BDC5101

Deterministic Operations Research Models

Zhenyu Hu

Semester II, 2018/2019

National University of Singapore

Dynamic Programming

- Overview of DP
- Deterministic DP
- Stochastic DP
- Further Applications

Overview of DP

AlphaGo

Source: https://qz.com/993147/the-awful-frustration-of-a-teenage-go-champion-playing-googles-alphago/

AlphaGo

NATURE | ARTICLE

日本語要約

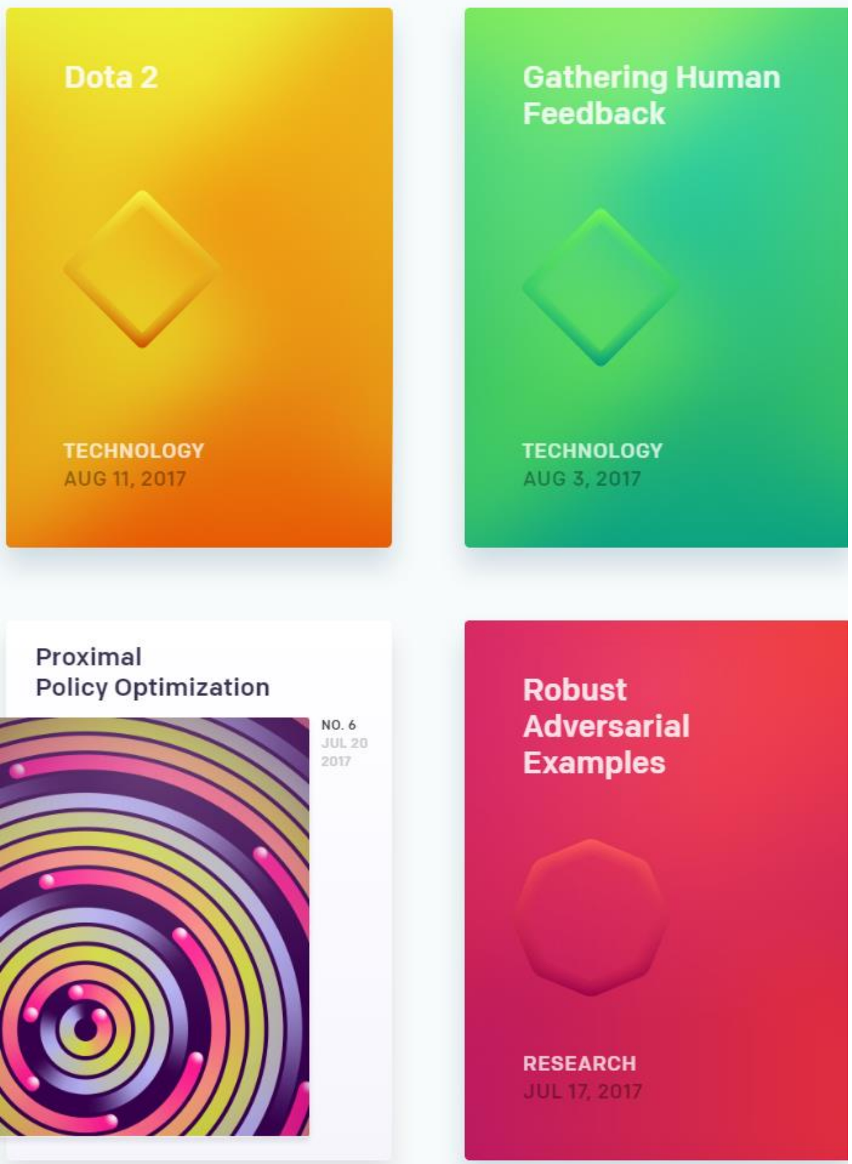
Mastering the game of Go without human knowledge

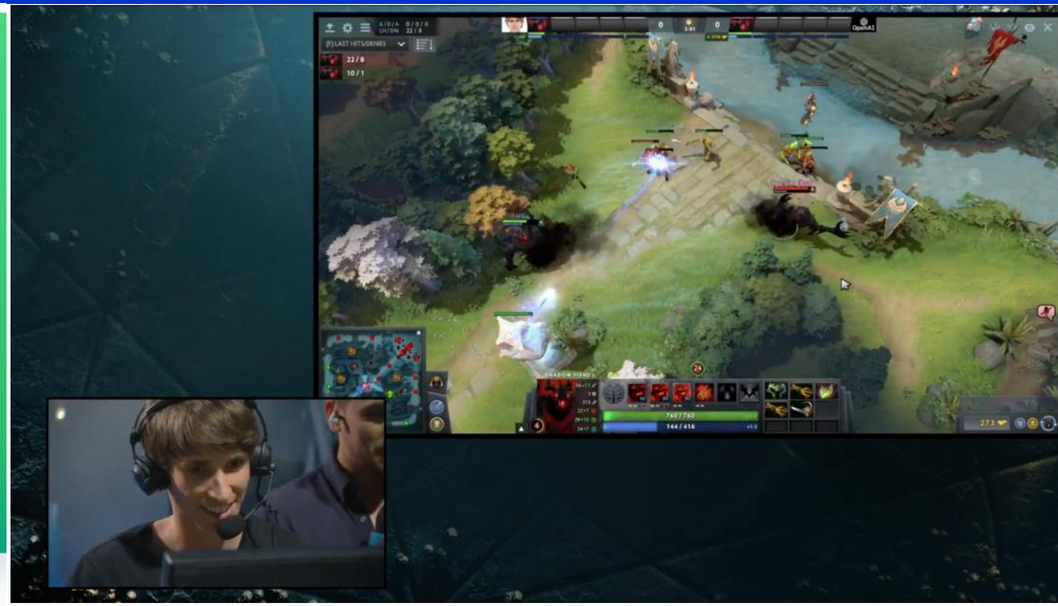
David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Affiliations | Contributions | Corresponding author

Nature **550**, 354–359 (19 October 2017) | doi:10.1038/nature24270
Received 07 April 2017 | Accepted 13 September 2017 | Published online 18 October 2017

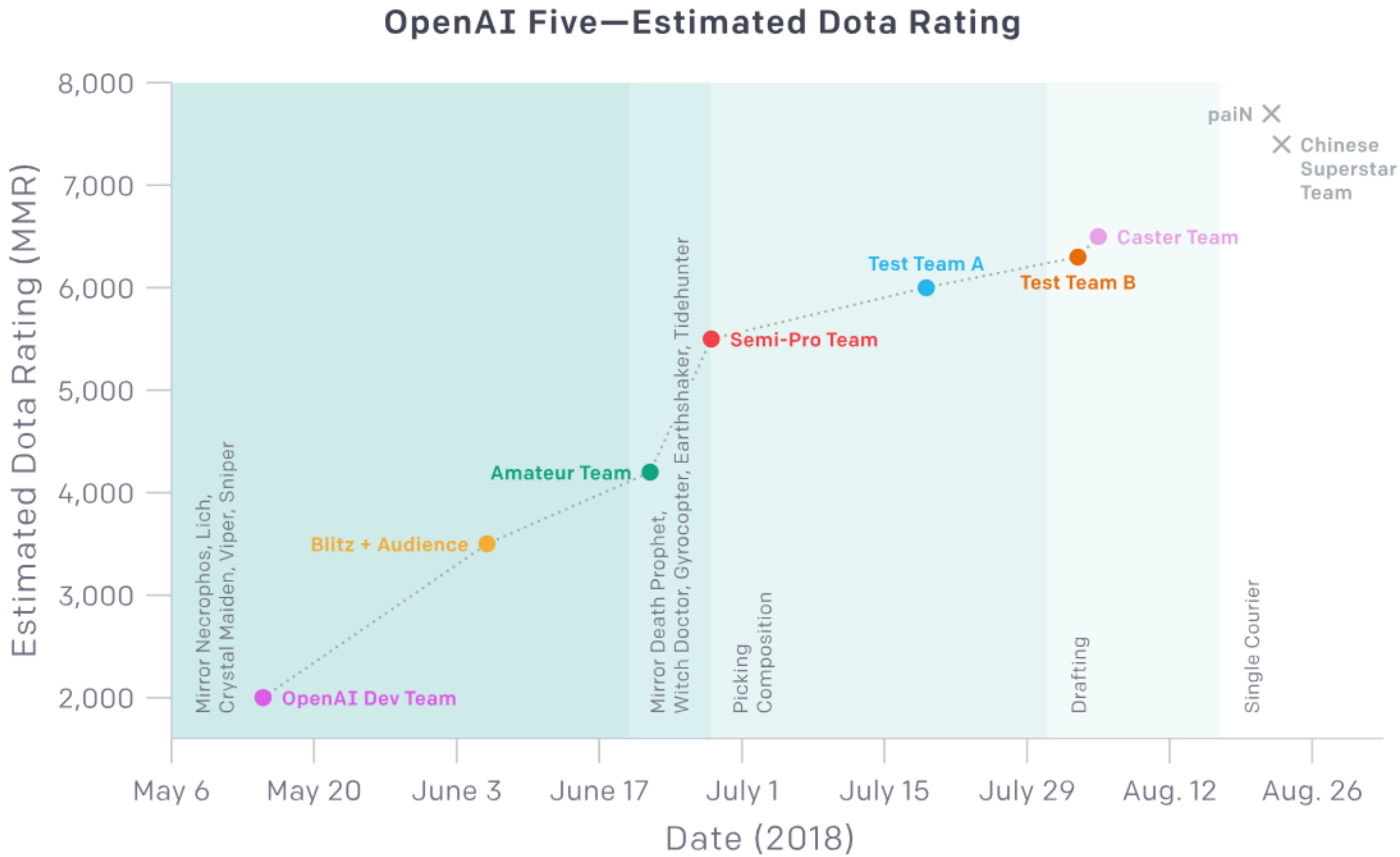
OpenAI





Source: https://blog.openai.com/dota-2/

OpenAI



Five's progress from extremely restricted Dota against low-ranked players in May to competing at Dota with the latest restriction set.

Source: https://openai.com/blog/the-international-2018-results/

Deterministic vs. Stochastic DP

- **Deterministic**
  - Used as a computational tool as well as modeling tool to various problems
  - All solutions can be determined beforehand
- **Stochastic DP**
  - Commonly used to model sequential decision problems
  - Solutions are determined sequentially as the randomness realize

Deterministic DP

A Computational Point of View

- **Fibonacci Numbers**

$$\mathbf{0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots}$$

- **In general**
  - $\boldsymbol{F_n = F_{n-1} + F_{n-2}}$
  - $\boldsymbol{F_0 = 0, F_1 = 1.}$

A Computational Point of View

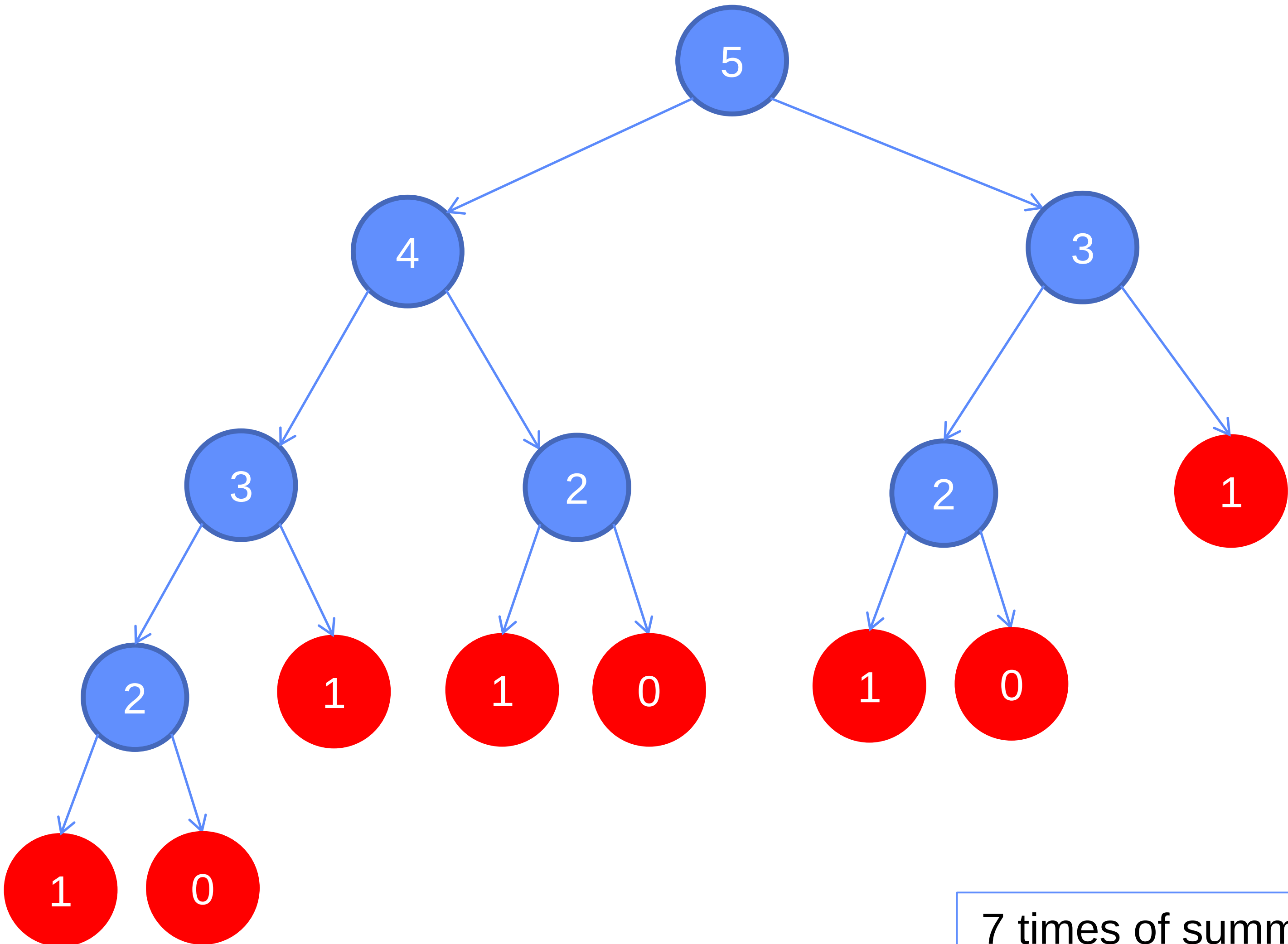
- **Algorithm 1**

$\underline{\textsc{RecFibo}(n):}$
    if $(n < 2)$
        return n
    else
        return $\textsc{RecFibo}(n-1) + \textsc{RecFibo}(n-2)$

- **Algorithm 2**

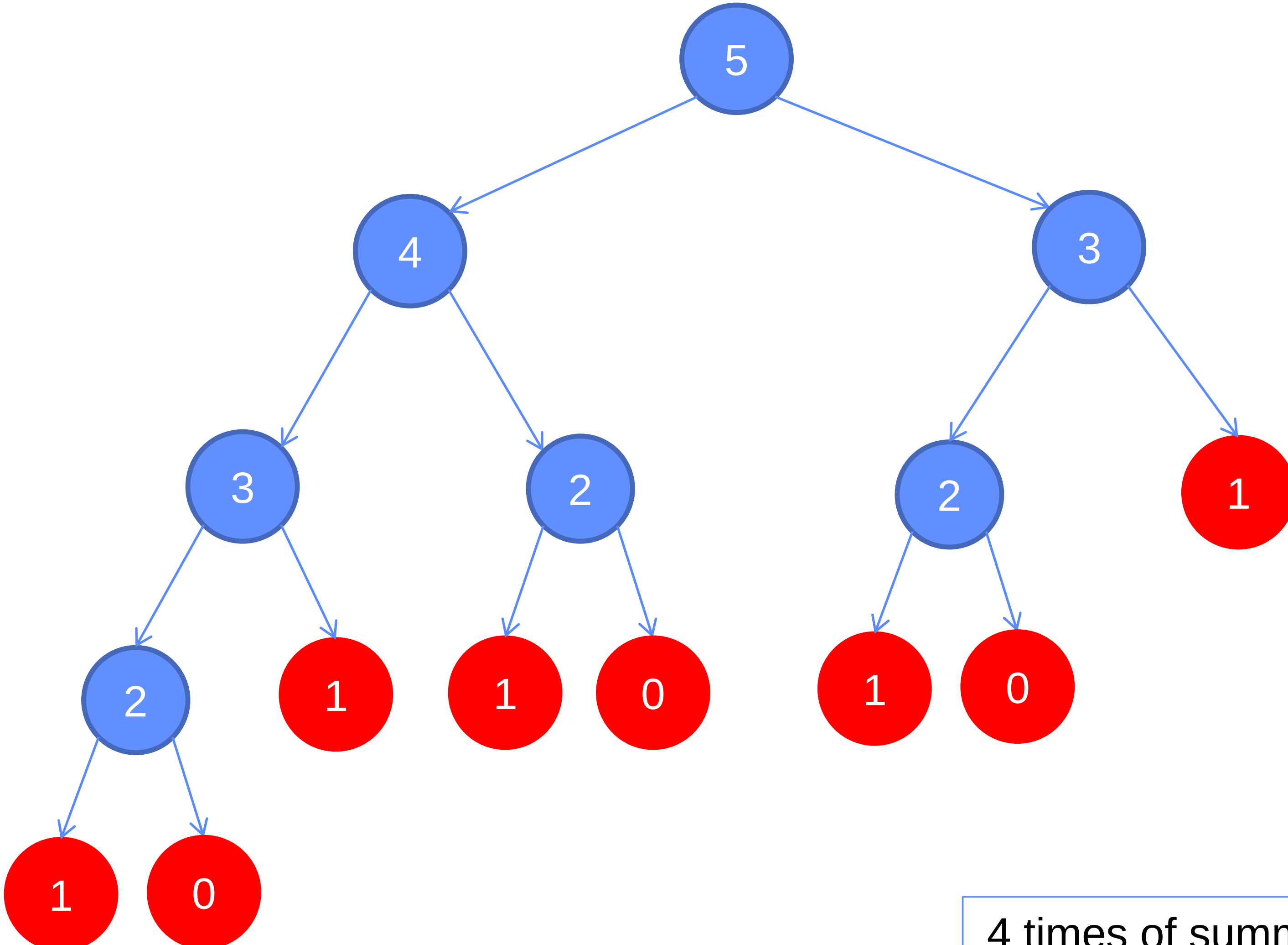
$\underline{\textsc{IterFibo}(n):}$
    $F[0] \leftarrow 0$
    $F[1] \leftarrow 1$
    for $i \leftarrow 2$ to n
        $F[i] \leftarrow F[i-1] + F[i-2]$
    return $F[n]$

First Algorithm: Tree View $n = 5$



7 times of summation operations

Second Algorithm: Tree View $n = 5$



4 times of summation operations

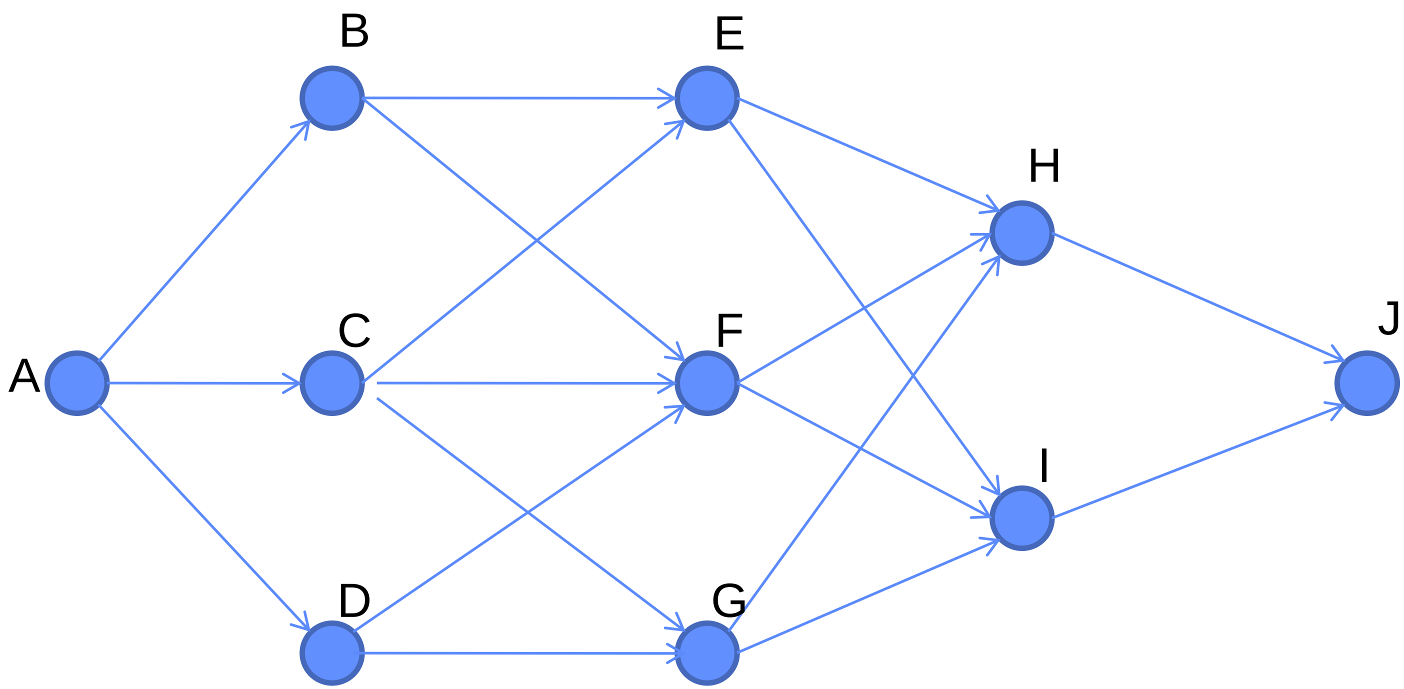
Key Idea: Value Function

- $F_k, 0 \le k \le n$ is called the values or value functions, k is called the stage.
  - Our goal is to solve the value for final stage F_n.
  - $F_k, 0 \le k \le n$ represents a (optimal) value for a sub-problem at stage k.
  - The crux of DP: all information we need is contained in F_k; after one solved sub-problem at stage k, one saves value in F_k and never revisits how it is solved.
- The relation used for computation:

$$F_k = F_{k-1} + F_{k-2}, 2 \le k \le n$$

is called the Bellman Equation with $F_0 = 0, F_1 = 1$ called the boundary conditions.

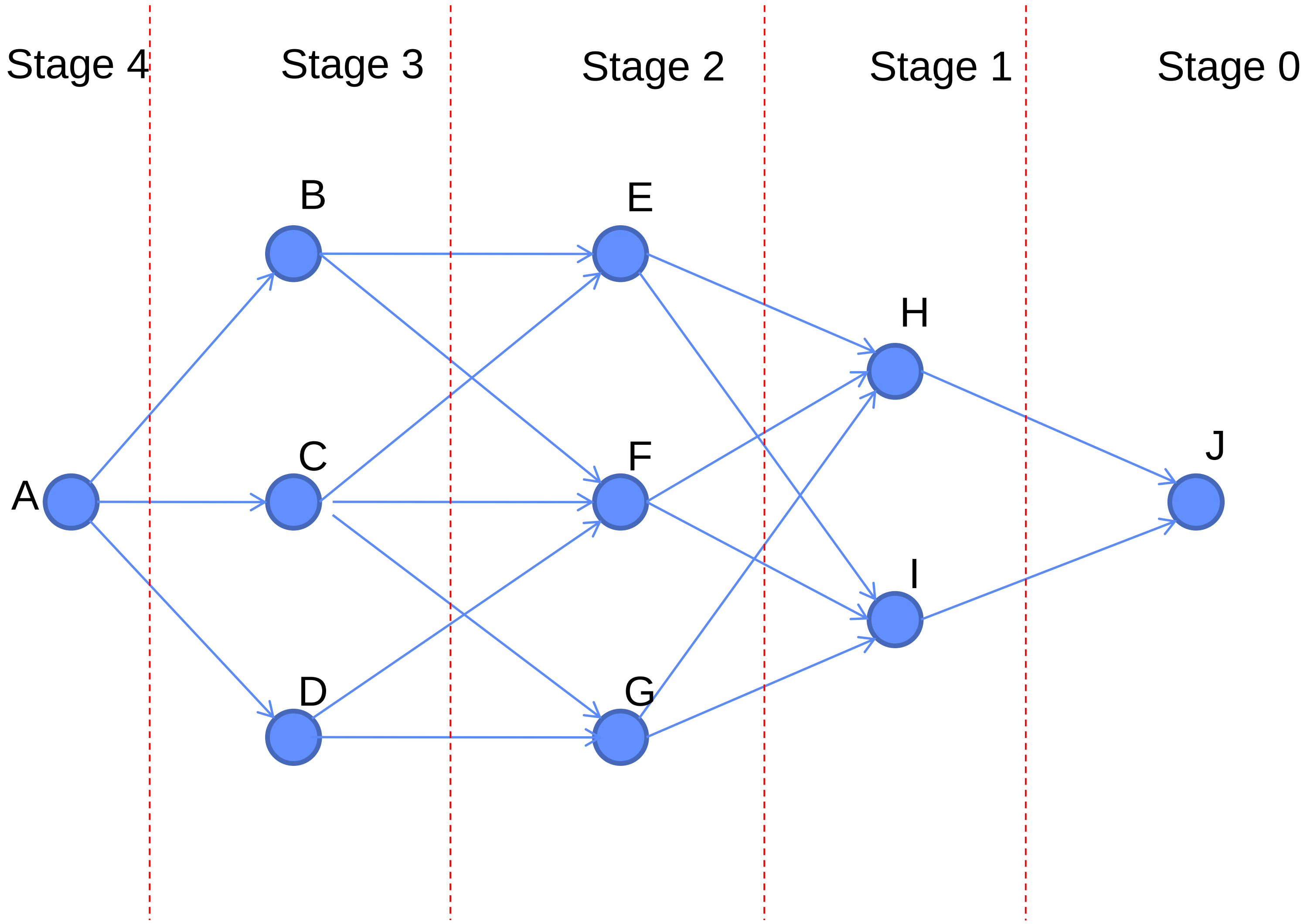
Shortest Path Problem



Cost on the Edges

	B	C	D	E	F	G	H	I	J
A	22	19	15						
B				16	15				
C				22	16	22			
D					19	22			
E							16	18	
F							18	11	
G							15	13	
H									23
I									29

Decomposition

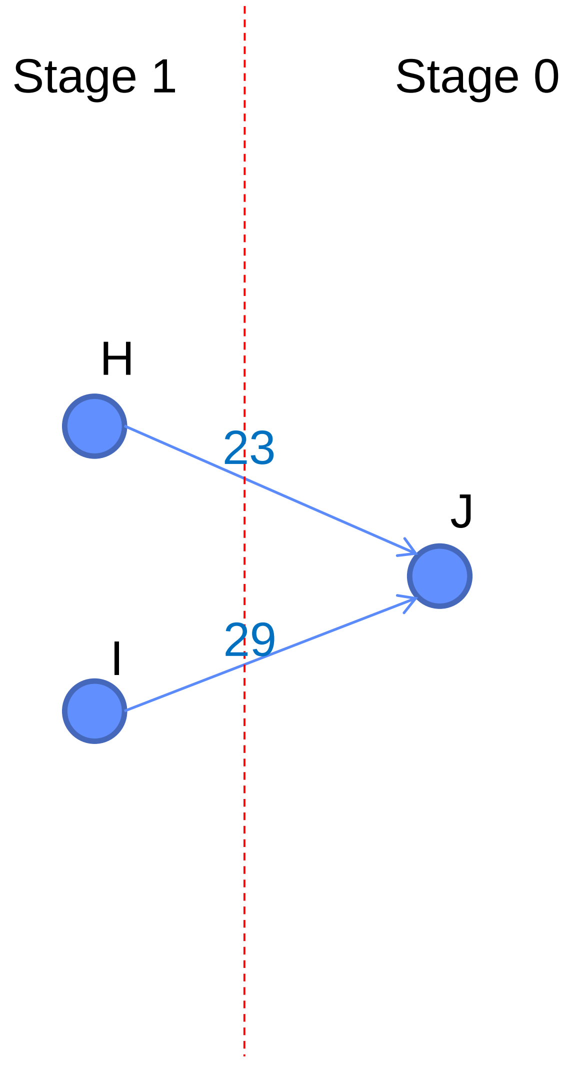


Decomposition

What is the shortest path from H to J?

What is the shortest path from I to J?

How do we save them down?

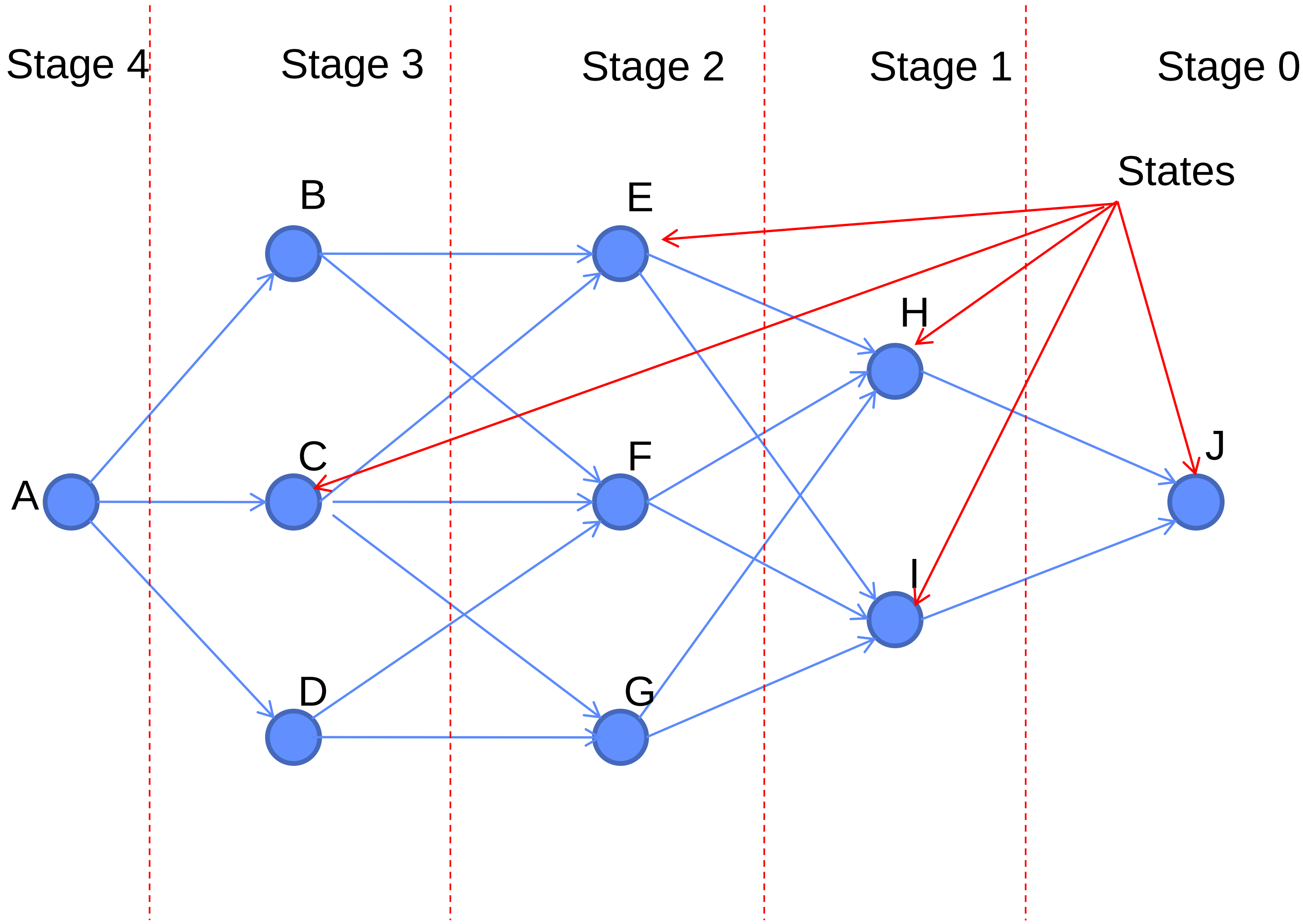


State

- **State: An information set upon which the decisions are based.**

- **Finding a good state representation is the art of dynamic programming**

State



Value Function

- **The value function now depends on both stage and state:**

$$V(k, i)$$

Stage k ***State i***

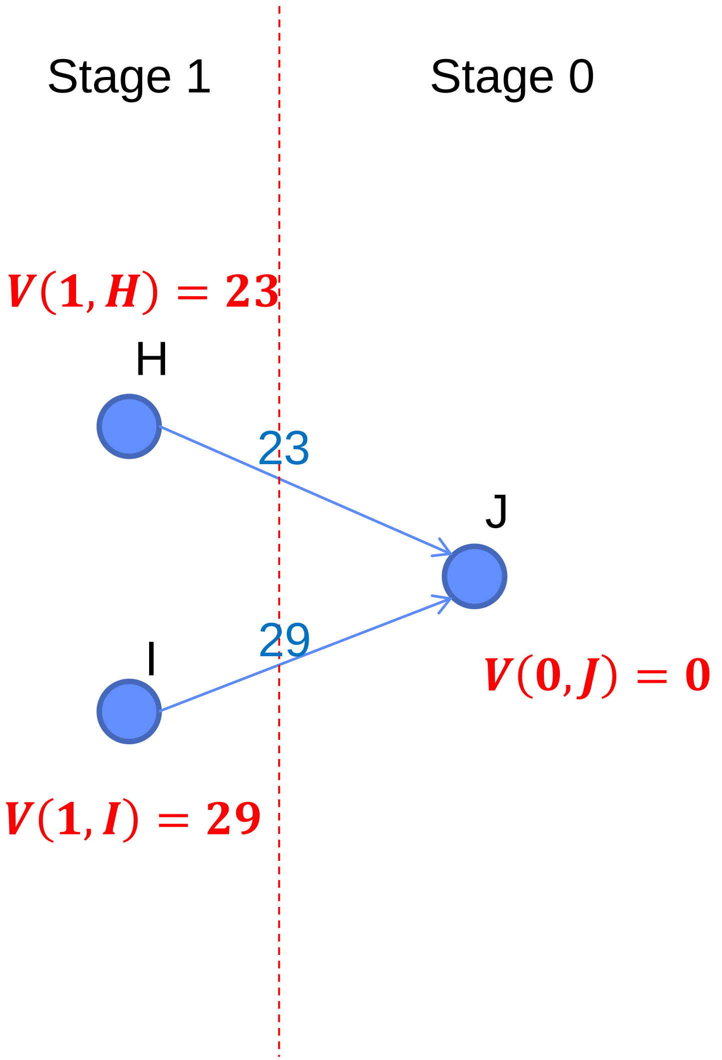
Some times also written as $V_k(i)$

Stage 1

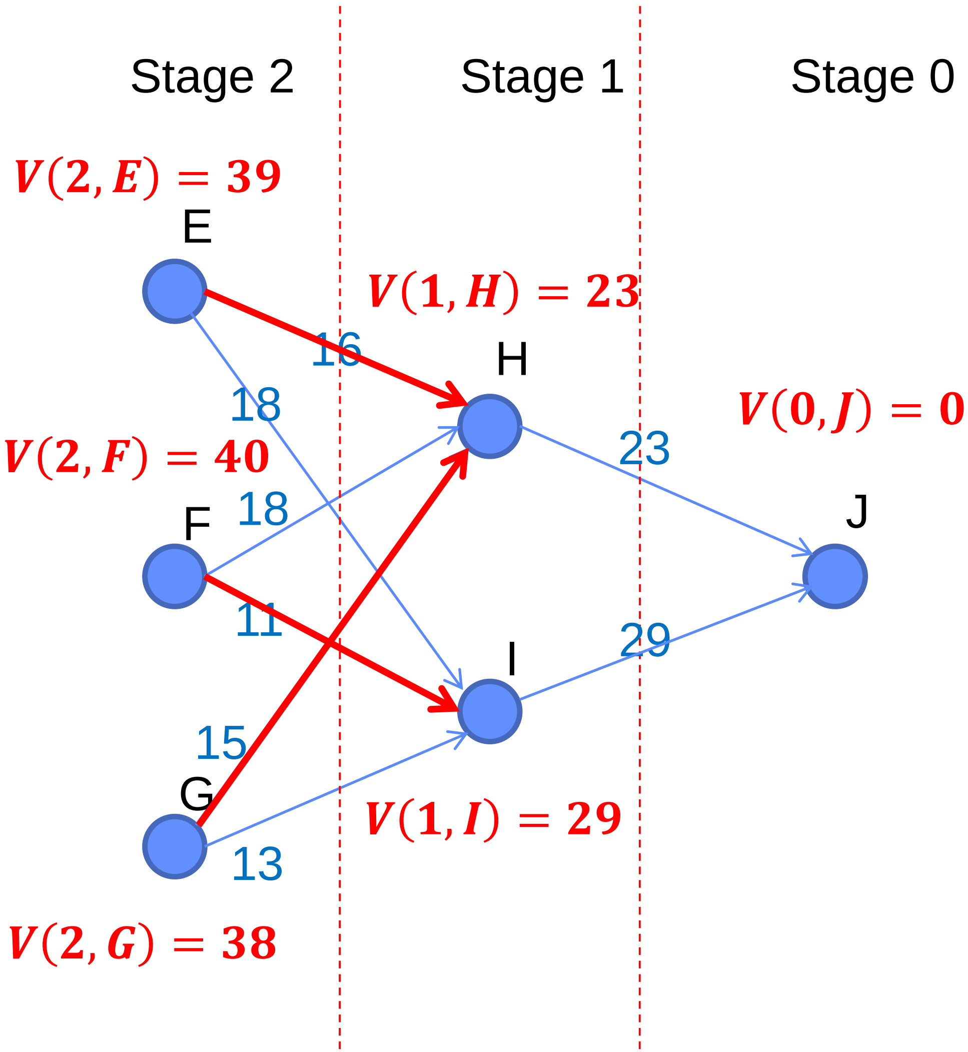
What is the shortest path from H to J?

What is the shortest path from I to J?

How do we save them down?

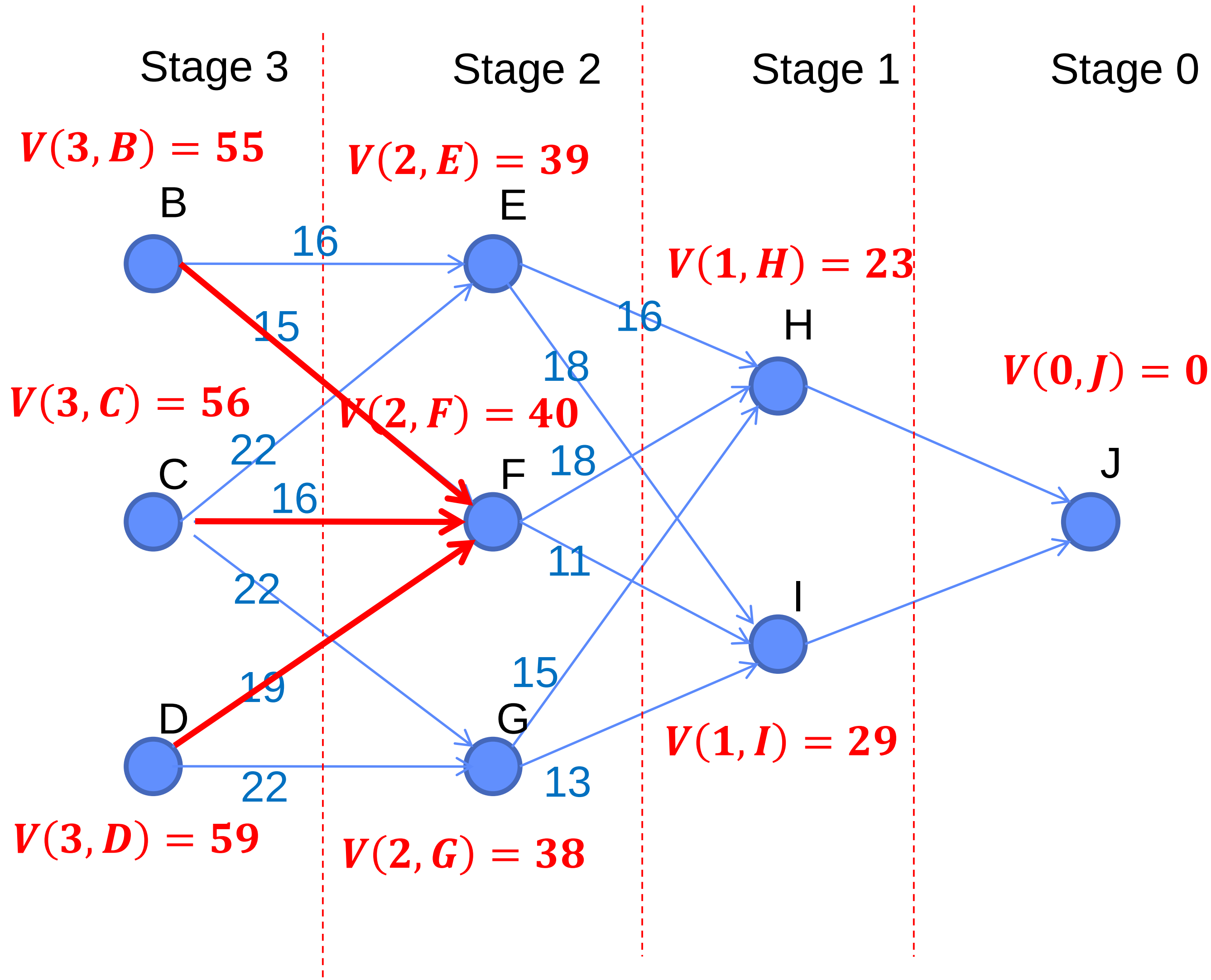


Stage 2



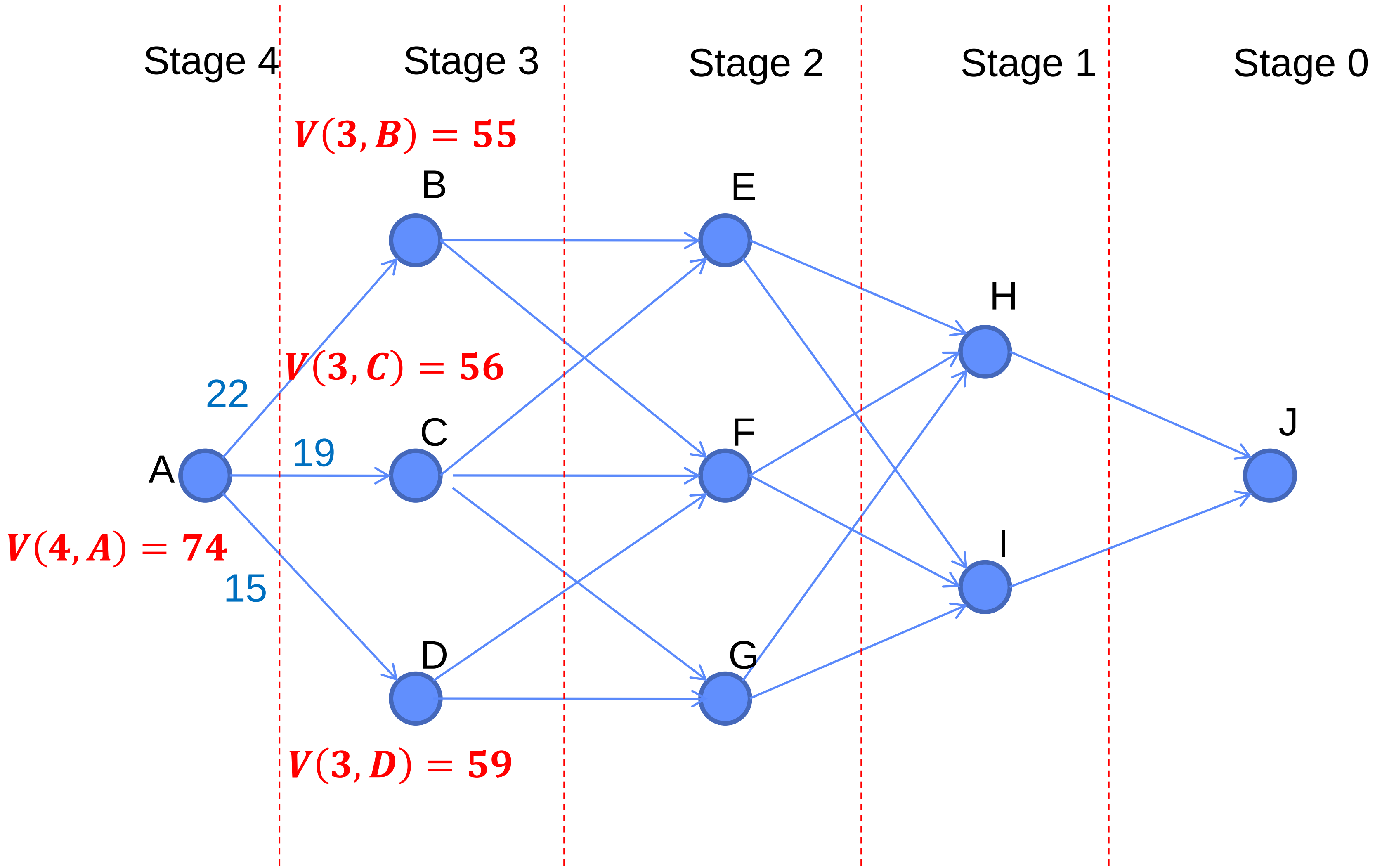
Stage 2
Stage 1
Stage 0
$V(2, E) = 39$
E
$V(1, H) = 23$
16
H
18
$V(0, J) = 0$
$V(2, F) = 40$
23
18
F
J
11
29
I
15
G
$V(1, I) = 29$
13
$V(2, G) = 38$

Stage 3



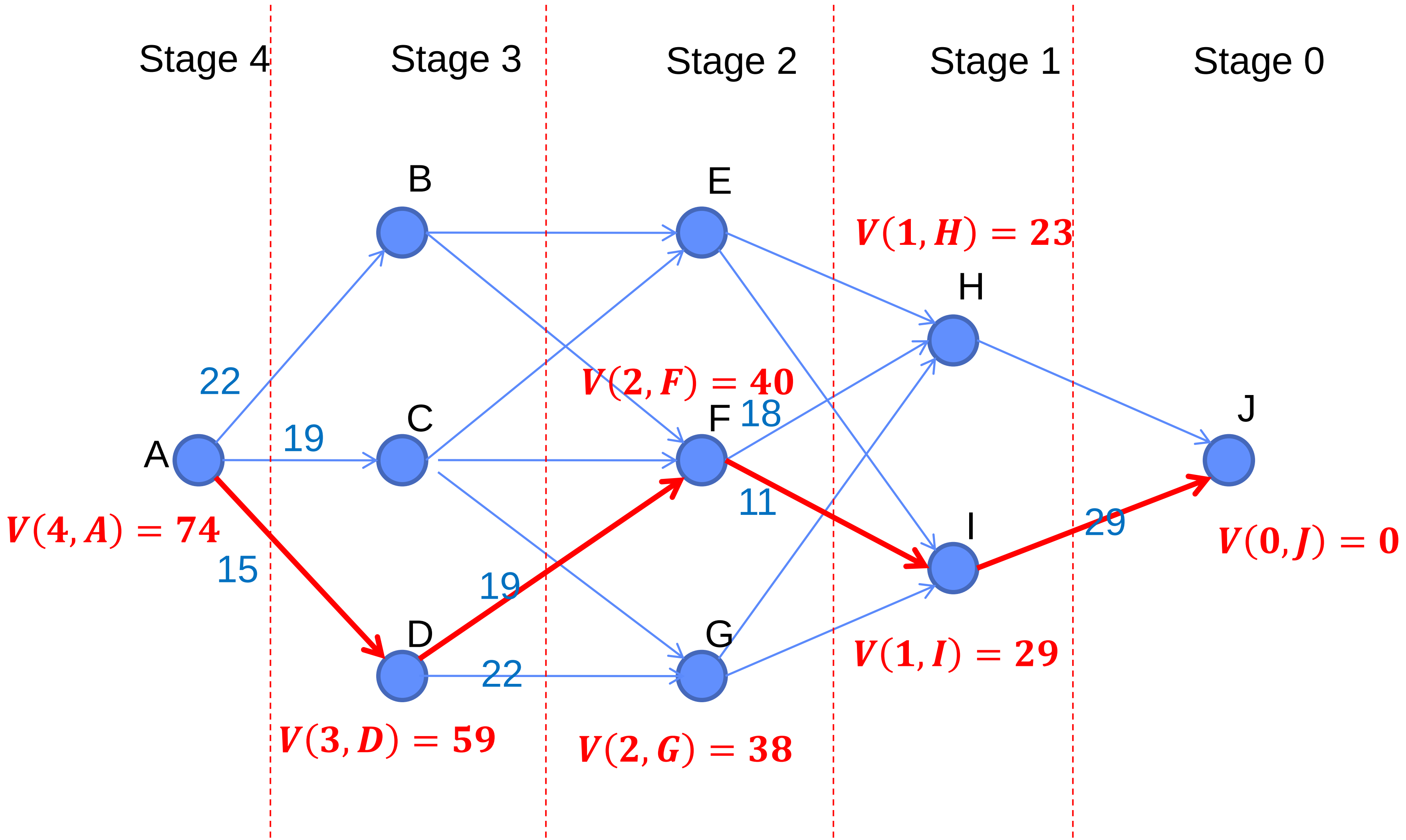
Stage 3
Stage 2
Stage 1
Stage 0
$V(3,B) = 55$
$V(2,E) = 39$
B
E
16
15
$V(1,H) = 23$
16
H
18
$V(0,J) = 0$
$V(3,C) = 56$
$V(2,F) = 40$
22
18
J
C
F
16
11
22
I
15
19
D
G
22
13
$V(1,I) = 29$
$V(3,D) = 59$
$V(2,G) = 38$

Final Stage



Stage 4
Stage 3
Stage 2
Stage 1
Stage 0
$V(3, B) = 55$
B
E
H
$V(3, C) = 56$
22
19
C
F
J
A
$V(4, A) = 74$
15
I
D
G
$V(3, D) = 59$

Solution



Stage 4
Stage 3
Stage 2
Stage 1
Stage 0
A
B
C
D
E
F
G
H
I
J
22
19
15
19
22
18
11
29
$V(4, A) = 74$
$V(3, D) = 59$
$V(2, F) = 40$
$V(2, G) = 38$
$V(1, H) = 23$
$V(1, I) = 29$
$V(0, J) = 0$

Shortest Path Problem: Summary

- $V(k, i)$ can be interpreted as the shortest distance to J when there are k stages remaining and we are at node i. We want to know $V(4, A)$.

- **Bellman Equation** gives a recursive relationship between value functions and we can use **backward induction** to solve the equations:

$$\boldsymbol{V(k,i) = \min_{j,(i,j)\in \mathrm{A}} \{c(i,j) + V(k-1,\ j)\},\ V(0,J) = 0}$$

A General Deterministic DP

- Stages (period): T,…,1 (or T,…, 0)
- State: s_t, action (control): x_t
- Cost per period: $c(s_t, x_t)$
- State transition: $s_{t-1} = f(s_t, x_t)$
- Original problem:

$$\min_{x_t}\{\, c(s_T, x_T) + c(s_{T-1}, x_{T-1}) + \cdots + c(s_1, x_1) \,\}$$

- Bellman Equation:

$$\boxed{V(t, s_t) = \min_{x_t}\{\, c(s_t, x_t) + V\big(t-1, f(s_t, x_t)\big)\}}$$

Shortest Path Problem

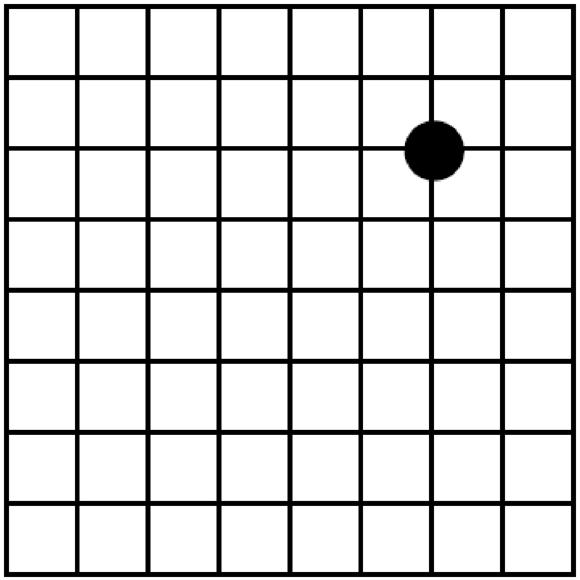
- **State:** $s_0 = J, s_1 = H, I, s_2 = E, F, G, \ldots$
- **Action (control):** $x_1 = J$, $x_2 = H, I$
- **Cost per period:** $c(s_t, x_t) = c(i, j)$ **for** $s_t = i, x_t = j$
- **State transition:** $s_{t-1} = x_t$

Summary: Components of DP

- **A problem is decomposed into *n stages*.**
- **At stage k, we will have different possible *states*.**
- **At each state, we focus *only* on the decision problem for the current stage and current state assuming all the sub-problems *before stage k* has been solved.**
- **We call a mapping from each state to a decision the *policy*.**

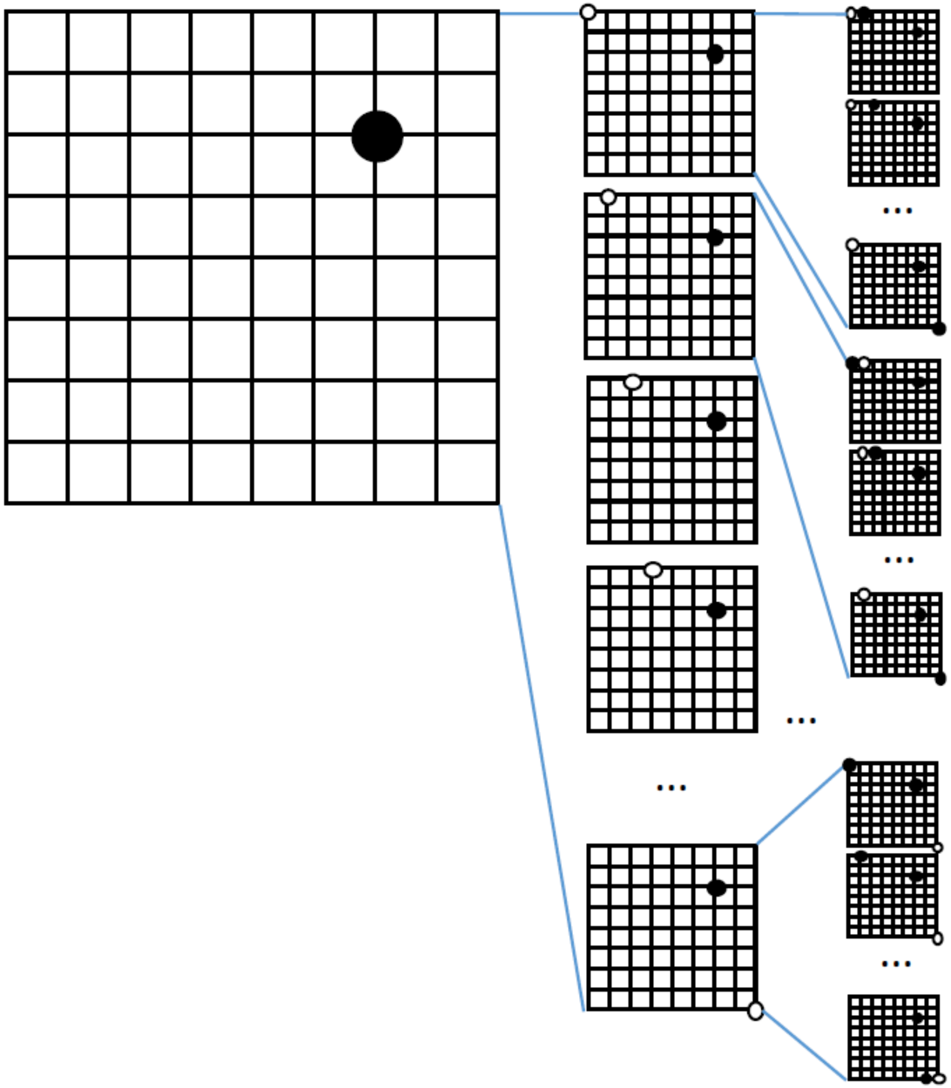
A Very Complicated Dynamic Problem: Go

- **Zero-sum, perfect-information, deterministic game**
- **State?**



$$= \begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

State Transition



Rough Estimate of Complexity

- **Average number of possible moves at each state: 250**
- **Average number of moves for a game: 150**
- **Complexity:** 250^{150} **(recall** 2^{50} **and** $50!$**; Chess:** 35^{80}**)**

Stochastic DP

Newsvendor Problem

- **Consider the newsvendor problem with the cost 40 cents and selling price $1.**

- **Demand is Bernoulli: 1 with probability ½ and 0 with probability ½.**

- **What is the optimal ordering quantity?**
  - **Do not order: cost is** 0
  - **Order 1: cost is** $0.4 + -1 \times \frac{1}{2} + 0 \times \frac{1}{2} = -0.1$ **(or profit is** 0.1**).**

Multi-Period Newsvendor Problem

- **Suppose the newsvendor can sell for two days. On each day the newsvendor has an opportunity to replenish his inventory, and the unsold copies from the first day can be used to satisfy demand in the second day.**

- **Demands D_1, D_2 are i.i.d. Bernoulli.**

- **What is the optimal ordering quantities in the first and second day respectively?**

Stochastic Optimization

- **Decision variables:** $x_1, x_2 \geq 0$

- **Objective:**
  - **First period sales:** $S_1 = \min\{x_1, D_1\}$
  - **Second period sales:** $S_2 = \min\{x_1 - S_1 + x_2, D_2\}$
  - **Minimize:** $z = c(x_1 + x_2) - pE[S_1 + S_2]$.

Stochastic Optimization

- **Demands:**

Demand (Day 1)	Demand (Day 2)	Probability
0	0	¼
0	1	¼
1	0	¼
1	1	¼

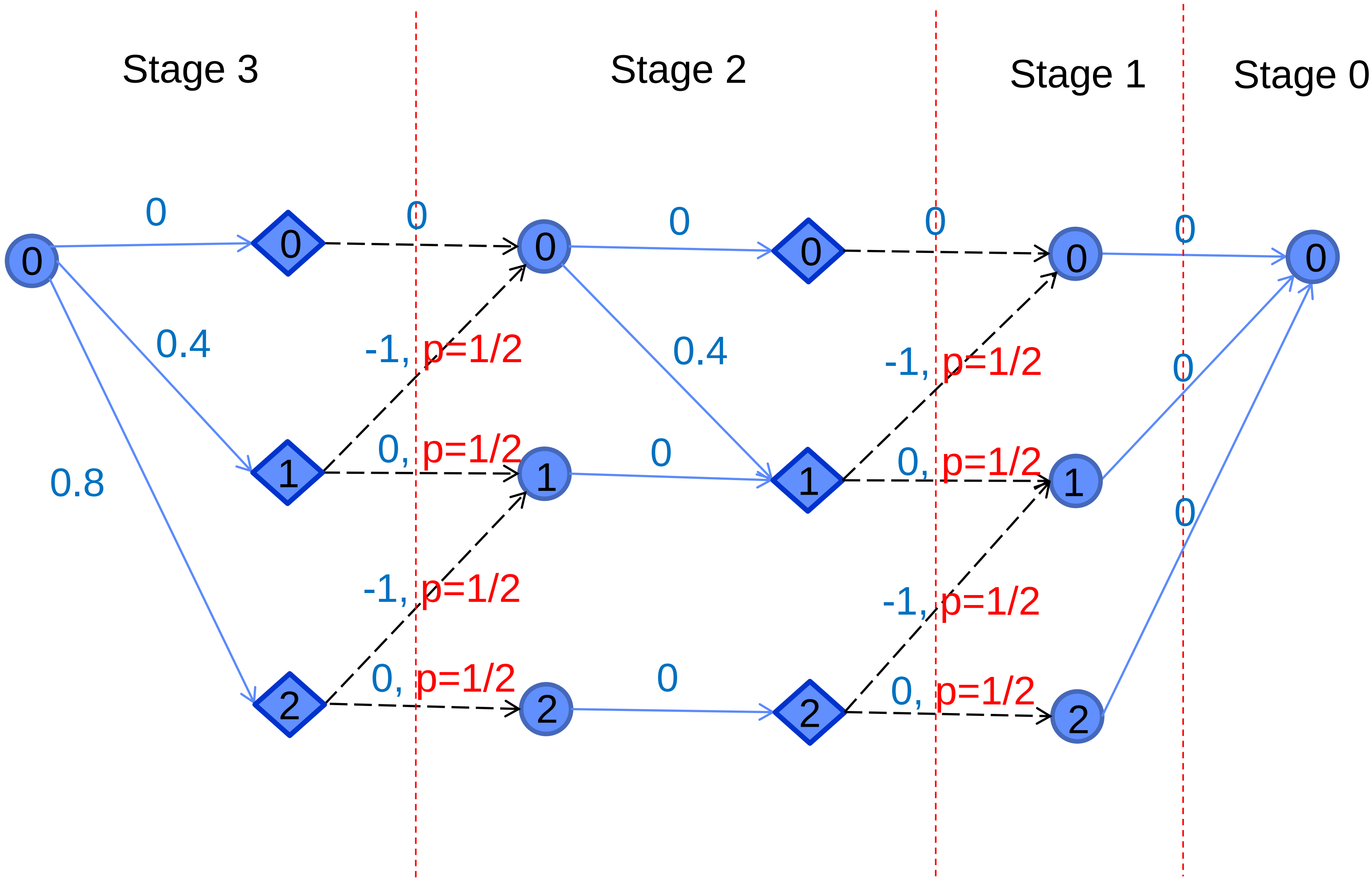
- **Decisions:**
  - $(x_1, x_2) = (0,0)$: $z = 0$
  - $(x_1, x_2) = (0,1)$: $z = 0.4 + -1 \times \frac{2}{4} + 0 \times \frac{2}{4} = -0.1$
  - $(x_1, x_2) = (1,0)$: $z = 0.4 + -1 \times \frac{3}{4} + 0 \times \frac{1}{4} = -0.35$
  - $(x_1, x_2) = (1,1)$: $z = 0.8 + -2 \times \frac{1}{4} + -1 \times \frac{1}{2} + 0 \times \frac{1}{4} = -0.2$
  - $(x_1, x_2) = (2,0)$: $z = 0.8 + -2 \times \frac{1}{4} + -1 \times \frac{1}{2} + 0 \times \frac{1}{4} = -0.2$

Open-loop Solution

The optimal solution is to order 1 unit in the first day and order nothing in the second day.
The expected profit is 35 cents.

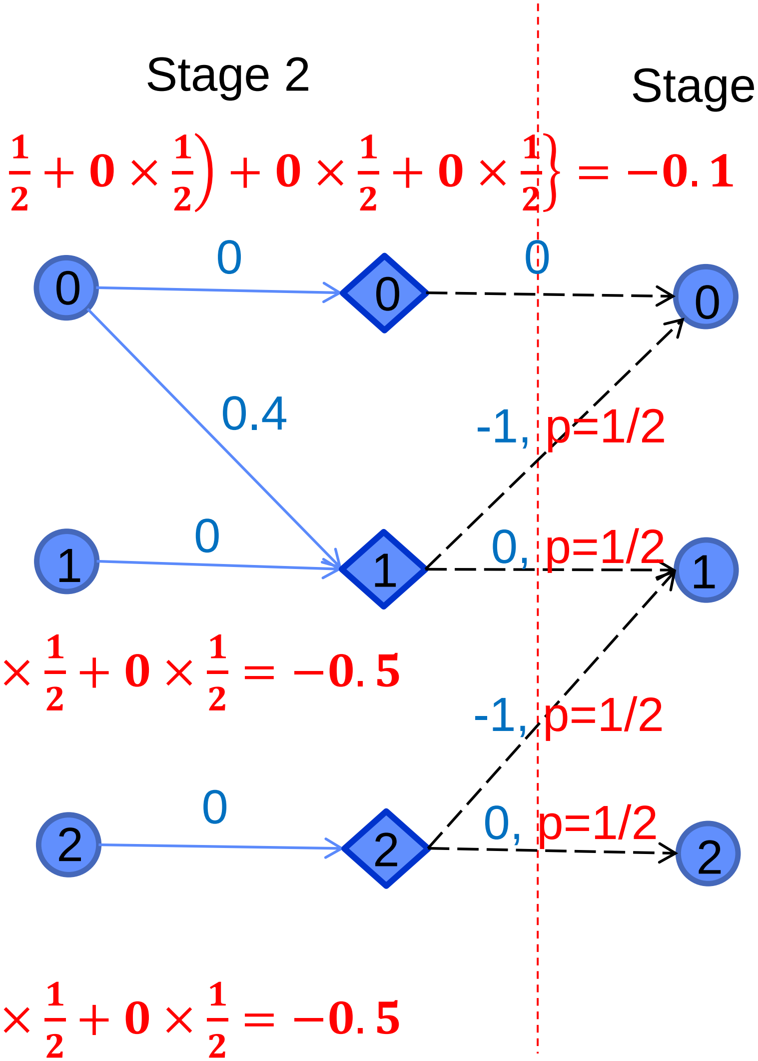
Information flow is not utilized in making decisions.

Stochastic Shortest Path Formulation



Stage 2 (Day 2)

$$V(2,0) = min\left\{0+0,\ 0.4+\left(-1\times\frac{1}{2}+0\times\frac{1}{2}\right)+0\times\frac{1}{2}+0\times\frac{1}{2}\right\} = -0.1$$



$$V(1,0) = 0$$

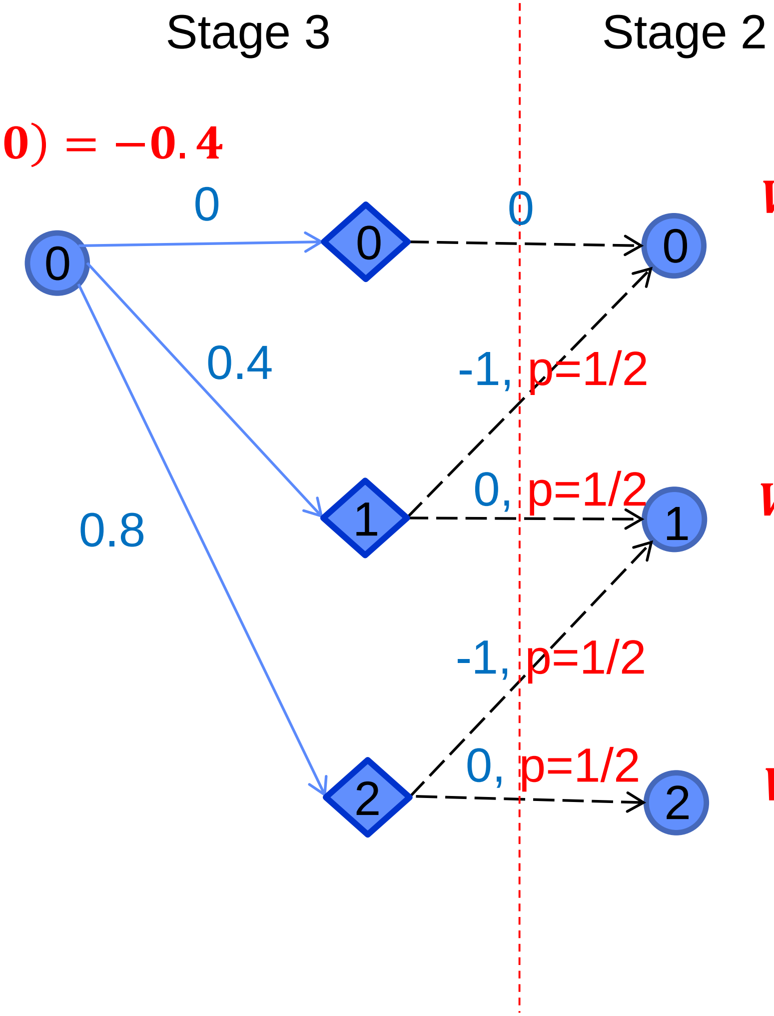
$$V(1,1) = 0$$

$$V(2,1) = 0+\left(-1\times\frac{1}{2}+0\times\frac{1}{2}\right)+0\times\frac{1}{2}+0\times\frac{1}{2} = -0.5$$

$$V(1,2) = 0$$

$$V(2,2) = 0+\left(-1\times\frac{1}{2}+0\times\frac{1}{2}\right)+0\times\frac{1}{2}+0\times\frac{1}{2} = -0.5$$

Last Stage (Day 1)



$\boldsymbol{V(3,0) = -0.4}$

$\boldsymbol{V(2,0) = -0.1}$ The first path (do not order):

$$0 - 0.1 = -0.1$$

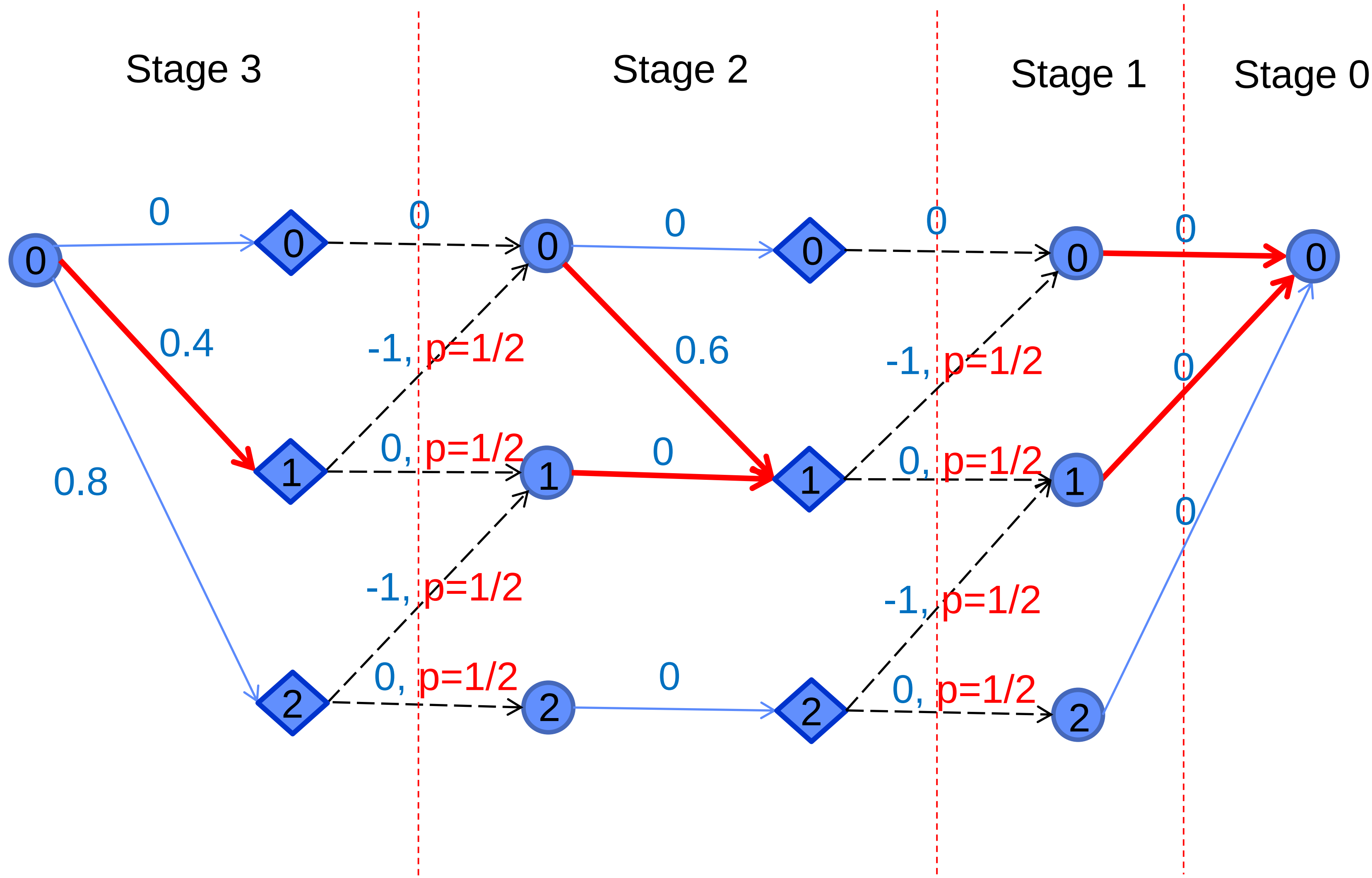
$\boldsymbol{V(2,1) = -0.5}$ The second path (order 1):

$$0.4 + -1 \times \frac{1}{2} + 0 \times \frac{1}{2} + -0.1 \times \frac{1}{2} + -0.5 \times \frac{1}{2}$$
$$= -0.4$$

$\boldsymbol{V(2,2) = -0.5}$ The third path (order 2):

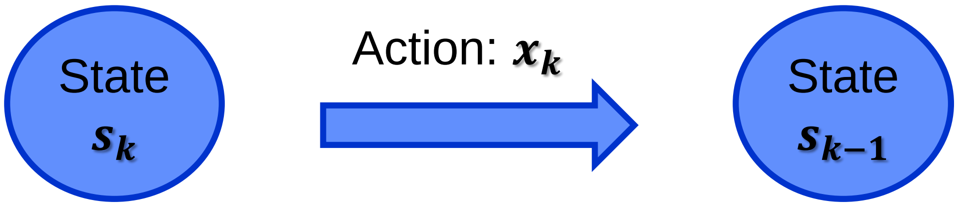
$$0.8 + -1 \times \frac{1}{2} + 0 \times \frac{1}{2} + -0.5 \times \frac{1}{2} + -0.5 \times \frac{1}{2}$$
$$= -0.2$$

Optimal Policy



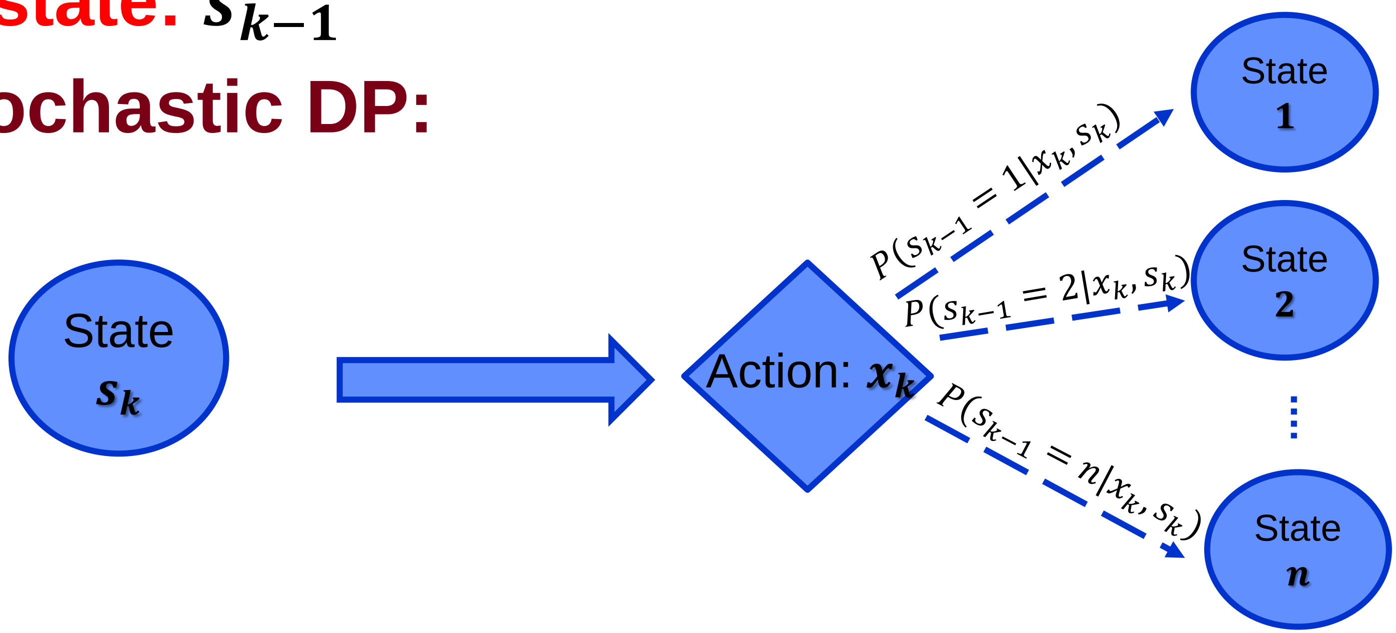
Deterministic DP vs Stochastic DP

- **Deterministic DP:**



  - **Choosing x_k is equivalent as choosing the next state: s_{k-1}**

- **Stochastic DP:**



Open-loop vs State Dependent Solution

- **Open-loop solution:**
  - The actions x_k are determined no matter what the state s_k is.
  - $(x_1, x_2) = (1,0)$ is determined no matter whether the first day demand is 1 or 0.
- **State dependent solution:**
  - The actions x_k depend on what the state s_k is.
  - $x_2(1) = 0, \ \ x_2(0) = 1$.

Multi-Period Newsvendor Problem: Summary

- $V(k,i)$ can be interpreted as the minimum cost (or maximum profit) when there are k stages remaining and we have i units of inventory. We want to know $V(3,0)$.

- **Bellman Equation** : let $S = \min\{i + x_k, D\}$, which is the random sales at stage k.

$$V(k,i) = \min_{x_k \geq 0}\{cx_k - pE_D[S] + E_D[V(k-1, i + x_k - S)]\},$$
$$V(1,i) = 0$$

Elements in A General Stochastic DP

- **Stages (period): T,…,1 (or T,…, 0)**
- **State:** $\boldsymbol{s_t = 1, \ldots, n}$**, action (control):** $\boldsymbol{x_t}$
- **Random shock:** $\boldsymbol{\epsilon_t}$
- **Expected cost per period:** $\boldsymbol{E[c(s_t, x_t, \epsilon_t)]}$
- **State transition:** $\boldsymbol{s_{t-1} = i}$ **with probability** $P(s_{t-1} = i | x_t, s_t)$
- **Stochastic Optimization Problem**

$$\min_{x_t}\{ E[c(s_T, x_T, \epsilon_T)] \ + E[c(s_{T-1}, x_{T-1}, \epsilon_{T-1})] + \cdots + E[c(s_1, x_1, \epsilon_1)]\}$$

- **Stochastic DP: Bellman Equation**

$$\boldsymbol{V(t, s_t)} = \min_{x_t}\{ \boldsymbol{E[c(s_t, x_t, \epsilon_t)]} \ + \textstyle\sum_i \boldsymbol{V(t-1, i) P(s_{t-1} = i | x_t, s_t)} \}$$

Multi-Period Newsvendor Problem

- **State:** $s_t = 0,1,2$
- **Action (control):** $x_t = 0,1,2$
- **Random shock:** $P(\epsilon_t = 0) = P(\epsilon_t = 1) = 1/2$
- **Cost per period:** $c(s_t, x_t, \epsilon_t) = cx_t - pE_D\left[\min\{s_t + x_t, \epsilon_t\}\right]$
- **State transition: e.g.** $P(s_{t-1} = 0 | x_t = 0, s_t = 1) = 1/2$

Further Applications

Investment Problem

- **Consider the investment problem with the following four financial products and a total budget of 10K.**

Product number (i)	1	2	3	4
Investment (K): w_i	2	3	4	6
Return (K): v_i	8	12	15	19

Think in a Sequential Way

- Think about the case when the salesperson for product 1 to product 4 comes sequentially. At each stage, we have to make the decision whether to buy the product or not.
- State:

The remaining budget.

Decomposition

Budget Available	Stage 4	Stage 3	Stage 2	Stage 1
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Single Stage Problem: Only Product 4

Budget Available	Stage 1 $v_4 = 19, w_4 = 6$
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Stage 1

Budget Available		Stage 1 $v_4 = 19, w_4 = 6$
0		0
1		0
2		0
3		0
4		0
5		0
6		19
7		19
8		19
9		19
10		19

Stage 2

Budget Available	Stage 2 $v_3 = 15, w_3 = 4$	Stage 1
0		0
1		0
2		0
3		0
4		0
5		0
6		19
7		19
8		19
9		19
10		19

Stage 2

Budget Available	Stage 2 $v_3 = 15, w_3 = 4$	Stage 1
0	0	0
1	0	0
2	0	0
3	0	0
4	15	0
5	15	0
6	19	19
7	19	19
8	19	19
9	19	19
10	34	19

Stage 3

Budget Available	Stage 3 $v_2 = 12, w_2 = 3$	Stage 2
0		0
1		0
2		0
3		0
4		15
5		15
6		19
7		19
8		19
9		19
10		34

Stage 3

Budget Available
0
1
2
3
4
5
6
7
8
9
10

Stage 3 $v_2 = 12, w_2 = 3$	Stage 2
0	0
0	0
0	0
12	0
15	15
15	15
19	19
27	19
27	19
31	19
34	34

Final Stage

Budget Available	Stage 4 $v_1 = 8, w_1 = 2$	Stage 3
0		0
1		0
2		0
3		12
4		15
5		15
6		19
7		27
8		27
9		31
10		34

Final Stage

Budget Available	Stage 4 $v_1 = 8, w_1 = 2$	Stage 3
0	0	0
1	0	0
2	8	0
3	12	12
4	15	15
5	20	15
6	23	19
7	27	27
8	27	27
9	35	31
10	35	34

Complete Solution

Budget Available	Stage 4 $v_1 = 8, w_1 = 2$	Stage 3 $v_2 = 12, w_2 = 3$	Stage 2 $v_3 = 15, w_3 = 4$	Stage 1 $v_4 = 19, w_4 = 6$
0	0	0	0	0
1	0	0	0	0
2	8	0	0	0
3	12	12	0	0
4	15	15	15	0
5	20	15	15	0
6	23	19	19	19
7	27	27	19	19
8	27	27	19	19
9	35	31	19	19
10	35	34	34	19

Knapsack Problem: Bellman Equation

- **The value of each cell: $V(k,i)$ is the maximum return one can get when there are k stages (k products) remaining and there is i budget left. Our goal is $V(4,10)$.**

- **Bellman Equation:**

$$V(k,i)=\begin{cases} max\{V(k-1,i),\ v_k+V(k-1,i-w_k)\}, if\ w_k \le i \\ V(k-1,i), if\ w_k \ge i \end{cases}$$

$$V(0,i)=0, i=1,\dots,10$$

Secretary Problem

- There is a single secretary position to fill
- There are n applicants for the position
- The applicants, if seen altogether, can be ranked from best to worst.
- Applicants are interviewed sequentially in random order with each order being equally likely.
- After an interview, an offer or rejection must be made, based on the applicants interviewed so far.
- Objective is to choose the best candidate:
  - Maximize the probability that the chosen candidate is the best

Naïve Method: Open-loop Solution

Before the interviews begin, choose the k-th applicant, k = 1,…,n such that the probability of selecting the best candidate is maximized.

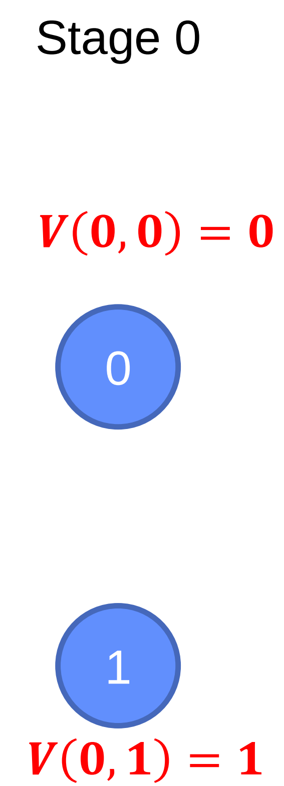
What is the best k and what is the optimal probability ?

Information flow is not utilized in making decisions.

Stochastic DP

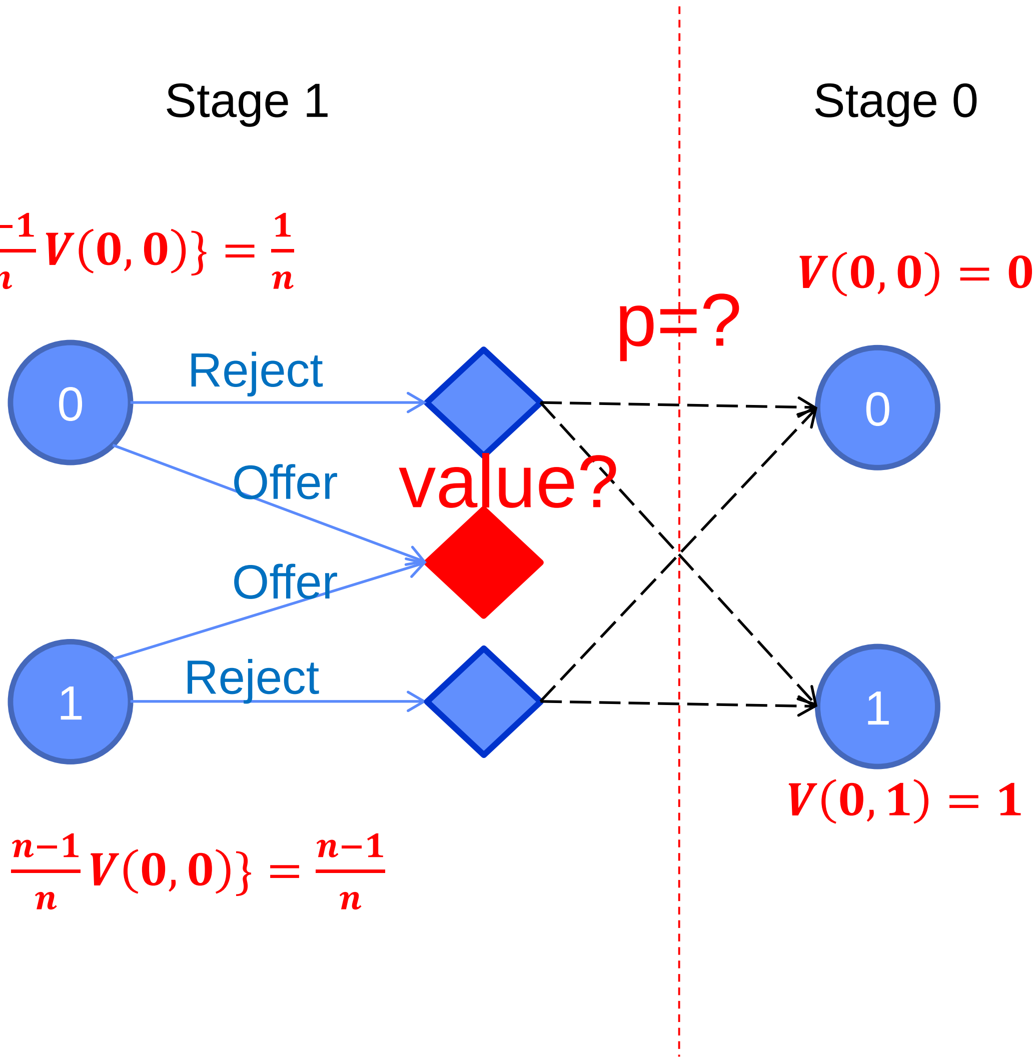
- **State:**
  - It is sufficient to know whether or not the current applicant is the best among all those interviewed before. Suppose 0 means it is not the best, 1 means it is the best.

Stage 0 (the n-th candidate)



Stage 1 (the n-1-th candidate)

$$V(1,0) = max\,\{0, \frac{1}{n}V(0,1) + \frac{n-1}{n}V(0,0)\} = \frac{1}{n}$$



$$V(1,1) = max\,\{\frac{n-1}{n}, \frac{1}{n}V(0,1) + \frac{n-1}{n}V(0,0)\} = \frac{n-1}{n}$$

Stage n-k (the k-th candidate)

- Let $V(n-k,i)$ be the best value we can expect when applicant k is at state i, $i=0,1$. We want to know $V(n-1,1)$

- Bellman Equation:

$$V(n-k,0)=max\{\,0,\frac{1}{k+1}V(n-k-1,1)+\frac{k}{k+1}V(n-k-1,0)\}$$

$$V(n-k,1)=max\{\frac{k}{n},\ \frac{1}{k+1}V(n-k-1,1)+\frac{k}{k+1}V(n-k-1,0)\,\}$$
$$=max\{\frac{k}{n},V(n-k,0)\}$$

$$V(0,0)=0,V(0,1)=1$$

Secretary Problem: n = 4

- k = 4

$$V(0,0) = 0, V(0,1) = 1$$

- k = 3

$$V(1,0) = \frac{1}{4}1 + \frac{3}{4}0 = \frac{1}{4}, V(1,1) = max\left\{\frac{3}{4}, \frac{1}{4}\right\} = \frac{3}{4}$$

- k = 2

$$V(2,0) = \frac{1}{3}\frac{3}{4} + \frac{2}{3}\frac{1}{4} = \frac{5}{12}, V(2,1) = max\left\{\frac{2}{4}, \frac{5}{12}\right\} = \frac{1}{2}$$

- k = 1

$$V(3,0) = \frac{1}{2}\frac{1}{2} + \frac{1}{2}\frac{5}{12} = \frac{11}{24}, V(3,1) = max\left\{\frac{1}{4}, \frac{11}{24}\right\} = \frac{11}{24}$$

- 1: Reject; 2-4: Accept if the best seen so far.

Secretary Problem: In General

- **There is a threshold k^* such that if $k > k^*$**

$$\frac{k}{n} \geq V(n-k, 0)$$

That is, we accept the candidate as long as the candidate is the best seen so far (state 1).

- **If $k < k^*$, $\frac{k}{n} < V(n-k, 0)$ and we reject the candidate.**

$$\boxed{k^* \approx \frac{1}{e} n \approx 0.37n}$$