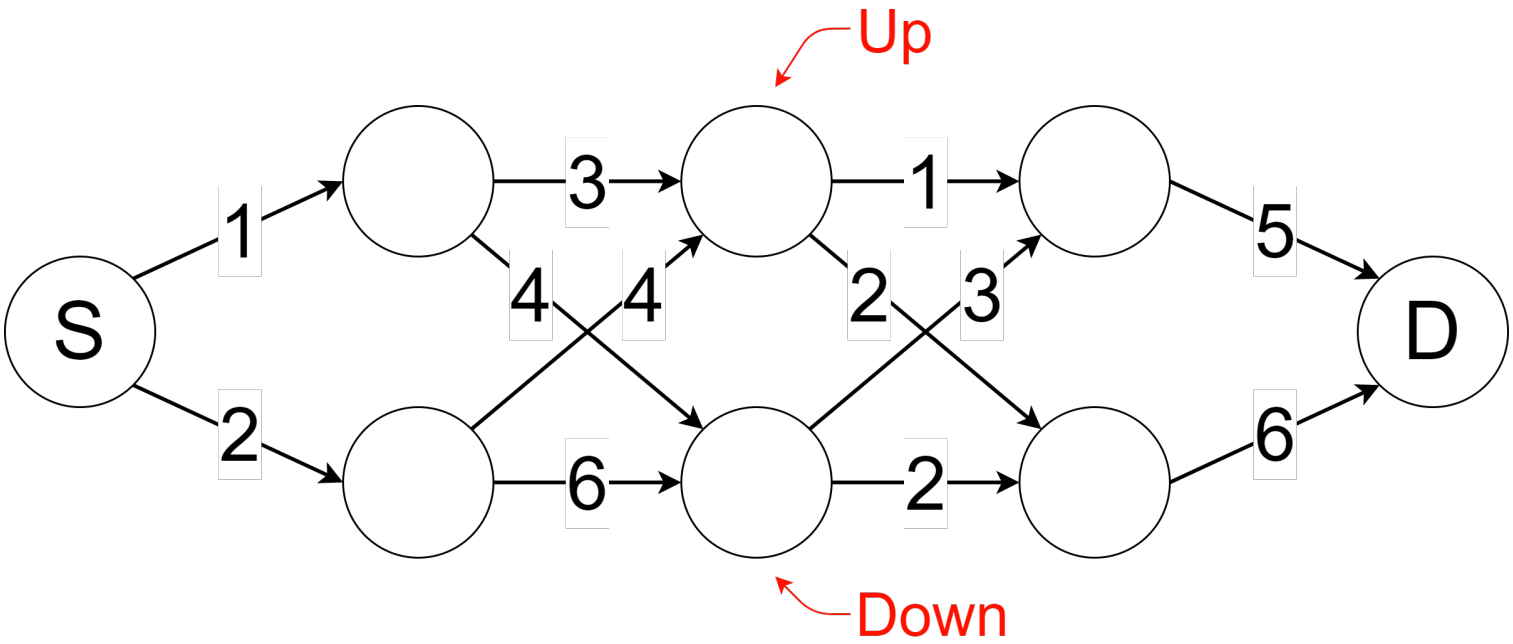
Lecture 2: Finite Horizon Stochastic DP

Course: Reinforcement Learning Theory
Instructor: Lei Ying
Department of EECS
University of Michigan, Ann Arbor

A Stochastic DP example

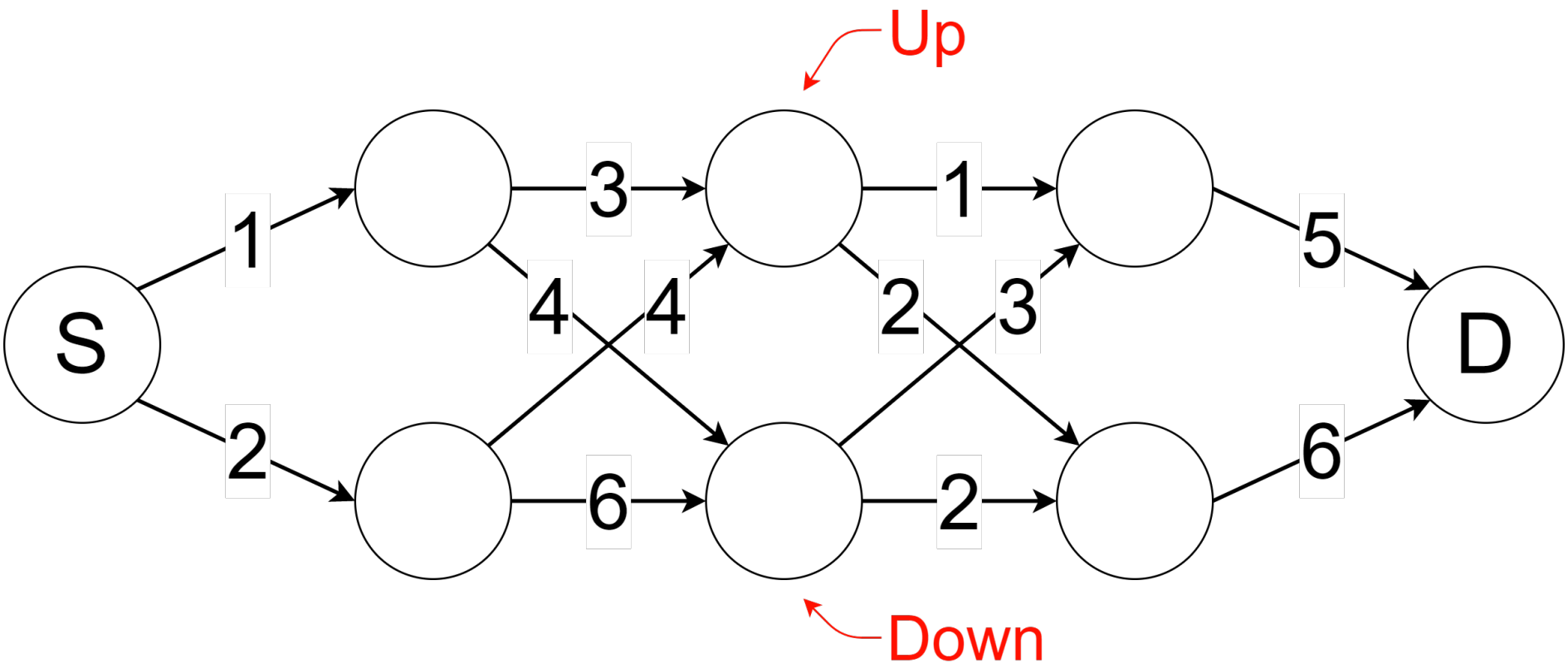


- Find the shortest path from S to D. Actions $\in$ (up,down).

$$u_k = \text{up} \quad \rightarrow \begin{cases} P(x_{k+1} = \text{top}) = 0.6 \\ P(x_{k+1} = \text{bottom}) = 0.4 \end{cases}$$

$$u_k = \text{down} \quad \rightarrow \begin{cases} P(x_{k+1} = \text{top}) = 0.4 \\ P(x_{k+1} = \text{bottom}) = 0.6 \end{cases}$$

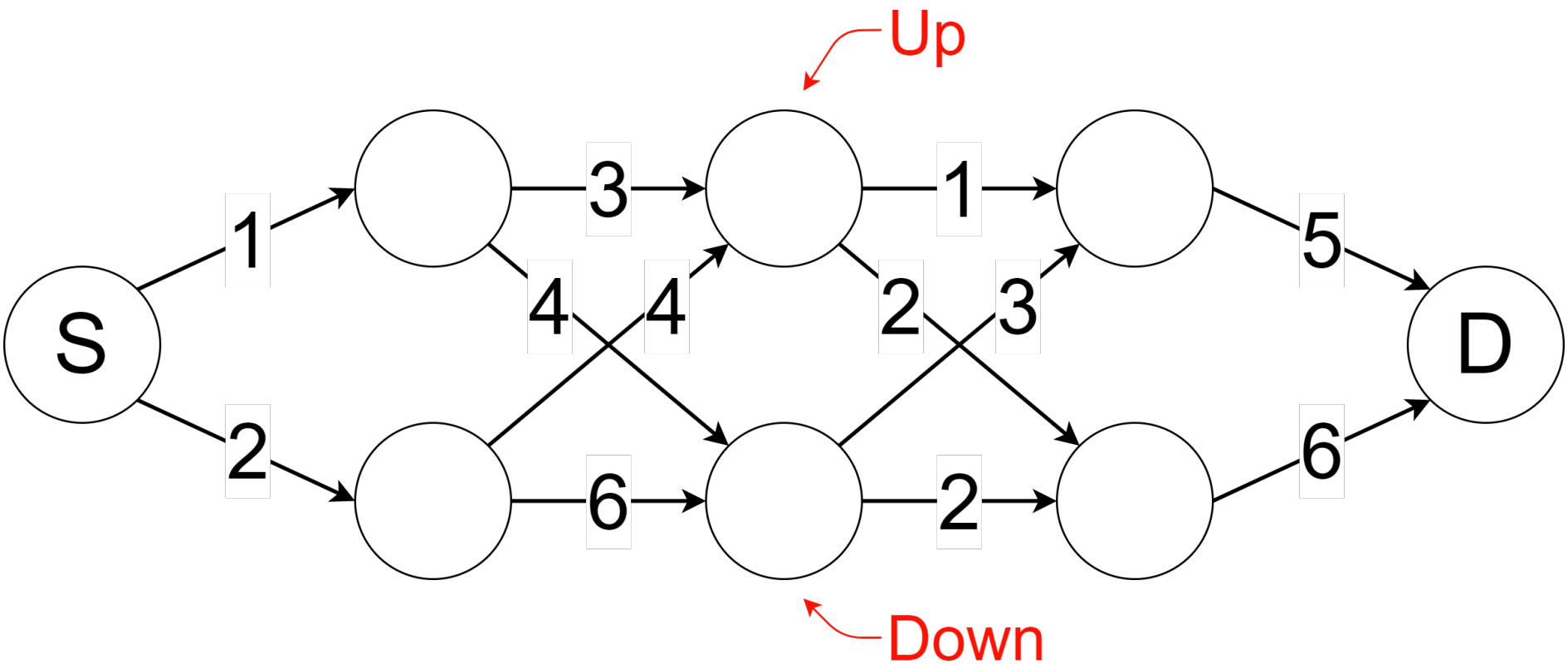
Stochastic DP example



Optimal policy:

A fixed sequence $(u_0, u_1, \ldots, u_{N-1})$?

Stochastic DP example

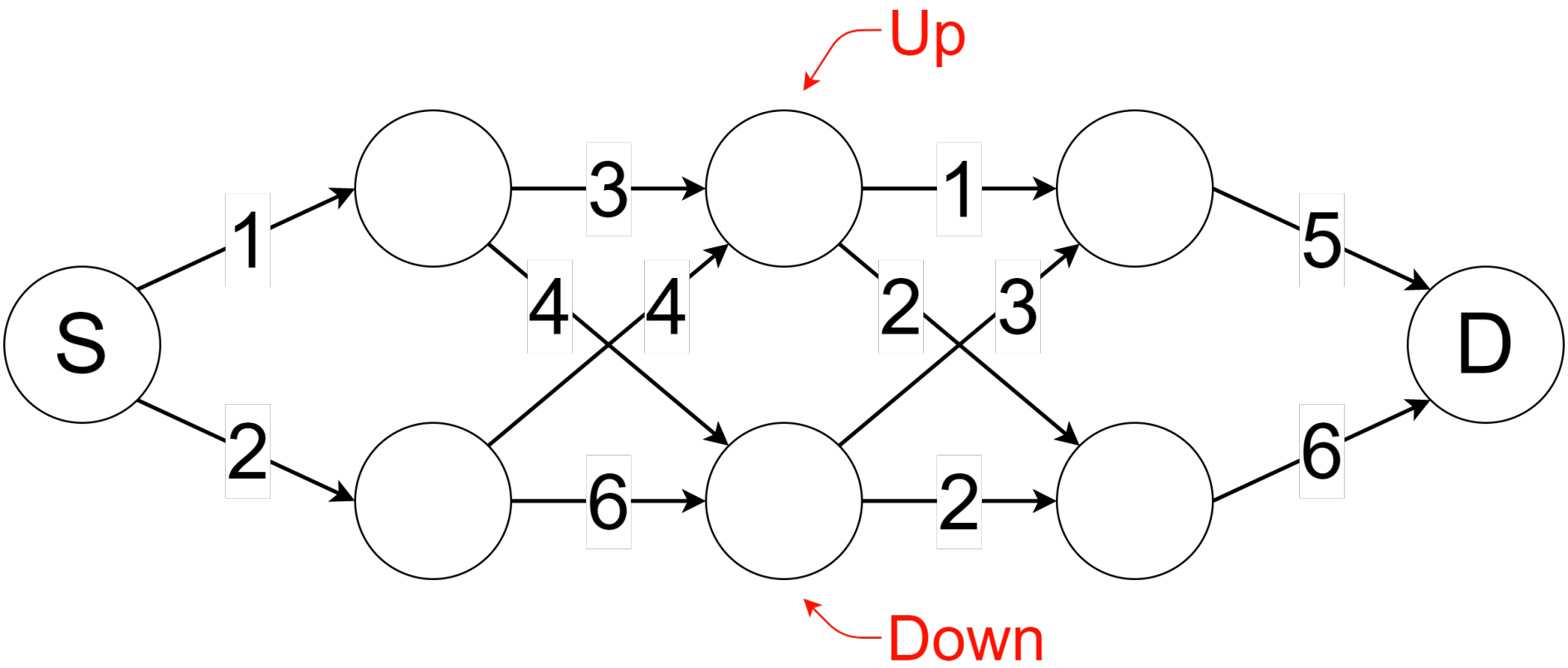


Optimal policy:

A fixed sequence $(u_0, u_1, \ldots, u_{N-1})$?

NO.

Stochastic DP example

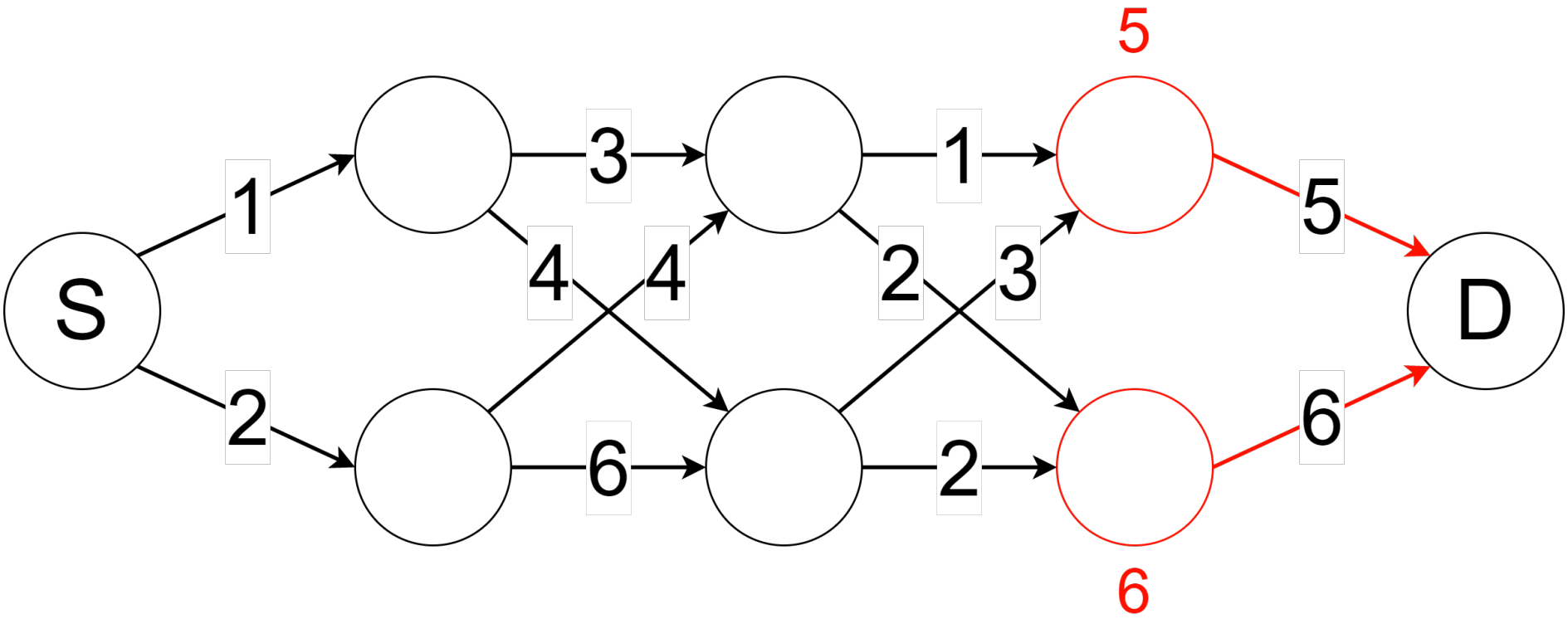


Optimal policy:

~~Fix sequence $(u_0, u_1, \ldots, u_{N-1})$?~~

State dependent policy: $\mu_k^*(x_k),\ \forall x_k$

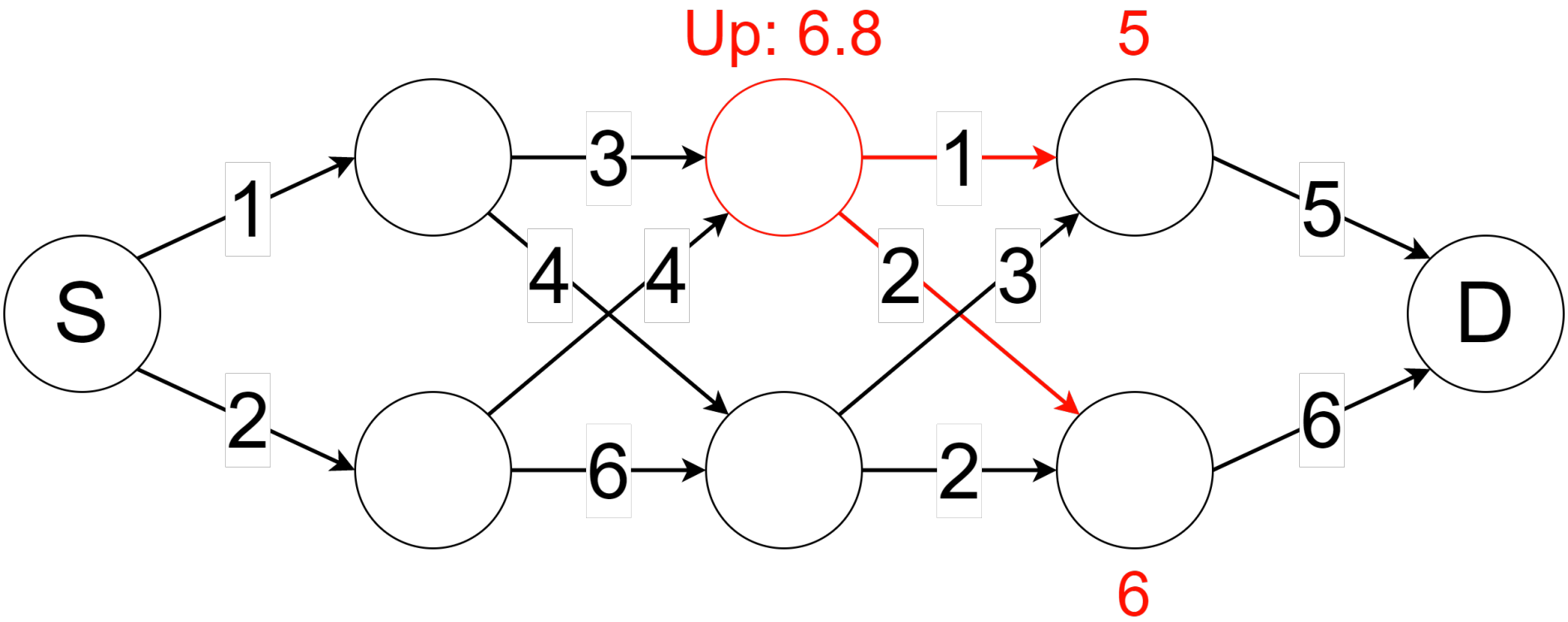
Stochastic DP example



The red nodes have only one possible path to reach node D.

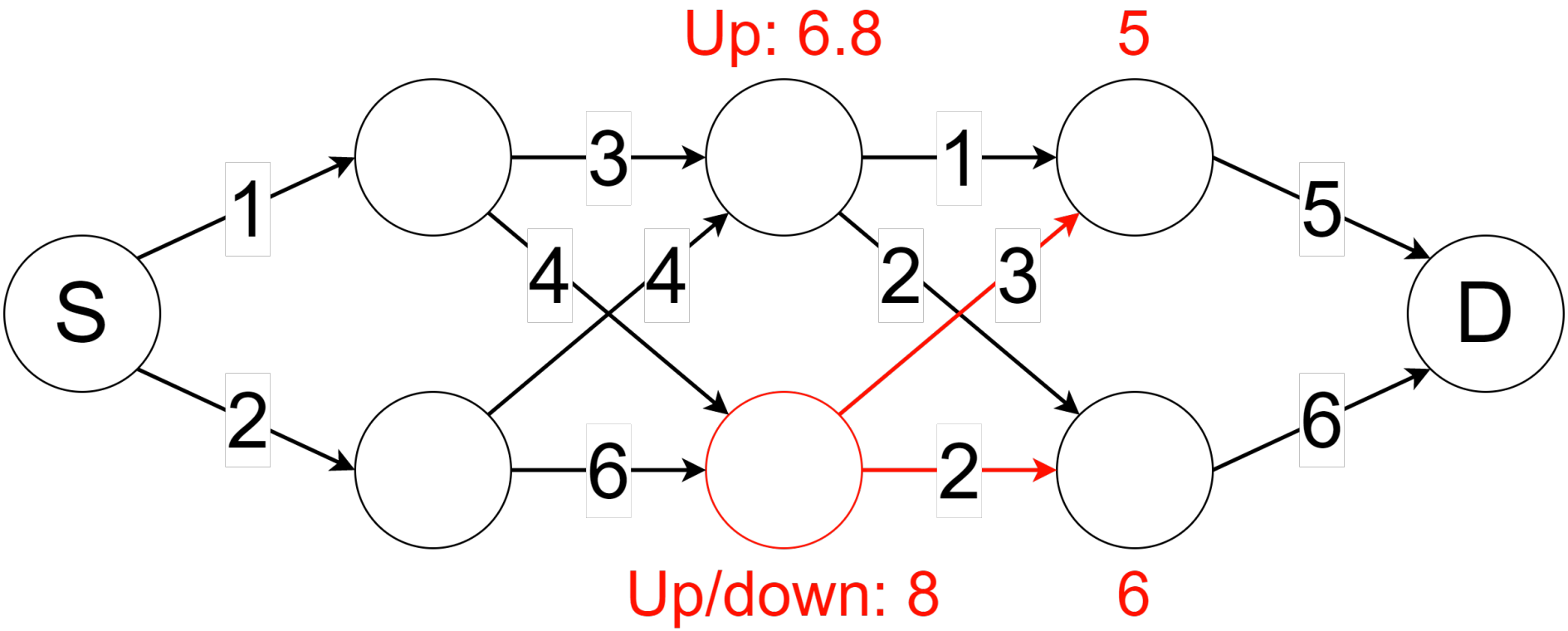
- Cost-to-go from the top node: 5
- Cost-to-go from the bottom node: 6

Stochastic DP example



- Cost-to-go with action Up:
  $0.6 \times (1+5) + 0.4 \times (2+6) = 3.6 + 3.2 = 6.8$
- Cost-to-go with action Down: $0.4 \times 6 + 0.6 \times 8 = 7.2$
- $\min\{\text{up}, \text{down}\} = 6.8 \implies u^*_{N-2}(\text{top}) = \textbf{up}$

Stochastic DP example



- Cost-to-go (Up): $0.6 \times (3+5) + 0.4 \times (2+6) = 8$
- Cost-to-go (Down): $0.4 \times 8 + 0.6 \times 8 = 8$
- $\min\{\text{up}, \text{down}\} = 8 \implies u^*_{N-2}(\text{bottom})$ can be either up or down

Stochastic DP

- Random transitions $P(x_{k+1}|x_k, u_k)$
- Cost function:

$$E\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)\right]$$

- Given initial state x_0, find a policy

$$\pi = \{\mu_0, \mu_1(x_1), \ldots, \mu_{N-1}(x_{N-1})\}$$

that minimizes

$$J(x_0) = \min_{\pi} E\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k))\right]$$

Stochastic DP

Backward pass for calculating the optimal cost-to-go.
Starting from $J_N^*(x_N) = g_N(x_N)\ \forall x_N$, calculate

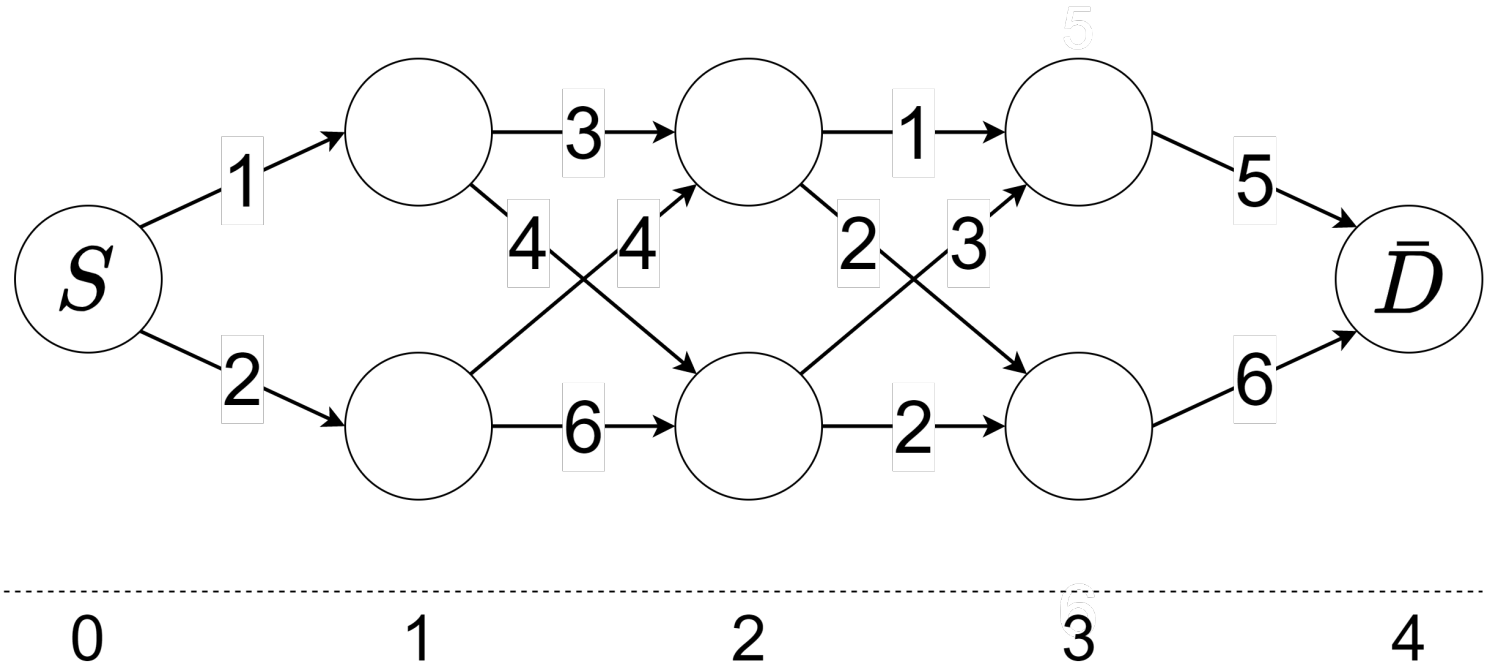
$$J_k^*(x_k) = \min_{u_k} E[\underbrace{g_k(x_k, u_k) + J_{k+1}^*(x_{k+1})}_{\text{both are random variables based on } P(x_{k+1}|x_k, u_k)}].$$

Forward pass to find the optimal policy:
Given state x_k and $J_{k+1}^*(x_{k+1})$,

$$u_k^* \in \arg\min_{u_k} E[g_k(x_k, u_k) + J_{k+1}^*(x_{k+1})]$$

Backward-forward algorithm: First compute J_k^* backward and then find u_k^* forward.

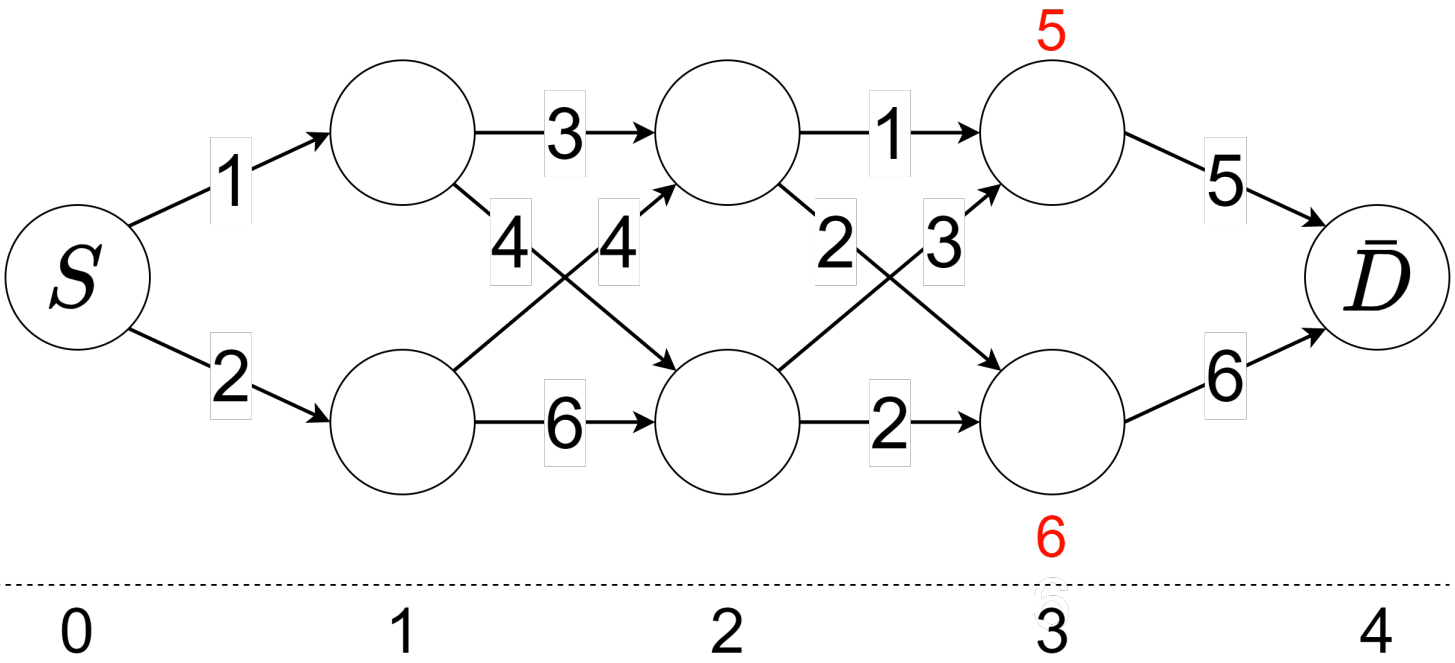
Example



T: top, B: bottom, U: up, D: down.

$$J_4(\bar{D}) = 0$$

Example

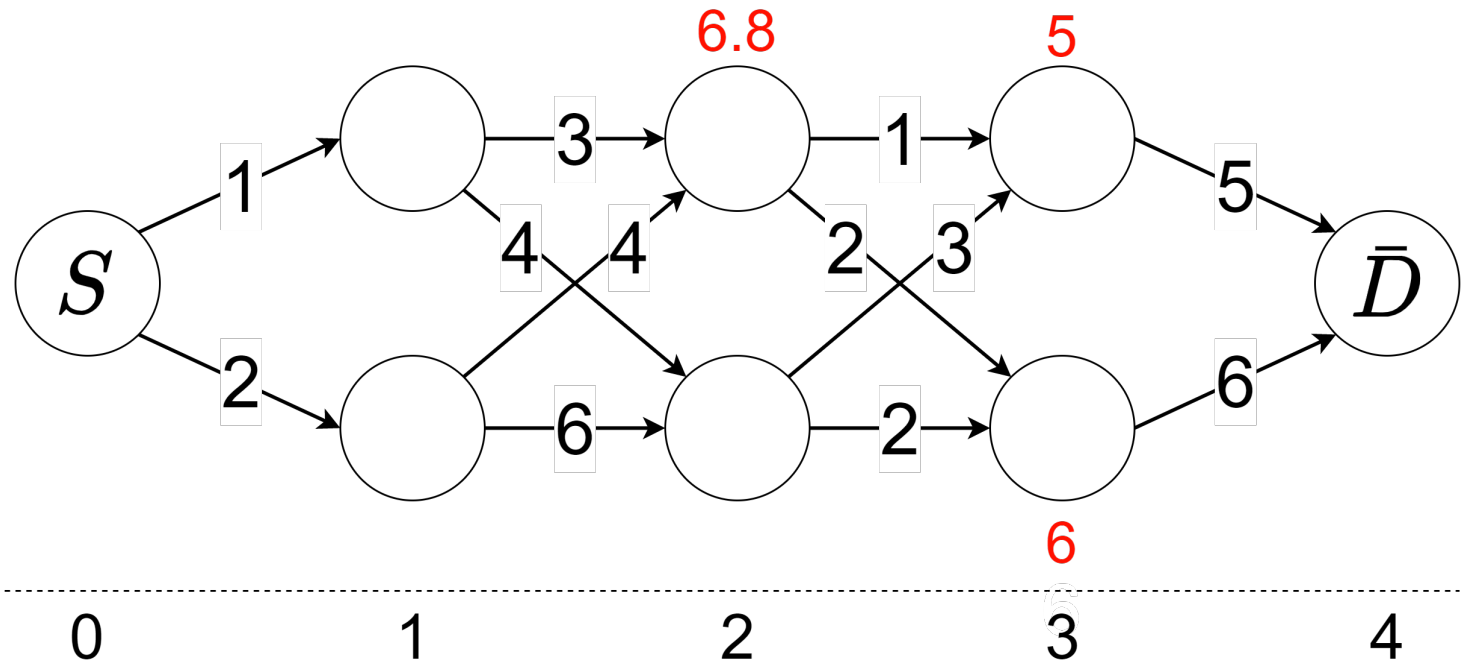


T: top, B: bottom, U: up, D: down.

$$J_4(\bar{D}) = 0$$

$$J_3(T) = 5,\ J_3(B) = 6$$

Example



T: top, B: bottom, U: up, D: down.

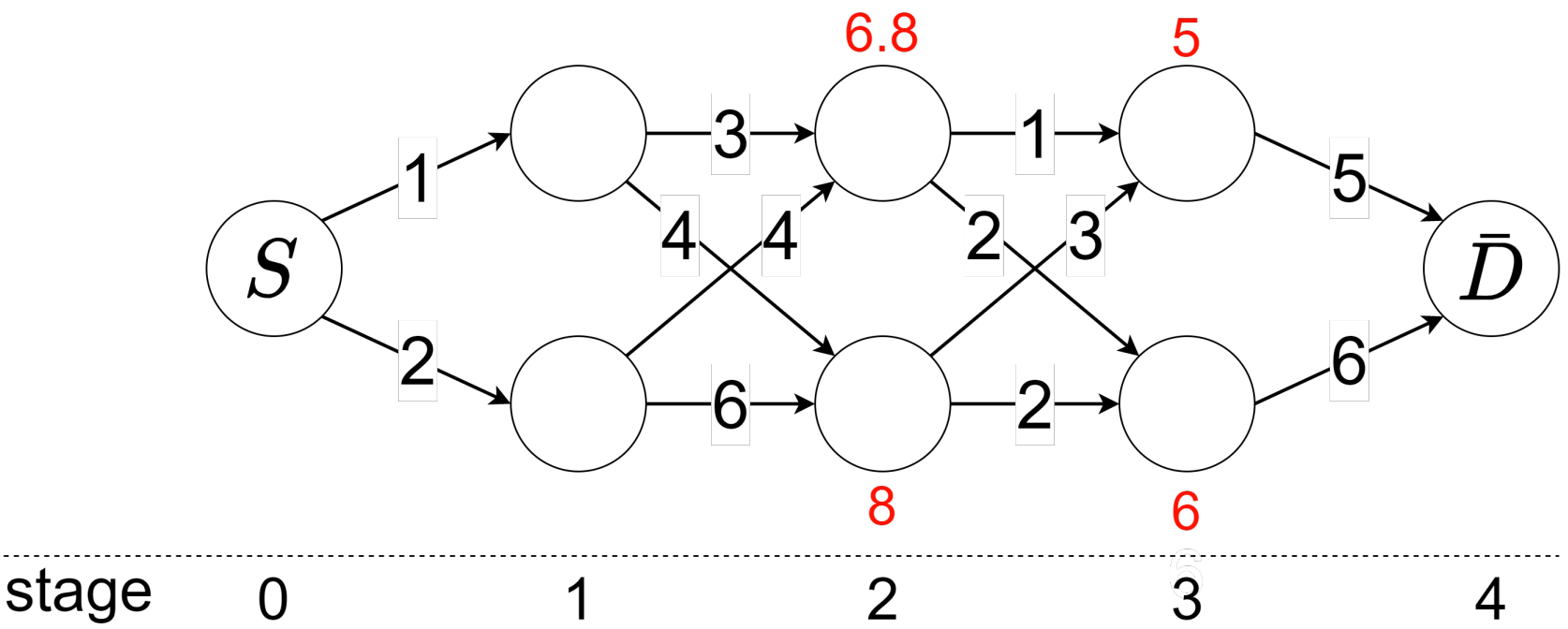
$$J_4(\bar{D}) = 0$$

$$J_3(T) = 5,\ J_3(B) = 6$$

$$J_2(T) = \min\{0.6(1+5) + 0.4(2+6), 0.4(1+5) + 0.6(2+6)\}$$

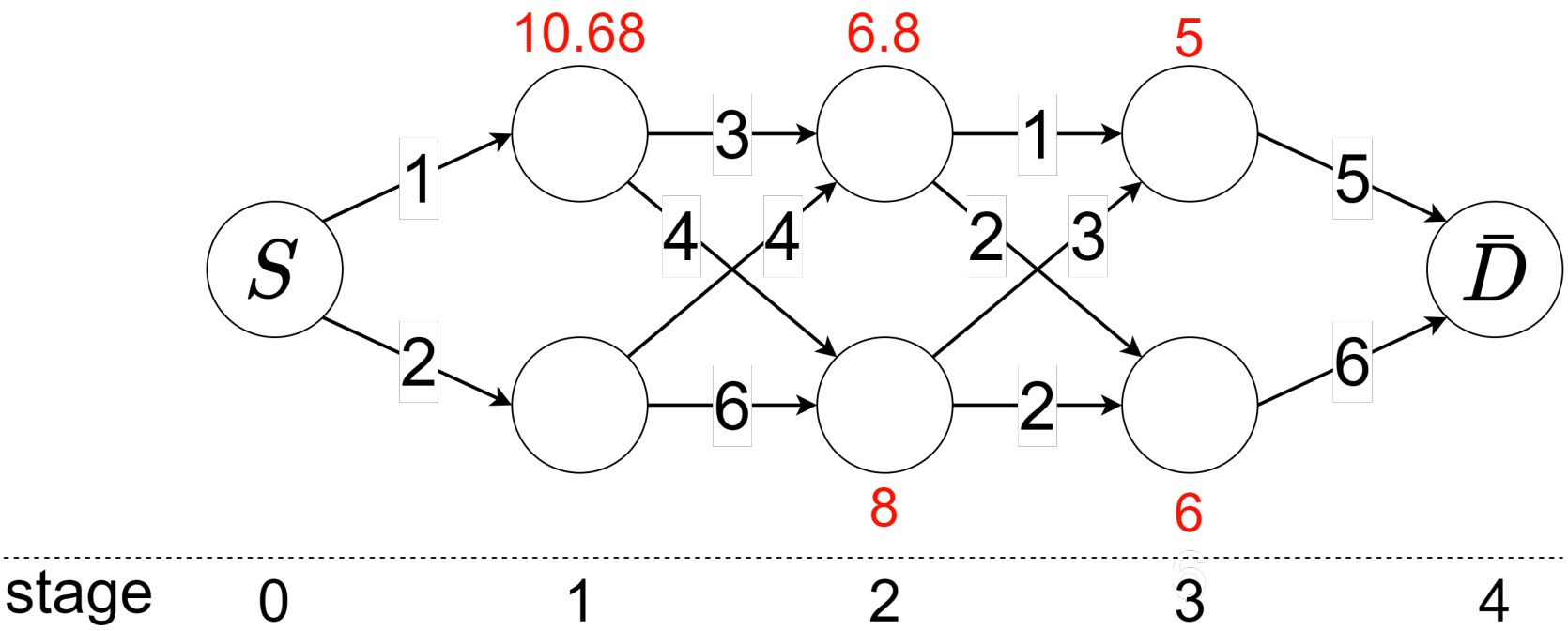
$$= \min\{6.8, 7.2\} = 6.8\ (U)$$

Example



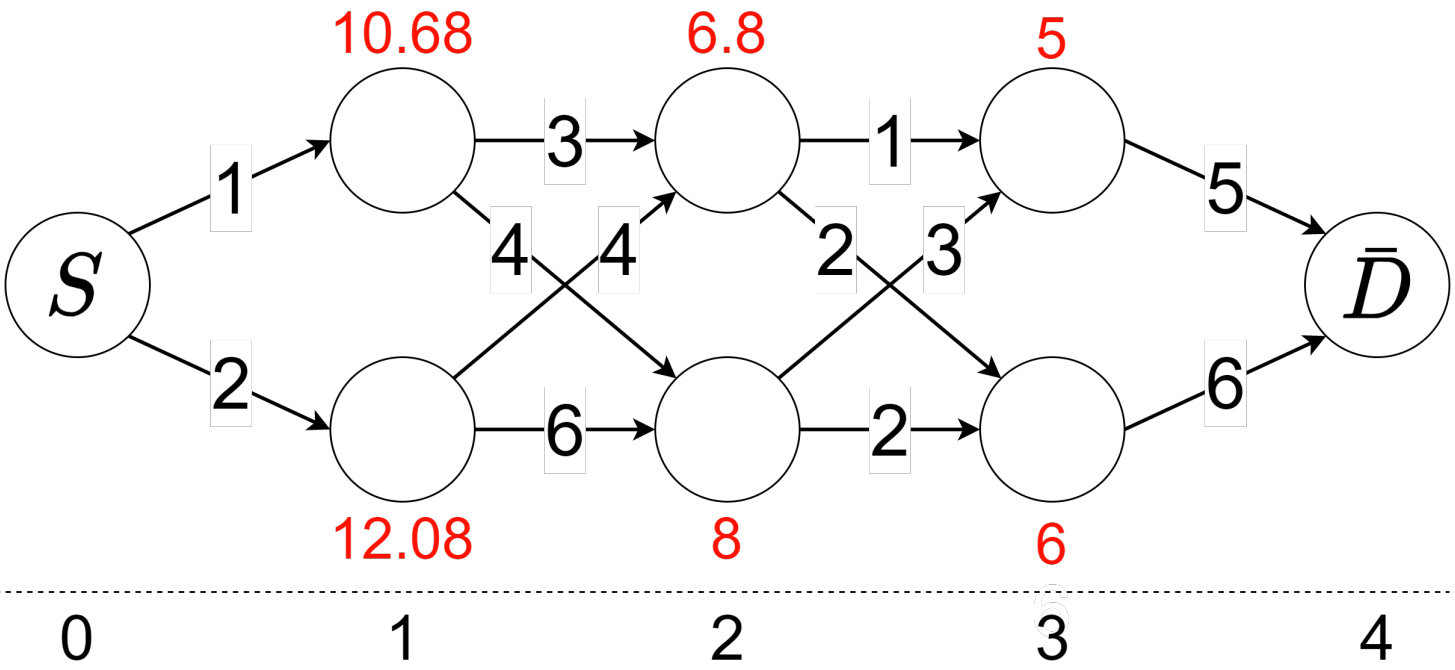
$$J_2(B) = \min\{0.6(3+5) + 0.4(2+6), 0.4(3+5) + 0.6(2+6)\}$$
$$= 8 \ (U/D)$$

Example



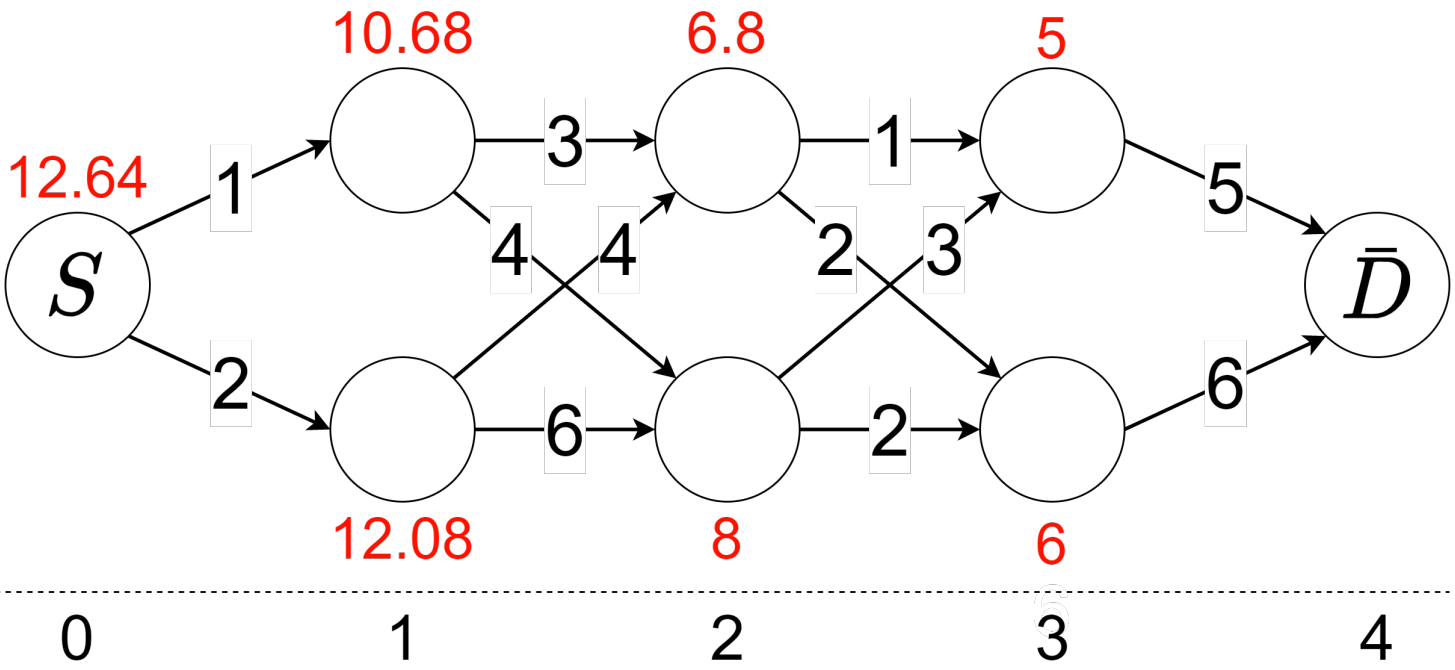
$$J_2(B) = \min\{0.6(3+5) + 0.4(2+6), 0.4(3+5) + 0.6(2+6)\}$$
$$= 8 \ (U/D)$$
$$J_1(T) = \min\{0.6(3+6.8) + 0.4(4+8), 0.4(3+6.8) + 0.6(4+8)\}$$
$$= 10.68 \ (U)$$

Example



$$J_1(B) = \min\{0.6(4+6.8)+0.4(6+8), 0.4(4+6.8)+0.6(6+8)\}$$
$$= 12.08\ (U)$$

Example



$$J_1(B) = \min\{0.6(4+6.8)+0.4(6+8), 0.4(4+6.8)+0.6(6+8)\}$$
$$= 12.08\ (U)$$
$$J_0(S) = \min\{0.6(1+10.68)+0.4(2+12.08), 0.4(1+10.68)+0.6(2+12.0$$
$$= 12.64\ (U)$$

Stochastic DP and the Bellman Equation

The Bellman Equation:

$$
\begin{aligned}
J_k^*(x_k) &= \min_{u_k} E\left[g_k(x_k, u_k) + J_{k+1}^*(x_{k+1})\right] \\
&= \min_{u_k} \sum_g g P(g_k = g | x_k, u_k) + \sum_x J_{k+1}^*(x) P(x_{k+1} = x | x_k, u_k)
\end{aligned}
$$

Example

You have \$2 and have to play a game 3 times. For each game, the chance of winning is 0.4, and chance of losing is 0.6.

- Goal: Find a policy that maximizes your chance of ending up with \$4.

Example

You have \$2 and have to play a game 3 times. For each game, the chance of winning is 0.4, and chance of losing is 0.6.

- Goal: Find a policy that maximizes your chance of ending up with \$4.

DP formulation:

- stage i: the ith game
- state at stage i: $x_i =$ money available at the beginning of game i
- action at stage i: $y_i =$ how much to bet (must be an integer)
- reward at stage i: when $i = 4$, $r_4(x_4) = \begin{cases} 1, & x_4 \geq 4 \\ 0, & \text{otherwise} \end{cases}$

  when $i = 1, 2, 3$, $r_i(x_i, y_i) = 0$
- Objective: $J_1^*(x_1 = 2) = \min_{\mu_k} -E\left[\sum_{k=1}^4 r_k(x_k) \middle| x_1 = 2\right]$.

Example

The Bellman Equation:

$$J_k^*(x_k) = \max_y E[J_{k+1}^*(x_{k+1})]$$
$$= \max_y \{J_{k+1}^*(x_k + y_k) \times 0.4 + J_{k+1}^*(x_k - y_k) \times 0.6\}$$

- For $k = 4 : J_4^*(x_4) = \begin{cases} 1, & x_4 = 4, 5, \ldots, 16 \\ 0, & x_4 = 0, 1, 2, 3 \end{cases}$
- For $k = 3$: $J_3^*(x_3) = \max_y \{J_4^*(x_3 + y) \times 0.4 + J_4^*(x_3 - y) \times 0.6\}$

  Note that $y_k^* = 0$ when $x_k \geq 4$, so we only need to consider $x_k \in \{1, 2, 3\}$.

  $$J_3^*(x_3 = 1) = \max\{ \underset{\substack{\uparrow \\ y=0}}{0} , \underset{\substack{\uparrow \\ y=1}}{0} \} = 0$$

Example

- $J_3^*(x_3 = 2) = \max\{\underset{y=0}{0}, \underset{y=1}{0}, \underset{y=2}{0.4}\} = 0.4$
- $J_3^*(x_3 = 3) = \max\{0, \underset{y=1}{0.4}, \underset{y=2}{0.4}, \underset{y=3}{0.4}\} = 0.4$
- $J_3^*(x_3 = k) = 1,\ k \geq 4$
- $J_2^*(x_2 = 0) = 0$
- $J_2^*(x_2 = 1) = \max\{J_3^*(x_3 = 1) = 0,$

$$0.4J_3^*(x_3 = 2) + 0.6J_3^*(x_3 = 0) = 0.16\}$$

$$= 0.16$$

- $J_2^*(x_2 = 2) = \max\{J_3^*(x_3 = 2) = 0.4,$

$$0.4J_3^*(x_3 = 3) + 0.6J_3^*(x_3 = 1) = 0.16,$$

$$0.4J_3^*(x_3 = 4) + 0.6J_3^*(x_3 = 0) = 0.4\}$$

$$= 0.4$$

Example

- $J_2^*(x_2 = 3) = \max\{J_3^*(x_3 = 3) = 0.4,$
$$0.4J_3^*(x_3 = 4) + 0.6J_3^*(x_3 = 2) = 0.64,$$
$$0.4J_3^*(x_3 = 5) + 0.6J_3^*(x_3 = 1) = 0.4,$$
$$0.4J_3^*(x_3 = 6) + 0.6J_3^*(x_3 = 0) = 0.4\}$$
$$= 0.64$$
- $J_2^*(x_2 = 4) = 1$
- $J_1^*(x_1 = 2) = \max\{J_2^*(x_2 = 2) = 0.4,$
$$0.4J_2^*(x_2 = 3) + 0.6J_2^*(x_2 = 1) = 0.352,$$
$$0.4J_2^*(x_2 = 4) + 0.6J_2^*(x_2 = 0) = 0.4\}$$
$$= 0.4$$

Example

- $J_2^*(x_2 = 3) = \max\{J_3^*(x_3 = 3) = 0.4,$
$$0.4J_3^*(x_3 = 4) + 0.6J_3^*(x_3 = 2) = 0.64,$$
$$0.4J_3^*(x_3 = 5) + 0.6J_3^*(x_3 = 1) = 0.4,$$
$$0.4J_3^*(x_3 = 6) + 0.6J_3^*(x_3 = 0) = 0.4\}$$
$$= 0.64$$
- $J_2^*(x_2 = 4) = 1$
- $J_1^*(x_1 = 2) = \max\{J_2^*(x_2 = 2) = 0.4,$
$$0.4J_2^*(x_2 = 3) + 0.6J_2^*(x_2 = 1) = 0.352,$$
$$0.4J_2^*(x_2 = 4) + 0.6J_2^*(x_2 = 0) = 0.4\}$$
$$= 0.4$$

Optimal strategy: bet once with \$2

Bellman Equation

The Bellman Equation:

$$J_k^*(x_k) = \min_{u_k} E\left[g_k(x_k, u_k) + J_{k+1}^*(x_{k+1})\right]$$
$$= \min_{u_k} \sum_g g P(g_k = g | x_k, u_k) + \sum_x J_{k+1}^*(x) P(x_{k+1} = x | x_k, u_k)$$

Difficulties in solving the Bellman equation:

1. Model-based: need to know $P(X_{k+1}|X_k, U_k)$.
   $\rightarrow$ model free, data-driven methods
2. Curse-of-dimensionality: large state and action spaces (the 9×9 Go game has 10^{35} states).
   $\rightarrow$ function approximation
3. Find the minimum.
   $\rightarrow$ actor-critic, policy gradient

Reference

- Chapter 1.2 of Dimitri P. Bertsekas, *Reinforcement Learning and Optimal Control*, Athena Scientific, 2019. Slides and lectures available at `https://web.mit.edu/dimitrib/www/RLbook.html`

Acknowledgements: I would like to thank Alex Zhao for helping prepare the slides, and Honghao Wei and Zixian Yang for correcting typos/mistakes.