Team Airline
midterm presentation

Our Project

pricing simulation: mimics a market for selling airplane tickets

- goal: compare performance and time consumption of different methods
- implemented several DP methods to calculate optimal solution
- suitable for all stable baselines algorithms
- comprehensive monitoring to analyze pricing strategies and compare RL-based policies with the optimal one

The pricing problem

- monopoly scenario: agent vs customers
- dynamic pricing

- **problem context**:

    time horizon: finite (→ take off)

    action: price offer

    demand: price and time-dependent

    rewards: sales revenue

    goal: maximize profits

The pricing problem

- monopoly scenario: agent vs customers
- dynamic pricing

- **problem context**:

  time horizon: finite (→ take off)

  action: price offer

  demand: price and time-dependent

  rewards: sales revenue

  goal: maximize profits

- **environment**:

  state space: (booking time, current occupancy)

  (0, 12) → (10, 0)

  action space: discrete (0, 1, .., 20)

  event: tickets sold per step (0, 10)

  reward: $action * event$ (no final reward)

  $\mathrm{p} = \left(1 - \frac{a}{20}\right) * (1 + t)/T$

  demand: $Binomial(10,\ k,\ p)$

  episode: one booking period

The pricing problem

- monopoly scenario: agent vs customers
- dynamic pricing
- MDP
- **problem context**:

  time horizon: finite (→ take off)

  action: price offer

  demand: price and time-dependent

  rewards: sales revenue

  goal: maximize profits

- **environment**:

  state space: (booking time, current occupancy)

  (0, 12) → (10, 0)

  action space: discrete (0, 1, .., 20)

  event: tickets sold per step (0, 10)

  reward: $action * event$ (no final reward)

  $\mathrm{p} = \left(1 - \frac{a}{20}\right) * (1 + t)/T$

  demand: $Binomial(10,\ k,\ p)$

  episode: one booking period

The pricing problem

- monopoly scenario: agent vs customers
- dynamic pricing
- MDP
- **problem context**:
  - time horizon: finite (→ take off)
  - action: price offer
  - demand: price and time-dependent
  - rewards: sales revenue
  - goal: maximize profits

- **environment**:
  - state space: (booking time, current occupancy) (0, 12) → (10, 0)
  - action space: discrete (0, 1, .., 20)
  - event: tickets sold per step (0, 10)
  - reward: $action * event$ (no final reward)
  - $\mathrm{p} = \left(1 - \frac{a}{20}\right) * (1 + t)/T$
  - demand: $Binomial(10,\ k,\ p)$
  - episode: one booking period

What would a good policy look like?

A simulation scenario

- **optimal solution** (full knowledge)

  → calculated by dynamic programming
- **approximate DP solution**
- **RL solution** (limited knowledge)
  - Q-Learning (lr=0.8, eps=0.5)
  - DQN, DDPG, TD3, A2C, SAC, PPO

    (default stable baselines parameters)
- setup: training 100, 500 and 1000 episodes per method (n=5)
- evaluation: policy, cumulated reward, initial value of policy

A simulation scenario

- **optimal solution** (full knowledge)

  → calculated by dynamic programming
- **approximate DP solution**
- **RL solution** (limited knowledge)
  - Q-Learning (lr=0.8, eps=0.5)
  - DQN, DDPG, TD3, A2C, SAC, PPO

    (default stable baselines parameters)
- setup: training 100, 500 and 1000 episodes per method (n=5)
- evaluation: policy, cumulated reward, initial value of policy

- **environment**:

  state space: (booking time, current occupancy)

  (0, 12) → (10, 0)

  action space: discrete (0, 1, .., 20)

  event: tickets sold per step (0, 10)

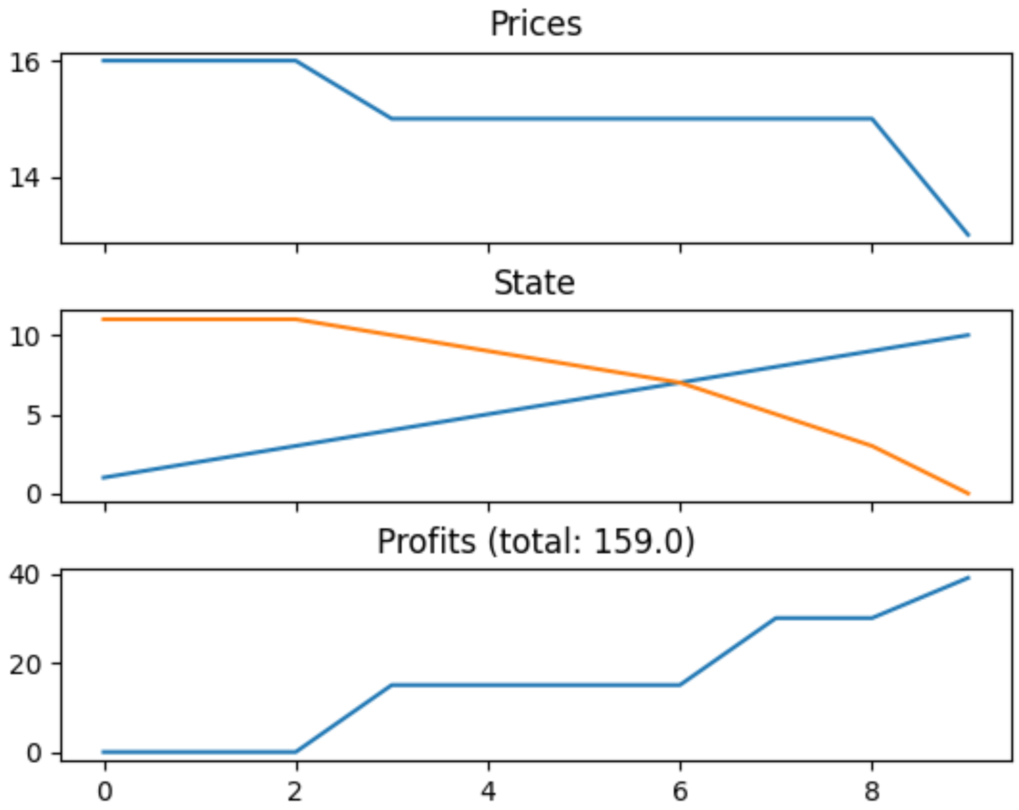
  reward: $action * event$ (no final reward)

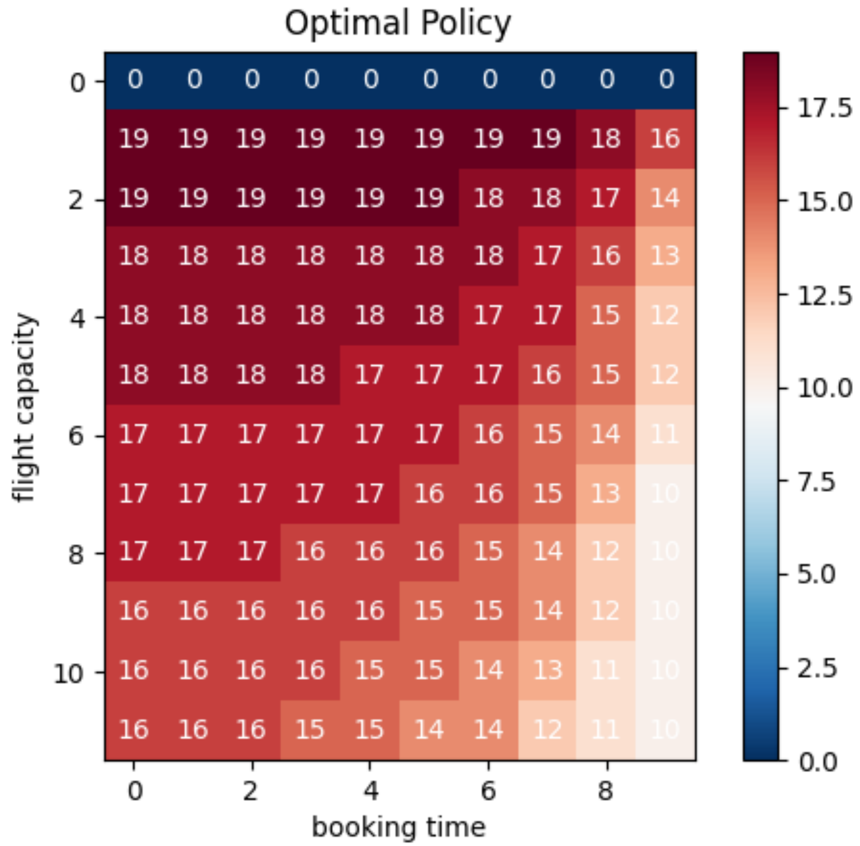
  $p = \left(1 - \frac{a}{20}\right) * (1 + t)/T$

  demand: $Binomial(10,\ k,\ p)$

  episode: one booking period

A simulation scenario – optimal policy



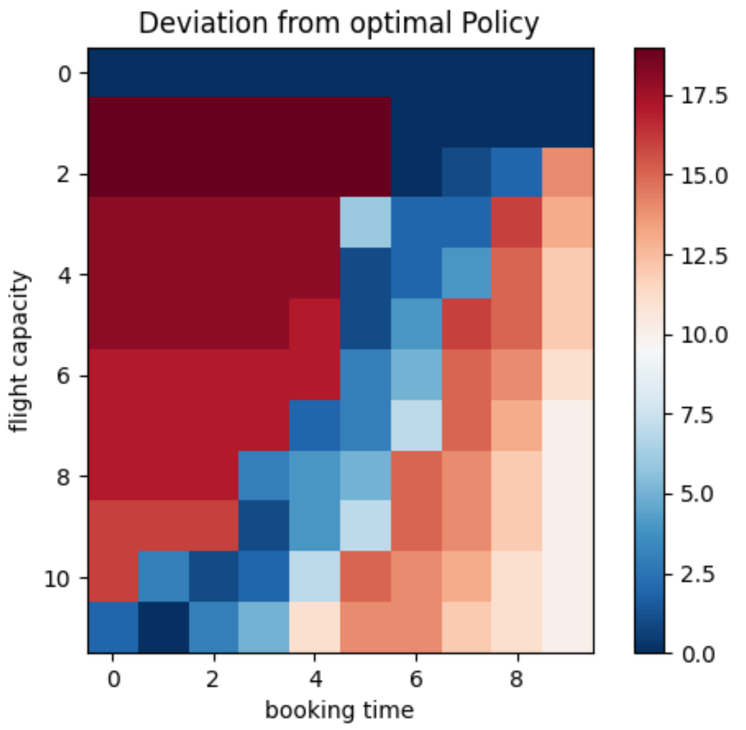


Backward Induction: 0.577s, Policy Iteration: 4.637s, Value Iteration: 6.487s

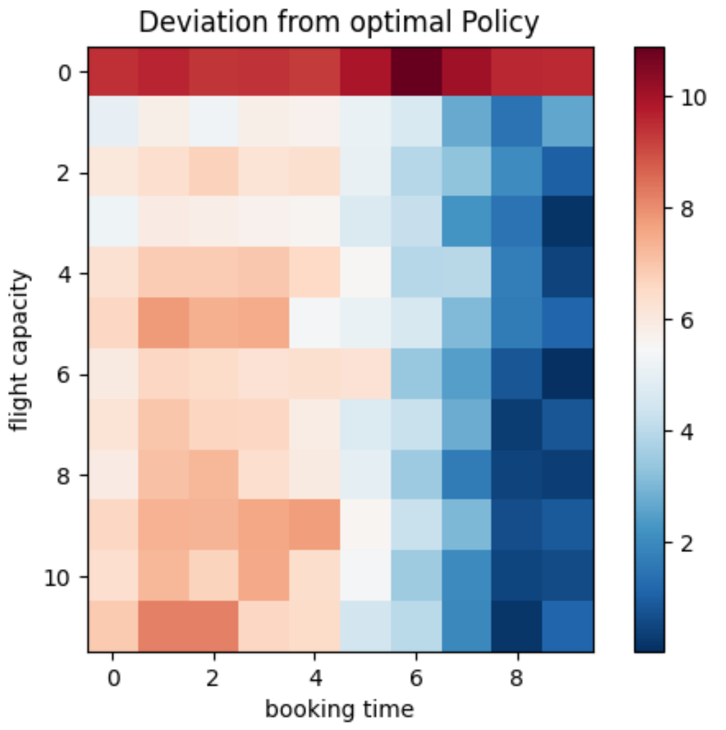
A simulation scenario – selected policies based on limited knowledge

After 100 Training Episodes

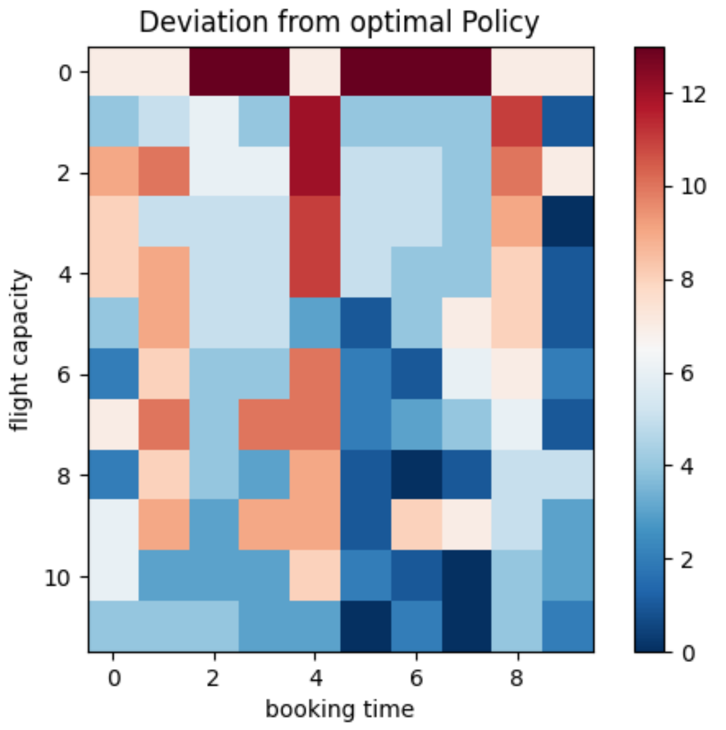
Approximate DP



SAC

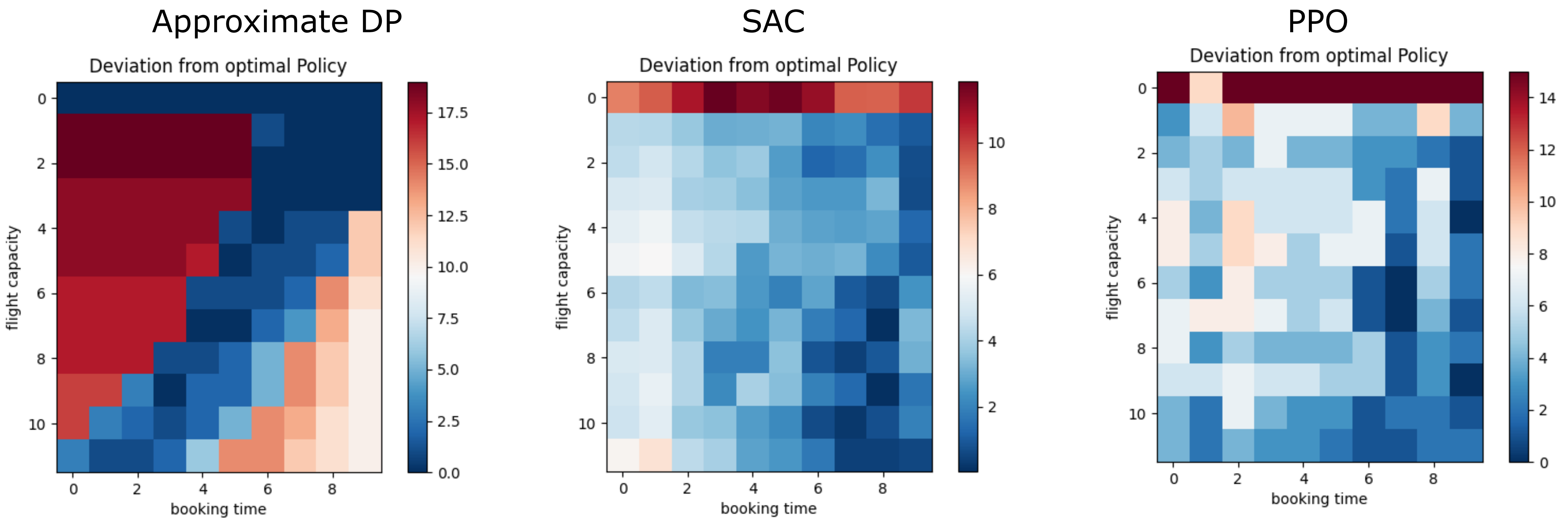


PPO



A simulation scenario – selected policies based on limited knowledge

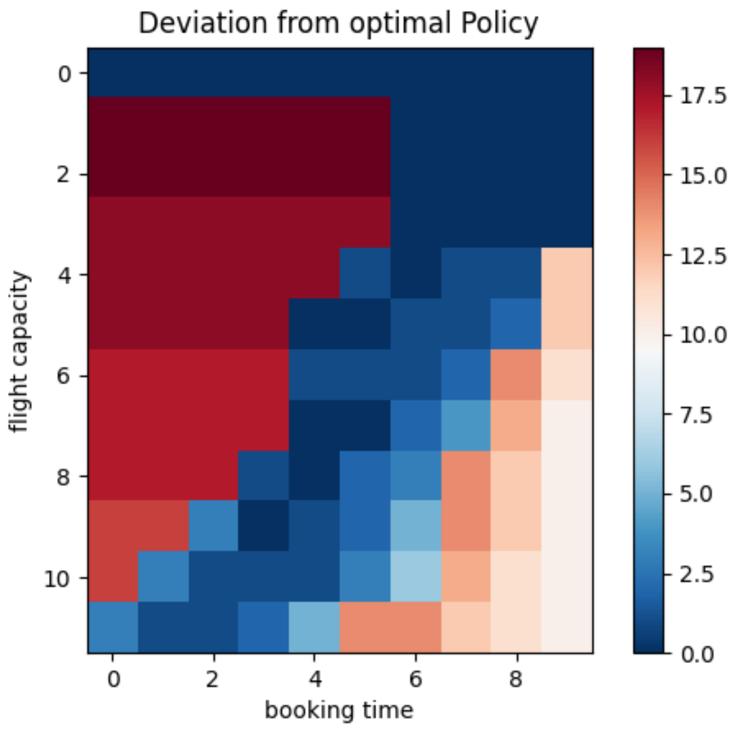
After 500 Training Episodes



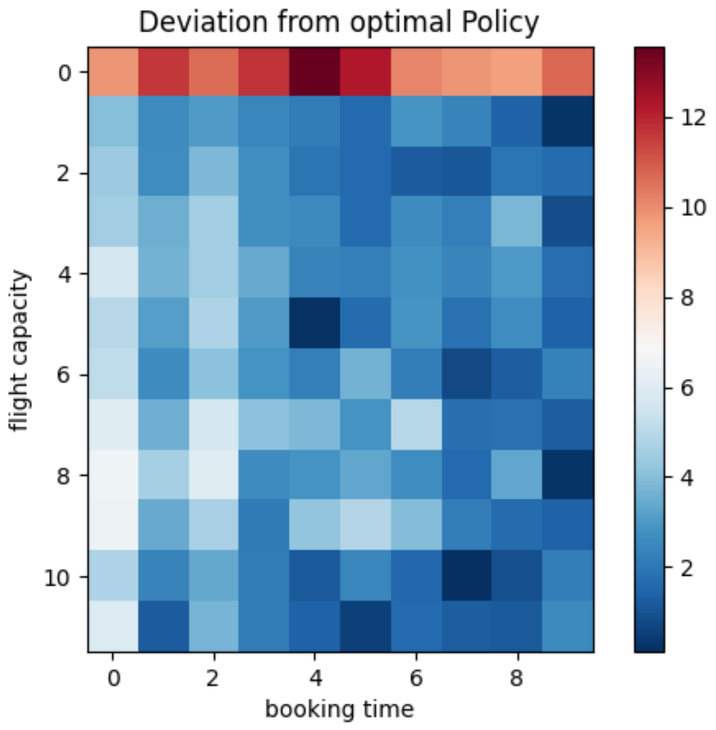
A simulation scenario – selected policies based on limited knowledge

After 500 Training Episodes

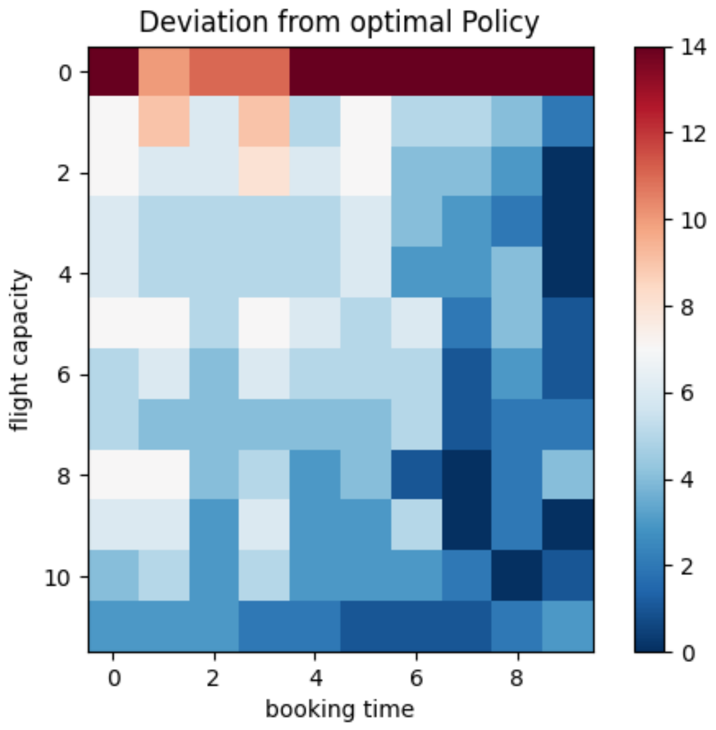
Approximate DP



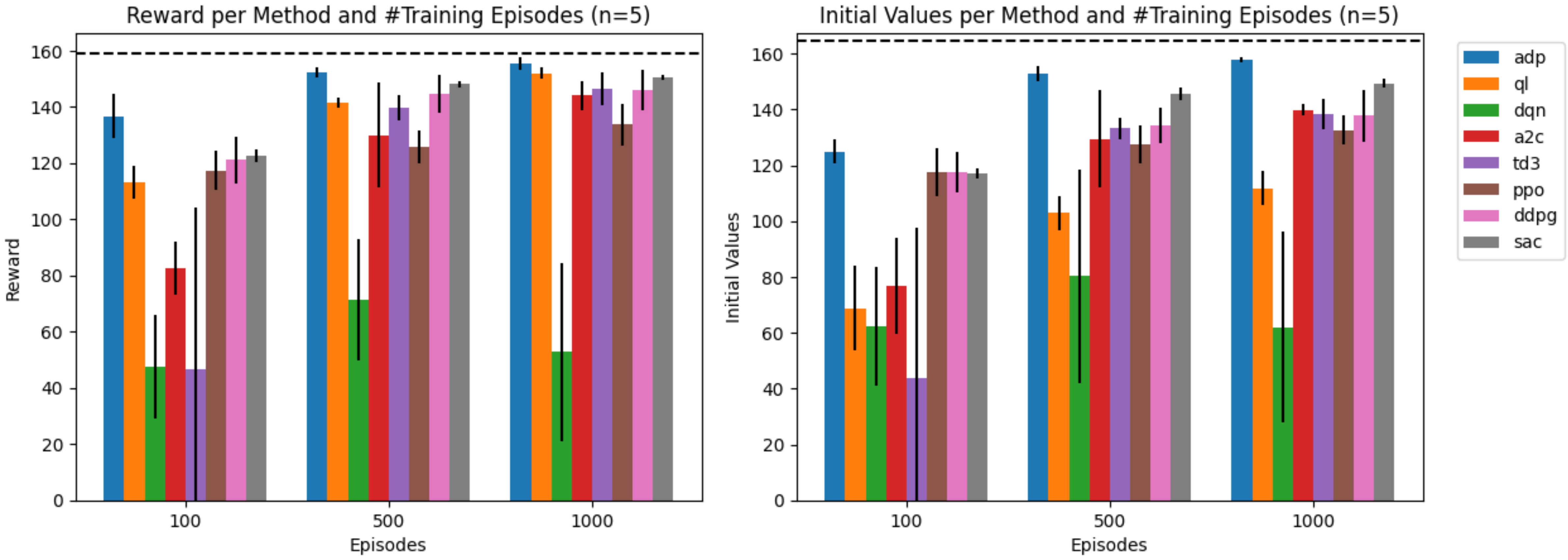
SAC



PPO



A simulation scenario – total rewards



Outlook

- more complex customer behavior (various customer types)
- agent predicts demand based on historical data (regressive)
- multidimensional state space to support multiple flights
  (demand substitution effects)
- introduce competition to agent

What do you think is the most interesting direction?


D-AISP
HB-IJU
OO-SNF