_1} = 4x_1 + 20 + 4(15 + x_1) = 8x_1 + 80$
< 0 if $x_1 < -15$

$\Rightarrow g(x_1)$ is decreasing if we increase x_1
$\Rightarrow g(x_1)$ is minimized at $x_1 = -15$, i.e., in Case 2

Solution procedure: backward induction (stage 1)

Need: $f_1^*(255)$

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**

$$f_1(s_1 = 255, x_1) = 2x_1^2 + 20(255 + x_1 - 220) + f_2^*(255 + x_1)$$

$$f_1^*(s_1) = \min_{x_1 : s_1 + x_1 \geq 220} f_1(s_1, x_1)$$

<u>Case 2</u>:

If $s_2 = s_1 + x_1 \geq 240$, i.e., $x_1 \geq -15$, then

$$f_1(s_1, x_1) = 2x_1^2 + 20(35 + x_1) + 2\left(5 + \frac{1}{3}x_1\right)^2 + 20(60 + x_1) + 2\left(\frac{1}{3}x_1\right)^2 + 2\left(5 - \frac{1}{3}x_1\right)^2$$

$$f_1^*(s_1) = \min_{x_1 : s_1 + x_1 \geq 220} \{f_1(s_1, x_1)\}$$

$$g(x_1) := f_1(s_1, x_1)$$

$$g(x_1) = 2x_1^2 + 20(35 + x_1) + 2\left(5 + \frac{1}{3}x_1\right)^2 + 20(60 + x_1) + 2\left(\frac{1}{3}x_1\right)^2 + 2\left(5 - \frac{1}{3}x_1\right)^2$$

$$\frac{dg(x_1)}{dx_1} = 4x_1 + 20 + 4\left(5 + \frac{1}{3}x_1\right)\left(\frac{1}{3}\right) + 20 + 4\left(\frac{1}{3}x_1\right)\left(\frac{1}{3}\right) + 4\left(5 - \frac{1}{3}x_1\right)\left(-\frac{1}{3}\right)$$

$$\frac{dg(x_1)}{dx_1} = \frac{16}{3}x_1 + 40 \Rightarrow \frac{dg(x_1)}{dx_1} = 0 \Rightarrow x_1 = -\frac{15}{2}$$

$$\frac{d^2g(x_1)}{dx_1^2} = \frac{16}{3} > 0$$

$\Rightarrow g(x_1)$ is minimized at $x_1 = -\frac{15}{2}$ $\Rightarrow x_1^* = -\frac{15}{2}$ and $f_1^*(255) = 1850$

Optimal Policy

	beginning state	hire/fire	ending state	requirement
1. Summer	s_1 (=255)	x_1	s_1+x_1	**220**
2. Fall	s_2 (=s_1+x_1)	x_2	s_2+x_2	**240**
3. Winter	s_3 (=s_2+x_2)	x_3	s_3+x_3	**200**
4. Spring	s_4 (=s_3+x_3)	x_4	s_4+x_4=255	**255**

$x_1^* = -7.5$ $\quad x_2^* = 80 - \frac{s_2}{3}$ $\quad x_3^* = 125 - \frac{s_3}{2}$ $\quad x_3^* = 255 - s_4$

$f_1^*(255) = 1850$

Optimum cost is $1,850

Optimum policy:

	beginning state	hire/fire	ending state	requirement	surplus
1. Summer	255	-7.5	247.5	220	27.5
2. Fall	247.5	-2.5	245	240	5
3. Winter	245	2.5	247.5	200	47.5
4. Spring	247.5	7.5	255	255	0

PROBABILISTIC DP

… problems where the transitions between states are probabilistic

REJECT ALLOWANCE

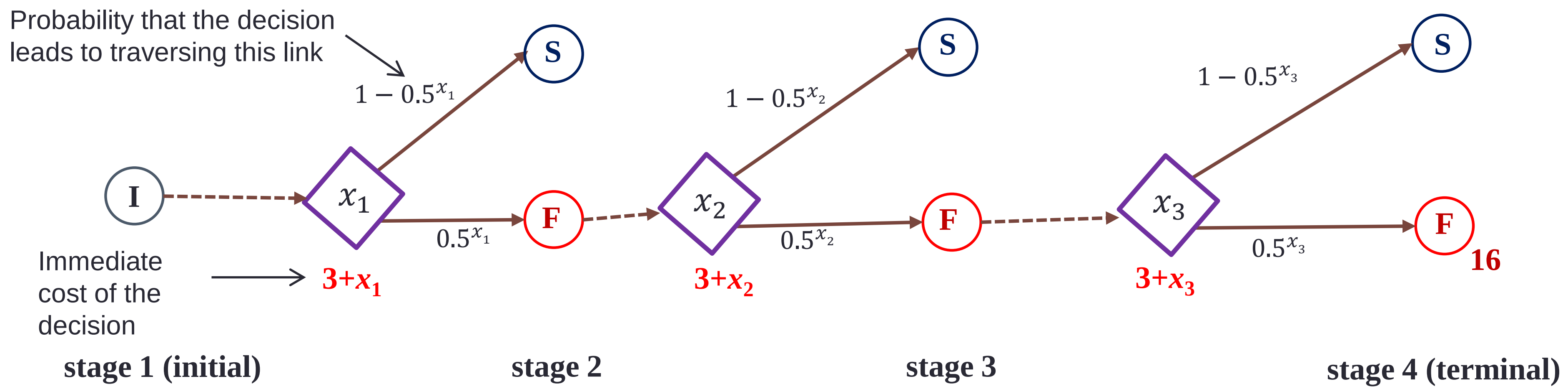
Example 4: Reject allowance

- A customer orders one unit of the product with the option to reject it if it is defective
- To sell a single unit, the manufacturer produces a batch (containing several units of the product), hoping that at least one unit from the batch will NOT be defective
- Each produced unit will be defective with probability **½**
- If the entire batch is defective, the manufacturer can run another round of production to produce another batch
- The maximum possible number of production runs is **three**
- Each production run costs **$3** to set up and each unit costs **$1** to produce
- If all three runs fail to produce a non-defective unit, the process ends and the manufacturer loses **$16** of revenue from the customer

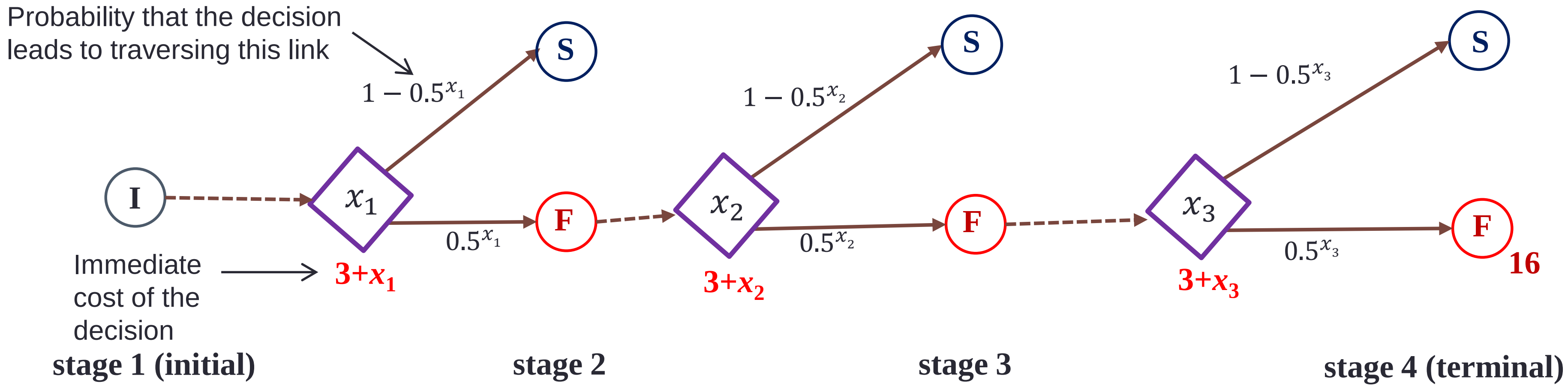
Question: Determine the number of units to produce in each production run in order to minimize the total expected cost

Stages, States, Decision Vars, Optimality Criterion

- Stages: 3 stages representing decision making for each production run and a terminal stage
- States: S (success) if we do not need anymore items in stage i,
  F (failure) if we still need an item in stage i or later
- Decision Variables: x_i ≔ size of the i'th production run
- Optimality criterion: minimize total expected cost



Value function

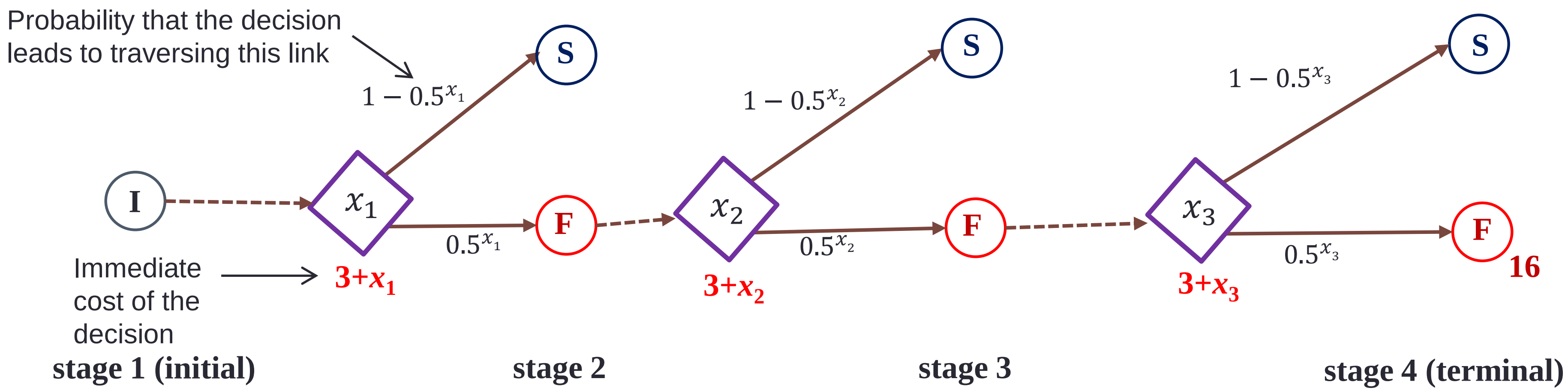


$f_n(s_n, x_n)$ = Expected cost of starting from state s_n in stage n, deciding to produce x_n and making optimal decisions thereafter

$$f_n^*(s_n) = \min_{x_n=0,1,2,3,\ldots} f_n(s_n, x_n)$$

Need: $f_1^*(I)$

Value function: final stage



$f_n(s_n, x_n)$ = Expected cost of starting from state s_n in stage n, deciding to produce x_n and making optimal decisions thereafter

$$f_n^*(s_n) = \min_{x_n=0,1,2,3,\ldots} f_n(s_n, x_n)$$

Need: $f_1^*(I)$

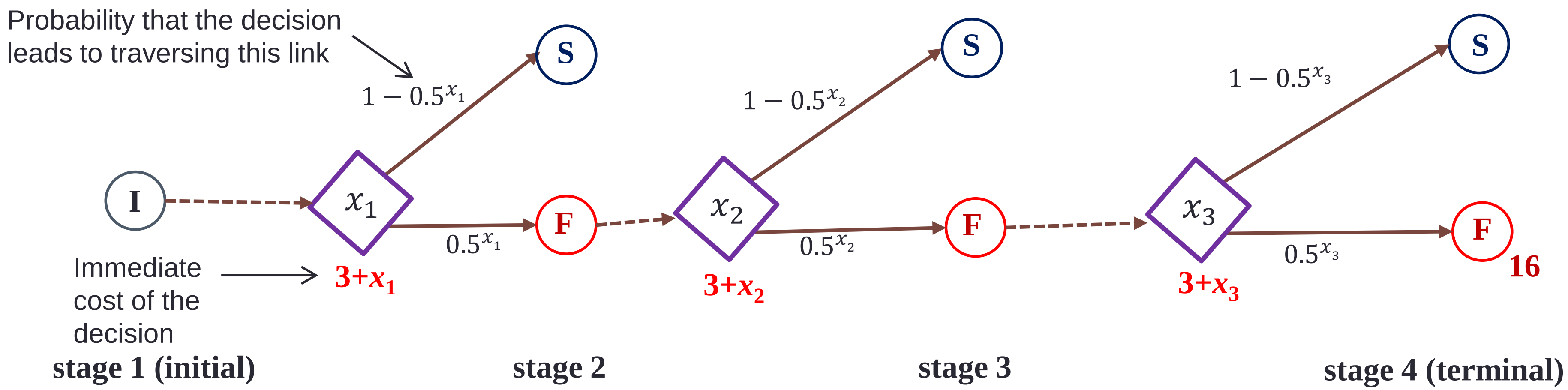
Value function if we reach success state in final stage (product accepted by the customer, no more cost)

$$f_4^*(\mathrm{S}) = 0$$

Value function at failed state in final stage (terminal cost)

$$f_4^*(\mathrm{F}) = 16$$

Value function: recursive relationship



$f_n(s_n, x_n)$ = Expected cost of starting from state s_n in stage n, deciding to produce x_n and making optimal decisions thereafter

$$f_n^*(s_n) = \min_{x_n=0,1,2,3,\ldots} f_n(s_n, x_n)$$

Need: $f_1^*(I)$

Value function at a successful state in stage 2 or 3 (product accepted by the customer, no more cost)

$$f_2^*(\mathrm{S}) = f_3^*(\mathrm{S}) = 0$$

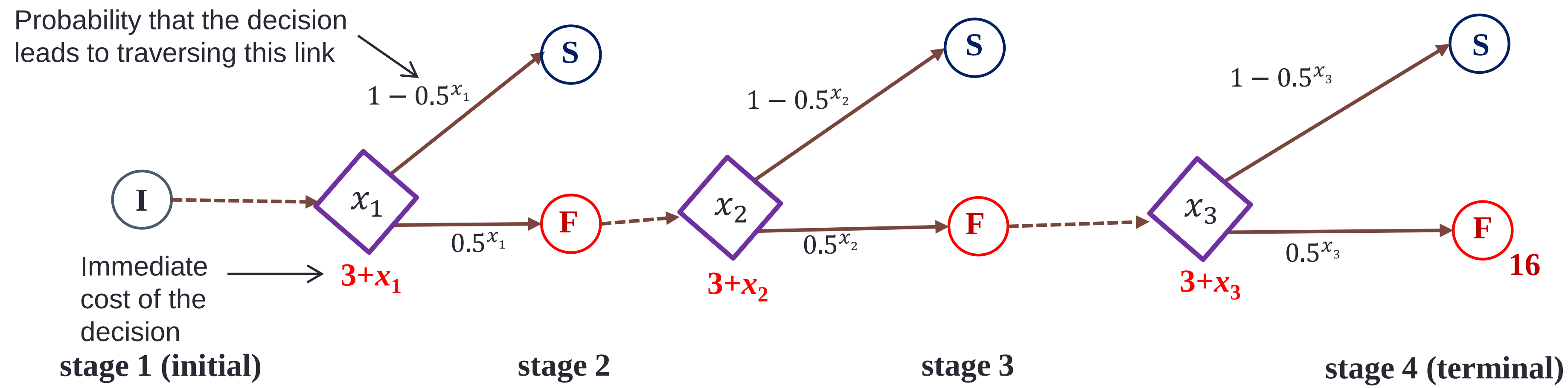
Value function at a failed state in stage 2 or 3 (expected cost of another round of production)

$$f_3(\mathrm{F}, x_3) = 3 + x_3 + 0.5^{x_3} \times f_4^*(F)$$

$$f_3^*(\mathrm{F}) = \min_{x_3=0,1,2,3,\ldots} \{3 + x_3 + 0.5^{x_3} \times f_4^*(F)\}$$

$f_n(s_n, x_n) =$
Cost of producing x_n in round n (Immediate cost)
$+$
Probability(Success | produce x_n) × Optimal Expected cost in the remaining stages given that we had a success
$+$
Probability(Failure | produce x_n) × Optimal Expected cost in the remaining stages given that we had a failure

Value function: recursive relationship



$f_n(s_n, x_n)$ = Expected cost of starting from state s_n in stage n, deciding to produce x_n and making optimal decisions thereafter

$$f_n^*(s_n) = \min_{x_n=0,1,2,3,\ldots} f_n(s_n, x_n)$$

Need: $f_1^*(I)$

Value function at a successful state in stage 2 or 3 (product accepted by the customer, no more cost)

$$f_2^*(\mathrm{S}) = f_3^*(\mathrm{S}) = 0$$

Value function at a failed state in stage 2 or 3 (expected cost of another round of production)

$$f_3^*(\mathrm{F}) = \min_{x_3=0,1,2,3,\ldots} \{3 + x_3 + 0.5^{x_3} \times f_4^*(F)\}$$

$$f_2^*(\mathrm{F}) = \min_{x_2=0,1,2,3,\ldots} \{3 + x_2 + 0.5^{x_2} \times f_3^*(F)\}$$

Value function at the initial state (similar to the value function in a failed state, need to run production)

$$f_1^*(I) = \min_{x_1=0,1,2,3,\ldots} \{3 + x_1 + 0.5^{x_1} \times f_2^*(F)\}$$

Solution Procedure: Backward induction

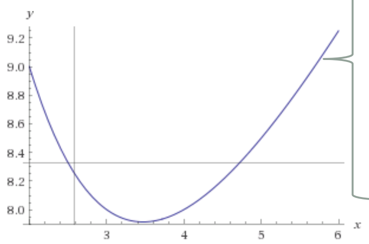
Need: $f_1^*(I)$

stage 4

$f_4^*(\mathrm{F}) = 16$

stage 3

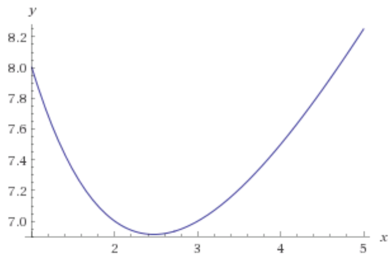
$$f_3^*(\mathrm{F}) = \min_{x_3=0,1,2,3,\ldots} \{3 + x_3 + 0.5^{x_3} \times f_4^*(F)\} = \min_{x_3=0,1,2,3,\ldots} \{3 + x_3 + 0.5^{x_3} \times 16\}$$



x_3	0	1	2	3	4	5	6	$f_3^*(F)$	x_3^*
$3 + x_3 + 0.5^{x_3} \times 16$	19	12	9	8	8	8.5	9.25	8	3 or 4

stage 2

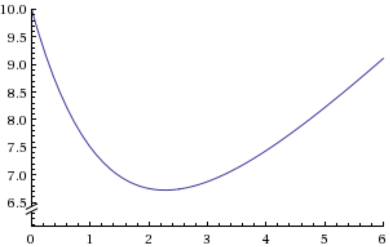
$$f_2^*(\mathrm{F}) = \min_{x_2=0,1,2,3,\ldots} \{3 + x_2 + 0.5^{x_2} \times f_3^*(F)\} = \min_{x_2=0,1,2,3,\ldots} \{3 + x_2 + 0.5^{x_2} \times 8\}$$



x_2	0	1	2	3	4	5	$f_2^*(F)$	x_2^*
$3 + x_2 + 0.5^{x_2} \times 8$	12	8	7	7	7.5	8.25	7	2 or 3

stage 1

$$f_1^*(I) = \min_{x_1=0,1,2,3,\ldots} \{3 + x_1 + 0.5^{x_1} \times f_2^*(F)\} = \min_{x_1=0,1,2,3,\ldots} \{3 + x_1 + 0.5^{x_1} \times 7\}$$



x_1	0	1	2	3	4	$f_1^*(I)$	x_1^*
$3 + x_1 + 0.5^{x_1} \times 7$	10	7.5	6.75	6.875	7.4375	6.75	2

Optimal Policy

Need: $f_1^*(I)$

Optimal expected cost is 6.75$
An optimal policy is to produce
2 units in the first run
2 units in the second run
3 units in the third run

stage 4

$f_4^*(\mathrm{F}) = 16$

stage 3

$$f_3^*(\mathrm{F}) = \min_{x_3=0,1,2,3,\dots} \{3 + x_3 + 0.5^{x_3} \times f_4^*(F)\} = \min_{x_3=0,1,2,3,\dots} \{3 + x_3 + 0.5^{x_3} \times 16\}$$

x_3	0	1	2	3	4	5	6	$f_3^*(F)$	x_3^*
$3 + x_3 + 0.5^{x_3} \times 16$	19	12	9	8	8	8.5	9.25	8	3 or 4

stage 2

$$f_2^*(\mathrm{F}) = \min_{x_2=0,1,2,3,\dots} \{3 + x_2 + 0.5^{x_2} \times f_3^*(F)\} = \min_{x_2=0,1,2,3,\dots} \{3 + x_2 + 0.5^{x_2} \times 8\}$$

x_2	0	1	2	3	4	5	$f_2^*(F)$	x_2^*
$3 + x_2 + 0.5^{x_2} \times 8$	12	8	7	7	7.5	8.25	7	2 or 3

stage 1

$$f_1^*(I) = \min_{x_1=0,1,2,3,\dots} \{3 + x_1 + 0.5^{x_1} \times f_2^*(F)\} = \min_{x_1=0,1,2,3,\dots} \{3 + x_1 + 0.5^{x_1} \times 7\}$$

x_1	0	1	2	3	4	$f_1^*(I)$	x_1^*
$3 + x_1 + 0.5^{x_1} \times 7$	10	7.5	6.75	6.875	7.4375	6.75	2

WINNING IN VEGAS

Example 5: Winning in Vegas

- You would like to show-off your casino skills to your friend
- Friend challenges you to the following game:
  - You start with **3** chips and you are allowed to play **3** rounds
  - In each round,
    - you can bet either none or some, or all of the chips that you have at the beginning of that round
    - Casino odds: 1/3 probability of losing all chips that you bet
2/3 probability of doubling the chips that you bet
- You win the game if you end up with at least 5 chips after 3 rounds; otherwise you lose

Question: determine the number of chips to bet in each round
to win the game against your friend