Towards RL for Operations?

Jim Dai

Cornell University & CUHK-Shenzhen

June 19, 2024

RL4SN, ENSEEIHT, Toulouse

Computational Experiences with Proximal Policy Optimization

- Queueing Network Controls via Deep Reinforcement Learning (2021)
  - J. G. Dai and Mark Gluzman, *Stochastic Systems*
- Scalable Deep Reinforcement Learning for Ride-Hailing (2021)
  - Jiekun Feng, J. G. Dai and Mark Gluzman, *IEEE Control Systems Letters*
- Inpatient Overflow Management with Proximal Policy Optimization (2024)
  - J. G. Dai, Pengyi Shi, Jingjing Sun
- Deep RL algorithms are scalable in solving MDP problems modeling these operations

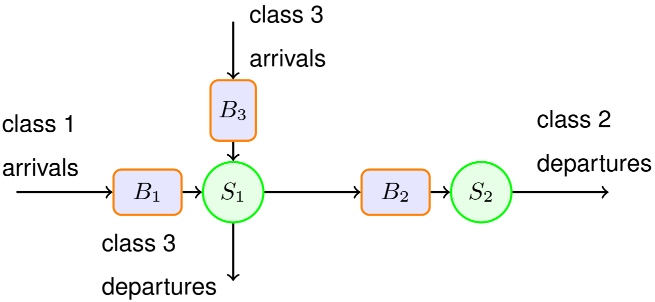
Queueing Network Controls via Deep RL

Mark Gluzman

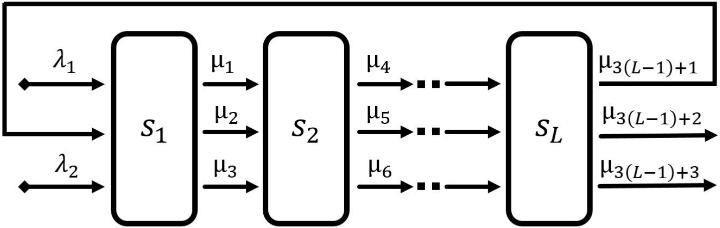
Meta

- Stochastic processing network (SPN) examples
- Proximal Policy Optimization (PPO) Algorithm in countable state space
- Numerical examples

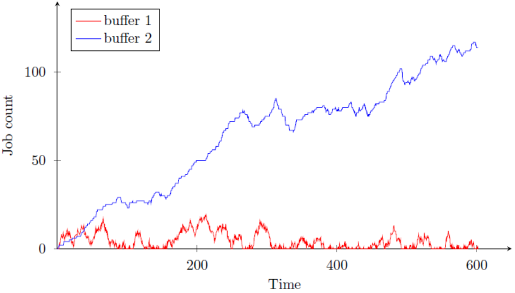
SPN example I, criss-cross queueing network



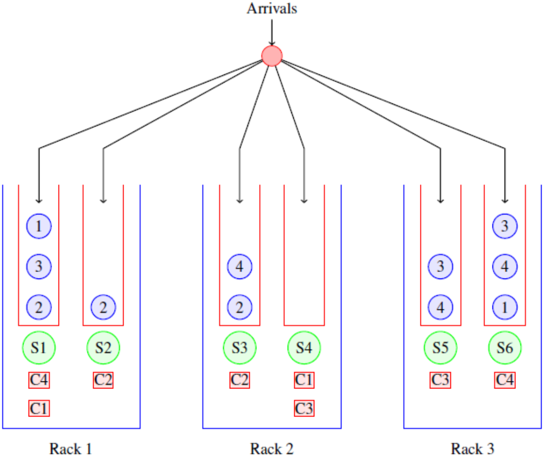
SPN example II: Re-entrant queueing networks (PR Kumar 1993)



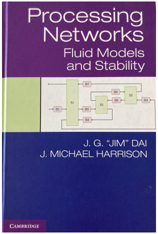
- Many apparently "good" policies can be unstable. Bramson, Lu-Kumar, Rybko-Stolyar, ...



SPN application I: data-intensive server farm



Dai-Harrison (2020), Processing Networks: Fluid Models and Stability, Cambridge University Press.



PPO algorithm, general formulation

- We consider an MDP with countable state space $\mathcal{X}$, finite action space $\mathcal{A}$, one-step cost $g(x) \geq 0$, and transition function $P(y|x, a)$.
- Consider a class of randomized Markovian policies π_θ, $\theta \in \Theta$. Under the policy π_θ, the transition matrix:

$$P_\theta(y|x) = \sum_{a \in \mathcal{A}} \pi_\theta(a|x) P(y|x, a) \text{ for } x, y \in \mathcal{X}.$$

- Assume each Markov chain P_θ is irreducible and aperiodic (not essential).
- Find $\theta \in \Theta$ to minimize the long-run average cost

$$\limsup_{N \to \infty} \frac{1}{N} \mathbb{E}_{\pi_\theta} \left[\sum_{k=0}^{N-1} g(x^{(k)}) \right], \tag{1}$$

which is independent of the initial state, $x^{(k)}$ is the state of the Markov chain P_θ after k timesteps.

Poisson equation

- Assume that the Markov chain P_θ has the (unique) stationary distribution, which is denoted by μ_θ.
- Long-run average cost in (1) is equal to $\mu_\theta^T g = \sum_{x \in \mathcal{X}} \mu_\theta(x) g(x)$.
- Assume that Poisson equation has a solution $h_\theta = h$

$$g(x) - \mu_\theta^T g + \sum_{y \in \mathcal{X}} P_\theta(y|x) h(y) - h(x) = 0 \quad \text{for each } x \in \mathcal{X}.$$

- An advantage function $A_\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ of policy π_θ:

$$A_\theta(x, a) := \mathop{\mathbb{E}}_{y \sim P(\cdot|x,a)} \left[g(x) - \mu_\theta^T g + h_\theta(y) - h_\theta(x) \right].$$

- When $\mathcal{X}$ is finite, both assumptions are satisfied.

When the state space is infinite: drift condition

- Drift condition: $\exists\, V : \mathcal{X} \to [1, \infty)$, $b \in (0,1)$, $d \geq 0$, a finite $C \subset \mathcal{X}$ such that

$$\sum_{y \in \mathcal{X}} P_\theta(y|x) V(y) \leq bV(x) + d\, \mathbb{I}_C(x), \quad \text{for each } x \in \mathcal{X}.$$

- (Meyn-Tweedie) If P_η satisfies the drift condition with some Lyapunov function $\mathcal{V} \geq 1$, P_η has a unique stationary distribution μ_η.
- Assume further $g \leq \mathcal{V}$. Poisson equation has the fundamental solution

$$h_\eta(x) := \mathbb{E}\left[\sum_{k=0}^{\infty} \left(g(x^{(k)}) - \mu_\eta^T g\right) \mid x^{(0)} = x\right] \text{ for each } x \in \mathcal{X}. \tag{2}$$

where $x^{(k)}$ is the state of the Markov chain P_η after k timesteps.

Some notations

- Assume policy π_η, $\eta \in \Theta$ satisfies the drift condition with Lyapunov function V.
- For a vector ν on $\mathcal{X}$, V-norm is defined as

$$\|\nu\|_{\infty,V} := \sup_{x\in\mathcal{X}} \frac{|\nu(x)|}{V(x)}.$$

- For a matrix M, $V-$norm is defined as

$$\|M\|_V := \sup_{x\in\mathcal{X}} \frac{1}{V(x)} \sum_{y\in\mathcal{X}} |M(x,y)|V(y).$$

- Fundamental matrix

$$Z_\eta := \sum_{k=0}^{\infty} (P_\eta - \Pi_\eta)^k \,,$$

where, for each $x, y \in \mathcal{X}$, $\Pi_\eta(x,y) := \mu_\eta(y)$.

Kakade-Langford-type lemma: long-run average version

Lemma 1

- Suppose P_η satisfies the drift condition with Lyapunov function V.
- $|g| \le V$.
- $\theta \in \Theta$ is close to η such that

$$D_{\theta,\eta} := \|(P_\theta - P_\eta)Z_\eta\|_V < 1.$$

Then, P_θ has a stationary distribution μ_θ, and

$$\begin{aligned}
\mu_\theta^T g - \mu_\eta^T g &= \mathop{\mathbb{E}}_{x\sim\mu_\theta, a\sim\pi_\theta(\cdot|x)}[A_\eta(x,a)] \qquad (3)\\
&= \mathop{\mathbb{E}}_{x\sim\mu_\eta, a\sim\pi_\eta(\cdot|x)}\left[\frac{\pi_\theta(x,a)}{\pi_\eta(x,a)}A_\eta(x,a)\right] + \Delta(\theta,\eta)\\
&\equiv L(\theta,\eta) + \Delta(\theta,\eta).
\end{aligned}$$

A bound on $\Delta(\theta, \eta)$

Theorem

Under the assumptions of Lemma 1, we have

$$\Delta(\theta,\eta) \le \delta(\theta,\eta) := \frac{D^2_{\theta,\eta}}{1 - D_{\theta,\eta}} \times$$
$$\times \left(1 + \frac{D_{\theta,\eta}}{(1 - D_{\theta,\eta})}(\mu_\eta^T V)\|I - \Pi_\eta + P_\eta\|_V \|Z_\eta\|_V\right) \left\|g - (\mu_\eta^T g)e\right\|_{\infty,V} (\mu_\eta^T V).$$

- Thus, we have

$$\mu_\theta^T g - \mu_\eta^T g \le \underbrace{L(\theta,\eta) + \delta(\theta,\eta)}_{\text{Surrogate function}}.$$

- When $D_{\theta,\eta}$ is small, $L(\theta,\eta) = O(D_{\theta,\eta})$ and $\delta(\theta,\eta) = O(D^2_{\theta,\eta})$.
- Conservative update: minimize$_\theta L(\theta,\eta)$ while keeping $D_{\theta,\eta}$ is small.

Proximal Policy Optimization

- Constrained optimization: minimize $L(\theta, \eta)$ while keeping $D_{\theta,\eta}$ small.

Lemma 2

Define ratio $r_{\theta,\eta}(a|x) := \frac{\pi_\theta(a|x)}{\pi_\eta(a|x)}$ and $G_\eta(x,a) := \frac{1}{V(x)} \sum_{y \in \mathcal{X}} \pi_\eta(a|x) P(y|x,a) V(y)$.

$$D_{\theta,\eta} \leq \|Z_\eta\|_V \sup_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} |r_{\theta,\eta}(a|x) - 1| \, G_\eta(x,a).$$

- The lemma says that $D_{\theta,\eta}$ is small when the ratio $r_{\theta,\eta}(a|x)$ is close to 1.
- Following Schulman et al. (2017), we solve an unconstrained optimization problem by minimizing the clipped surrogate objective over θ

$$L^\epsilon(\theta,\eta) := \mathop{\mathbb{E}}_{\substack{x \sim \mu_\eta \\ a \sim \pi_\eta(\cdot|x)}} \max \left[r_{\theta,\eta}(a|x) A_\eta(x,a),\ \text{clip}(r_{\theta,\eta}(a|x), 1-\epsilon, 1+\epsilon) A_\eta(x,a) \right],$$

where $\epsilon > 0$ is a hyper-parameter.

PPO as a Policy Improvement Algorithm

- Given policy π_η; find an improved policy π_{η^*}. Define ratio $r_{\theta,\eta}(a|x) := \frac{\pi_\theta(a|x)}{\pi_\eta(a|x)}$
- Fix $\epsilon > 0$ as a hyper-parameter. Define

$$L^\epsilon(\theta, \eta) := \mathop{\mathbb{E}}_{\substack{x\sim\mu_\eta \\ a\sim\pi_\eta(\cdot|x)}} \max\left[r_{\theta,\eta}(a|x)A_\eta(x,a),\ \text{clip}\Big(r_{\theta,\eta}(a|x), 1-\epsilon, 1+\epsilon\Big)A_\eta(x,a)\right],$$

- μ_η is the stationary distribution of P_η.
- Policy improvement: from π_η to π_{η^*}, where

$$\eta^* = \text{argmin}_\theta\, L^\epsilon(\theta, \eta).$$

PPO: Markov chain Monte Carlo

- Under policy π_η an episode is generated:
$$E = \left\{ x^{(0)}, a^{(0)}, x^{(1)}, a^{(1)}, \cdots, x^{(K-1)}, a^{(K-1)} \right\}. \tag{4}$$

- Based on the generated episode (4), the Monte-Carlo estimate of $L^\epsilon(\theta, \eta)$ is
$$\hat{L}^\epsilon(\theta, \eta, E) := \frac{1}{K} \sum_{k=0}^{K-1} \max\Big[r_{\theta,\eta}(a^{(k)}|x^{(k)}) A_\eta(x^{(k)}, a^{(k)}), \\ \text{clip}\left(r_{\theta,\eta}(a^{(k)}|x^{(k)}), 1-\epsilon, 1+\epsilon \right) A_\eta(x^{(k)}, a^{(k)}) \Big].$$

- We use ADAM to find $\eta^* = \operatorname{argmin}_\theta \hat{L}^\epsilon(\theta, \eta, E)$.

- Open problem: π_η-drift condition implies π_{η^*}-drift condition?

Numerical results for the re-entrant networks

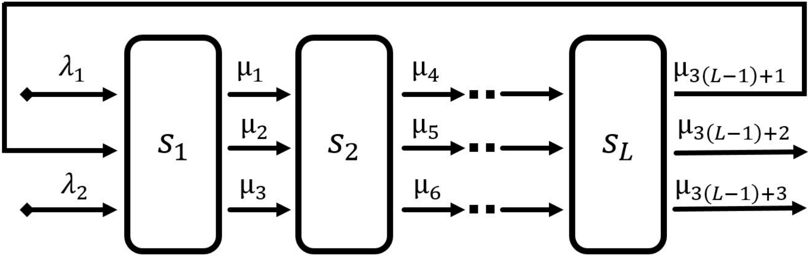


Figure: The extended six-class network.

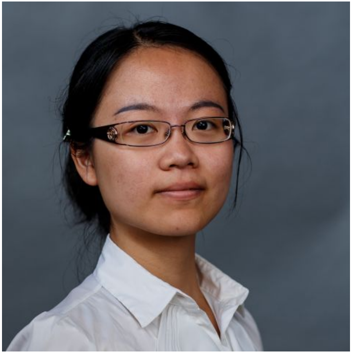
PPO wins in a challenging queueing control problem

$\#$ of classes, $3L$	LBFS	FCFS	FP	RFP	Our method
6	15.749	40.173	15.422	15.286	14.130 ± 0.208
9	25.257	71.518	26.140	24.917	23.269 ± 0.251
12	34.660	114.860	38.085	36.857	32.171 ± 0.556
15	45.110	157.556	45.962	43.628	39.300 ± 0.612
18	55.724	203.418	56.857	52.980	51.472 ± 0.973
21	65.980	251.657	64.713	59.051	55.124 ± 1.807

Table: Simulation results for the heavy-loaded ($\rho_\ell = 0.9$) re-entrant networks.

- Robust fluid policy (RFP): Bertsimas-Nasrabadi-Paschalidis (2015).
- Fluid policy (FP): Avram-Bertsimas-Ricard (1995).

Scalable Deep Reinforcement Learning for Ride-Hailing



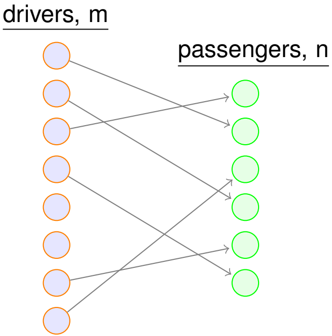
Aurora Feng

Teza Technologies

Mark Gluzman

Meta

Matching and repositioning: Curse of dimension on # of matches



- Without repositioning, $\frac{m!}{(m-n)!}$ possible “batch” actions.

- Hierarchical decisions
  - One driver at a time, sequentially
  - trip-type (RL algorithm)
  - (o,d) pair of regions
  - driver-passenger (platform)

Scalability: atomic policy to generate trip types

- At each epoch, for example, 9:01am
- RL generates trip-types sequentially, following a trained atomic policy
- The term "atomic actions" was coined in Feng-D-Gluzman (2021)

Theorem (D-Wu-Zhang 2024)

An optimal atomic policy is optimal for the original problem.

Manxi Wu

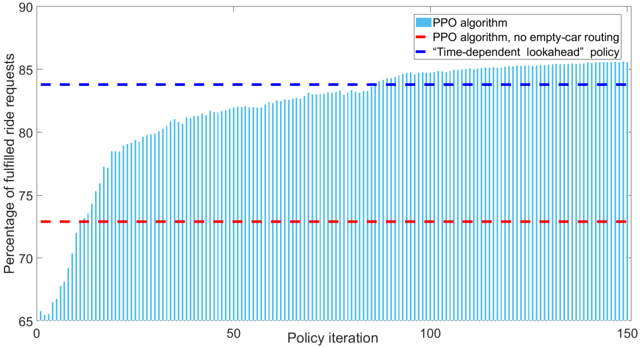
Cornell ORIE

Zhanhao Zhang

Cornell ORIE

Experiments: 9-region network

- The 9-region transportation networks from Braverman et al. 2019 [1] is based on the real-world data released by the Didi Research Institute.

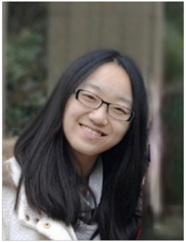


A transportation network consisting of $R = 9$ regions, $N = 2000$ cars, and $H = 240$ minutes.

[1] A. Braverman, J. G. Dai, X. Liu, and Y. Lei, Empty-car routing in ridesharing systems, Operations Research, 2019.

Inpatient Overflow Management with Proximal Policy Optimization

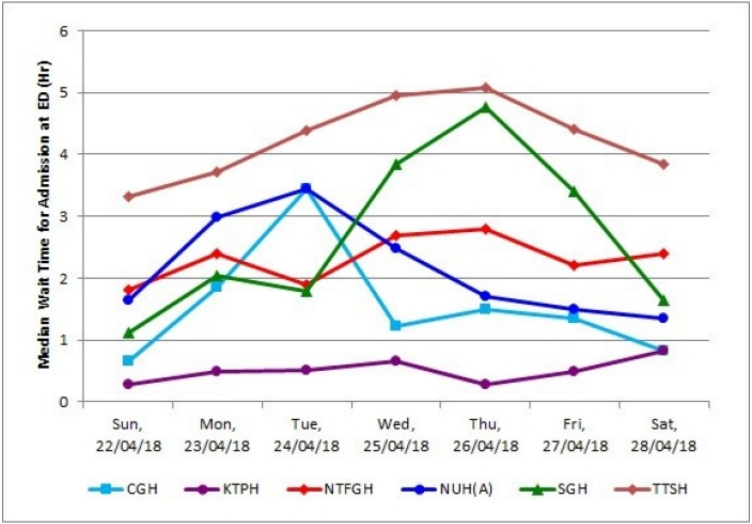
Pengyi Shi
Krannert, Purdue

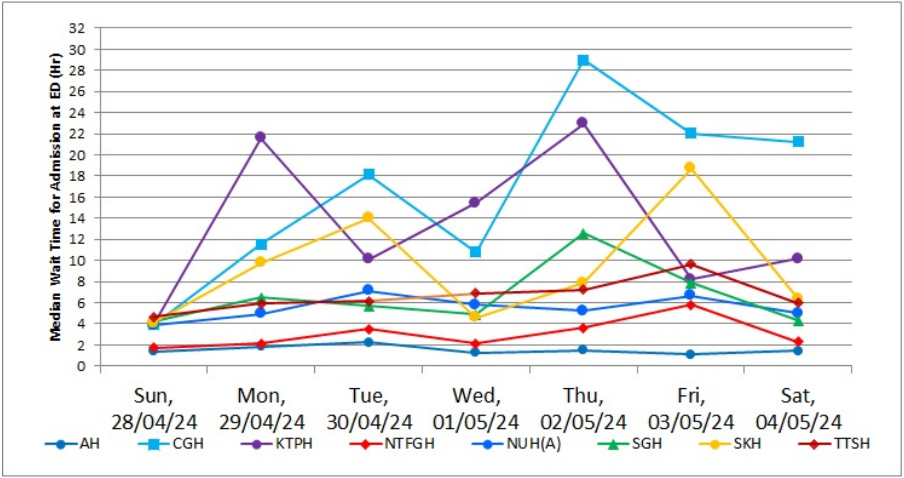


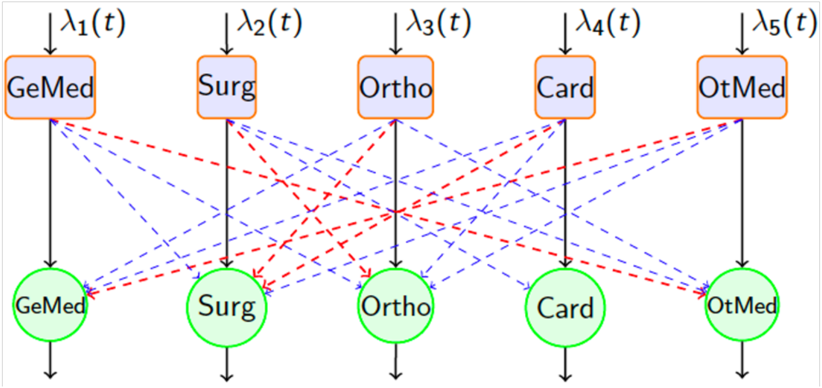
Jingjing Sun
CUHK-Shenzhen

Hospital inpatients, from ED to wards

Waiting time for beds in Singapore (MOH website: April 2018)

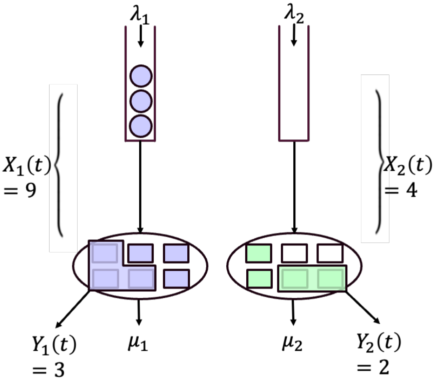



Median Wait Time for Admission at ED (Hr)
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
Sun, 28/04/24
Mon, 29/04/24
Tue, 30/04/24
Wed, 01/05/24
Thu, 02/05/24
Fri, 03/05/24
Sat, 04/05/24
AH
CGH
KTPH
NTFGH
NUH(A)
SGH
SKH
TTSH

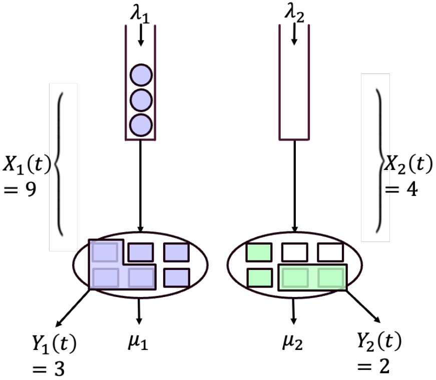

$\lambda_1(t)$
$\lambda_2(t)$
$\lambda_3(t)$
$\lambda_4(t)$
$\lambda_5(t)$
GeMed
Surg
Ortho
Card
OtMed
GeMed
Surg
Ortho
Card
OtMed

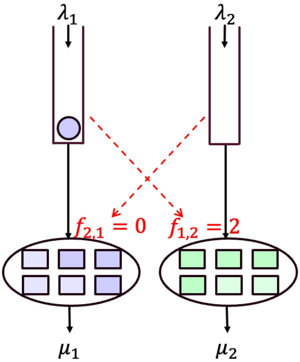
MDP: States

$$S(t) = \Big(\underbrace{X_1(t), \cdots, X_J(t)}_{\text{Patient count}}, \quad \underbrace{Y_1(t), \cdots, Y_J(t)}_{\text{to-be-discharge count}}, \quad \underbrace{h(t)}_{\text{time}} \Big)$$



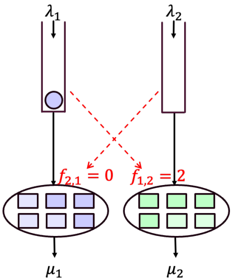
MDP: Actions

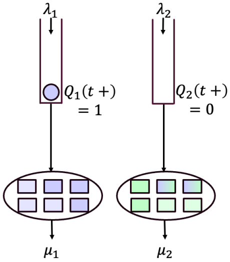




Actions: $f(t) = \{f_{12}(t), f_{21}(t)\}$

MDP: One-Step Cost and Objective





- One-step cost:

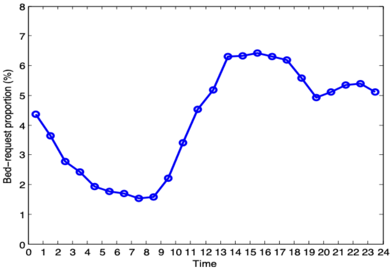
$$g\big(S(t), f(t)\big) = \sum_{i \neq j} B_{ij} f_{i,j}(t) + \sum_j C_j Q_j(t+).$$

- Average-cost objective:

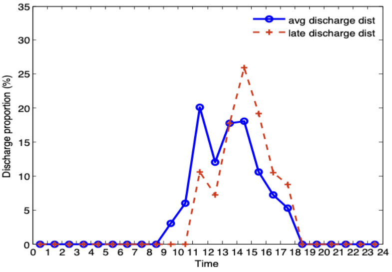
$$\textbf{Min}_f \lim_{N\to\infty} \frac{1}{N} \mathbb{E}\Big[\sum_{t=1}^{N} g\big(S(t), f(t)\big)\Big].$$

Periodic MDP

The bed request (arrival) and discharge (departure) pattern is periodic.



(a) Empirical bed-request distribution



(b) Empirical discharge time distribution

Consider m decision epochs a day and denote the kth decision epoch at day t and t_k, our objective can be rewritten as

$$\textbf{Min}_f \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{m} g\big(S(t_k), f(t_k)\big)\right]$$

Prior Work: Dai and Shi (2019)-ADP approach

- Value function approximation with queueing based features: $\hat{V}_\pi(s) = \langle \beta_\pi, \phi(s) \rangle$
- At each state, search an action in the large action space to update policy:

$$\min_{f \in \mathcal{A}} \left[g(s, f) + \mathbb{E}_{s' \sim P(\cdot|s,f)} \hat{V}_\pi(s') \right].$$

- Works well in 5-pool system.

Challenge: Large Action Space

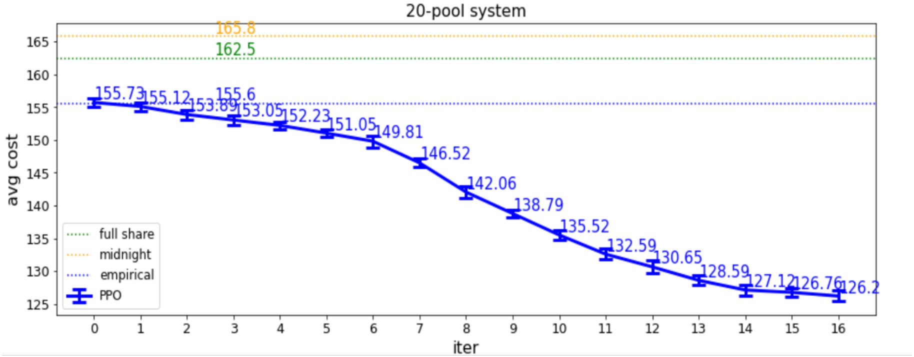
Example: In a twenty-pool system, assume the capacity in each pool is 60. At state with

$$(x_1, ..., x_{20}) = (65, 63, 62, 50, 50, 50, ..., 50),$$

where there are 10 waiting customers from three classes, and 10 idle servers in 17 partially-occupied server pools, the action space size is:

$$C_{17}^{5+17} \cdot C_{17}^{3+17} \cdot C_{17}^{2+17} \approx 5.13 \cdot 10^9$$

PPO + atomic policies on a 20-pool model



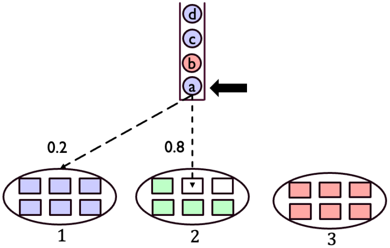
Key Factors Contributed to the PPO Success

- **Atomic policies:** Decompose actions into a sequence of atomic actions. Atomic policies have small action space.
- **Periodic MDP:** Policy NN design.
- **Value function approximation[2]:** Use LSTD method to reduce estimation variance in the long-run average cost setting.

Randomized atomic policy

- At 3pm, state $s = (9, 4, 7, y_1, y_2, y_3, 15:00)$, with four waiting patients.
- A sequence of "atomic actions" will be taken at 3pm, one step at a time.
- In step 1, set $s^1 = s$. Suppose

$$\pi(a^1 = 1 \mid s^1, c^1 = 1) = 0.2 \quad \text{waiting}$$
$$\pi(a^1 = 2 \mid s^1, c^1 = 1) = 0.8 \quad \text{pool 2}$$



- Suppose 1st step atomic action $a^1 = 1$, patient "a" continues to wait, and $s^2 = s^1$

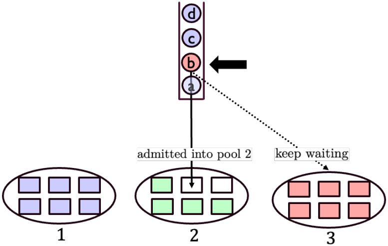
Second Step Atomic Action

- In step 2, updated state $s^2 = (9, 4, 7, y_1, y_2, y_3, 15:00)$.

Suppose

$\pi(a^2 = 3 \mid s^2, c^2 = 3) = 0.6$ waiting

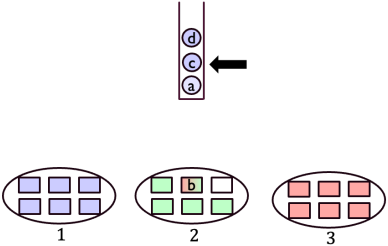
$\pi(a^2 = 2 \mid s^2, c^2 = 3) = 0.4$ pool 2



- And 2nd step atomic action $a^2 = 2$.

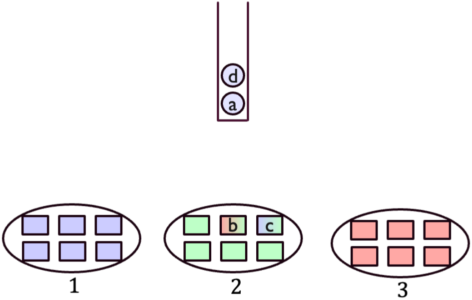
Patient ‘b” is routed to pool 2

$s^3 = (9, 5, 6, y_1, y_2, y_3, 15:00)$



After 4th Atomic Action

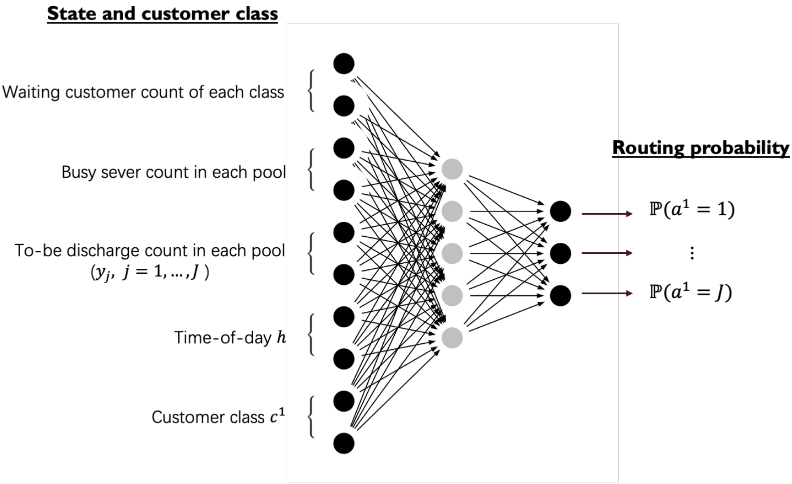
- Post-action state: $s^5 = (8, 6, 6, y_1, y_2, y_3, 15:00)$



- At 15:30 (next decision epoch), a new sequence of atomic actions will be taken.

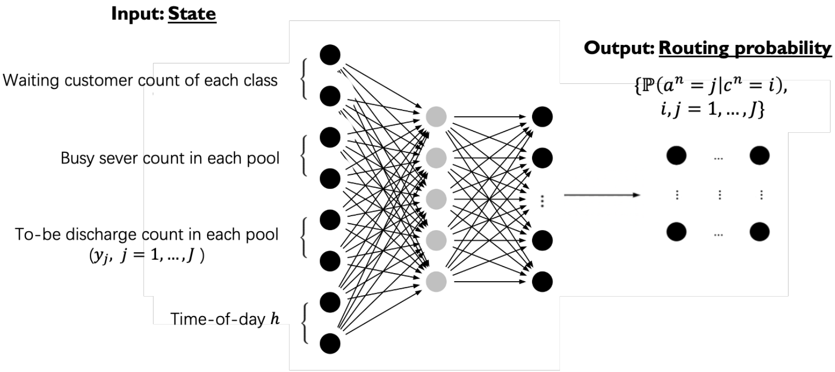
Randomized Policy Parameterized by NN

- Neural network θ to parameterize randomized policy π_θ

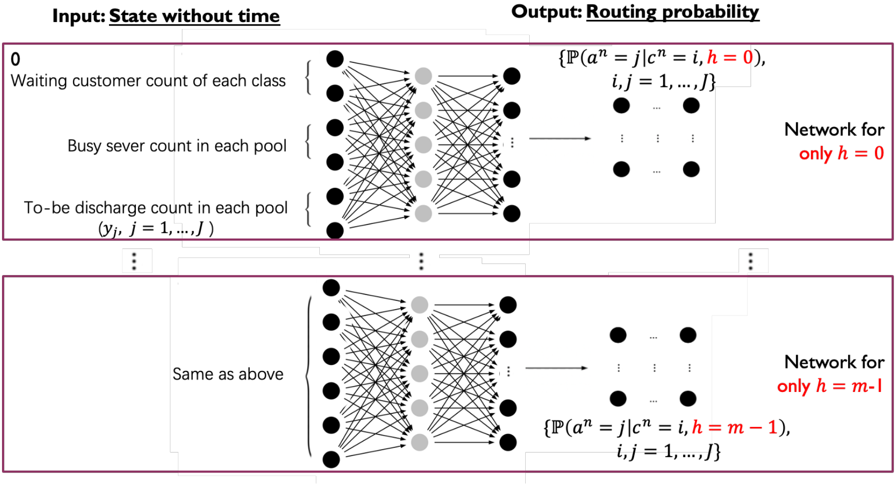


PPO Hyper-parameter: NN Structure

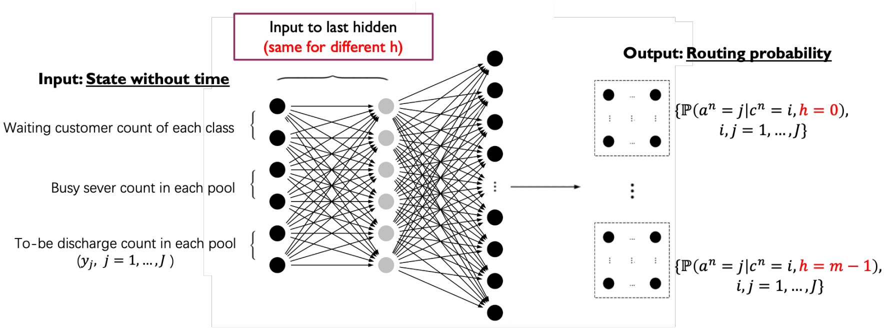
Choice 1: Fully-connect



Choice 2: Fully-separate



Choice 3: Partial-share



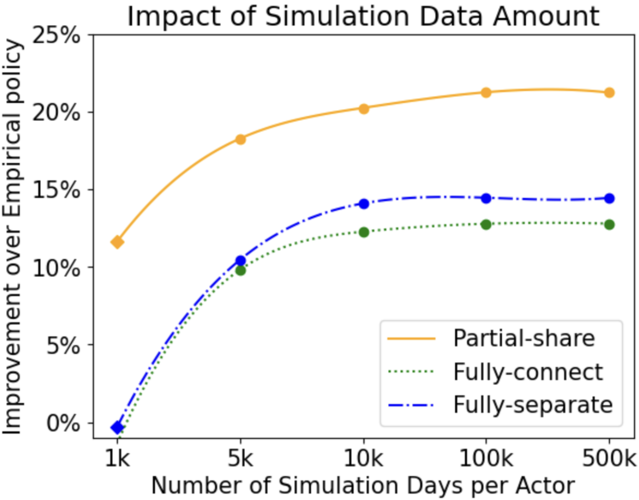
Numerical results: Baseline Hyper-parameters

Conduce experiments in 10-pool model with baseline PPO hype-parameters:

Parameters	Baseline Choice
NN depth	One hidden layer
NN hidden layer width	34 neurons
Basis function	$(X, X^2, Y, Y^2, XY, V_s)$ from Dai and Shi (2019)
Initial policy	Complete-overflow policy
Simulation days per actor	10,000
Number of actors	10
Number of training epochs	15
Clipping parameter (ϵ)	0.5

Table: Baseline choice of PPO hyper-parameters in ten-pool setting.

Numerical results: Partial-share Policy Network is Sample Efficient



Summary

- Atomic polices can be optimal (D-Wu-Zhang 2024)
- PPO + atomic policy can drastically reduce discrete action space
- PPO is a policy improvement algorithm
  - Policy evaluation is data-expensive
  - Can be implemented by using operational data? + synthetic data?
- For periodic MDPs, policy NN design improves sample efficiency

Some reflections

- Is RL relevant to operations?
  - Game
  - Robotics
  - LLM
  - Inventory
  - FE
  - ...
  - Load
  - Time-varying
  - Behavior
- Killer application?

Some further thoughts

- Diffusion control problems
  - Han-Jentzen-E (2018), PNAS
  - Ata, Harrison, Si (2023, 2024)

- Standard test problems?

- TSP Test Data

  U Waterloo

Appendix

N-model

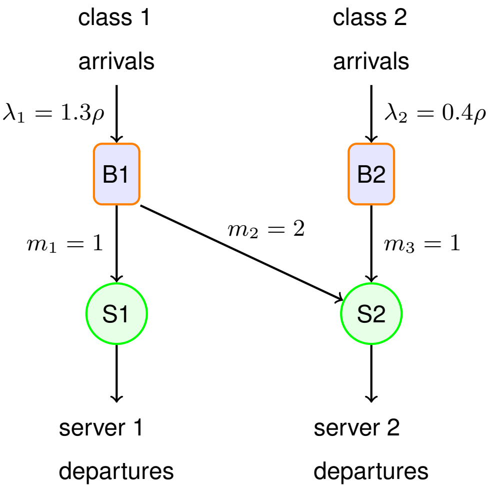
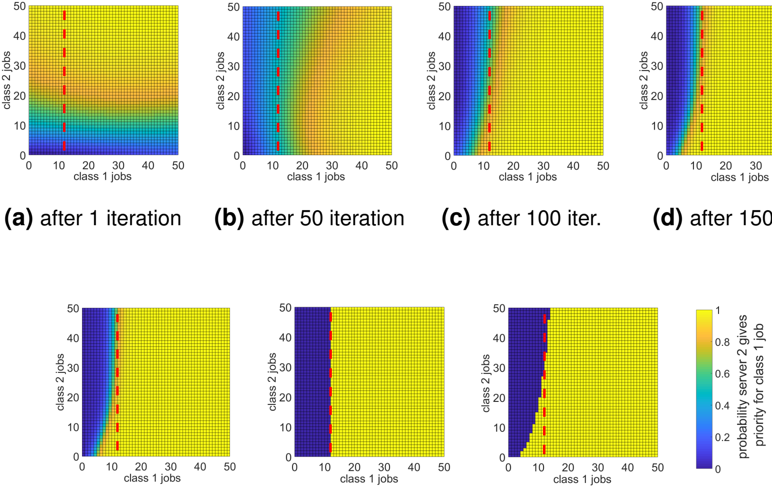


Figure: N-model network with $\rho = 0.95$

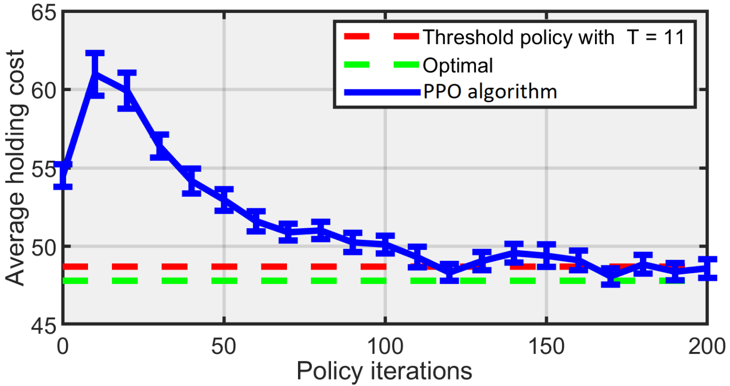
Numerical results for N-model



(a) after 1 iteration **(b)** after 50 iteration **(c)** after 100 iter. **(d)** after 150 iter.

(e) after 200 iter. **(f)** Threshold policy **(g)** Optimal policy

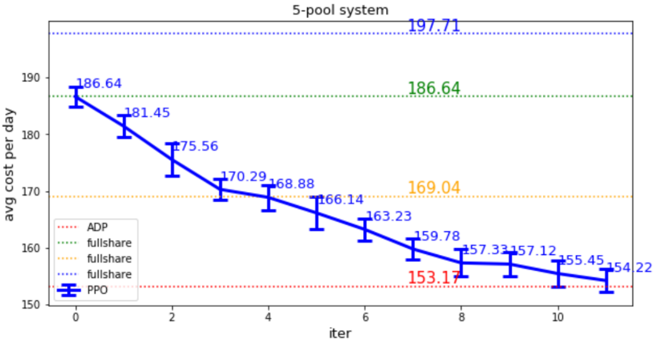
Numerical results for N-model



Numerical results: 5-pool

- PPO method improve greatly over naive policies
- PPO method obtain comparable results with ADP method
- Time: 2.5h per iteration for PPO

  10h per iteration for ADP



PPO Hyper-parameter: NN Structure

Tailored NN structure designs to improve sample efficiency.

- Input excludes customer class c^n

  Output $\mathbb{P}(a^n = j | c^n = i), \ i, j = 1, ..., J;$

- Input excludes time-of-day h

  Output $\mathbb{P}(a^n = j | c^n = i, h = l), \ i, j = 1, ..., J, l = 0, ..., m - 1;$

Numerical results: PPO Hyper-parameters