Deep Reinforcement Learning for Inventory Management under Stochastic Demand

Yixin WANG [1] Yang YU [1] Joshua Zoen-Git HIEW [2] Vincent Tsz Fai CHOW [1, †]

[1]Faculty of Business, The Hong Kong Polytechnic University

[2]Department of Mathematical and Statistical Sciences, University of Alberta

yixinwang@polyu.edu.hk

December 8, 2023

Outline

Background

- Inventory management is the process of managing the flow of goods from the source to the customer.
- It involves ordering, storing, pricing, and selling, etc., of products at the right time and place.

Background

- **Uncertainty resources**:
  - Demand side (stochastic demand, censored demand,...);
  - Supply side (supply uncertainty, multiple supply resources,...);
  - Logistic (delivery lead time, disruptions,...).

- **Dimensional curse** (consider more factors: multiple products, fulfillment,...)

Motivation

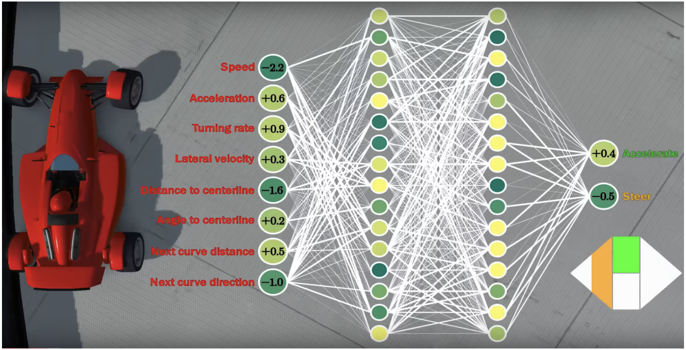
- Heuristics V.S. Deep Reinforcement Learning
  - No general heuristic rules;
  - Inventory policy optimization is non-linear and high-dimensional;
  - Make inventory decisions automatically by DRL.

AlphaGo

ChatGPT

Deep reinforcement learning

- The policy is represented by DNN to solve complicated problems;
- The policy is updated by constantly interacting with the environment and getting rewards or penalties.



https://www.youtube.com/watch?v=Dw3BZ6O_8LY

Inventory management model

- Stochastic demand environment
  - Seasonal/Poisson demand environments
  - Explore a DRL-based policy and access more available demand data to reduce the demand uncertainty.
- Inventory system with nonzero lead time

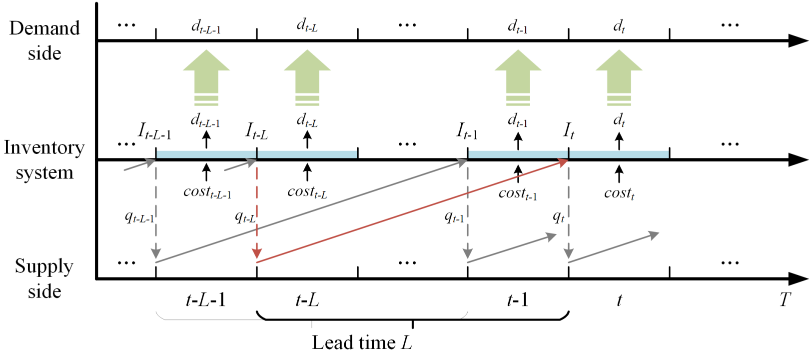


Figure 1: Dynamics of inventory management system.

Inventory management model

- ***States***:

$$s_t = (D_t, I_t, Q_t)$$

where D_t is a demand window at time t including the demand information from period $t - \tau$ to $t - 1$, $\{d_{t-\tau}, d_{t-\tau+1}, ..., d_{t-1}\}$, I_t represents the current inventory position, and Q_t is the undelivered orders $\{q_{t-l+1}, ..., q_{t-1}\}$

- ***Actions***: replenishment quantity, $a_t = q_t$;
- ***Rewards***: profit-to-go function
  - In lost-sales inventory system[1]:

$$\mathcal{R}_t(s_t, q_t, d_t) = pd_t - c_k + c_o q_t + c_h[I_t - d_t]^+ + c_p[d_t - I_t]^+$$

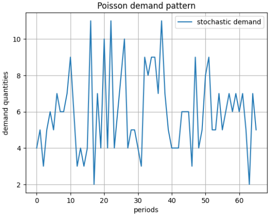
- ***Objective***: maximize the long-term profit

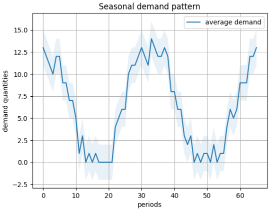
$$\mathcal{V}(s_t) = \max_{q_t \in \mathcal{Q}_t} \left\{ \mathcal{R}_t(s_t, q_t) + \gamma \sum_{s' \in \mathcal{S}_{t+1}} \mathbb{P}(s_{t+1} = s' \mid s_t, q_t) \mathcal{V}_{t+1}(s') \right\}$$

[1] *Lawrence V Snyder and Zuo-Jun Max Shen. Fundamentals of supply chain theory. John Wiley & Sons, 2019.*

Instances

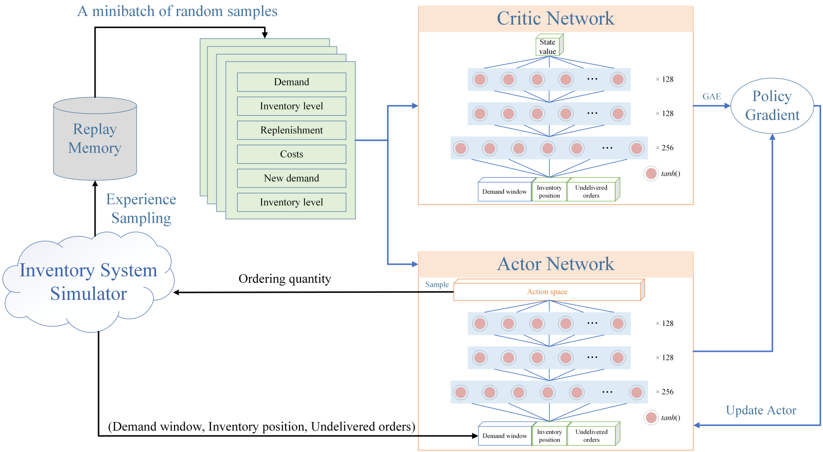
- Inventory management simulator
  - Lost-sale inventory system
  - finite-horizon length: 66 (weeks)
  - fixed costs c_k: 0
  - ordering cost c_o: 2
  - holding cost c_h: 1
  - selling price p: 10
  - **lost-sales cost c_p: *4, 8***
  - **lead time l: *4, 8, 12***
  - **demand pattern: *Poisson, Seasonal***





Optimization with DRL

- Proximal policy optimization with actor-critic[2]



[2] John Schulman et al. "Proximal policy optimization algorithms". In: arXiv preprint arXiv:1707.06347 (2017).

Training

Table 1: The hyperparameters of PPO-AC

Hyperparameters	Values
Discount factor γ	0.99
GAE weight parameter λ	0.95
Buffer size	256
Epoches K	16
Mini-batch size m	64
Hidden layers	3
Clipped parameter ϵ	0.2
Entropy coefficient β	1e-2
Activation function	*tanh()*
Hidden neural units	**[256, 128, 128]**
Actor network learning rate	**1e-2, 1e-4, 1e-6**
Critic network learning rate	**1e-2, 1e-4, 1e-6**
Training steps T	150000

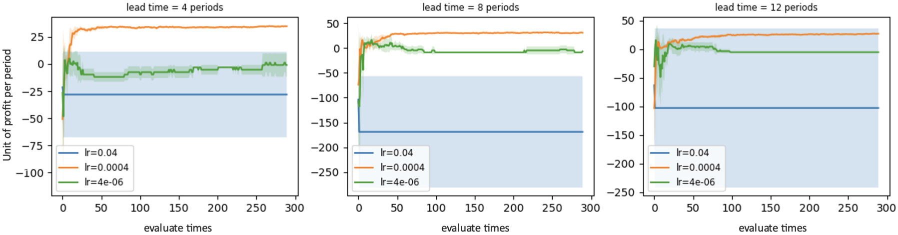


Figure 2: Learning curve (Poisson demand)

Behaviors

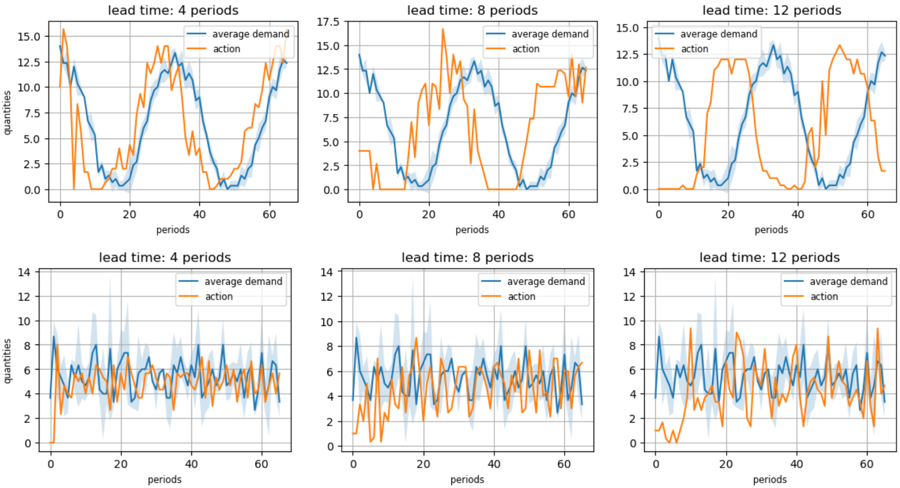


Figure 3: Policy trajectory

Behaviors

- Remove the demand information: $s_t = (I_t, Q_t)$

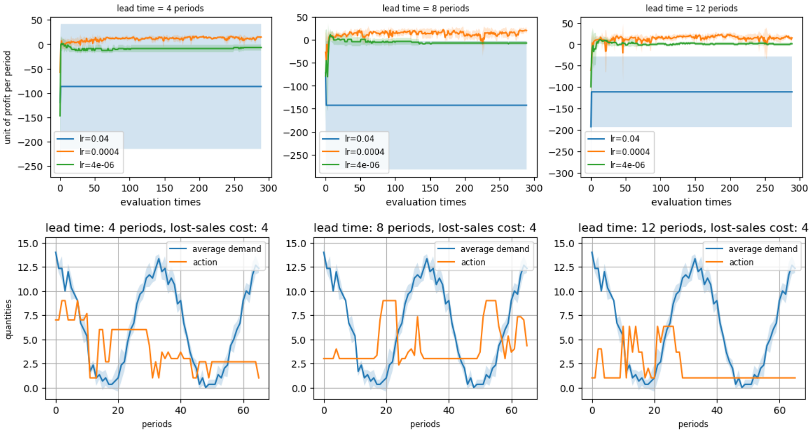


Figure 4: Learning curves and policy trajectory without demand information

Future work

- Compare DRL-based policy with the SOTA heuristic method
- Explore an effective representation and learning model in more complicated supply chain networks, e.g., multi-echelon, multi-sourcing

Thank you!