Week 5

Policy Iteration (Page 1-8)

Value Iteration (Page 9-17)

Model-Free RL (Page 18-24)

Prof. Subrahmanya Swamy

Iterative Policy Evaluation

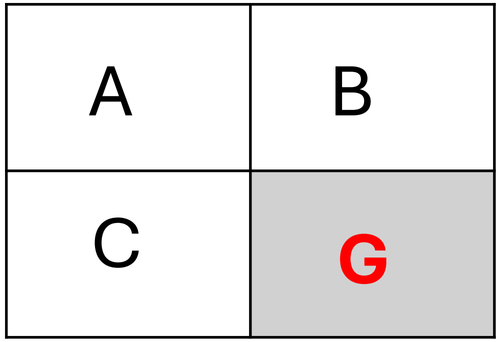
- How to find V_π of a given policy π ?
- Iteratively apply the BE equation

$$V_{k+1}(s) = R_s^{\pi} + \sum_{s'} P_{ss'}^{\pi} V_k(s')$$

Repeat till

$$V_{k+1} = V_k \implies V_k = V_\pi$$

Grid Example: Policy Evaluation



A	B
C	G

- **Deterministic** state transitions
- $R_t = -1$ on all transitions
- Terminal state value $V_\pi(G) = 0$
- Discount factor $\gamma = 1$
- **Uniform** Random Policy π

Uniform Policy Dynamics:

$$P^\pi_{A,A} = \frac{1}{2}, \quad P^\pi_{A,B} = \frac{1}{4}, \quad P^\pi_{A,C} = \frac{1}{4}$$

$$P^\pi_{B,A} = \frac{1}{4}, \quad P^\pi_{B,B} = \frac{1}{2}, \quad P^\pi_{B,G} = \frac{1}{4}$$

$$P^\pi_{C,A} = \frac{1}{4}, \quad P^\pi_{C,G} = \frac{1}{4}, \quad P^\pi_{C,C} = \frac{1}{2}$$

Iterative Policy Evaluation:

$$V_{k+1}(s) = R^\pi_s + \sum_{s'} P^\pi_{ss'} V_k(s')$$

$$\underset{V_0}{\begin{bmatrix}0\\0\\0\end{bmatrix}} \longrightarrow$$

Initial estimate

$$V_1(A) = -1 + \tfrac{1}{2}V_0(A) + \tfrac{1}{4}V_0(B) + \tfrac{1}{4}V_0(C)$$

$$V_1(B) = -1 + \frac{1}{4}V_0(A) + \frac{1}{2}V_0(B) + \frac{1}{4}V_0(G)$$

$$V_1(C) = -1 + \tfrac{1}{4}V_0(A) + \tfrac{1}{4}V_0(G) + \tfrac{1}{2}V_0(C)$$

Update at k=0

$$\longrightarrow \underset{V_1}{\begin{bmatrix}-1\\-1\\-1\end{bmatrix}} \xrightarrow[k=1]{\text{update}} \underset{V_2}{\begin{bmatrix}-2\\-1.75\\-1.75\end{bmatrix}} \xrightarrow[k=2]{\text{update}} \cdots$$

Repeat till $V_{k+1} = V_k$ $\Longrightarrow$ $V_k = V_\pi$

Policy Evaluation gives V_π

How to find Optimal Policy V^* ?

$\Downarrow$

Policy Iteration

Policy Iteration Algorithm

$$\pi_0 \xrightarrow[\text{Policy Evaluation}]{} V_{\pi_0} \xrightarrow[\text{Policy Improvement}]{} \pi_1 \to V_{\pi_1} \to \pi_2 \to V_{\pi_2} \to \ldots$$

Repeat till

$$\pi_{k+1} = \pi_k$$

$$\Downarrow$$

$$\pi_k = \pi_*$$

- Policy Evaluation: Iteratively apply BE equation $V_{k+1}(s) = R_s^{\pi} + \sum_{s'} P_{ss'}^{\pi} V_k(s')$
- Policy Improvement: $\pi_{i+1}(s) \coloneqq argmax_a \; R_s^a + \sum_{s'} P_{ss'}^a V_{\pi_i}(s')$

Policy Iteration Algorithm

Repeat till

$$\pi_{k+1} = \pi_k$$

$$\Downarrow$$

$$\pi_k = \pi_*$$

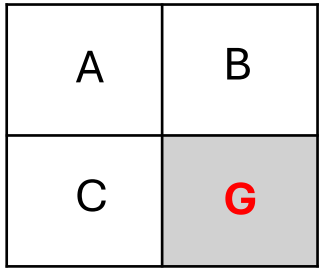
$$\pi_0 \rightarrow V_{\pi_0} \rightarrow \pi_1 \rightarrow V_{\pi_1} \rightarrow \pi_2 \rightarrow V_{\pi_2} \rightarrow \ldots$$

Policy Evaluation

Policy Improvement

- Policy Evaluation: Iteratively apply BE equation $V_{k+1}(s) = R_s^{\pi} + \sum_{s'} P_{ss'}^{\pi} V_k(s')$
- Policy Improvement: $\pi_{i+1}(s) \coloneqq argmax_a\ R_s^a + \sum_{s'} P_{ss'}^{a} V_{\pi_i}(s')$
- $V_{\pi_{i+1}} \geq V_{\pi_i}$ due to Policy Improvement Theorem

Grid Example: **Policy Iteration**



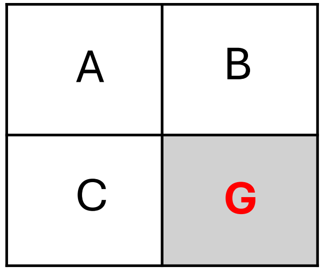
A	B
C	**G**

Deterministic Transitions

$$\pi_0 \underset{\substack{\uparrow \\ \text{Policy} \\ \text{Evaluation}}}{\rightarrow} V_{\pi_0} \underset{\substack{\uparrow \\ \text{Policy} \\ \text{Improvement}}}{\rightarrow} \pi_1 \rightarrow V_{\pi_1} \rightarrow \pi_2 \rightarrow V_{\pi_2} \rightarrow \ldots$$

Exercise

- Take π_0 as a uniform random policy
- Apply Policy iteration algorithm
- Show the sequence of policies that we get



Deterministic Transitions

Value Iteration

Value Iteration

Bellman Optimality (BO)

$$V^*(s) = \max_{\mathrm{a}} R_s^a + \sum_{s'} P_{ss'}^a V^*(s')$$ (Optimal Substructure)

Value Iteration

Iteratively apply BO equation till $V_{k+1} = V_k$

$$V_{k+1}(s) = \max_{\mathrm{a}} R_s^a + \sum_{s'} P_{ss'}^a V_k(s')$$

Optimal Policy from V^*

$$\pi^*(s) = \arg\max_{\mathrm{a}} R_s^a + \sum_{s'} P_{ss'}^a V^*(s')$$

Implementation Details

- Issue: Takes a very long time to see $V_{k+1} = V_k$
- Solution: Stop when $||V_{k+1} - V_k||$ is small
- Typically, max-norm is used

Implementation Details

- Synchronous updates

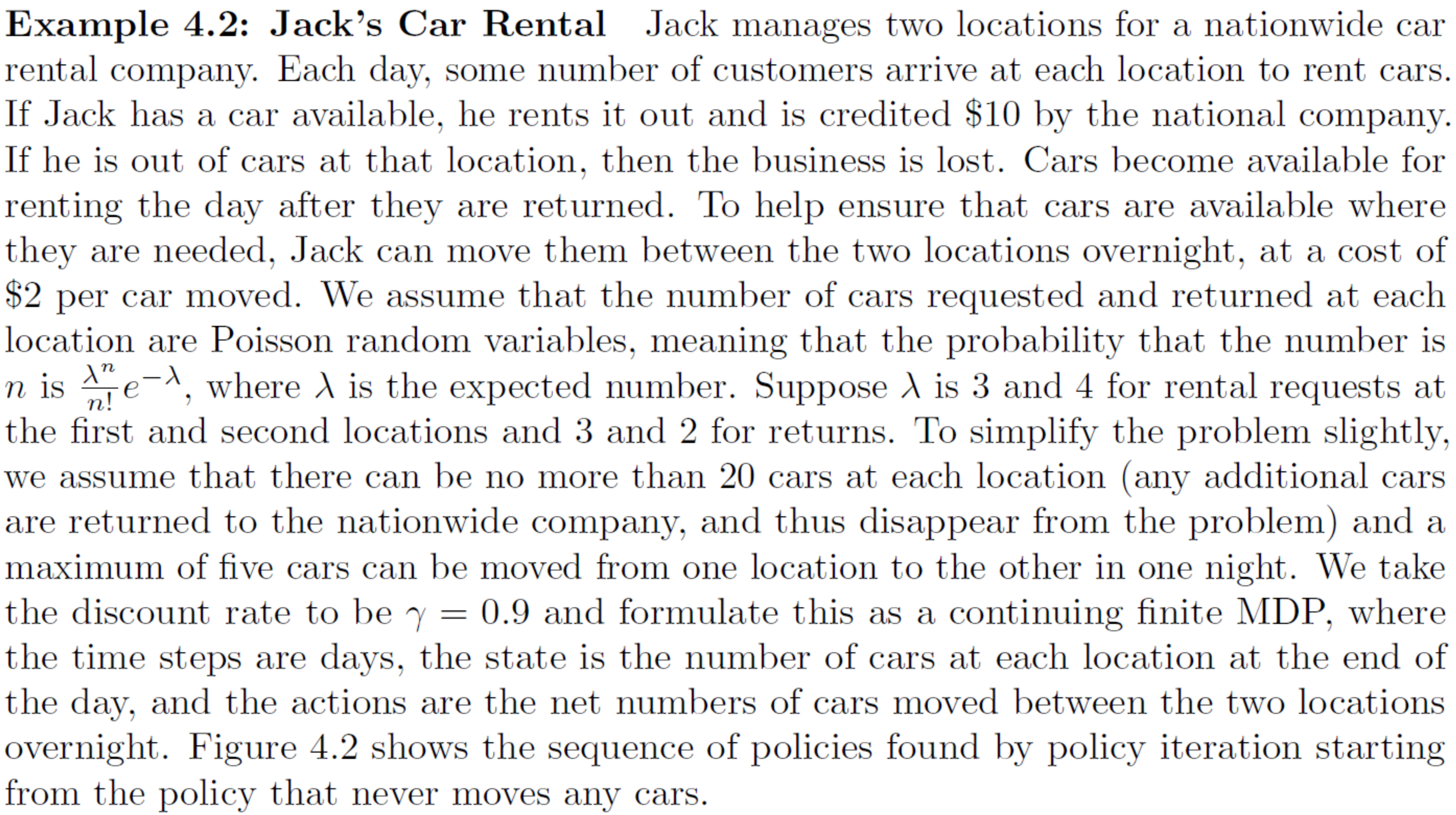
- In-place updates

- Asynchronous updates

Car rental Example

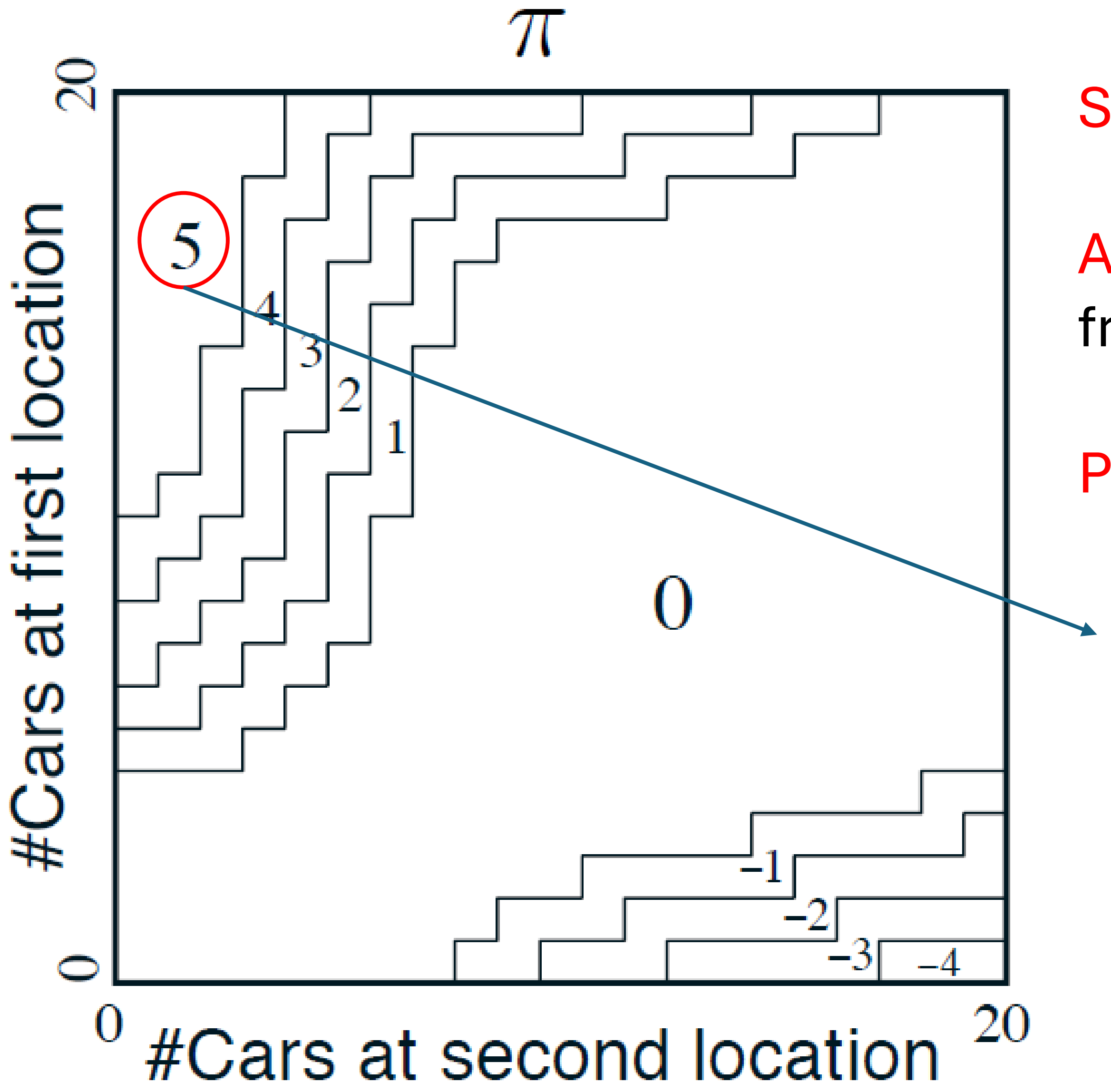
- A car rental company operates in two cities
- Customers arrive at these cities and rent a car for $10. If a customer arrives when cars are unavailable: Business Loss
- #car requests and returns are Poisson random variables
- At most, 20 cars can be parked at each location
- Upto 5 Cars can be transferred overnight between cities at a 3$ cost
- Problem to solve: How many cars should be transferred to maximise profit?

Car Rental Example



Example 4.2: Jack's Car Rental Jack manages two locations for a nationwide car rental company. Each day, some number of customers arrive at each location to rent cars. If Jack has a car available, he rents it out and is credited \$10 by the national company. If he is out of cars at that location, then the business is lost. Cars become available for renting the day after they are returned. To help ensure that cars are available where they are needed, Jack can move them between the two locations overnight, at a cost of \$2 per car moved. We assume that the number of cars requested and returned at each location are Poisson random variables, meaning that the probability that the number is n is $\frac{\lambda^n}{n!}e^{-\lambda}$, where λ is the expected number. Suppose λ is 3 and 4 for rental requests at the first and second locations and 3 and 2 for returns. To simplify the problem slightly, we assume that there can be no more than 20 cars at each location (any additional cars are returned to the nationwide company, and thus disappear from the problem) and a maximum of five cars can be moved from one location to the other in one night. We take the discount rate to be $\gamma = 0.9$ and formulate this as a continuing finite MDP, where the time steps are days, the state is the number of cars at each location at the end of the day, and the actions are the net numbers of cars moved between the two locations overnight. Figure 4.2 shows the sequence of policies found by policy iteration starting from the policy that never moves any cars.

Car Rental Problem: An Example Policy



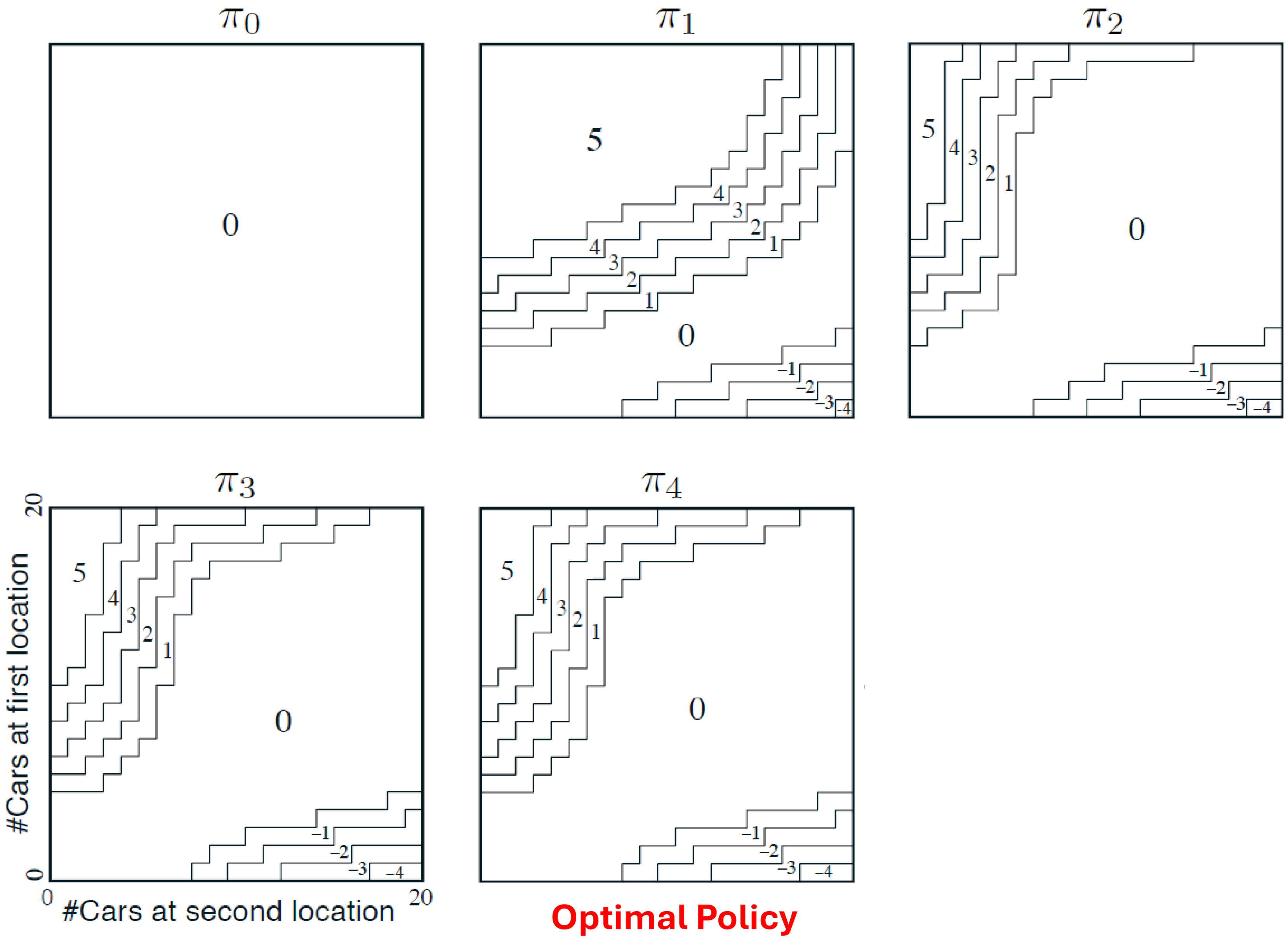
States: (#cars at loc_1, #cars at loc_2)

Actions: How many cars to transfer from loc_1 to loc_2

Policy: Mapping from state to action

If location 1 has many cars (e.g., loc_1 =20), and location 2 has a few cars (e.g., loc_2= 0), transfer 5 cars.

Policy Iteration



Optimal Policy

Model-Free RL

Prof. Subrahmanya Swamy

How to find V_π for a given π?

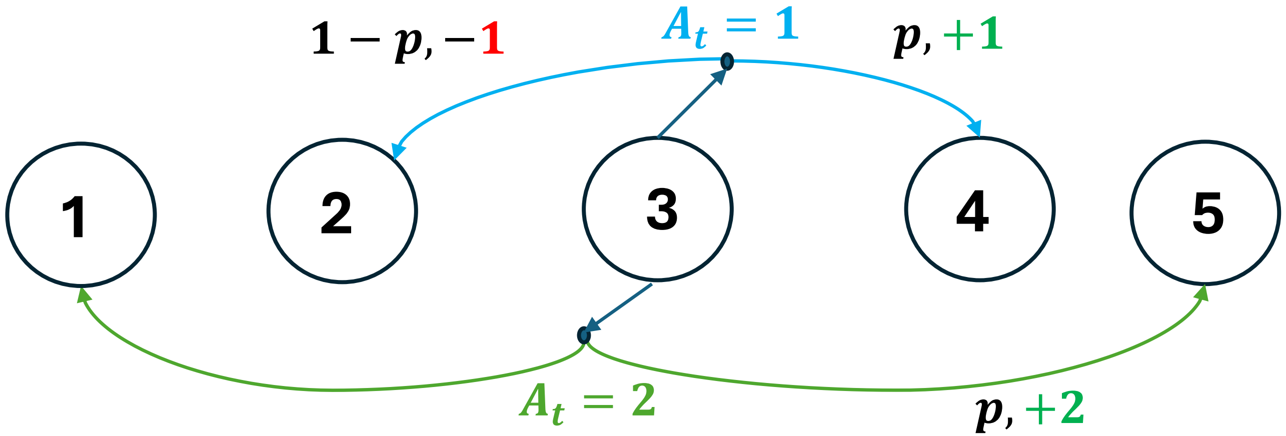
Policy Evaluation

$$V_{k+1}(s) = R_s^\pi + \sum_{s'} P_{ss'}^\pi \, V_k(s')$$

$$R_s^\pi = \sum_a \pi(a \mid s) \; R_{ss'}^a$$

$$P_{ss'}^\pi = \sum_a \pi(a \mid s\,) \; P_{ss'}^a$$

Requires complete knowledge of the environment: MDP model



Model-Free RL: Unknown $R_s^a, P_{ss'}^a$

Task	Model Available	Model Unknown
Policy Evaluation V_π	Iterative Policy Evaluation	??
Optimal Policy π^*	Policy Iteration, Value Iteration	??

Learn through real-time interaction with environment

$(S_{t+1,} R_{t+1})$

Environment

Agent

Action (A_t)

State (S_t)

Monte-Carlo (MC) method to estimate V_π

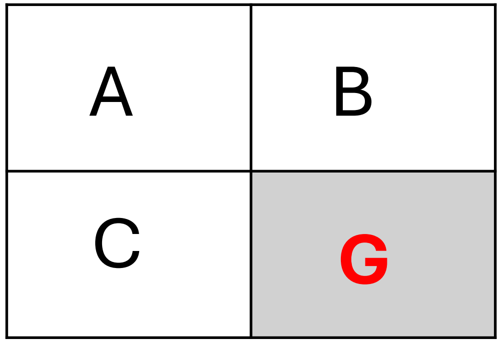
- $V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$

- Interact with the environment and generate multiple episodes of data
  - Episode 1: $S_0 = s,\ A_0 \sim \pi,\ R_1,\ S_1,\ A_1 \sim \pi,\ R_2,\ S_2\ , \ldots,\ S_T$
  - Episode 2: $S_0 = s,\ A_0 \sim \pi,\ R_1,\ S_1,\ A_1 \sim \pi,\ R_2,\ S_2\ , \ldots,\ S_T$
  - …
  - …

- Compute sample returns of each episode from state s
  - $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots$

- $V_\pi(s) \approx$ sample avg of the returns

Monte-Carlo: Grid Example



A	B
C	G

Uniform Random Policy

Sample Episode

$A, Right, -1, B, Left, -1, A, Down, -1, C, Right, -1, G$

MC First-Visit

- Two states: $\{A, B\}$

- Observed episodes:
  - A, 1, B, -2, B, 4, A, 0, B, -2 → Terminated
  - B, -1, B, 3, A, 2, B, 0, A , -3 → Terminated

- Observed returns
  - Episode 1: Return from first-visit of state A: $1-2+4+0-2=1$
  - Episode 1: Return from first-visit of state B: $-2+4+0-2=0$
  - Episode 2: Return from first-visit of state A: $2+0-3=-1$
  - Episode 2: Return from first-visit of state B: $-1+3+2+0-3=1$

- MC estimates (average of observed returns):
  - $V(A) \approx \frac{1}{2}(1-1)=0$
  - $V(B) \approx \frac{1}{2}(0+1)=\frac{1}{2}$

MC Every-Visit

- Two states: $\{A, B\}$

- Observed episodes:
  - A, 1, B, -2, B, 4, A, 0, B, -2 → Terminated
  - B, -1, B, 3, A, 2, B, 0, A , -3 → Terminated

- Observed returns
  - Episode 1: Returns from all-visits of state A:
  - Episode 1: Return from all-visits of state B:
  - Episode 2: Return from all-visits of state A:
  - Episode 2: Return from all-visits of state B:

- MC estimates (average of observed returns):
  - $V(A) \approx$
  - $V(B) \approx$