Plan for today

- Dynamic Programming
  - Stagecoach Problem
  - Terminologies
  - Project Assistant Allocation Problem
  - Employment Scheduling Problem

DYNAMIC PROGRAMMING

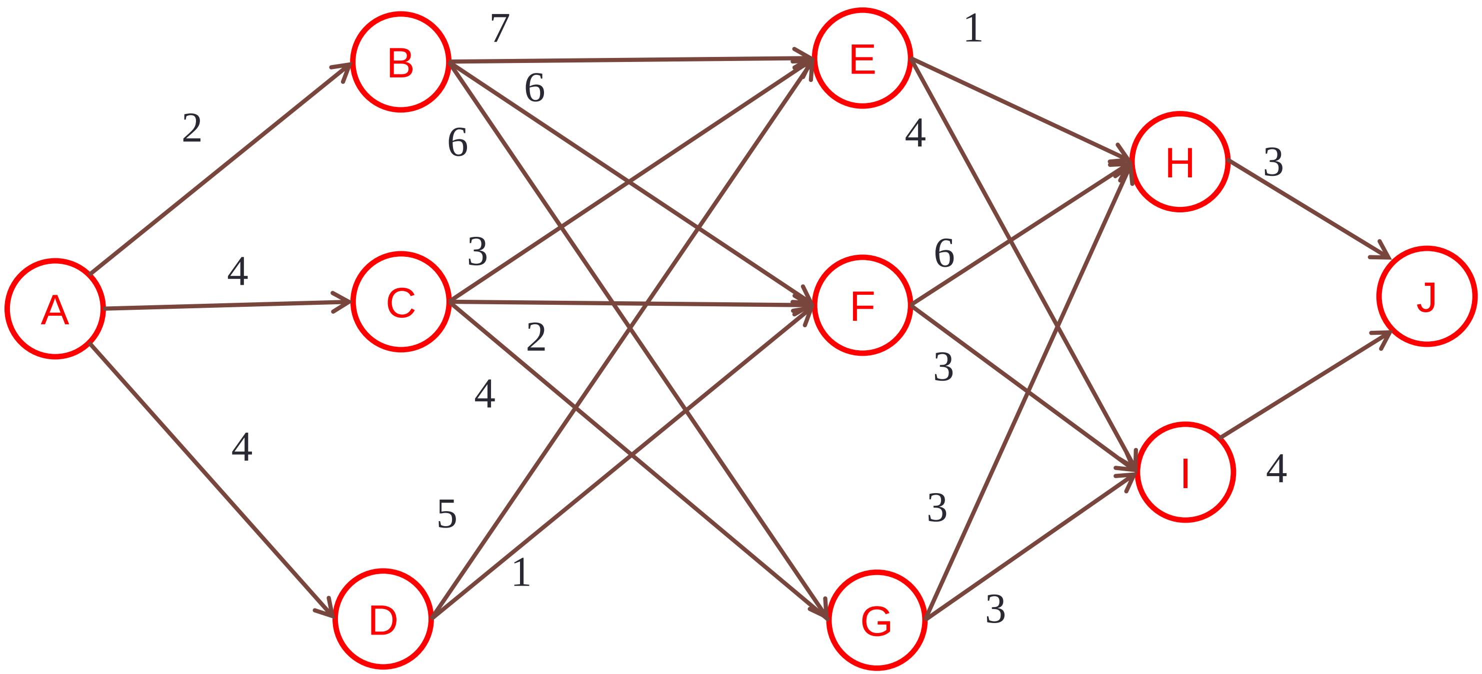
… where we see the algorithmic technique of dynamic programming (through examples)

Recap

STAGECOACH PROBLEM

Motivating Example

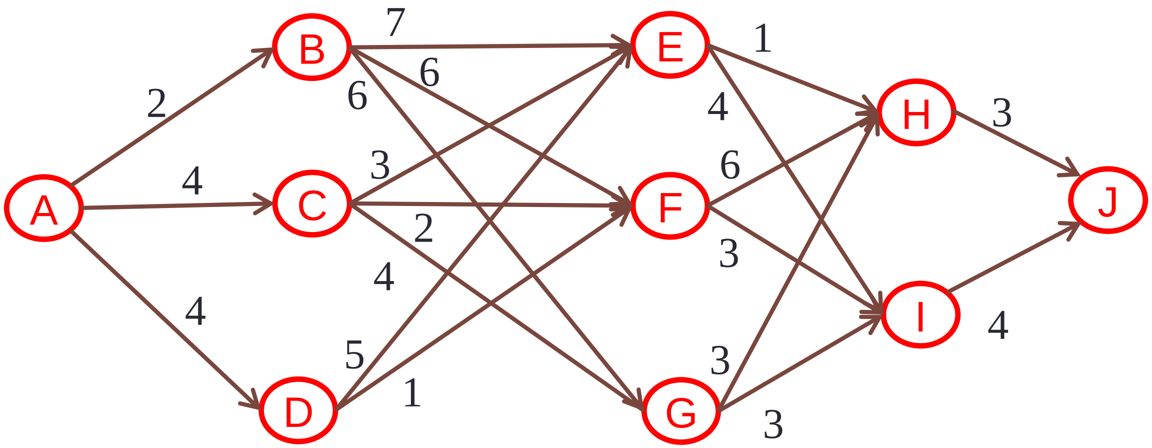
Motivating example: Stagecoach Problem



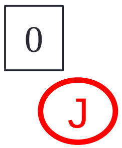
The numbers on the link are toll rates

Question: find the cheapest route from A to J

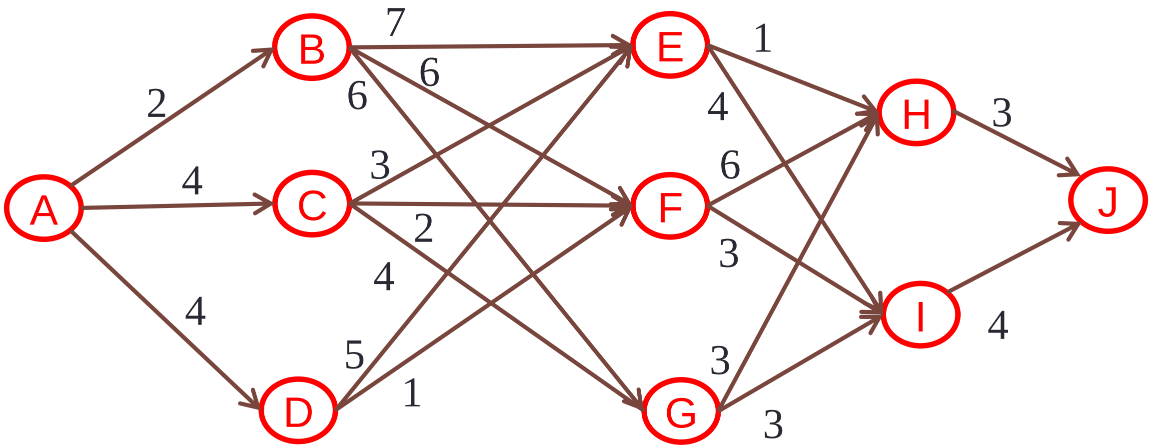
Dynamic Programming: solving the example



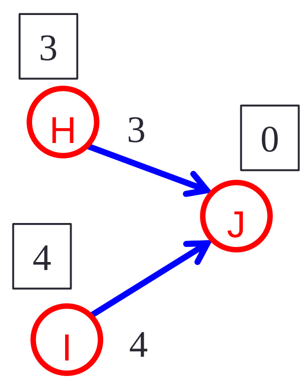
Stage 5



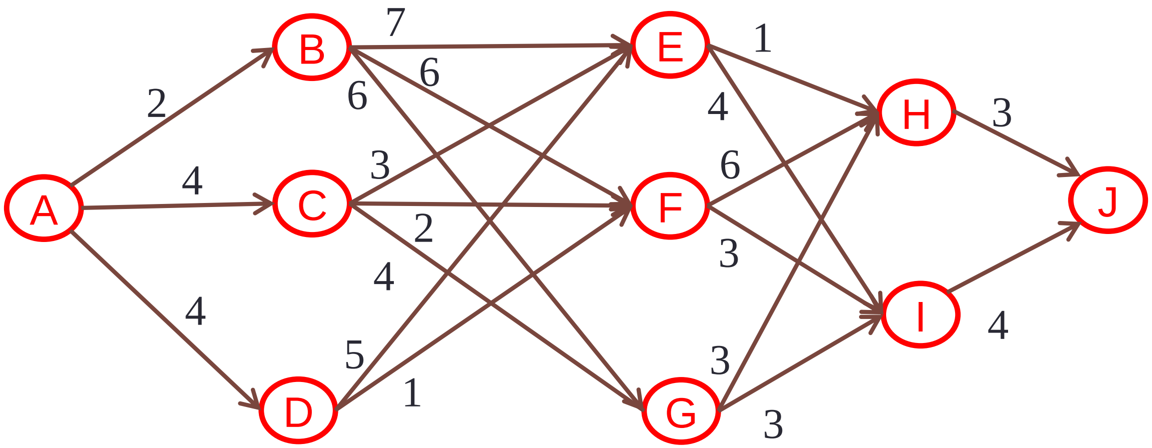
Dynamic Programming: solving the example



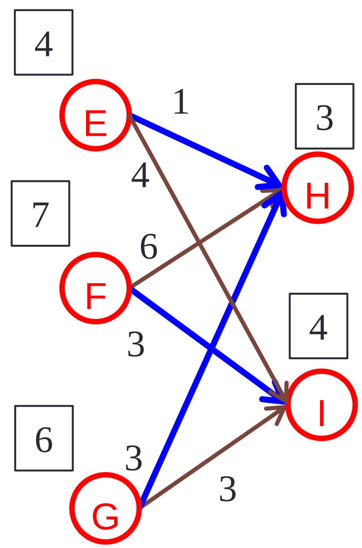
Stage 4



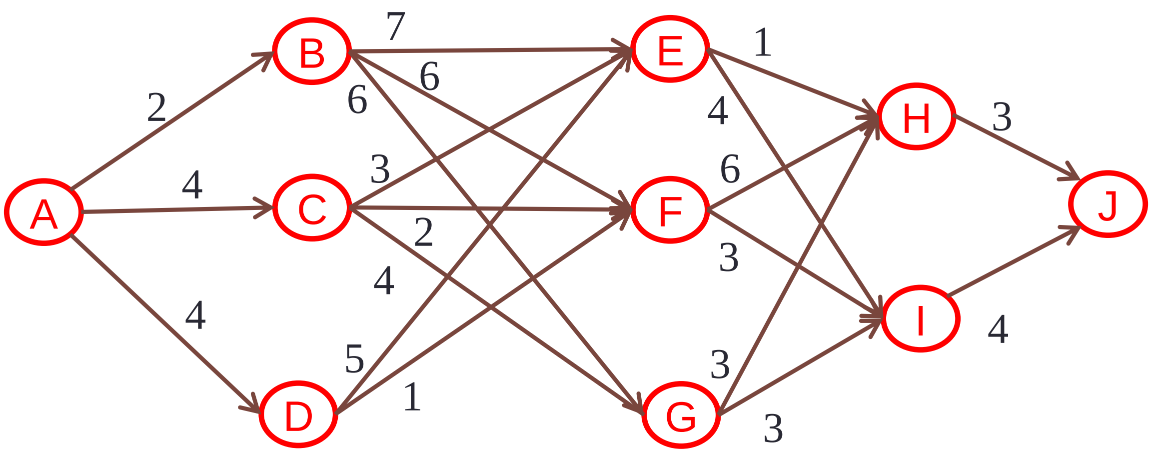
Dynamic Programming: solving the example



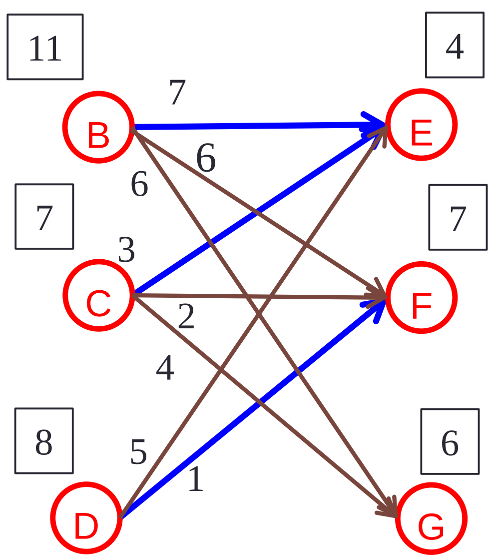
Stage 3



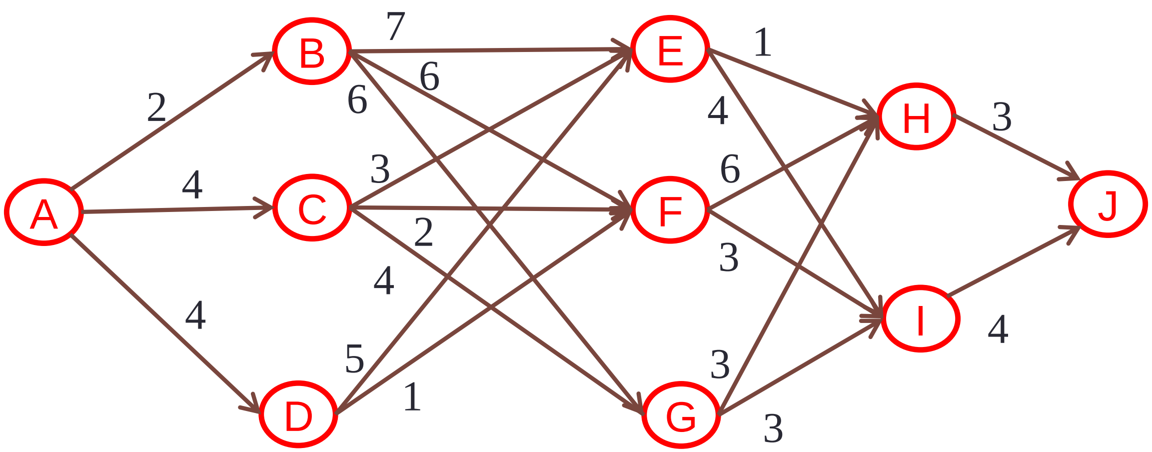
Dynamic Programming: solving the example



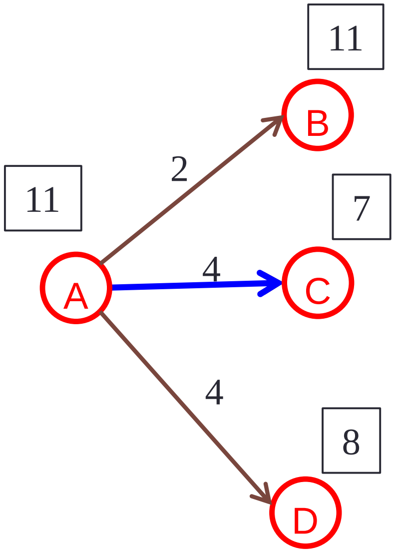
Stage 2



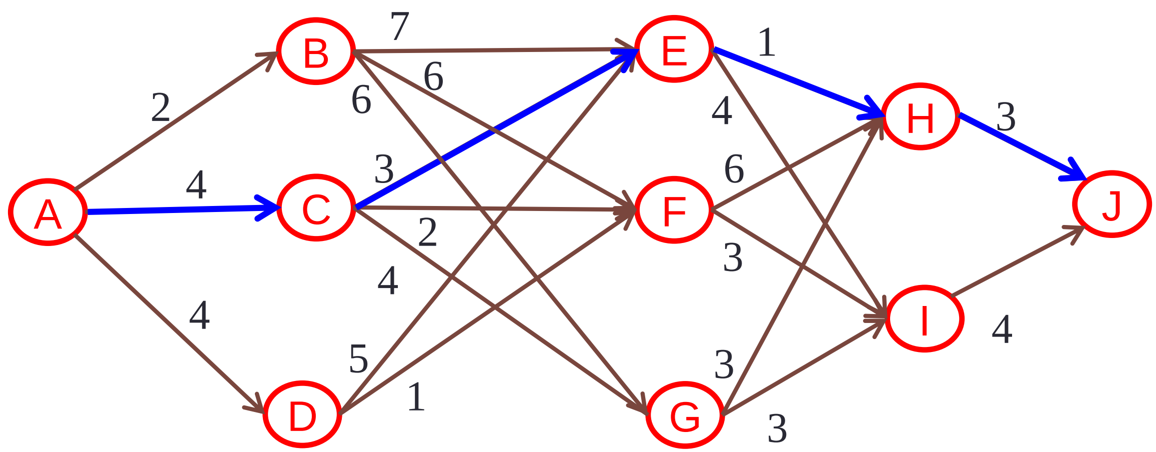
Dynamic Programming: solving the example

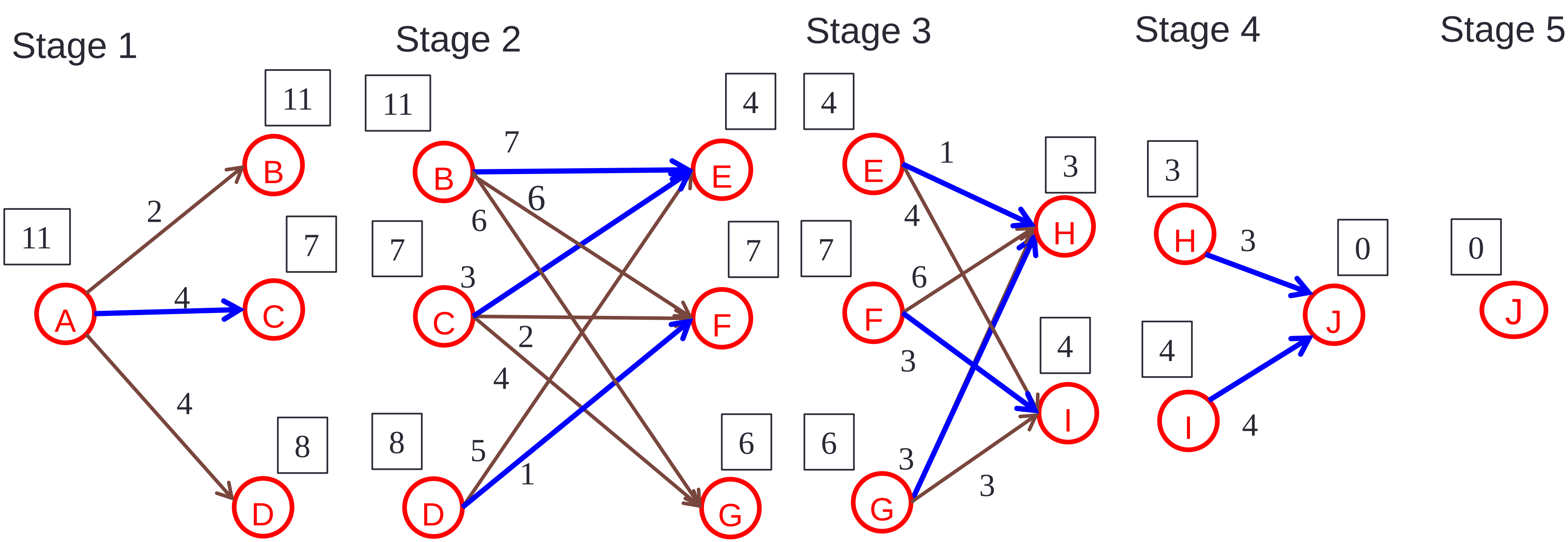


Stage 1



Dynamic Programming: solving the example





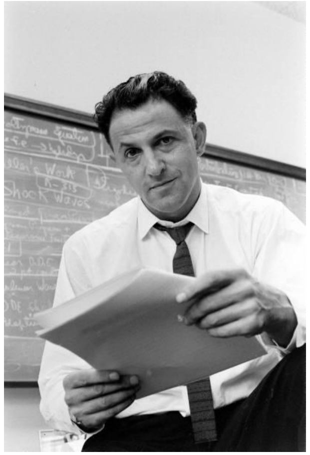
DYNAMIC PROGRAMMING

- terminologies
- when can it be applied?

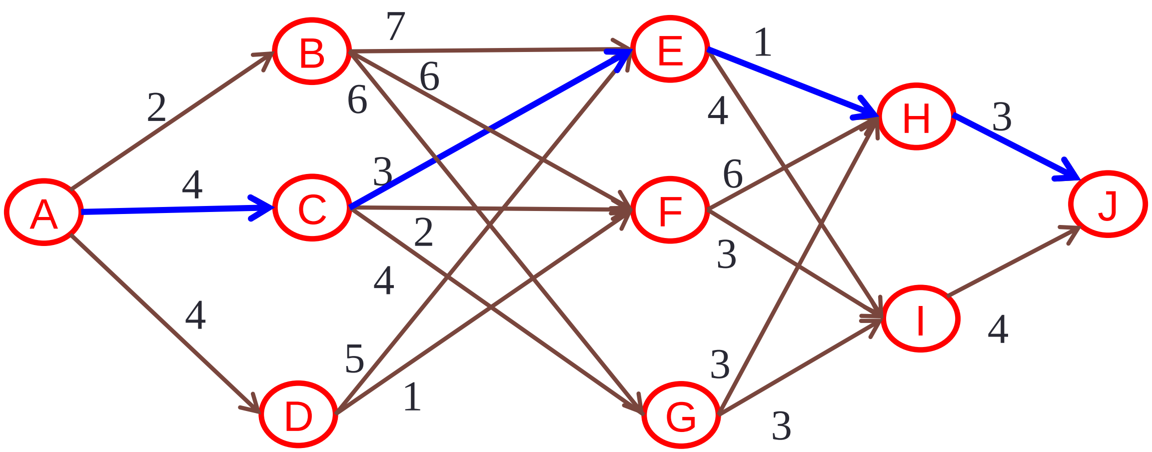
Dynamic Programming: terminologies

- Stages (~ 5 stages in the example)
- States (~ cities in the example)
- Policy/Decision variables
  (~ where to go next from the current state)
- Optimality criterion
  (~ minimize total cost)

- In order to use the DP technique to solve an optimization problem, we first need to formulate the problem in a fashion that facilitates the use of DP
  - Identify Stages, States, Decision Variables, Optimality criterion



Dynamic Programming: When can it be applied?



<u>Bellman's Principle/Principle of Optimality</u>: Optimal policy from a state in stage n can be obtained if we know the optimal policies for all states in stage $n+1$

Dynamic Programming technique is applicable only to problems which satisfy the principle of optimality

Value function: recursive relationship

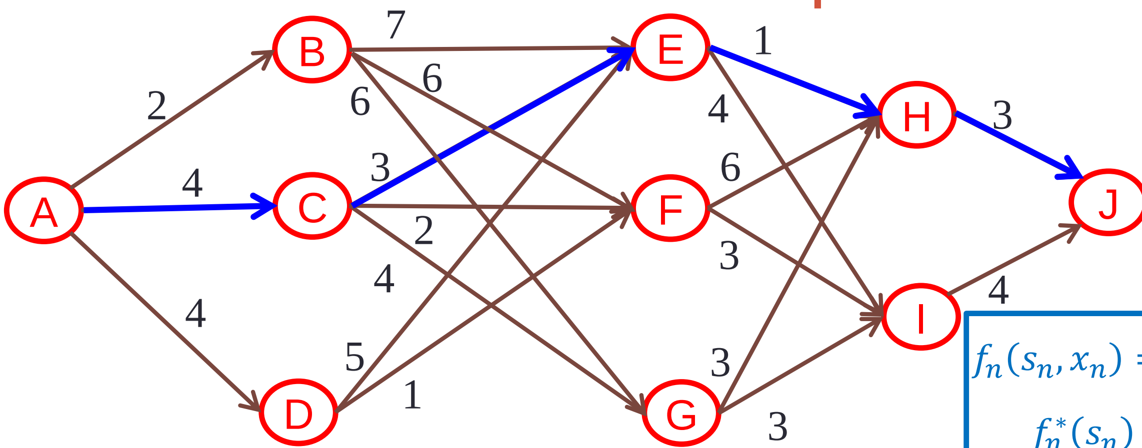
- Suppose our problem has N stages
- Let s_n = current state for stage $n \in \{1, 2 \ldots, N\}$
- x_n = decision variable for stage n

- **Value function:** $f_n(s_n, x_n)$ = contribution of stages $n, n+1, \ldots, N$ to obj if system starts in state s_n of stage n and takes decision x_n and optimal decisions are made thereafter
  - In the example, $f_n(s_n, x_n)$ represents the cost of starting in state s_n of stage n, taking the link to x_n and following the optimal route from x_n to the destination

- **Optimal value function:** $f_n^*(s_n) = \min_{x_n} f_n(s_n, x_n)$
- By Bellman's principle, $f_n(s_n, x_n)$ is usually a **recursive function** that depends on $f_{n+1}^*(s)$
  - In example, $f_n(s_n, x_n) = \text{cost}(s_n \to x_n) + f_{n+1}^*(x_n)$

- x_n^* = optimal decision x_n from s_n, i.e., $x_n^* \coloneqq \operatorname{argmin}_{x_n} f_n(s_n, x_n)$

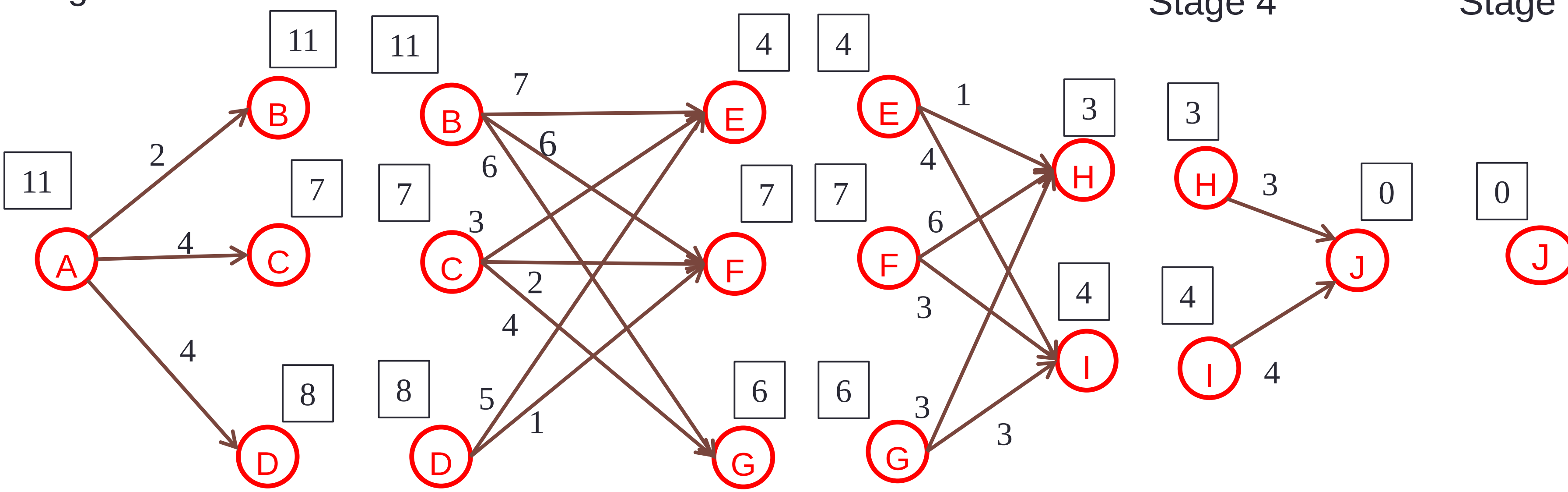
Value function: recursive relationship



$$f_n(s_n, x_n) = \text{cost}(s_n \rightarrow x_n) + f^*_{n+1}(x_n)$$

$$f^*_n(s_n) = \min_{x_n} f_n(s_n, x_n)$$

Stage 1 Stage 2 Stage 3 Stage 4 Stage 5



$f^*_1(A) = 11$

$f^*_2(B) = 11$
$f^*_2(C) = 7$
$f^*_2(D) = 8$

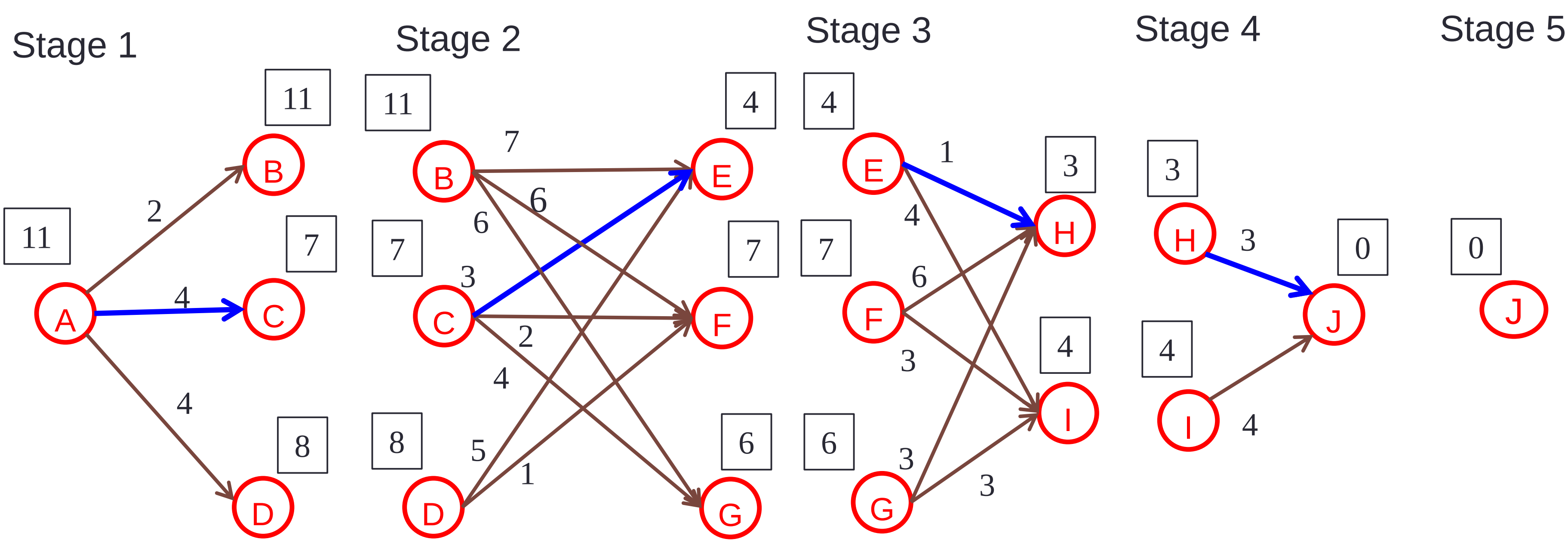
$f^*_3(E) = 4$
$f^*_3(F) = 7$
$f^*_3(G) = 6$

$f^*_4(H) = 3$
$f^*_4(I) = 4$

$f^*_5(J) = 0$

Solution Procedure: Backward Induction

- **Backward induction:** starting from the *final stage*, proceed *backward* stage by stage
  - Find optimal policy for each state in that stage
    - Use recursive relationship to compute all value functions
  - Until optimal policy for initial stage is found
- **Optimal policy:** follow the states with the best value at each stage
  - start from the initial state, use the best decision to transition to an adjacent state in the next stage and repeat



$f_1^*(A) = 11$

$f_2^*(B) = 11$
$f_2^*(C) = 7$
$f_2^*(D) = 8$

$f_3^*(E) = 4$
$f_3^*(F) = 7$
$f_3^*(G) = 6$

$f_4^*(H) = 3$
$f_4^*(I) = 4$

$f_5^*(J) = 0$

Dynamic Programming: Standard Procedure

1. Identify Stages, States, Decision Variables, Optimality Criterion
2. Find the optimal value function of the states in the final stage
3. Write the value function of the states in the n'th stage as a recursive relationship
   - the value function should depend only on the optimal values of states in future stages

     [DP algorithmic technique is applicable only if this property is satisfied]
4. Use backward induction to compute the optimal value function and the best decision for every state
5. Identify the optimal policy by tracing the best decision from the initial state

PROJECT ASSISTANT ALLOCATION PROBLEM

Project Assistant Allocation Problem

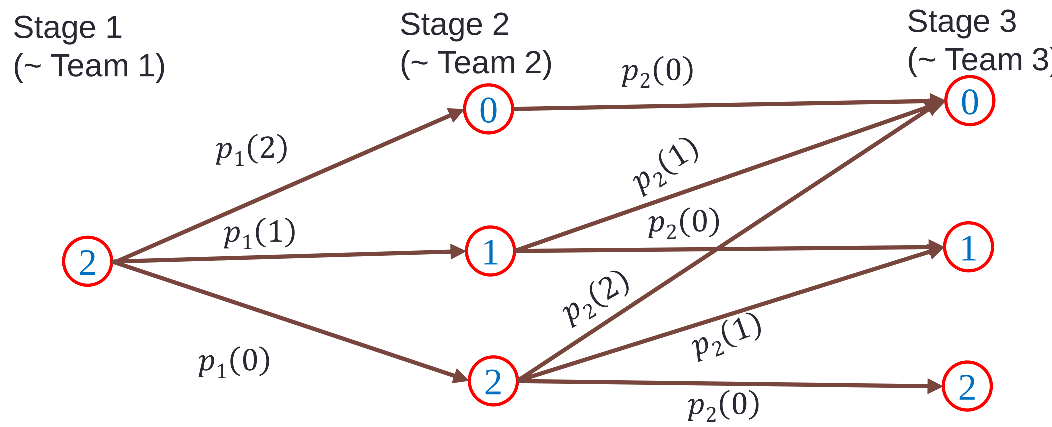
- Three teams work in parallel to complete a project
- Each team may fail
- The entire project fails if "all" 3 teams fail
- Two new scientists are available
- Adding new scientist(s) to a team reduces its failure probability (as shown)

Question: How many scientists should join each team to minimize the project's failure probability?

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30

DP Formulation

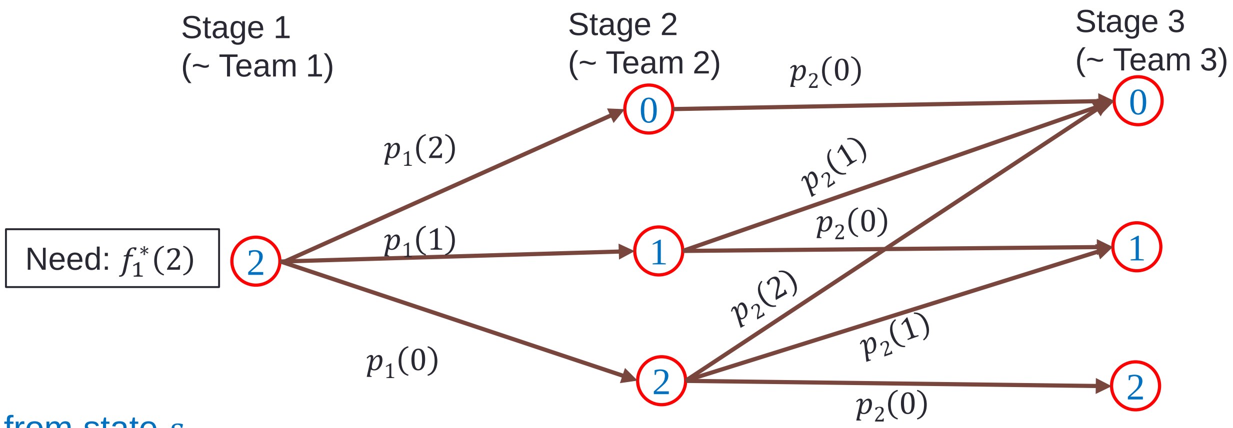
No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



1. Stages
   - One stage per research team (representing decision making regarding how many to be allocated to the team)
2. States
   - Number of unassigned scientists available for the current and subsequent stages (take values in $\{0,1,2\}$)
   - change of states between stages depend on decision
   - can only go from "higher" states to "lower" ones (cannot go from 0 to 2)
3. Decision variables/Policy (at each stage):
   - x_n = Number of scientists assigned to the team associated with stage n
4. Immediate cost of decision x_n is $p_n(x_n)$: Probability of failure of the team associated with stage n if x_n scientists are assigned to it
5. Optimality criterion: Minimize probability of failure of all three teams

Value Function

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_n(s_n, x_n)$ = Probability of failure starting from state s_n in stage n by allocating x_n to team n and making an optimum allocation thereafter

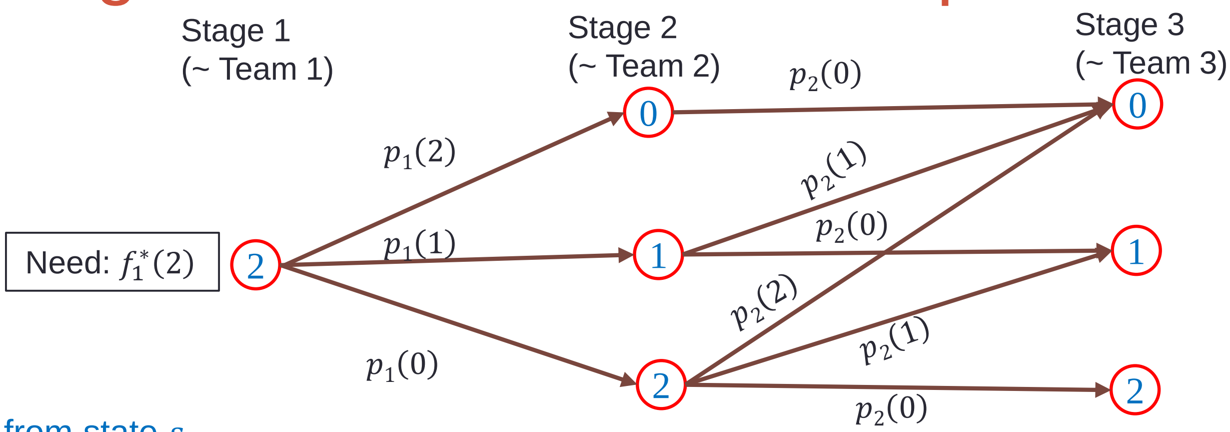
$$f_n^*(s_n) = \min_{x_n \in \{0,\dots,s_n\}} f_n(s_n, x_n)$$

$$x_n^*(s_n) = \underset{x_n \in \{0,\dots,s_n\}}{argmin} \; f_n(s_n, x_n)$$

Need: $f_1^*(2)$

Value Function: last stage and recursive relationship

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_n(s_n, x_n)$ = Probability of failure starting from state s_n in stage n by allocating x_n to team n and making an optimum allocation thereafter

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

$$x_n^*(s_n) = \underset{x_n \in \{0,\ldots,s_n\}}{argmin} f_n(s_n, x_n)$$

stage 3:

$$f_3^*(s_3) = \min_{x_3 \in \{0,\ldots,s_3\}} \{p_3(x_3)\}$$

$$x_3^*(s_3) = \underset{x_3 \in \{0,\ldots,s_3\}}{argmin} \{p_3(x_3)\}$$

stage 2:

$f_n(s_n, x_n) = p_n(x_n) \times f_{n+1}^*(s_n - x_n)$ for $n = 2,1$

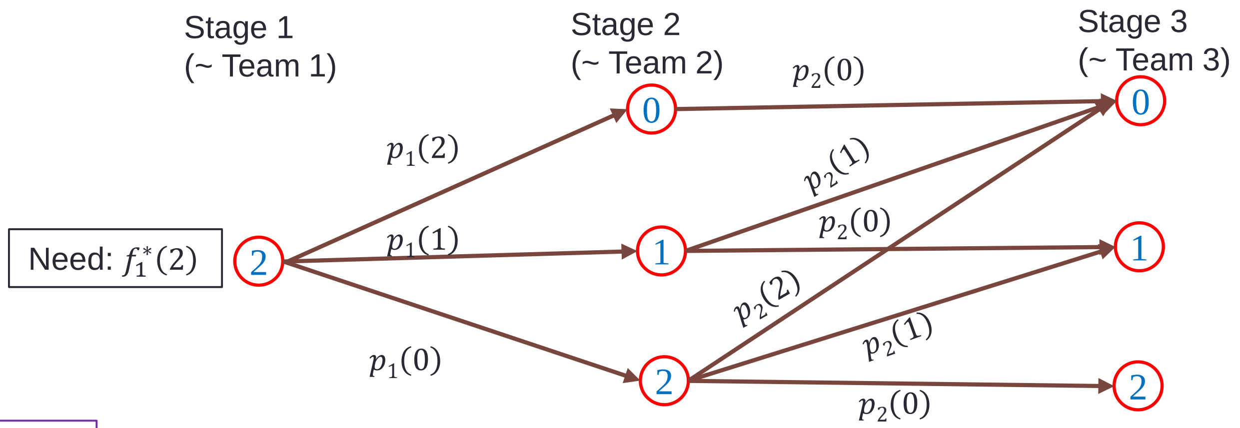
$f_n^*(s_n) = \min_{x_n \in \{0,\ldots,s_n\}} f_n(s_n, x_n)$ for $n = 2,1$

$x_n^*(s_n) = \underset{x_n \in \{0,\ldots,s_n\}}{argmin} f_n(s_n, x_n)$ for $n = 2,1$

$f_n(s_n, x_n) =$
Probability of failure of team n with x_n allocated scientists (immediate cost)
$\times$
the probability of failure if we do optimal allocation of the remaining scientists among the remaining stages (future cost)

Solution Procedure: backward induction

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



stage 3:

$$f_3^*(s_3) = \min_{x_3 \in \{0,\ldots,s_3\}} \{p_3(x_3)\}$$
$$x_3^*(s_3) = \underset{x_3 \in \{0,\ldots,s_3\}}{argmin} \{p_3(x_3)\}$$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

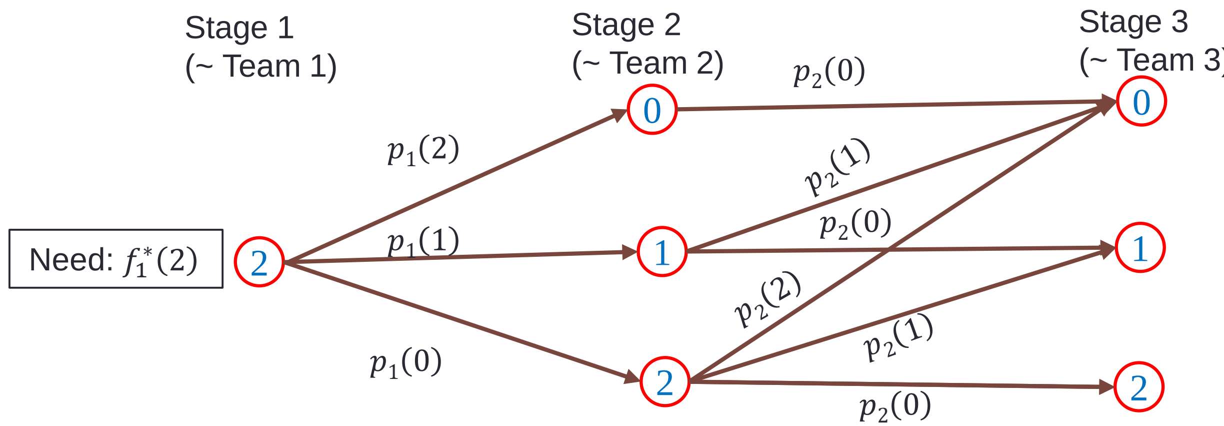
$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

States (s_3)	$f_3^*(s_3)$	$x_3^*(s_3)$
0	0.8	0
1	0.5	1
2	0.3	2

Solution Procedure: backward induction

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



stage 2:

$$f_n(s_n, x_n) = p_n(x_n) \times f_{n+1}^*(s_n - x_n) \text{ for } n = 2,1$$
$$f_n^*(s_n) = \min_{x_n \in \{0,\ldots,s_n\}} f_n(s_n, x_n) \text{ for } n = 2,1$$
$$x_n^*(s_n) = \underset{x_n \in \{0,\ldots,s_n\}}{argmin} f_n(s_n, x_n) \text{ for } n = 2,1$$

$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

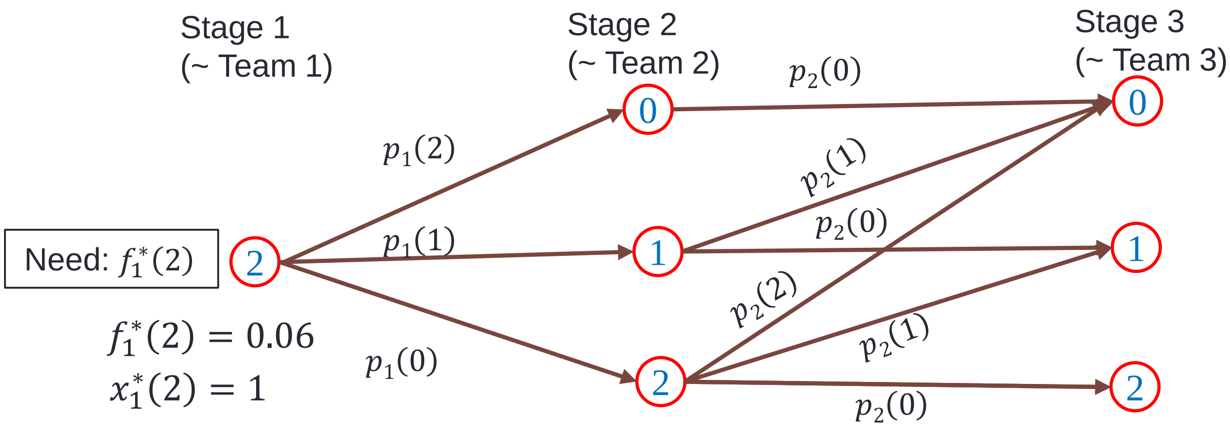
$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

s_2	x_2	$f_2(s_2, x_2)$	$f_2^*(s_2)$	x_2^*
0	0	$0.6 \times f_3^*(0) =$ 0.48	0.48	0
1	0	$0.6 \times f_3^*(1) =$ 0.3	0.3	0
	1	$0.4 \times f_3^*(0) =$ 0.32		
2	0	$0.6 \times f_3^*(2) =$ 0.18		
	1	$0.4 \times f_3^*(1) =$ 0.2		
	2	$0.2 \times f_3^*(0) =$ 0.16	0.16	2

Solution Procedure: backward induction

No. of new Scientist s	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

stage 1:

$f_n(s_n, x_n) = p_n(x_n) \times f_{n+1}^*(s_n - x_n)$ for $n = 2,1$

$f_n^*(s_n) = \min_{x_n \in \{0,\dots,s_n\}} f_n(s_n, x_n)$ for $n = 2,1$

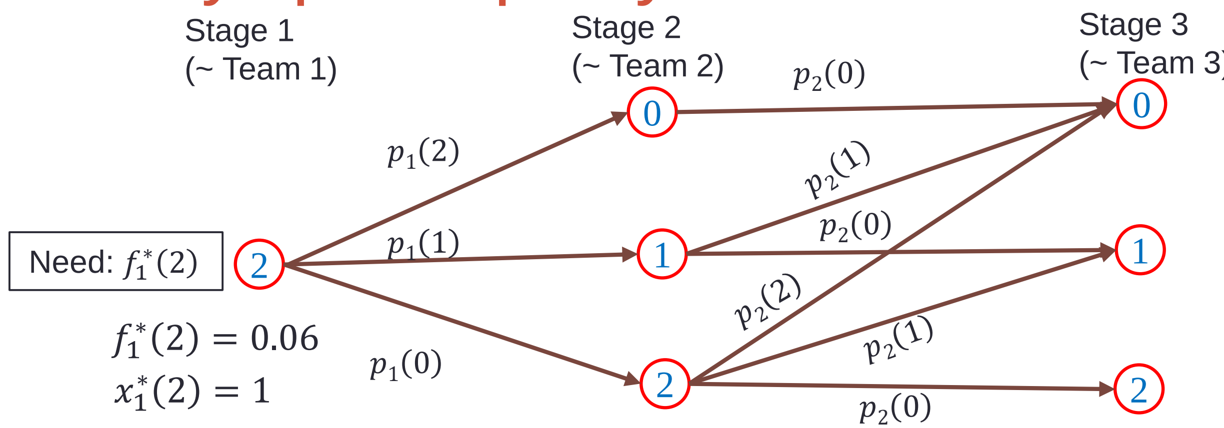
$x_n^*(s_n) = \underset{x_n \in \{0,\dots,s_n\}}{argmin}\ f_n(s_n, x_n)$ for $n = 2,1$

s_1	x_1	$f_1(s_1, x_1)$	$f_1^*(s_1)$	x_1^*
2	0	$0.4 \times f_2^*(2) = 0.064$		
	1	$0.2 \times f_2^*(1) = 0.06$	0.06	1
	2	$0.15 \times f_2^*(0) = 0.072$		

Min possible probability of failure is 0.06

Solution Procedure: Identify optimal policy

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

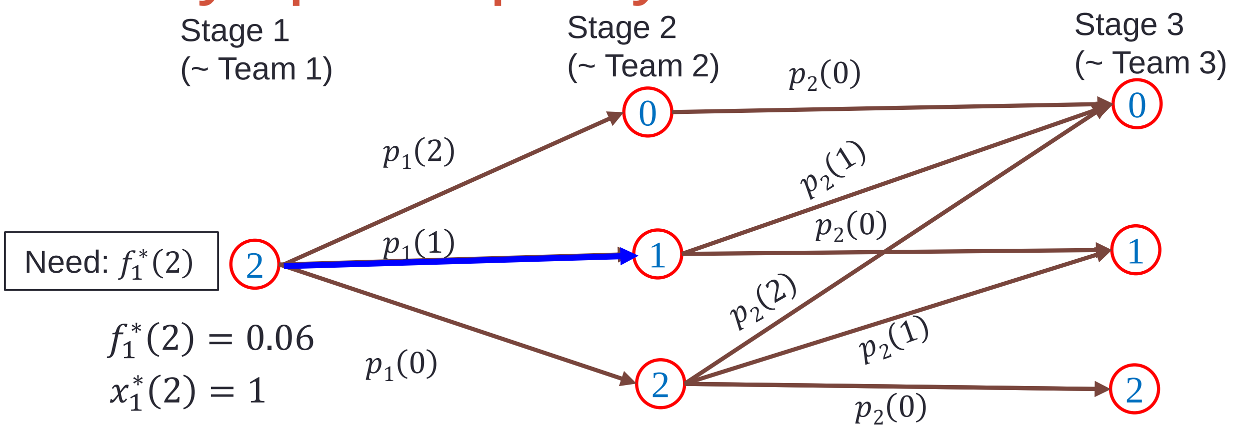
$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

Min possible probability of failure is 0.06

The optimal policy (follow the best value): start from the initial state, identify the best decision to the next stage and proceed

Solution Procedure: Identify optimal policy

No. of new Scientist s	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_1^*(2) = 0.06$
$x_1^*(2) = 1$

$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

stage 1:

$f_n(s_n, x_n) = p_n(x_n) \times f_{n+1}^*(s_n - x_n)$ for $n = 2,1$

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

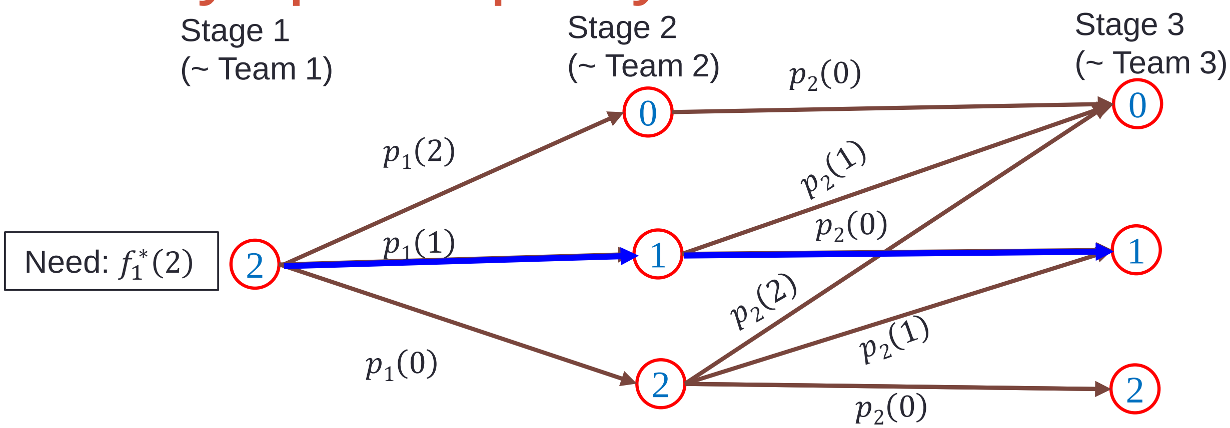
$x_n^*(s_n) = \arg\min_{x_n \in \{0,\ldots,s_n\}} f_n(s_n, x_n)$ for $n = 2,1$

s_1	x_1	$f_1(s_1, x_1)$	$f_1^*(s_1)$	x_1^*
2	0	$0.4 \times f_2^*(2) = 0.064$		
	1	$0.2 \times f_2^*(1) = 0.06$	0.06	1
	2	$0.15 \times f_2^*(0) = 0.072$		

Min possible probability of failure is 0.06

Solution Procedure: Identify optimal policy

No. of new Scientist s	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

stage 2:

$f_n(s_n, x_n) = p_n(x_n) \times f_{n+1}^*(s_n - x_n)$ for $n = 2,1$

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

$x_n^*(s_n) = \underset{x_n \in \{0,\ldots,s_n\}}{argmin} f_n(s_n, x_n)$ for $n = 2,1$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

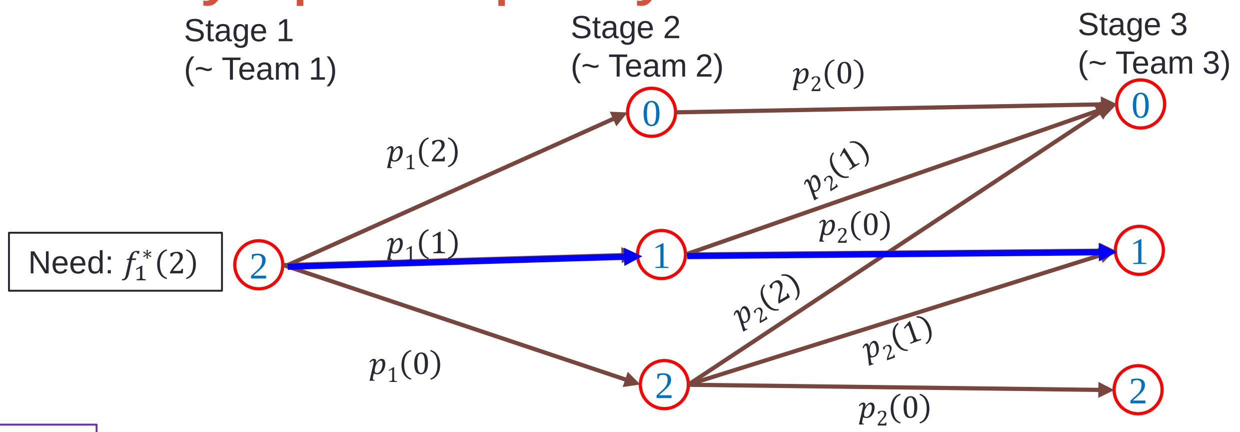
$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

s_2	x_2	$f_2(s_2, x_2)$	$f_2^*(s_2)$	x_2^*
0	0	$0.6 \times f_3^*(0) = 0.48$	0.48	0
1	0	$0.6 \times f_3^*(1) = 0.3$	0.3	0
	1	$0.4 \times f_3^*(0) = 0.32$		
2	0	$0.6 \times f_3^*(2) = 0.18$		
	1	$0.4 \times f_3^*(1) = 0.2$		
	2	$0.2 \times f_3^*(0) = 0.16$	0.16	2

Solution Procedure: Identify optimal policy

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



stage 3:

$$f_3^*(s_3) = \min_{x_n \in \{0,\ldots,s_n\}} \{p_3(x_3)\}$$

$$x_3^*(s_3) = \underset{x_n \in \{0,\ldots,s_n\}}{argmin} \{p_3(x_3)\}$$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

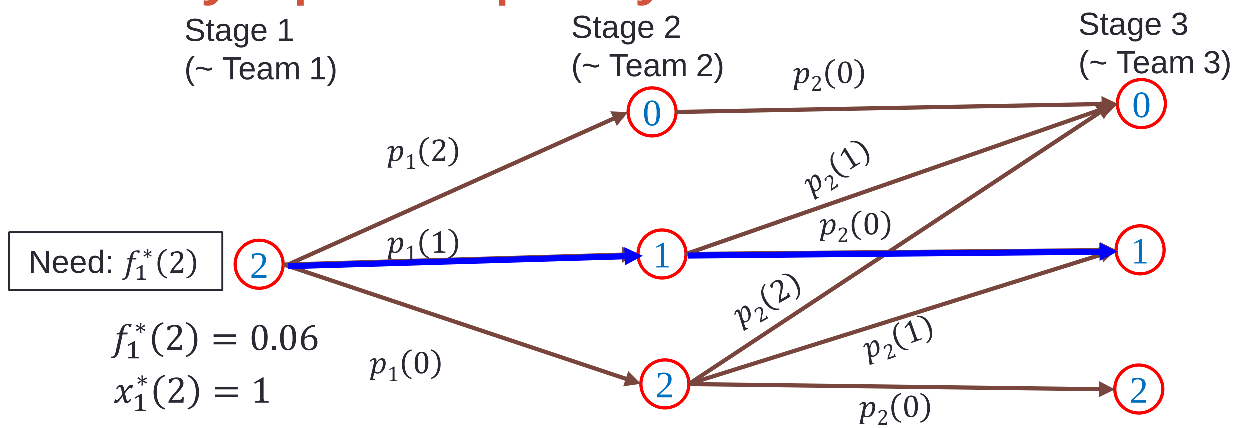
$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

States (s_3)	$f_3^*(s_3)$	$x_3^*(s_3)$
0	0.8	0
1	0.5	1
2	0.3	2

Solution Procedure: Identify optimal policy

No. of new Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30



$f_1^*(2) = 0.06$
$x_1^*(2) = 1$

$f_2^*(0) = 0.48$
$x_2^*(0) = 0$

$f_2^*(1) = 0.3$
$x_2^*(1) = 0$

$f_2^*(2) = 0.16$
$x_2^*(2) = 2$

$f_3^*(0) = 0.8$
$x_3^*(0) = 0$

$f_3^*(1) = 0.5$
$x_3^*(1) = 1$

$f_3^*(2) = 0.3$
$x_3^*(2) = 2$

Min possible probability of failure is 0.06

The optimal policy (follow the best value): start from the initial state, identify the best decision to the next stage and proceed

Optimal policy: one scientist for Team 1 and Team 3

Dynamic Programming: Standard Procedure

1. Identify Stages, States, Decision Variables, Optimality Criterion
2. Find the optimal value function of the states in the final stage
3. Write the value function of the states in the n'th stage as a recursive relationship
   - the value function should depend only on the optimal values of states in future stages

     [DP algorithmic technique is applicable only if this property is satisfied]
4. Use backward induction to compute the optimal value function and the best decision for every state
5. Identify the optimal policy by tracing the best decision from the initial state

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: $f_1^*(255)$

Need: $f_1^*(255)$

Value function: final 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**

$f_n(s_n, x_n)$ = 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: $f_1^*(255)$

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

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

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

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