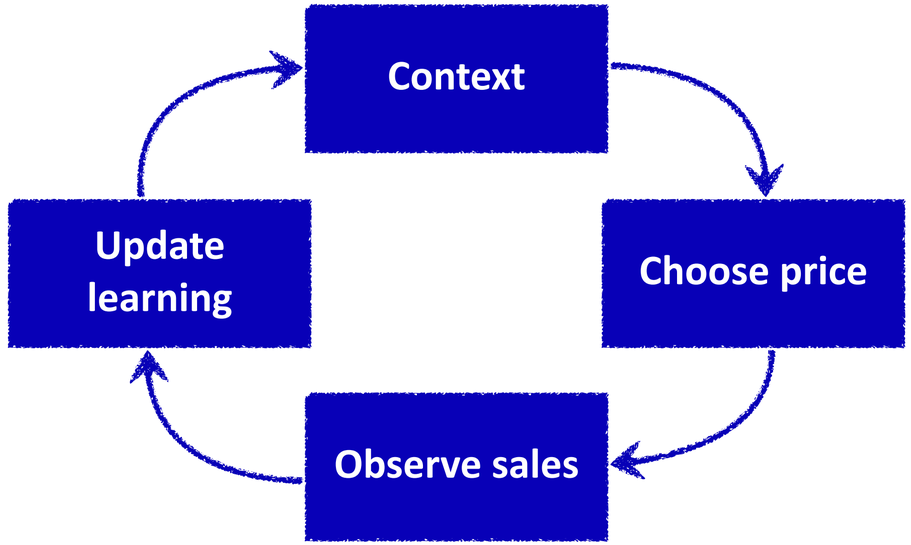
Learning to price ancillary seats with Bayesian Value Iteration

65th AGIFORS Annual Symposion, 2025

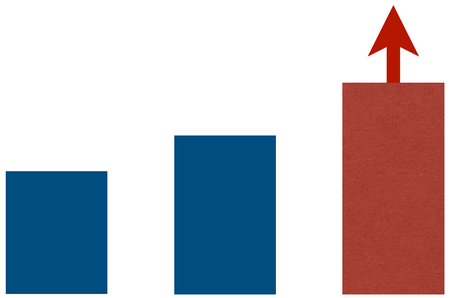
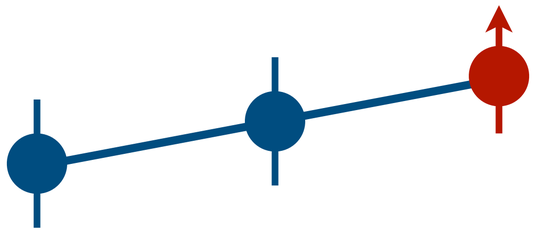
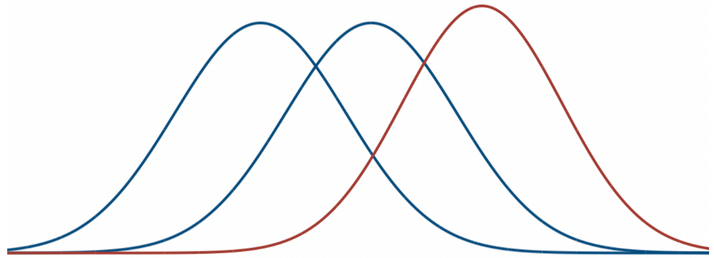
Kevin Duijndam (KLM), joint work with Ger Koole (VU), Rob van der Mei (CWI, VU)

Ancillary seat pricing as a sequential learning algorithm.

- Ancillary seat pricing could basically be seen as sequential decisions under uncertainty —> **Bandit algorithms.**
- Context (flight type, DBD, itinerary, …) matters in this learning environment.
- To learn, we need exploration, but we can’t afford too much revenue loss.
- Today’s main underlying methods for bandit problems are all heuristics to balance exploration-exploitation:
  - ε-greedy
  - UCB
  - Thompson Sampling



Three main underlying heuristic mechanisms to decide in bandit context.

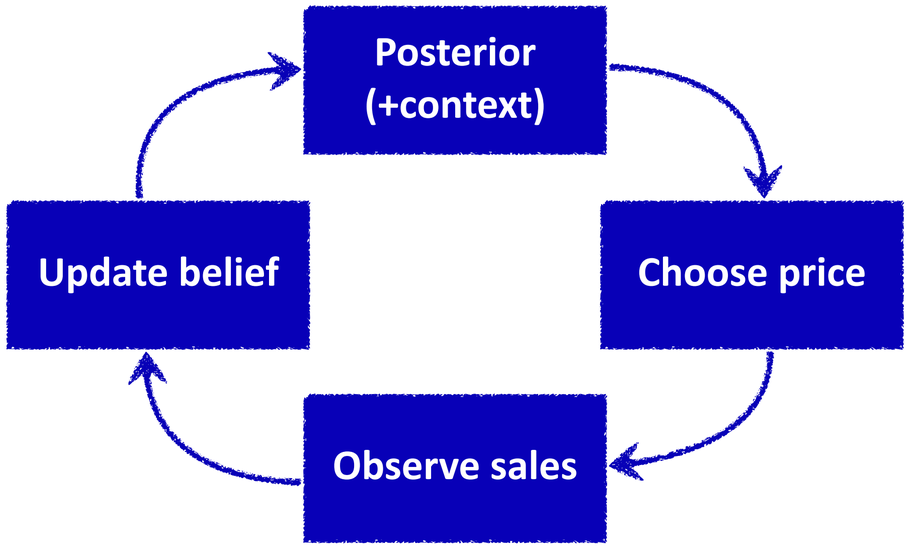
Epsilon-Greedy	Upper Confidence Bound (UCB)	Thompson Sampling
		
Choose the arm with the highest observed pay-off with probability 1-ε. With probability ε choose random arm to explore.	Track observed pay-off, and add optimistic exploration bonus. Pick the arm with highest optimistic estimate.	Maintain an estimated pay-off distribution per arm, take random sample from this posterior distribution for pay-off estimation in this round. Pick arm with highest pay-off.

But we could introduce a fourth:

from Contextual Bandit to Belief-MDP.

- Main point, treat knowledge of pay-off distribution as part of the state:
  - $\mathcal{X}$ an augmented state space, combining the state estimates for the arm pay-off distributions, and the context, so $\mathcal{X} = \mathcal{S} \times \mathcal{C}$ with a specific state represented as (s, c) with s the pay-off distribution estimates, and c the context.
- Then, we can analytically model the value of information gain in future states.
  - $P(s', c' \mid (s, c), a)$ is then the transition probability from state (s, c) to (s', c') when action a is taken, as determined by the Bayesian update.

With that, we can *plan* exploration instead of heuristically letting that happen, and we could use **value iteration to find the optimal policy.**



Value Iteration to solve a Contextual Bandit problem.

$$V(s,c) = \max_{a \in A} \left[R((s,c),a) + \gamma \mathbb{E}[V(s',c') \mid (s,c),a] \right]$$

Clear advantages:

- No need to balance exploration-exploitation heuristically with e.g. an artificial exploration bonus; in the expectation for the next state, the exact valuation of potential future states is calculated.

- Value function can be calculated offline, online only lookup necessary.

Contextual Value Iteration algorithm.

Algorithm 1 Contextual Value Iteration (finite-context variant)

Require: discount γ, horizon limit H, offline tolerance ϑ

Offline: run value iteration on a grid of (m, Σ, α) until $\|V_{n+1} - V_n\|_\infty < \vartheta$; store V^*
for $t = 1, \ldots, H$ **do**
    observe context c_t and posterior state s_t
    $a_t \leftarrow \arg\max_a \{R(x_t, a) + \gamma\, \mathbb{E}[V^*(x') \mid x_t, a]\}$
    play a_t, observe q_t and c_{t+1}
    update (m, Σ) via posterior calculation
    $\alpha_{c_t, c_{t+1}} \leftarrow \alpha_{c_t, c_{t+1}} + 1$
end for

Seat pricing model.

At each round t we observe a discrete context $c_t \in \{0,1\}$ with $c = 0$ denoting leisure and $c = 1$ denoting business demand. We choose a price $a_t \in \mathcal{P}$ from a small finite grid and realise seats sold

$$y_t \sim \text{Poisson}(\lambda_\theta(c_t, a_t)) \quad , \quad \lambda_\theta(c, a) = \exp\left(\theta^t \phi(c, a)\right)$$

with revenue $r_t = a_t y_t$. Context evolves exogenously via a Markov kernel $\mu(c' \mid c)$.

To capture that business is less price elastic, we use a 4-dimensional feature map

$$\phi(c, a) = \left[1, a, 1\{c = 1\}, 1\{c = 1\} \cdot a\right]^t$$

with corresponding parameter names $\theta = \left[\theta_0, \theta_p, \theta_{b0}, \theta_{bp}\right]$. This gives the log-intensity $\log \lambda_\theta(c, a) = \theta_0 + \theta_p \cdot a + \theta_{b0} 1(c = 1) + \theta_{bp} 1(c = 1) \cdot a$.

Seat pricing model.

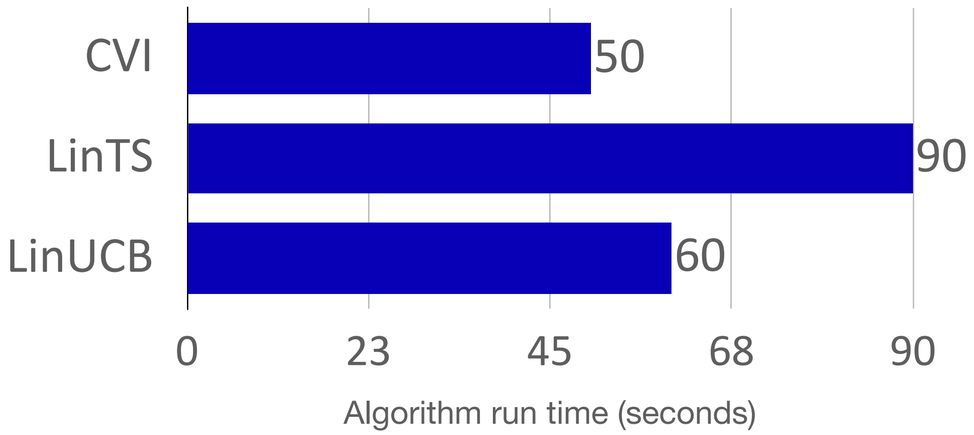
If we then maintain a Gaussian belief on $\theta : \theta \sim \mathcal{N}(m, \Sigma)$, then the Bellman update at (m, c) takes the form

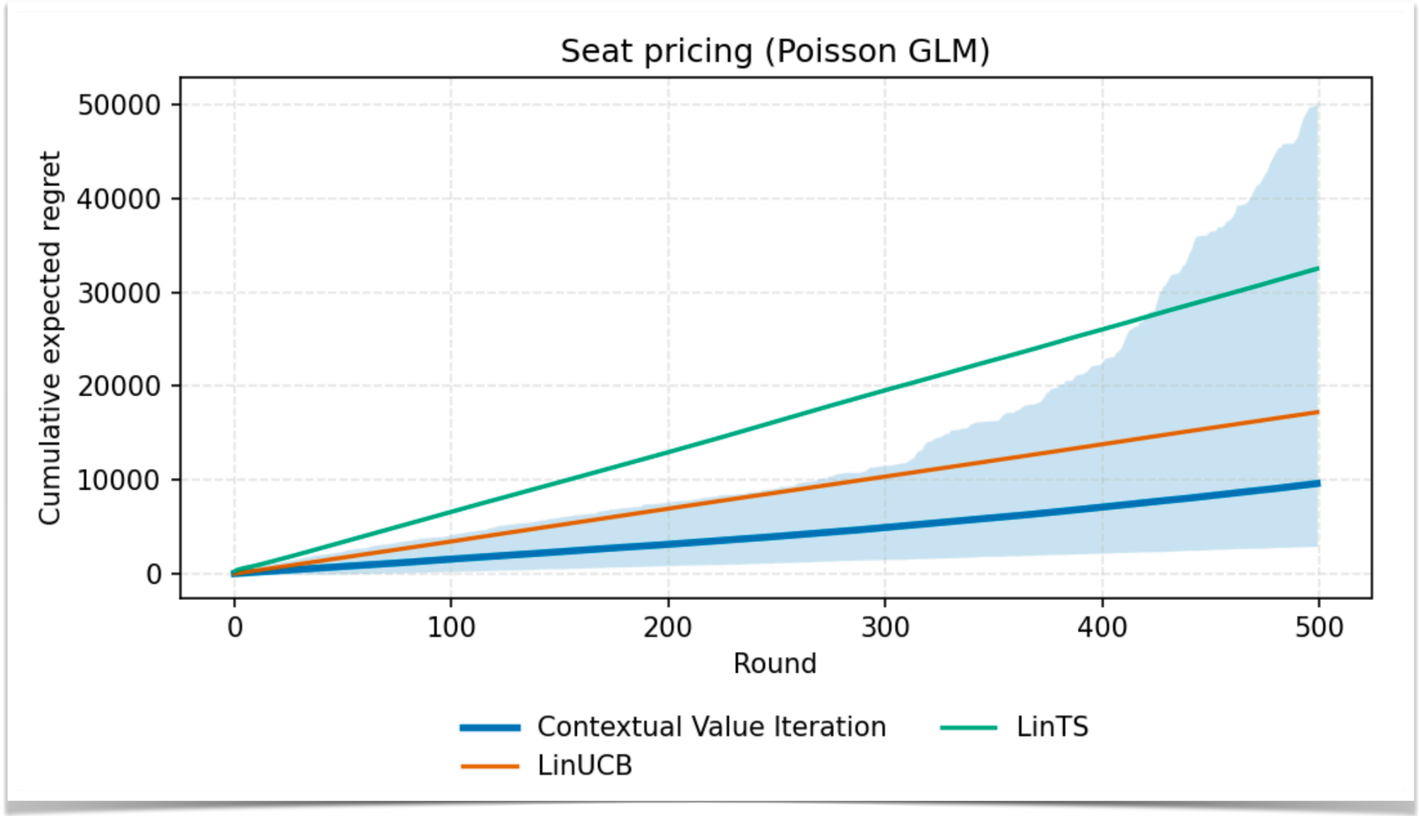
$$(TV)(m,c) = \max_{a \in \mathcal{P}} \left[a\lambda + \gamma \mathbb{E}_{y \sim \text{Poisson}(\lambda)} \left[\mathbb{E}_{c' \sim \mu(.,c)} V(m'(y), c') \right] \right]$$

Running value iteration on this finite state space yields a table V^* and a greedy policy $\pi^*(m, c)$ that we can use for online look-up.

Seat pricing model.

This can all be implemented and compared against relevant baselines.





But, we run into the curse of dimensionality when calculating the value function.

With this toy-sample, we see the approach works, but it's a very small sample. The value function grows in number of arms, parameters for pay-off distribution, and context dimensions.

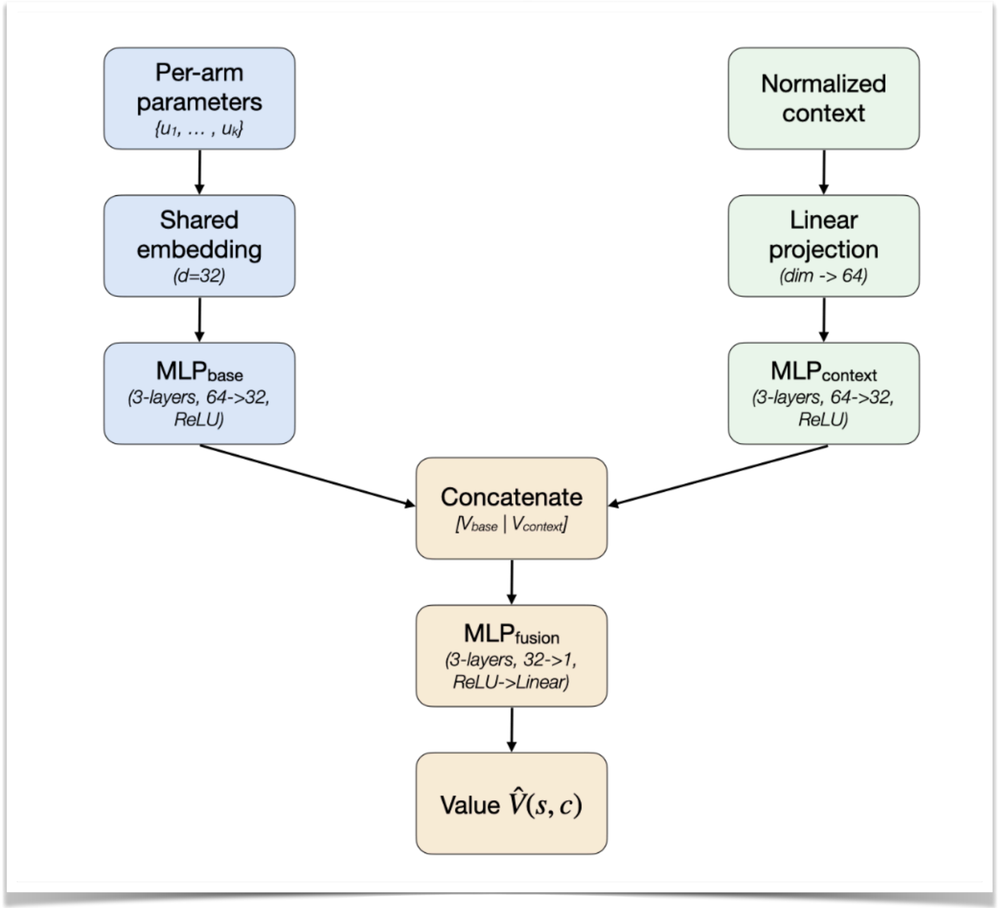
So, we need to approximate the value function in an efficient manner —> **Deep Learning**.

Deep Value Iteration using a dual-stream neural network.

Approximate the value function using a dual stream neural network.

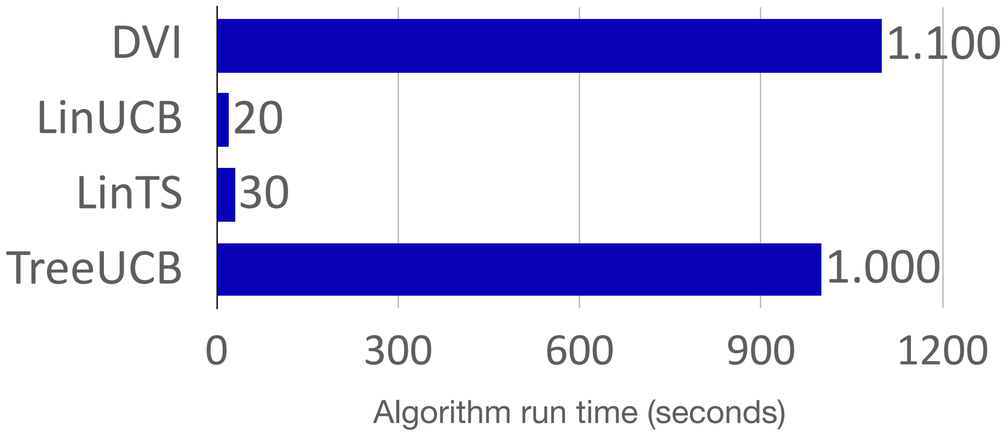
Base stream (left) learns per-arm uncertainty, and context stream (right) learns context corrections. Both streams are then combined to learn influence of context on per-arm uncertainty.

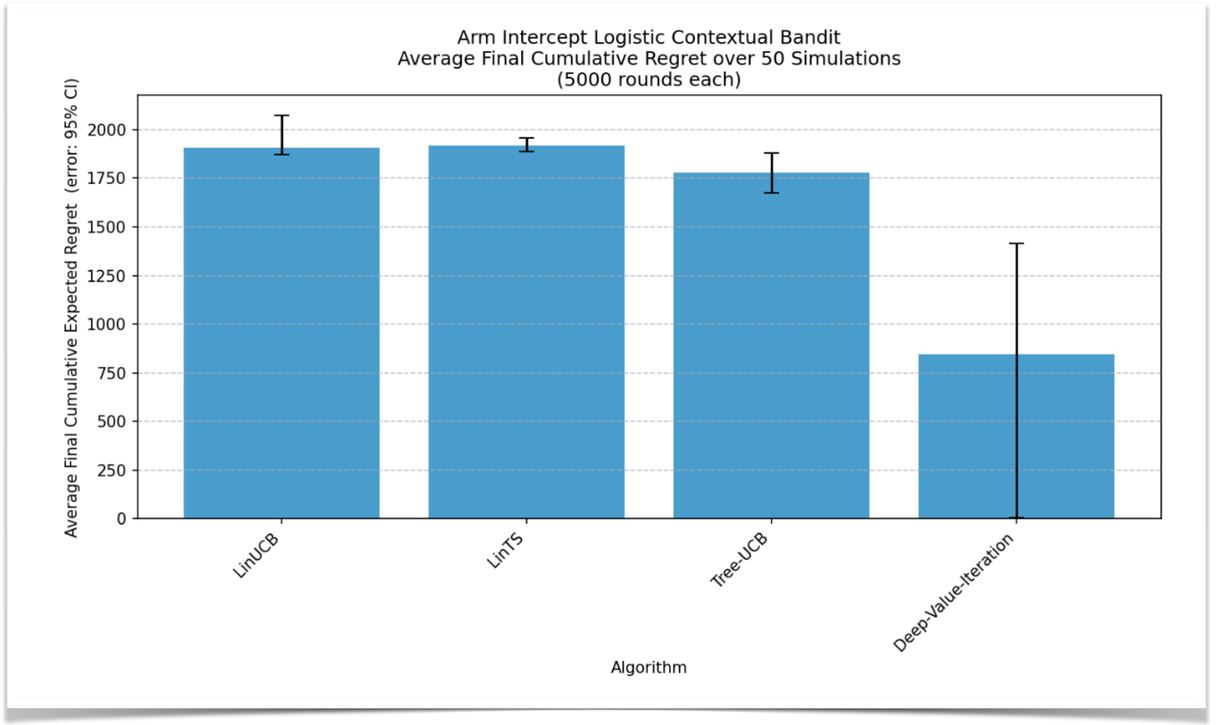
With this, value function can be approximated to arbitrary precision, and trade-off of training time vs precision can be chosen.



Deep Value Iteration using a dual-stream neural network.

With a synthetic problem with a much larger number of arm (20) and context (10) dimensions, Deep Value Iteration outperforms strong baselines.

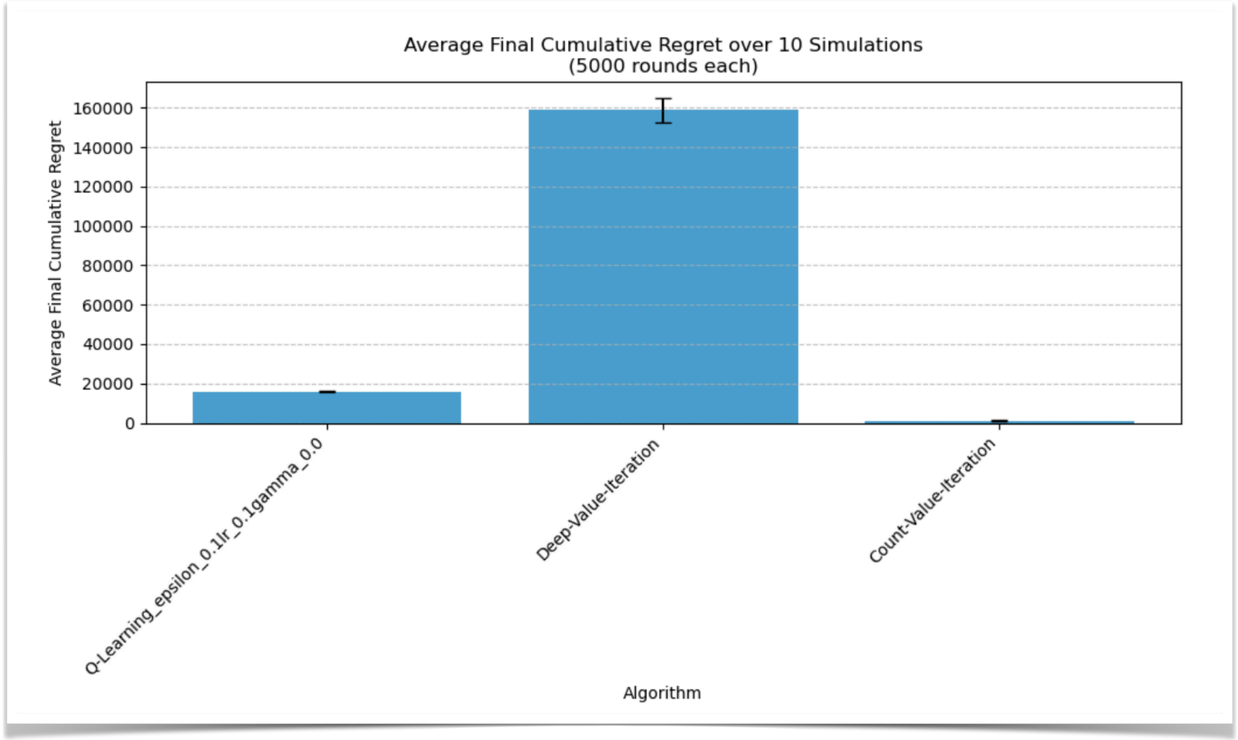




Deep Value Iteration does have its attention points.

Network design can impact performance linked to concrete problem, and without online testing, no clear way to know right design.

And more importantly, as in any Bayesian approach, prior misspecification can break performance.



Main take-aways.

- Ancillary seat pricing could logically be seen as a contextual multi-armed bandit problem.
- Instead of heuristic approaches to balance exploration-exploitation, we could also use an exact approach from OR, value iteration.
- On small problems, this can be calculated in an exact manner providing the optimal solution.
- On lager problem instances, we can approximate this through deep learning, to still find a much better approach than relevant baselines.
- However, we need to be careful in network design and prior choice to ensure good performance.

Learning to price ancillary seats with Bayesian Value Iteration

65th AGIFORS Annual Symposion, 2025

Kevin Duijndam (KLM), joint work with Ger Koole (VU), Rob van der Mei (CWI, VU)