Structured Actor-Critic for Managing Public Health Points-of-Dispensing

Daniel Jiang
joint work with Yijia Wang

Operations Management Seminar, April 9, 2021
Singapore Management University

Outline

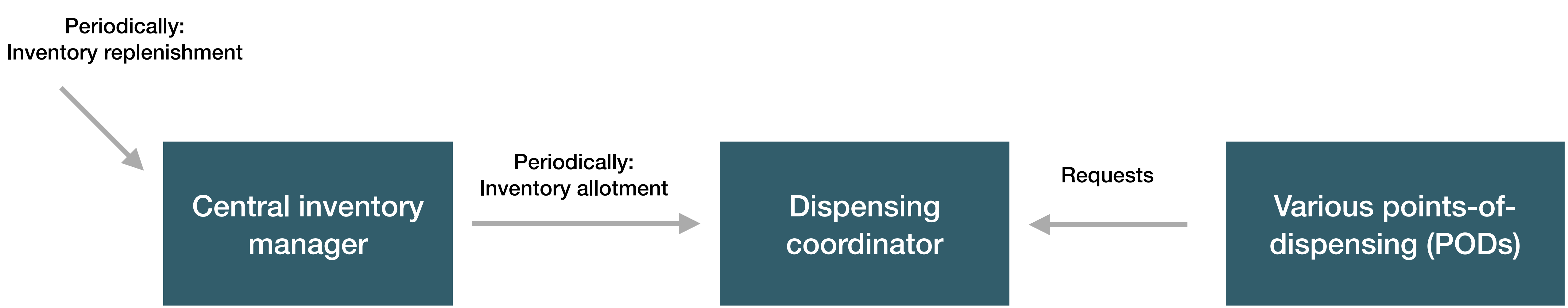
- Introduction to the public health problem
- Hierarchical MDP inventory and dispensing model
- The structured actor-critic approach
- Synthetic experiments
- Case study: *Naloxone for First Responders Program*

Introduction

- Public health organizations manage "points-of-dispensing" (PODs) for dispensing critical medical supplies during emergency situations.
  - Examples: vaccines, antibiotics, and others, such as **naloxone**, an opioid overdose reversal drug for harm reduction.
- Our problem: **optimal inventory control and dispensing** for a public health agency and "independent" PODs.

Problem preview

- Components of our problem:
  - A central inventory storage managed by the public health agency
    - Inventory is replenished periodically
  - A lower-level *dispensing coordinator* that interfaces with PODs
    - Receives inventory from central storage
    - Receives requests from arriving PODs (demands)

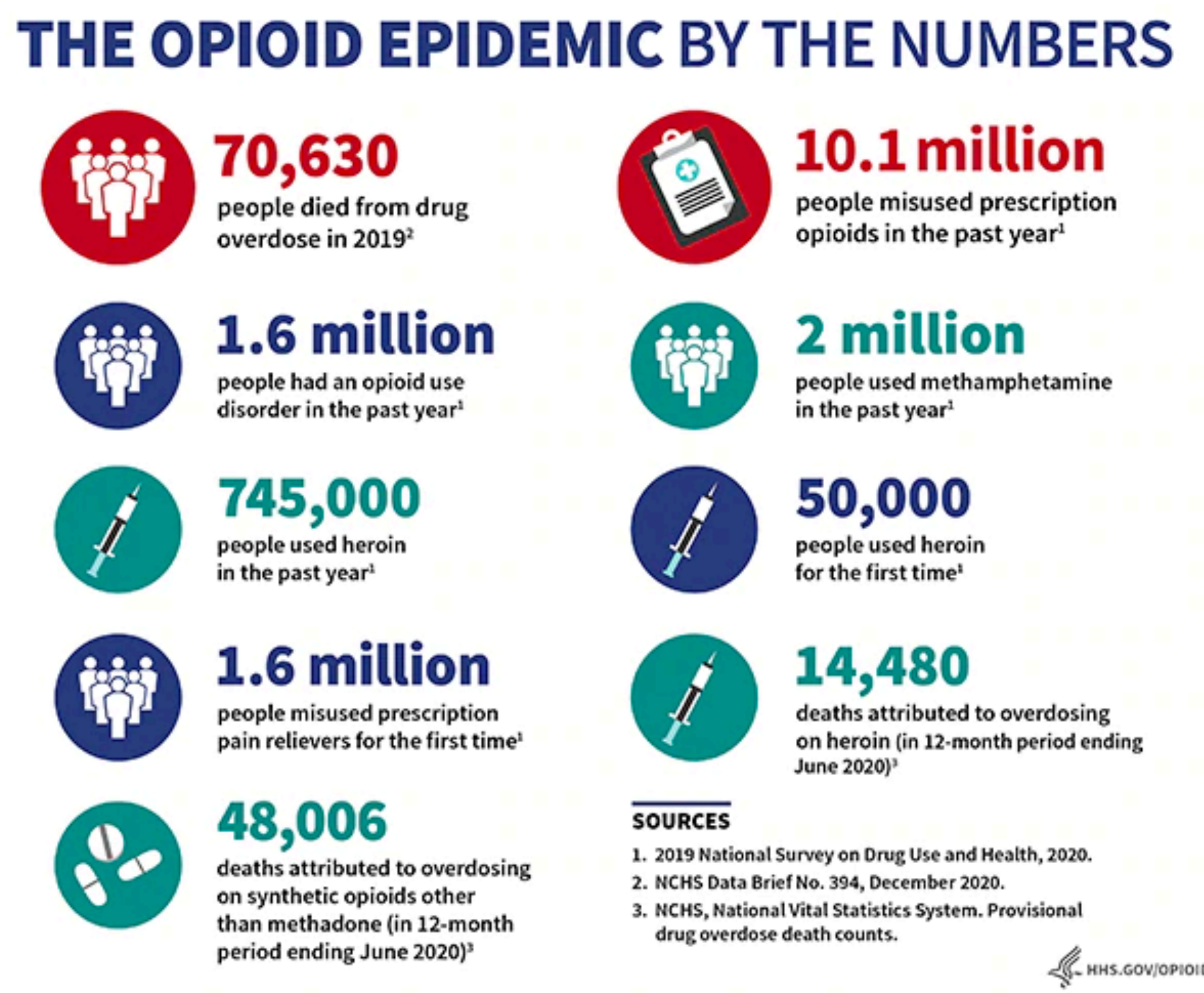


Problem preview

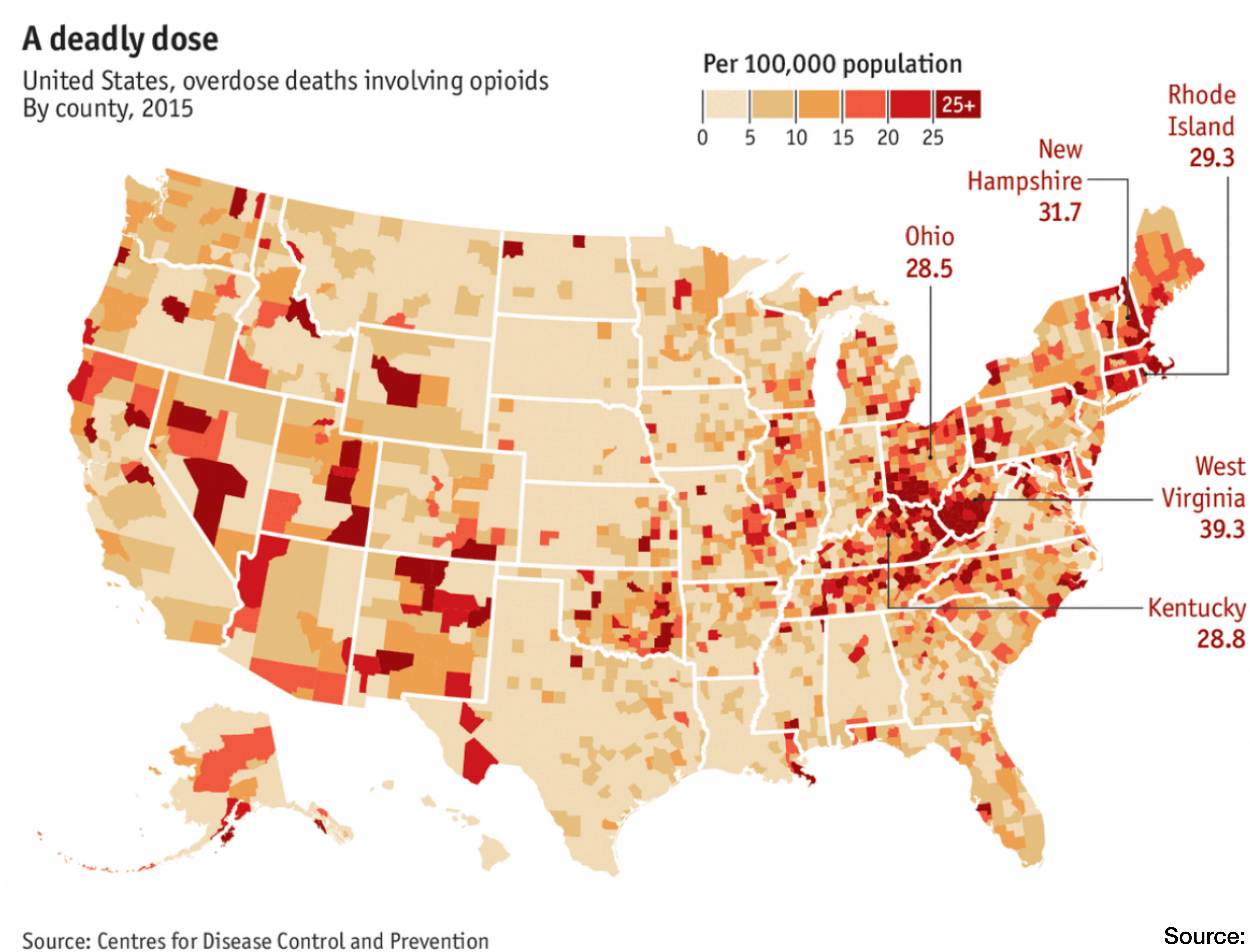
- Features of our problem:
  - Heterogeneous utility functions that depend on the requesting POD
    - Effectiveness of the public health intervention can vary across different groups of the affected population
    - Trade-off for the dispensing coordinator:
      - *Should we satisfy a lower-priority demand now, or save the inventory for a possible higher-priority demand in the future?*
  - Two timescales
    - Slower one for inventory replenishment (central inventory manager)
    - Faster one for dispensing decisions (dispensing coordinator)
  - Stochastic demands
  - Discrete inventory states

Example 1: Opioid overdose epidemic

- The U.S. Department of Health and Human Services (HHS) declared it a **public health emergency** in 2017.
- HHS: "*Increased prescription of opioid medications led to widespread misuse of both prescription and non-prescription opioids before it became clear that these medications could indeed be highly addictive.*"
    - Previously, pharmaceutical companies said that these drugs were not addictive.



Example 1: Opioid overdose epidemic



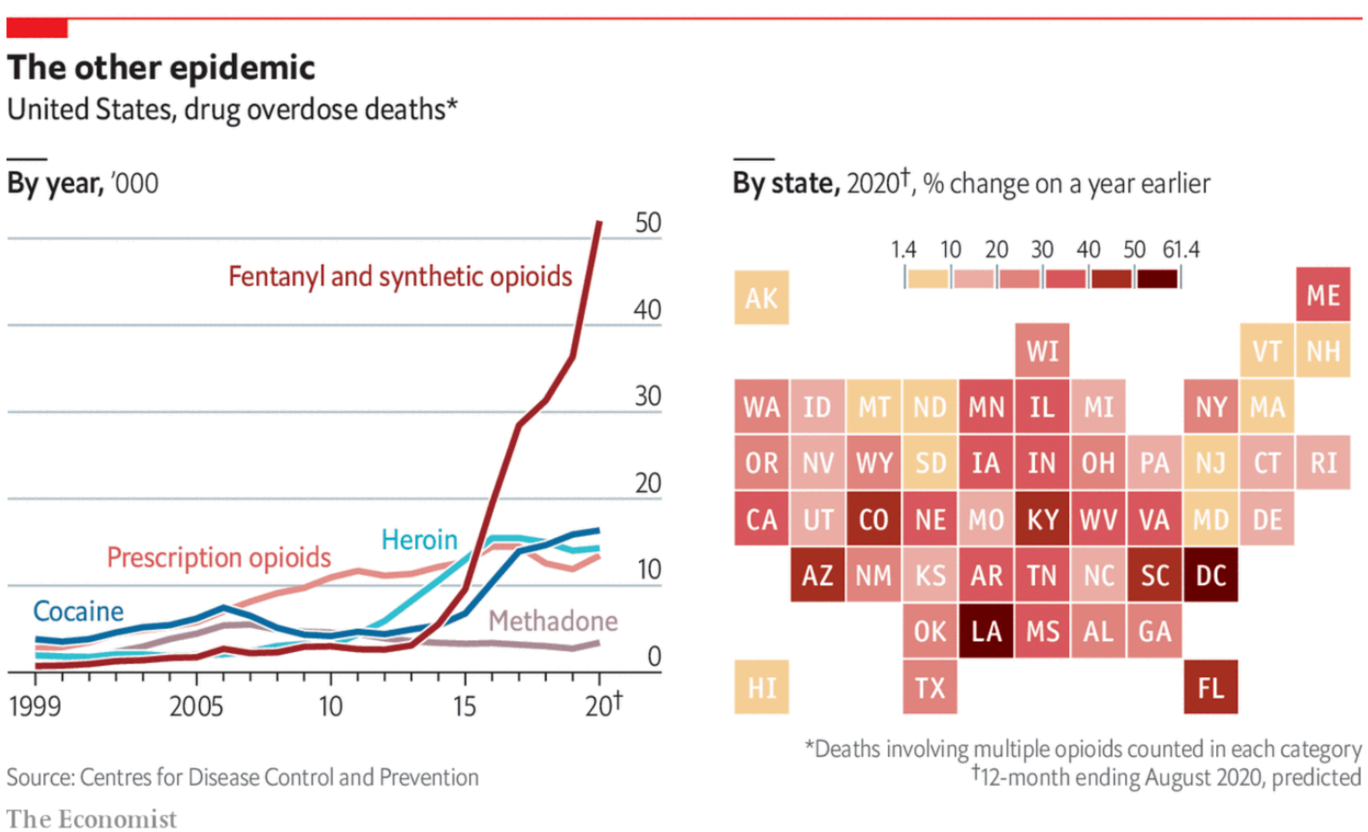
Source: Economist, 2017

Example 1: Opioid overdose epidemic

Daily chart

Opioid deaths in America reached new highs in the pandemic

Once a problem confined to the eastern part of the country, fentanyl has spread west



- Spreading to the western part of the country
- Job losses and social isolation may have worsened the situation
- Using drugs alone is more dangerous (no one to help)
- In King County (where Seattle is):
  - 2015 overdose deaths: 3
  - 2020 overdose deaths: 176

Source: Economist, 2021

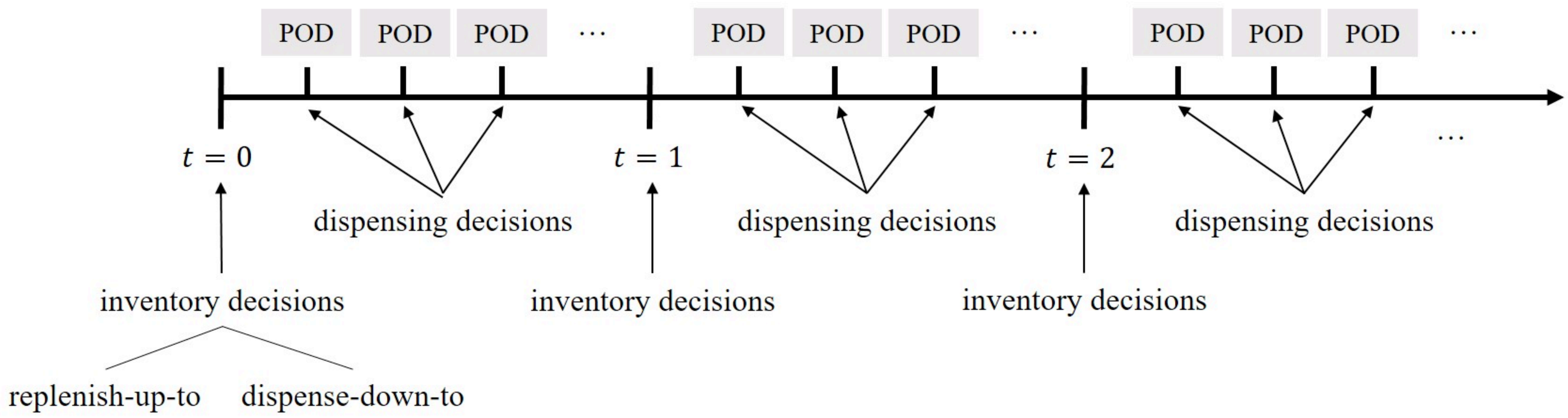
Example 1: Opioid overdose epidemic

- Naloxone is a drug that has the ability to reverse overdoses within minutes
  - To save lives, it is critical that this drug is widely distributed
- “Harm reduction” programs are distributing naloxone free of charge to first responders (incl. EMS, law enforcement, fire fighters, public transit drivers)
- Utility of naloxone varies across regions due to the varying levels of opioid usage in different populations
  - e.g., West Virginia DHHR distributes extra naloxone to high priority counties
- Utility of naloxone also varies across different types of first responders
  - e.g., law enforcement officers are “often a community’s first contact with opioid overdose victims after 9-1-1 services have been summoned” (Goodloe and Dailey (2014); Rando et al. (2015))

Example 2: Vaccine distribution, COVID-19 & H1N1

- Heterogenous utilities are very clear:
  - COVID-19: Compared with 5-17 age group, the rate of death is 1100 times higher in 65-74 age group, 2800 times higher in 75-84 age group, and 7900 times higher in 85 and older age group (CDC, 2021).
  - H1N1: The reported H1N1 cases from April 15 to July 24, 2009, show that the infected rate (number of cases per 100,000 population) of 0 to 4 age group is 17.6 times of the infected rate of 65 and older age group, and the rate of 5 to 24 age group is 20.5 times of the rate of 65 and older age group (CDC, 2009).

Sequence of events



- In each period, there are n sub-periods for which dispensing takes place
- Timing of events:
  - The central inventory manager decides **how much to replenish** and **how much to dispense** throughout the n sub-periods
  - The dispensing coordinator receives the inventory allotment and the sequentially receives POD requests and allocates inventory to maximize utility

Lower-level problem: Dispensing MDP

- The dispensing coordinator optimizes utility over n sub-periods (they want spend their allotment of inventory for this period optimally)
- In sub-period i of period t, the arriving POD is represented by an *attribute-demand* pair $(\xi_{t,i}, D_{t,i})$, with $D_{t,i} \in \{0,1,\ldots,D_{\text{max}}\}$.
  - When there is no arriving POD, demand is zero.
- The utility function of satisfying x_i units of demand is $u(x_i, \xi_{t,i})$
- Lower-level objective:

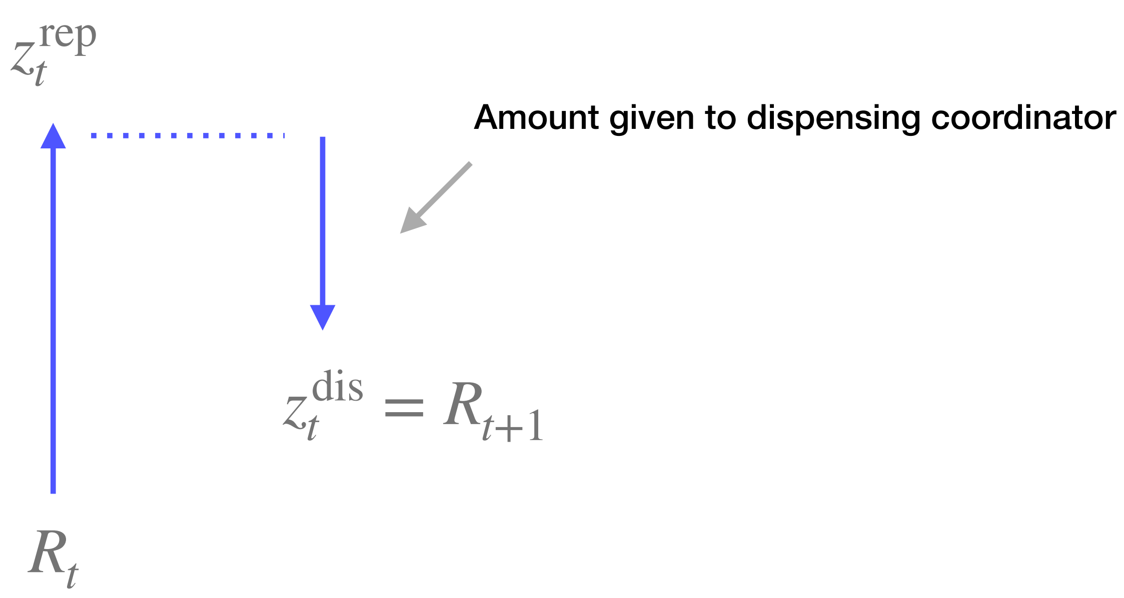
Lower-level dispensing policy

$$U_0(x, \xi \mid w) = \max_{\mu \in \mathscr{M}} \mathbf{E}\left[\sum_{i=0}^{n-1} u\Big(\min(\mu_i(x_i, \xi_i), D_i), \xi_i\Big) \,\Big|\, x_0 = x, \xi_0 = \xi, W_t = w\right].$$

An information state that transitions at the upper-level timescale

Upper-level problem: Inventory control MDP

- T planning periods, with two decision to be made in each period:
  - Replenish-up-to level z_t^{rep}
  - Dispense-down-to level z_t^{dis}
- The inventory state is R_t and information state is W_t
  - The information state may contain information such as past demands, current disease trends, or other dynamic information
- Holding cost h, ordering cost c_{W_t}



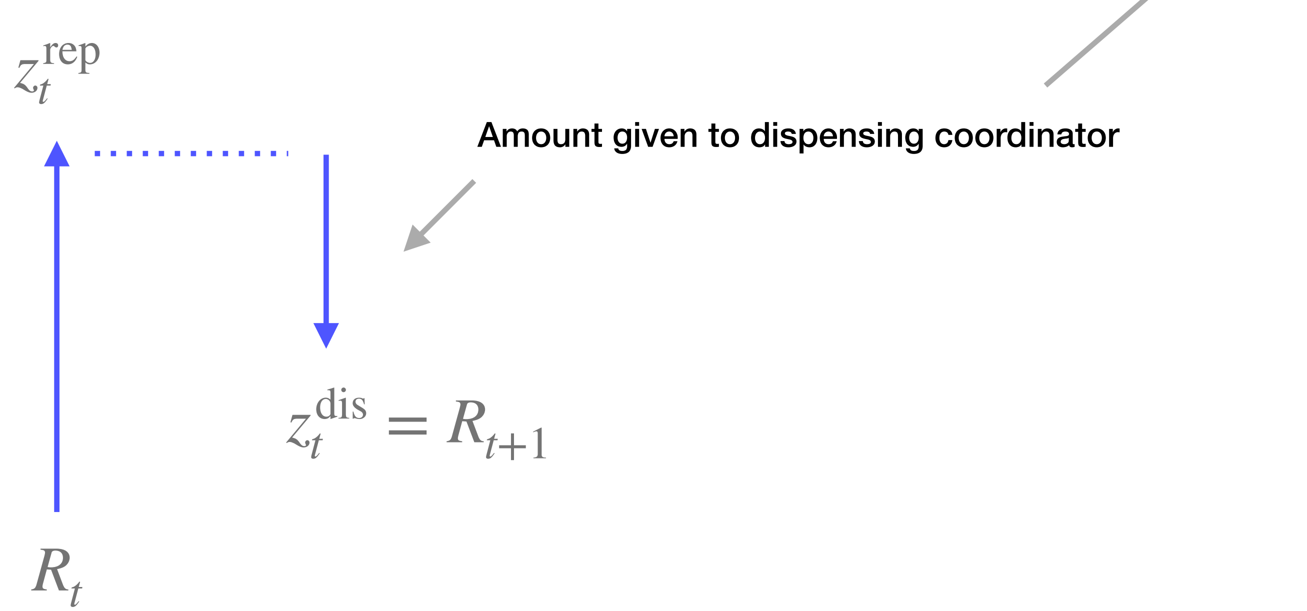
Upper-level problem: Inventory control MDP

- Objective is to maximize dispensing utility minus costs

$$\max_{\pi \in \Pi} \ \mathbf{E}\left[\sum_{t=0}^{T-1} \left(-hR_t - c_{W_t}\left(\pi_t^{\text{rep}}(R_t, W_t) - R_t\right) + U_0\left(\pi_t^{\text{rep}}(R_t, W_t) - \pi_t^{\text{dis}}(R_t, W_t), \xi_{t,0} \mid W_t\right)\right]\,.$$

- Bellman equation

$$V_t(r, w) = \max_{z^{\text{rep}}, z^{\text{dis}}} (c_w - h)\, r - c_w z^{\text{rep}} + \mathbf{E}_w\left[U_0\left(z^{\text{rep}} - z^{\text{dis}}, \xi_{t,0} \,|\, w\right) + V_{t+1}\left(z^{\text{dis}}, W_{t+1}\right)\right]\,.$$



Upper-level problem: Inventory control MDP

- Note that we can compute the Bellman step in two steps, one for replenishment and one for dispensing:

$$V_t(r,w) = \max_{z^{\text{rep}},z^{\text{dis}}} (c_w - h)\, r - c_w z^{\text{rep}} + \underbrace{\mathbf{E}_w\left[U_0\left(z^{\text{rep}} - z^{\text{dis}}, \xi_{t,0} \,|\, w\right) + V_{t+1}\left(z^{\text{dis}}, W_{t+1}\right)\right]}_{} .$$

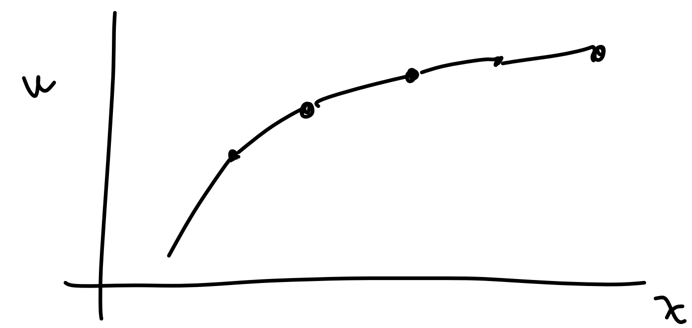
Can consider this the “dispensing” value function after replenishment is decided

- With a *post-decision* reformulation, we get the following:

$$\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = -\, c_w z^{\text{rep}} + \mathbf{E}_w\left[U_0\left(z^{\text{rep}} - \pi_t^{\text{dis},*}(z^{\text{rep}}, w), \xi_{t,0} \,|\, w\right)\right] + \tilde{V}_t^{\text{dis}}\left(\pi_t^{\text{dis},*}(z^{\text{rep}}, w), w\right)\Big],$$

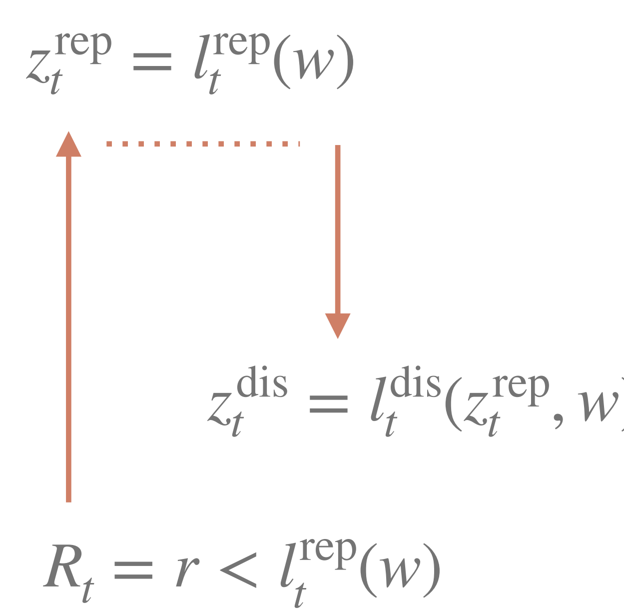
$$\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) = \mathbf{E}_w\left[(c_{W_{t+1}} - h) z^{\text{dis}} + \tilde{V}_{t+1}^{\text{rep}}\left(\pi_{t+1}^{\text{rep},*}(z^{\text{dis}}, W_{t+1}), W_{t+1}\right)\right]$$

- Policies (in red) and values (in blue) can be written in interleaving fashion

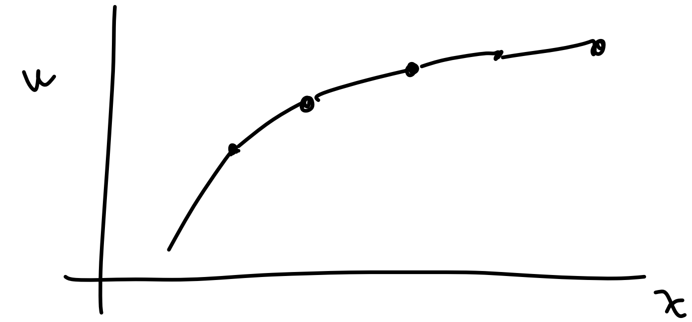


Structural properties of the MDP

- **Assumption**: For any ξ, the utility function $u\left(x, \xi\right)$ is discretely concave in x.
- **Proposition:**
  1. The lower-level MDP value function $U_i\left(x, \xi \mid w\right)$ is discretely concave in the inventory state x for all ξ, w, and i.
  2. The upper-level MDP value functions $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ are discretely concave in z^{rep} and z^{dis}, resp.
  3. Optimal policies are both state-dependent, discrete basestock policies:
     - $\pi_t^{\text{rep},*}(r, w) = \max\{r, l_t^{\text{rep}}(w)\}$,
     - $\pi_t^{\text{dis},*}(z^{\text{rep}}, w) = \min\{z^{\text{rep}}, l_t^{\text{dis}}(z^{\text{rep}}, w)\}$,
     - where $l_t^{\text{rep}}(w), l_t^{\text{dis}}(z^{\text{rep}}, w) \in \{0,1,\ldots,R_{\max}\}$.



Structural properties of the MDP



- **Assumption**: For any ξ, the utility function $u\left(x, \xi\right)$ is discretely concave in x.

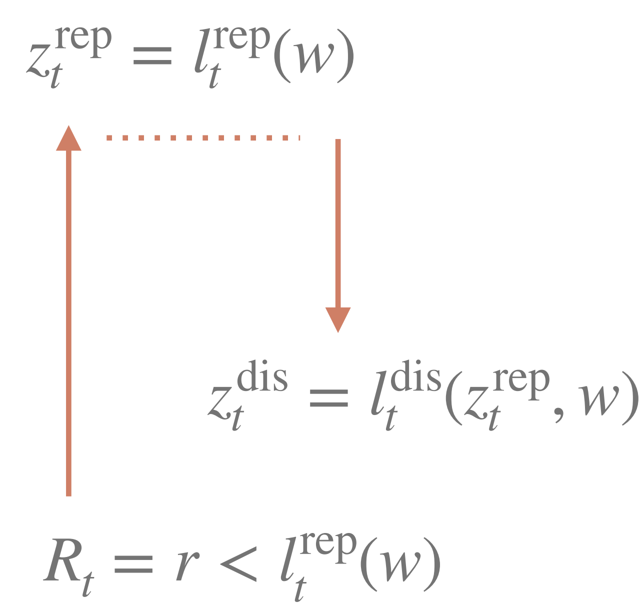
- **Proposition:**

  1. The lower-level MDP value function $U_i\left(x, \xi \mid w\right)$ is discretely concave in the inventory state x for all ξ, w, and i.

  2. The upper-level MDP value functions $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ are discretely concave in z^{rep} and z^{dis}, resp.

  3. Optimal policies are both state-dependent, discrete basestock policies:

     - $\pi_t^{\text{rep},*}(r, w) = \max\{r, l_t^{\text{rep}}(w)\}$,
     - $\pi_t^{\text{dis},*}(z^{\text{rep}}, w) = \min\{z^{\text{rep}}, l_t^{\text{dis}}(z^{\text{rep}}, w)\}$,
     - where $l_t^{\text{rep}}(w), l_t^{\text{dis}}(z^{\text{rep}}, w) \in \{0,1,\ldots,R_{\max}\}$.



Structural properties of the MDP

Main algorithmic research question:

In a data-driven setting, is it possible to take advantage of both the **structure in the policy** and **structure in the value function**?

Approximate dynamic programming (ADP)
Reinforcement learning (RL)

- ADP/RL algorithms can be classified into the following classes:

  1. **Value-based methods,** such as Q-learning (Watkins et al., 1989), use a combination of stochastic approximation and the Bellman equation to iteratively learn an *approximate value function* (or state-action values Q):

     - $Q_t^n(s, a) = (1 - \alpha_t^n)\, Q_t^{n-1}(s, a) + \alpha_t^n \cdot \text{observation}$

  2. **Policy-based methods**, such as policy gradient (Sutton et al., 1999), directly parameterize a class of *approximate policy functions* π_θ and optimize it via stochastic gradient methods.

  3. **Actor-critic methods** (Konda & Tsitsiklis, 2000) approximate both the policy and value function. Typically use linear models for function approximation.

     - Our method is falls here, but we utilize two types of structure.
     - "Actor" is the policy approximation, "critic" is the value approximation

Structured actor-critic algorithm

- Recall:

$$\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = -c_w z^{\text{rep}} + \mathbf{E}_w\Big[U_0\big(z^{\text{rep}} - \pi_t^{\text{dis},*}(z^{\text{rep}}, w), \xi_{t,0} \,|\, w\big)\Big] + \tilde{V}_t^{\text{dis}}\big(\pi_t^{\text{dis},*}(z^{\text{rep}}, w), w\big),$$

$$\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) = \mathbf{E}_w\Big[(c_{W_{t+1}} - h)z^{\text{dis}} + \tilde{V}_{t+1}^{\text{rep}}\big(\pi_{t+1}^{\text{rep},*}(z^{\text{dis}}, W_{t+1}), W_{t+1}\big)\Big]$$

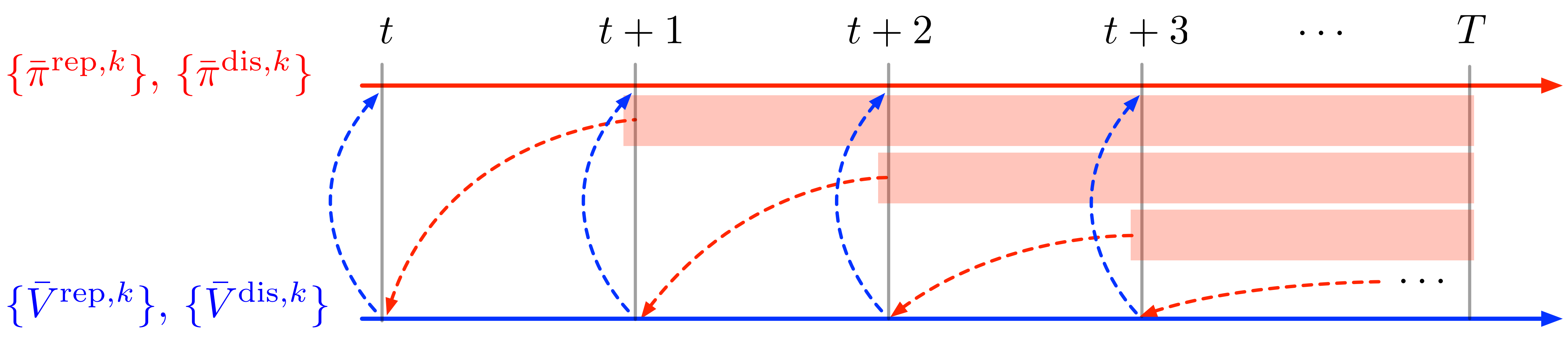
- Also, note that:

$$\pi_t^{\text{rep},*}(r, w) \in \operatorname{argmax}_{z^{\text{rep}}} \tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w),$$

$$\pi_t^{\text{dis},*}(z^{\text{rep}}, w) \in \operatorname{argmax}_{z^{\text{dis}}, z^{\text{rep}})} U_0\big(z^{\text{rep}} - z^{\text{dis}}, \xi_{t,0} \,|\, w\big) + \tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$$

- If the optimal policy and next stage value is known, we can write the current value
- If the optimal value is known, then we can write the current policy
- Let's apply these relationships in an alternating fashion

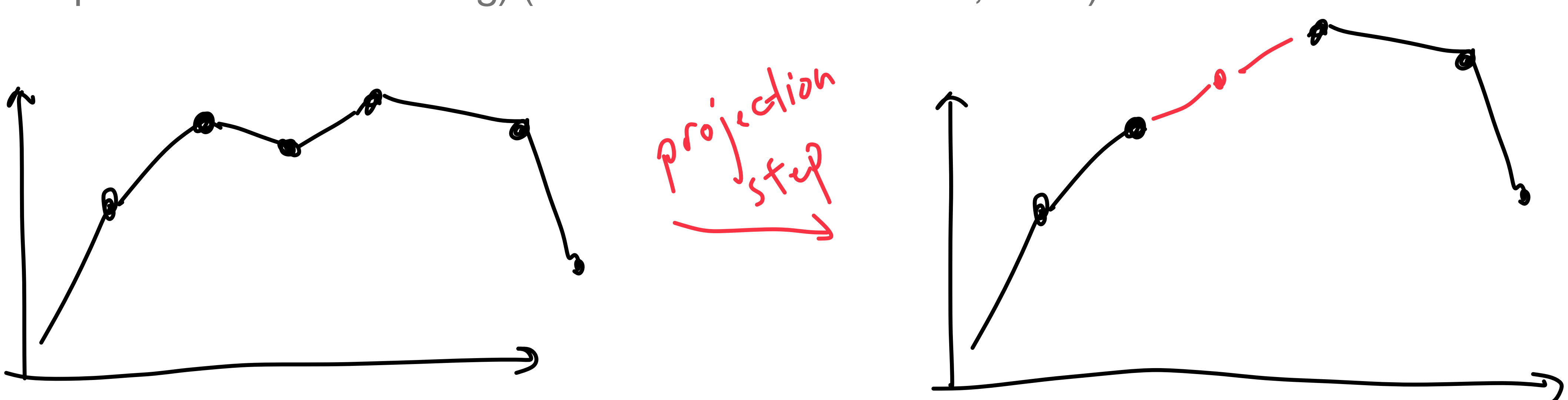
Structured actor-critic (S-AC) algorithm



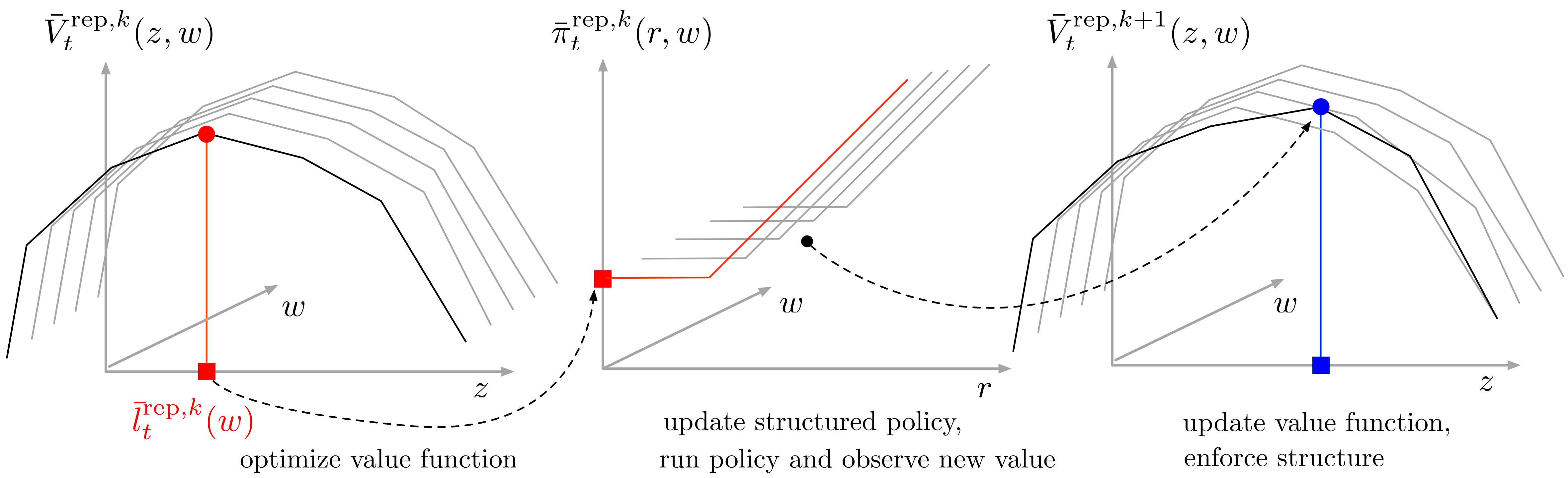
- **On policy update steps**:
    - Use the value function approximation to update the policy
- **On value function update steps**:
    - Simulate the policy approximation forward (red) to update the value function

How do we represent the structure?

- **For the policy**:
  - Only store the base-stock threshold $l_t^{\text{rep}}(w)$, $l_t^{\text{dis}}(z^{\text{rep}}, w)$ and then make use of the base-stock form when using the policy
  - In the case of $l_t^{\text{rep}}(w)$, reduces the need to store individual policies for each inventory state
- **On value function update steps**:
  - Store the value function as a sequence of **slopes** between points
  - After each observation, project the value function to **maintain concavity** (i.e., make sure the slopes are non-increasing) (Nascimento and Powell, 2009)



How do we represent the structure?



Structured actor-critic algorithm

- Input: random initial policies and piecewise concave value function
- At each iteration k, loop through all time periods t
- Simulate current policy forward to get new slope observations
- Update value function using the slope observations (and do concave proj.)
- The updated value function implies new basestock thresholds
- Update the policies
- Repeat

Algorithm 1: Structured Actor-Critic Method

Input: Lower level optimal policy μ^* (learned from backward dynamic programming). Initial policy estimates $\bar{l}^{\text{rep},0}$ and $\bar{\pi}^{\text{dis},0}$, and value estimates $\bar{v}^{\text{rep},0}$ and $\bar{v}^{\text{dis},0}$ (nonincreasing in z^{rep} and z^{dis} respectively). Stepsize rules $\tilde{\alpha}_t^k$ and $\tilde{\beta}_t^k$ for all t, k.

Output: Approximations $\bar{l}^{\text{rep},k}$, $\bar{\pi}^{\text{dis},k}$, $\bar{v}^{\text{rep},k}$, and $\bar{v}^{\text{dis},k}$.

for $k = 1, 2, \ldots$ **do**

  Sample initial states $z_0^{\text{rep},k}$ and $z_0^{\text{dis},k}$.

  for $t = 0, 1, \ldots, T-1$ **do**

    Observe w_t^k and $\xi_{t,1}^k$, then observe $\hat{v}_t^{\text{rep},k}$ and $\hat{v}_t^{\text{dis},k}$ according to (17) and (18) respectively.

    Perform SA step:

$$\tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) = \left(1 - \alpha_t^k(z^{\text{rep}}, w)\right) \bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) + \alpha_t^k(z^{\text{rep}}, w)\, \hat{v}_t^{\text{rep},k},$$
$$\tilde{v}_t^{\text{dis},k}(z^{\text{dis}}, w) = \left(1 - \alpha_t^k(z^{\text{dis}}, w)\right) \bar{v}_t^{\text{dis},k-1}(z^{\text{dis}}, w) + \alpha_t^k(z^{\text{dis}}, w)\, \hat{v}_t^{\text{dis},k}.$$

    Perform the concavity projection operation (19):

$$\bar{v}_t^{\text{rep},k} = \Pi_{z_t^{\text{rep},k}, w_t^k}(\tilde{v}_t^{\text{rep},k}), \quad \bar{v}_t^{\text{dis},k} = \Pi_{z_t^{\text{dis},k}, w_t^k}(\tilde{v}_t^{\text{dis},k}).$$

    Observe and update the replenish-up-to threshold:

$$\hat{l}_t^{\text{rep},k} = \arg\max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(0)} \textstyle\sum_{j=0}^{z^{\text{rep}}} \bar{v}_t^{\text{rep},k}(j, w_t^k),$$
$$\bar{l}_t^{\text{rep},k}(w) = \left(1 - \beta_t^k(w)\right) \bar{l}_t^{\text{rep},k-1}(w) + \beta_t^k(w)\, \hat{l}_t^{\text{rep},k}.$$

    Observe and update the dispense-down-to policy:

    for $z_t^{\text{rep}} = 0, 1, \ldots, R_{\max}$ **do**

$$\hat{\pi}_t^{\text{dis}} = \arg\max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z_t^{\text{rep}})} U_0^{\mu^*}\left(z_t^{\text{rep}} - z^{\text{dis}}, \xi_{t,0}^k \,|w_t^k\right) + \textstyle\sum_{j=0}^{z^{\text{dis}}} \bar{v}_t^{\text{dis},k}(j, w_t^k),$$
$$\bar{\pi}_t^{\text{dis},k}(z^{\text{rep}}, w) = \left(1 - \alpha^k(z^{\text{rep}}, w)\right) \bar{\pi}_t^{\text{dis},k-1}(z^{\text{rep}}, w) + \alpha^k(z^{\text{rep}}, w)\, \hat{\pi}_t^{\text{dis}}.$$

    end

    If $t < T-1$, take $z_{t+1}^{\text{rep},k}$ and $z_{t+1}^{\text{dis},k}$ according to the ϵ-greedy exploration policy.

  end

end

Almost sure convergence of S-AC

Theorem. Both the value function and policy approximations converge to their optimal counterparts almost surely. We have

$$\bar{v}_t^{\mathrm{rep},k}(z^{\mathrm{rep}}, w) \xrightarrow{k\to\infty} v_t^{\mathrm{rep},*}(z^{\mathrm{rep}}, w),$$

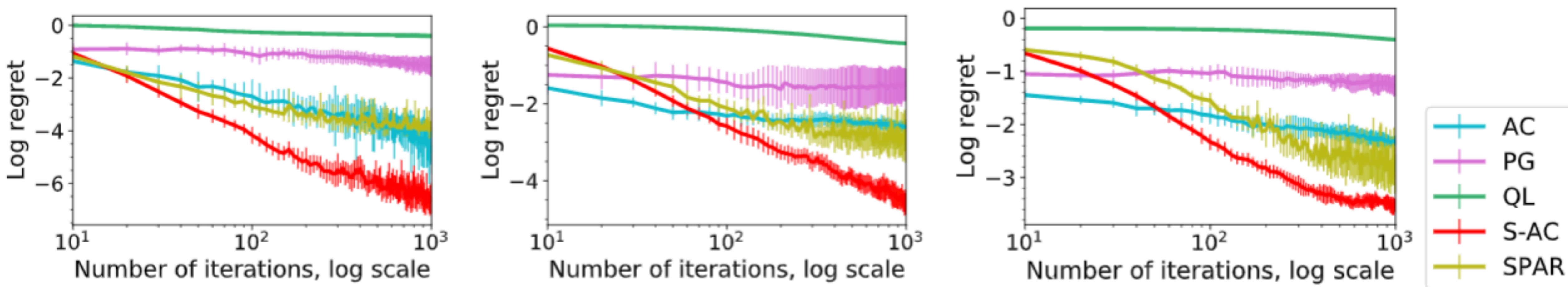
$$\bar{\pi}_t^{\mathrm{rep},k}(r, w) \xrightarrow{k\to\infty} \pi_t^{\mathrm{rep},*}(r, w),$$

almost surely. Same holds for the dispensing values and policies.

Baseline algorithms vs S-AC

- **Multi-stage version of SPAR** (Nascimento and Powell, 2009)
    - Uses concave value functions + a temporal difference to update slopes without a policy approximation
- **Actor-critic (AC)** with linear function approximations for both policy and value function
- **Monte-Carlo policy gradient (PG)** with the same policy function approximation as the AC algorithm
- **Q-learning (QL):** each state-action pair is updated independently
    - S-AC and SPAR lie in between the extremes of AC/PG and QL

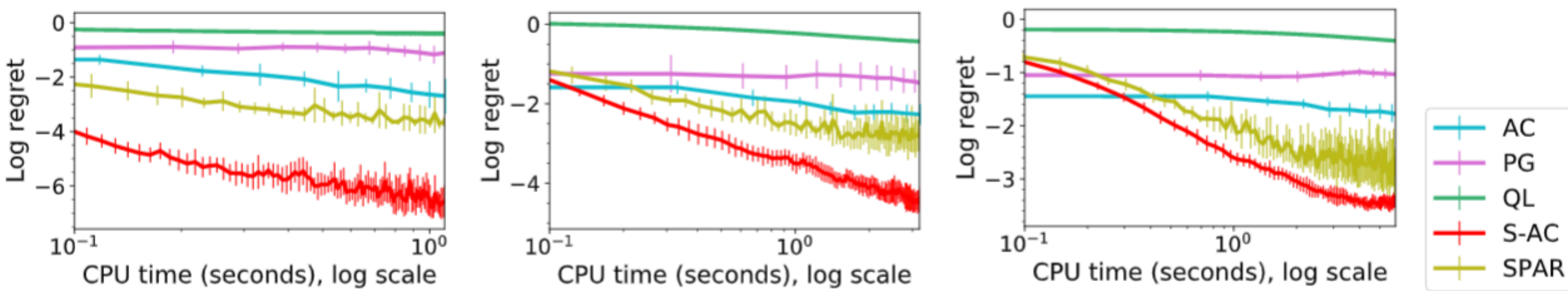
Synthetic experiments (iterations)



(a) $R_{\max} = 20, |\mathcal{W}| = 3$ (b) $R_{\max} = 40, |\mathcal{W}| = 9$ (c) $R_{\max} = 60, |\mathcal{W}| = 15$

- **Main takeaways:**
  - AC is the most competitive when accounting for the number of iterations
  - PG and AC do well initially, especially for the largest problem, likely due to the fact that they use stochastic policies (initially random) that encourage exploration early on
  - Vanilla Q-learning is largely ineffective

Synthetic experiments (CPU time)

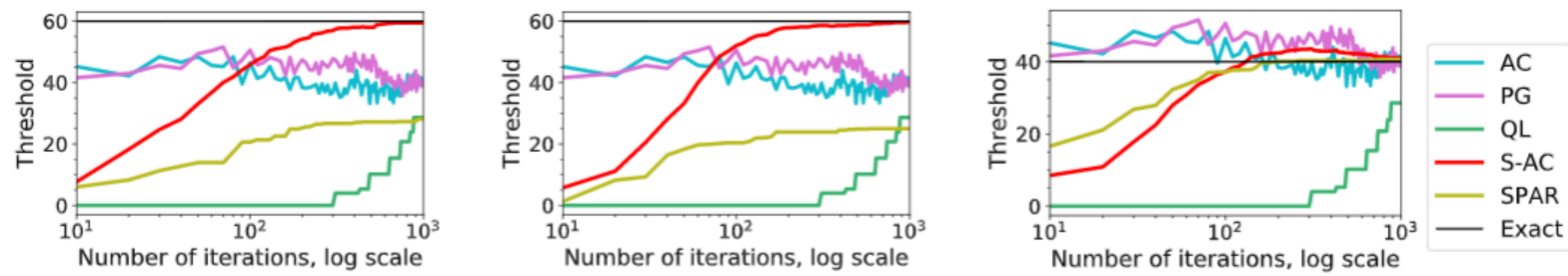


(a) $R_{\max} = 20, |\mathcal{W}| = 3$ (b) $R_{\max} = 40, |\mathcal{W}| = 9$ (c) $R_{\max} = 60, |\mathcal{W}| = 15$

- **Main takeaways:**
  - SPAR (value functions with concavity projection) is most competitive
  - PG and AC act on the original action space $(z^{\text{rep}}, z^{\text{dis}})$, so updates are slightly slower than SPAR and S-AC (which take advantage of structure)

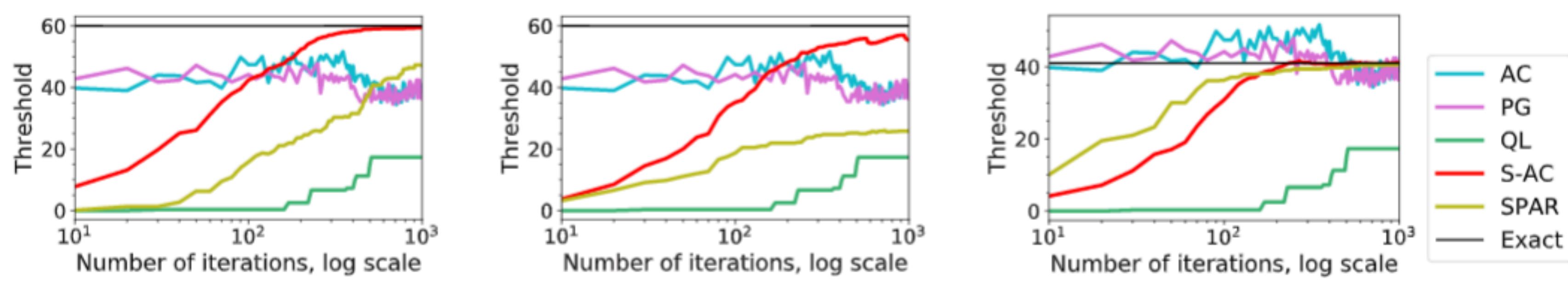
Synthetic experiments (convergence of thresholds)

Instance 1: 9 information states



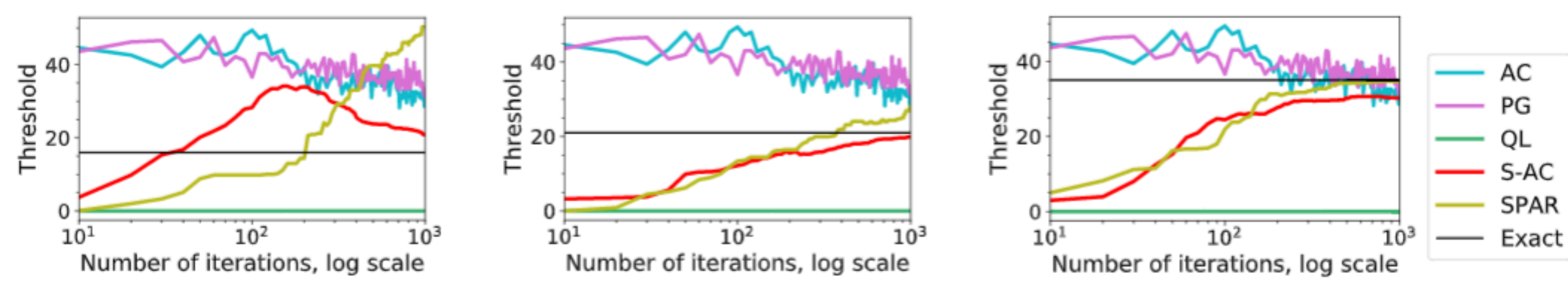
(a) $w_0 = 1$ (b) $w_0 = 4$ (c) $w_0 = 8$

Instance 2: 12 information states



(a) $w_0 = 2$ (b) $w_0 = 6$ (c) $w_0 = 10$

Instance 3: 15 information states



(a) $w_0 = 2$ (b) $w_0 = 7$ (c) $w_0 = 12$

- **Main takeaway:**
  - S-AC exhibits more stable convergence to the true threshold, compared to the “implied” thresholds of the other algorithms

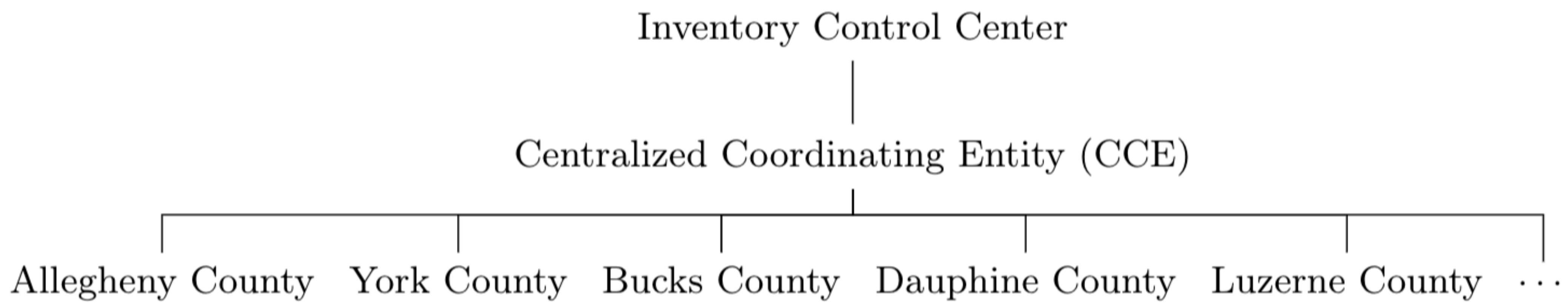
Synthetic experiments (sensitivity analysis)

Figure 8: Convergence of replenish-up-to thresholds at $t = 0$ for the $R_{\max} = 60, |\mathcal{W}| = 15$ instance.

Table 3: Impact of parameters on ADP algorithms for the $R_{\max} = 50, |\mathcal{W}| = 9$ instance.

Parameter	Value	AC	PG	QL	S-AC	SPAR	Exact
Mean total demand	30, Normal	19,037	16,009	7,287	**20,313**	19,077	21,332
	30, Uniform	18,113	15,142	8,476	**20,865**	20,098	21,332
	50, Normal	28,422	23,237	10,318	**29,080**	28,278	29,387
	50, Uniform	28,023	23,112	10,286	**29,077**	28,150	29,387
Mean ordering cost	30	30,914	25,488	15,125	**33,532**	32,671	34,647
	50	18,037	14,009	7,287	**20,313**	19,077	20,689
	70	11,257	8,660	6,032	**11,866**	11,553	11,984
Holding cost	5	18,037	14,009	7,287	**20,313**	19,077	20,689
	20	18,402	15,064	7,189	**19,839**	19,285	20,131
	35	17,807	14,498	5,855	**19,381**	18,784	19,592
	50	17,150	15,011	4,582	**18,988**	18,418	19,203
	65	16,575	13,708	2,954	**18,597**	17,931	18,835

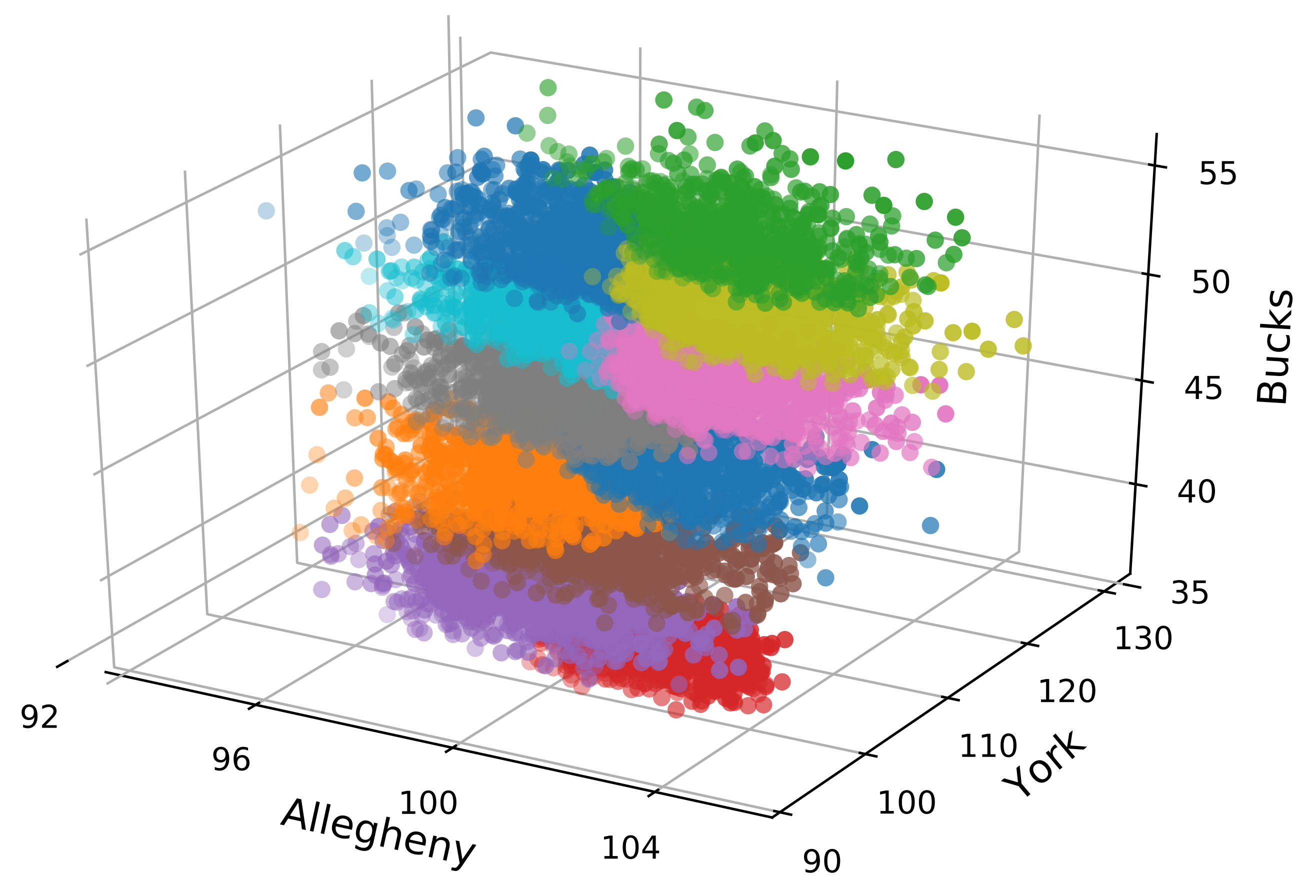
Case study: Naloxone for First Responders Program



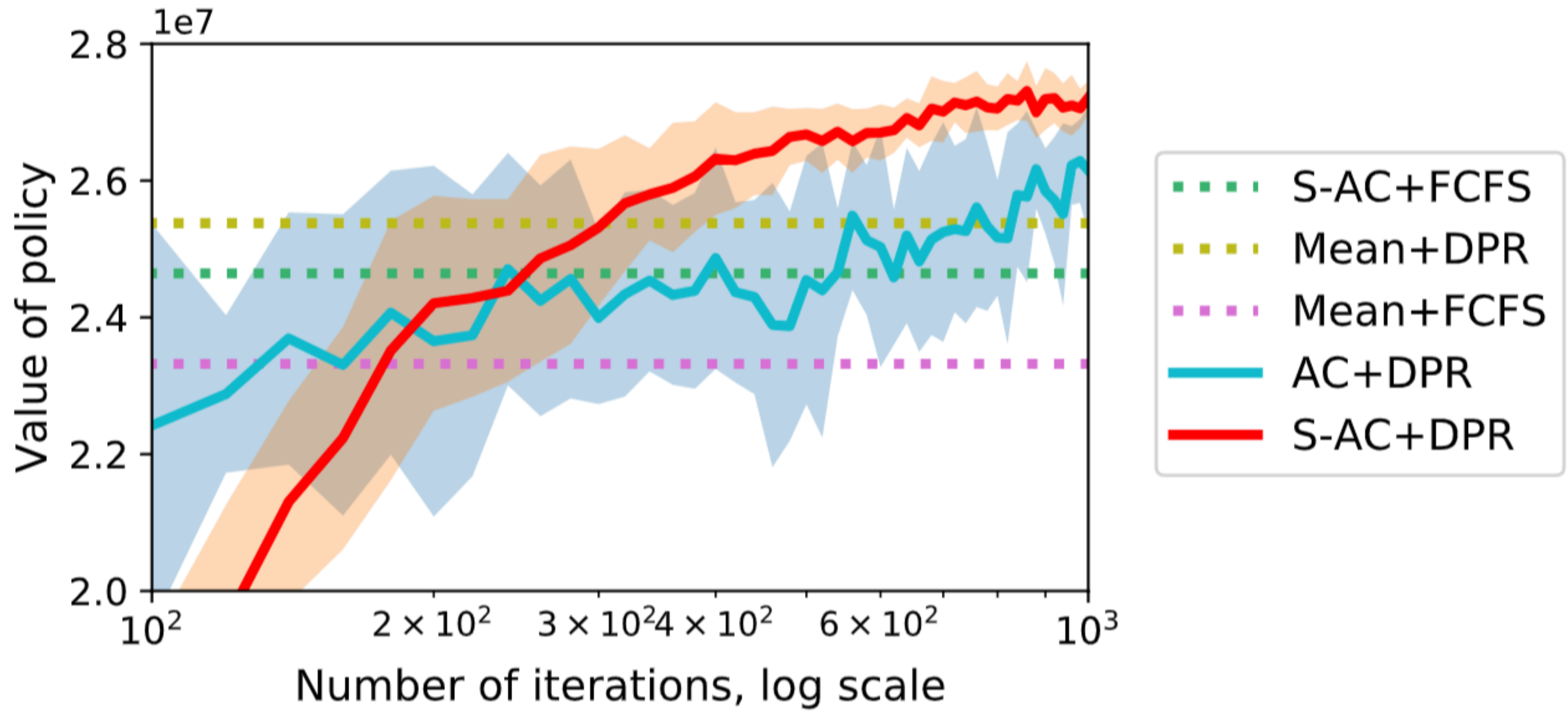
- **Setup:**
  - Motivation: the NFRP program's hierarchical structure relies on a centralized coordination entity (our "dispensing coordinator")
  - Data: monthly opioid overdose data from the five most-affected counties in Pennsylvania (these are the PODs)
  - Utility function of a county is based on proportion of incidents occurring there relative to the other counties

S-AC with clustered information states

- **Weakness of S-AC:** Exploits structure in the inventory dimension but not the information dimension
- The case study has a 5-dimensional information space, which becomes challenging to handle
- **Minor extension:** Perform k-means clustering in the information dimension and then run S-AC on an “aggregated MDP”
    - Similar to aggregation in Tsitsiklis and Van Roy, 1996, but *only* in the information state

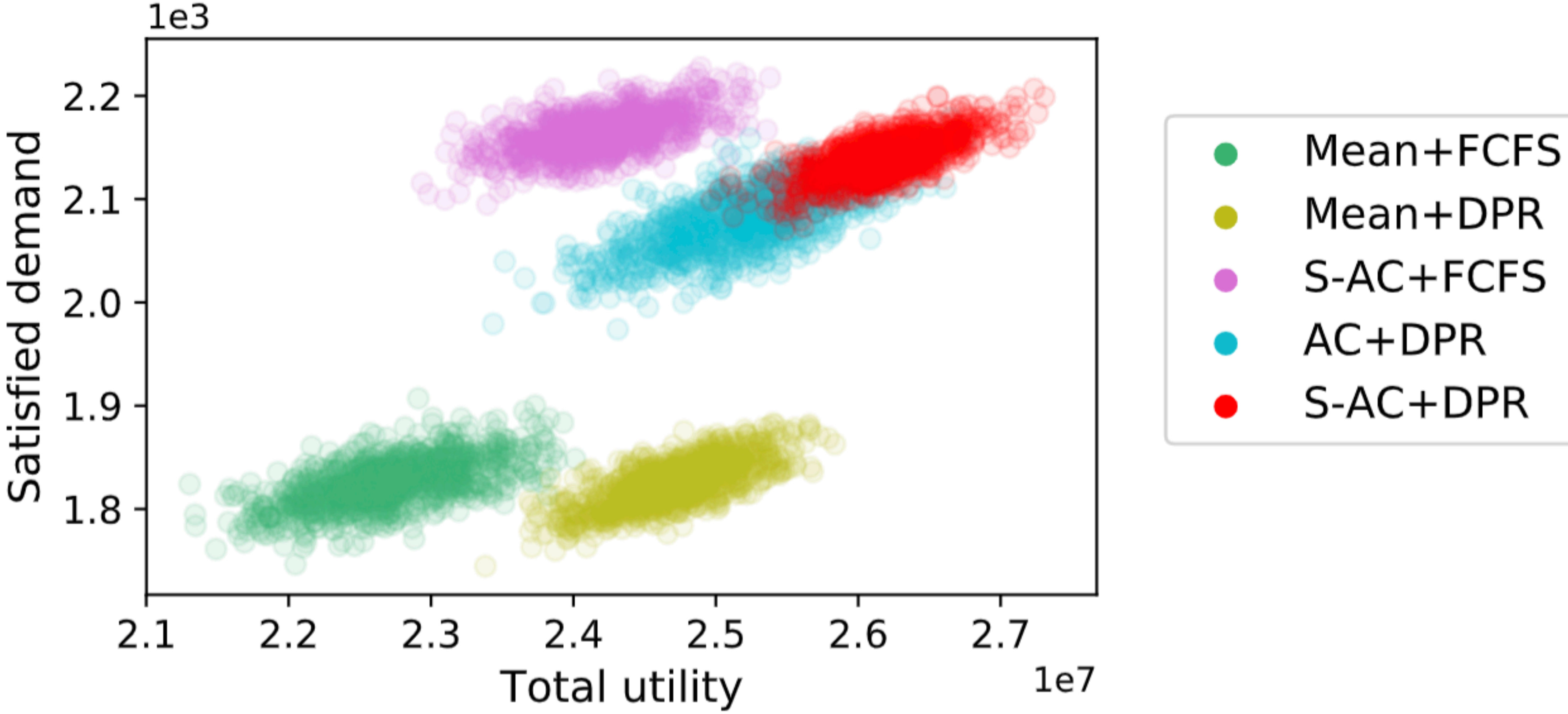


Case study: Naloxone for First Responders Program

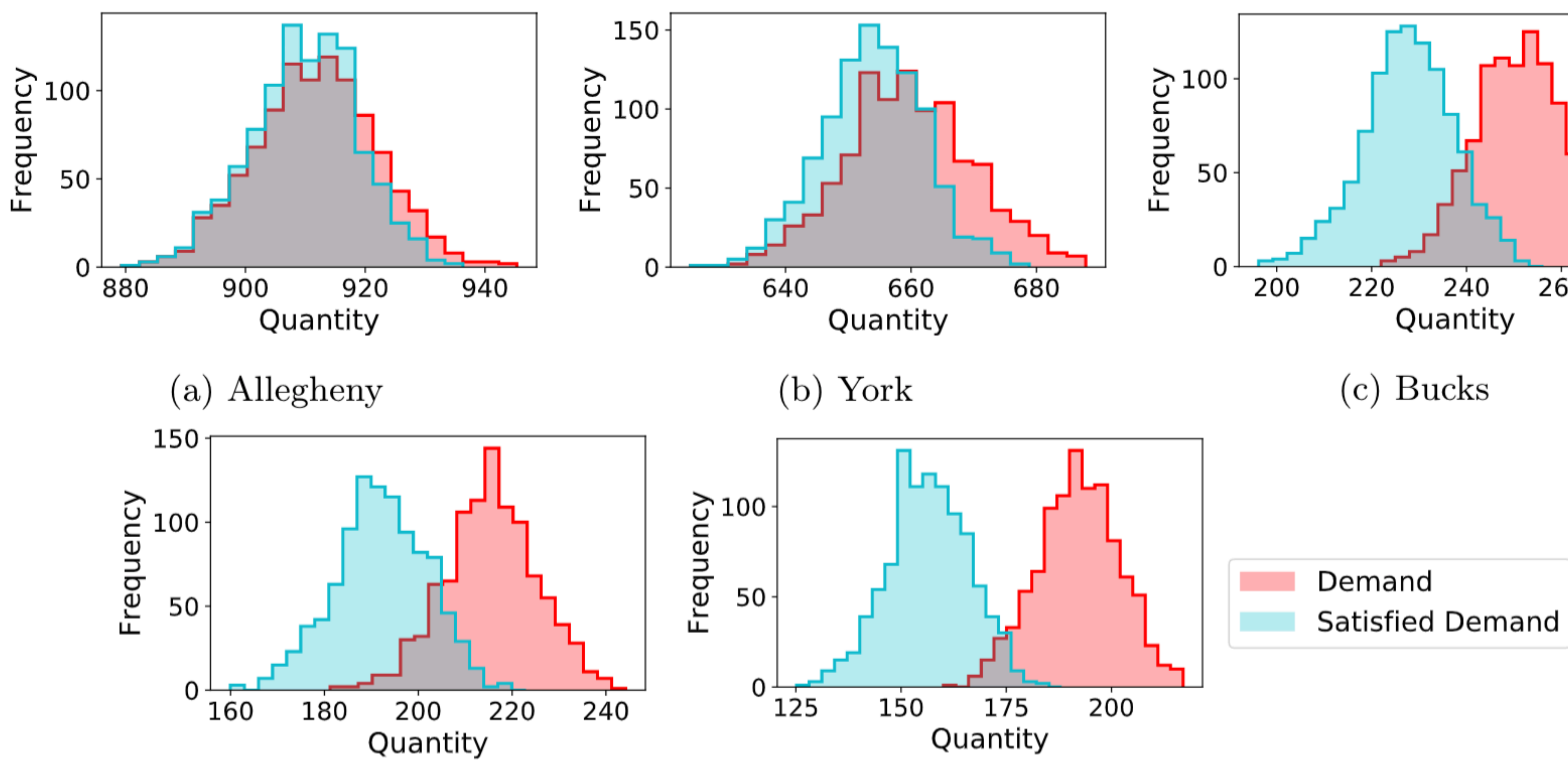


- **Heuristics**
  - **Upper-level:** Mean = replenish up to the mean demand
  - **Lower-level:** FCFS = dispense in a first-come-first-serve manner
  - **Lower-level:** DPR = solve using DP with discrete states, then regress on result

Case study: Naloxone for First Responders Program



Case study: Naloxone for First Responders Program



(a) Allegheny (b) York (c) Bucks

(d) Dauphin (e) Luzerne

Thank you! Questions?

Please feel free to email me at drjiang@pitt.edu for additional comments!

A revised version of this paper will be available soon.