A Reinforcement Learning Approach to Dual-Sourcing Inventory Problem

Tonghua Tian and Xumei Xi

School of Operations Research and Information Engineering
Cornell University

tt543@cornell.edu, xx269@cornell.edu

May 17, 2021

Outline

Setup

- Two suppliers:

	Regular R	Express E
Lead time	L_r	L_e
Cost	c_r	c_e

- Assume $L_r > L_e + 1$ and $c_r < c_e$.
- Demands: i.i.d. nonnegative $\{D_t, t \geq 0\}$.
- Inventory: I_t.
- Pipeline vectors: $\mathbf{q}_t^r = \left\{q_{t-i}^r, i \in [L_r]\right\}, \mathbf{q}_t^e = \left\{q_{t-i}^e, i \in [L_e]\right\}$ denote orders placed but not yet delivered with R and E.
- Unit holding and backorder costs are $h > 0$ and $b > 0$.

Dynamics

At time $t \geq 0$, a sequence of events happen in the following order.

1. On-hand inventory I_t is observed.
2. Policy π places the new orders q_t^r and q_t^e (**action**).
3. New inventory $q_{t-L_r}^r + q_{t-L_e}^e$ is delivered and added to I_t.
4. Demand D_t is realized. Update inventory and pipeline vectors (**state**) according to

$$
\begin{aligned}
I_{t+1} &= I_t + q_{t-L_r}^r + q_{t-L_e}^e - D_t, \\
\mathbf{q}_{t+1}^r &= \left(q_{t-L_r+1}^r, \ldots, q_{t-1}^r, q_t^r\right), \\
\mathbf{q}_{t+1}^e &= \left(q_{t-L_e+1}^e, \ldots, q_{t-1}^e, q_t^e\right).
\end{aligned}
$$

5. Cost is incurred as

$$C_t = c_r q_t^r + c_e q_t^e + h I_{t+1}^+ + b I_{t+1}^-.$$

Minimize long-run average cost: $C(\pi) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[C_t^\pi\right].$

Tailored Base-Surge (TBS) Policy

- A TBS policy $\pi_{r,S}$ orders r products from R and follows an order-up-to rule from E, where we maintain the express inventory position above S,

$$\begin{aligned} q_t^r &= r \\ q_t^e &= \max\left(0, S - \hat{I}_t\right), \end{aligned}$$

  with $\hat{I}_t := I_t + \sum_{i=t-L_e}^{t-1} q_i^e + \sum_{i=t-L_r}^{t-L_r+L_e} q_i^r$.
- Empirically, TBS performs well with an increase in the lead time difference. (Klosterhalfen et al. 2011)
- Theoretically, TBS is asymptotically optimal as L_r increases when L_e is fixed. (Xin & Goldberg. 2018)

Outline

Advantage Actor-Critic (A2C) for Discounted Reward

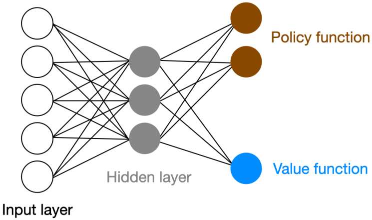
- Discount factor: $\gamma = 0.99$.
- Actor approximates policy function π_θ while critic approximates value function V_v.
- In each episode, obtain rollout trajectory $\{(s_t, a_t, r_t, s_{t+1})\}_{t=T}^{T+n}$. Minimize loss

$$L = L_{\text{actor}} + W \cdot L_{\text{critic}}\,, \quad \text{with}$$

$$\begin{cases} L_{\text{actor}} = -\sum_{t=0}^{n} \log \pi_\theta\left(s_{T+t}, a_{T+t}\right) \underbrace{\left(\sum_{i=t}^{n} \gamma^{i-t} r_{T+i} - V_v\left(s_{T+t}\right)\right)}_{\hat{A}(s_{T+t}, a_{T+t})}, \\ L_{\text{critic}} = \sum_{t=0}^{n} \Big[\underbrace{\sum_{i=t}^{n} \gamma^{i-t} r_{T+i}}_{V_{\text{target}}(s_{T+t})} - V_v\left(s_{T+t}\right) \Big]^2. \end{cases}$$

Implementation Details

- Share parameters: one neural network that has one softmax output for the policy and one linear output for the value function, sharing non-output layers.



- Initialization: supervised learning to make the NN policy close to an arbitrary TBS policy.
- Long rollout trajectory (1000).
- Tune W to control relative learning speed of the actor and critic: $L = L_{\text{actor}} + W \cdot L_{\text{critic}}$.
- Minimize loss using ADAM.

Outline

Dual-Sourcing Inventory Problem

Advantage Actor-Critic Algorithm

Empirical Results

Performance

Demands $D \sim \text{Poisson}(\lambda)$; fix $L_e = 1$, $c_r = 100$, $c_e = 105$, and $h = 1$. 100 simulations.

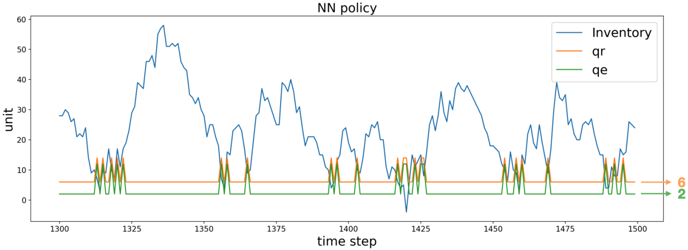
$(\Delta L, b)$	$(2, 99)$	$(10, 99)$	$(2, 19)$	$(10, 19)$
TBS	-515.31 ± 3.55	-516.67 ± 3.14	-515.21 ± 3.48	-516.48 ± 3.29
initial NN	-572.23 ± 40.91	-579.88 ± 41.55	-539.78 ± 3.25	-659.57 ± 3.52
NN	-539.58 ± 3.66	-547.30 ± 5.30	-520.80 ± 3.15	-553.76 ± 3.10

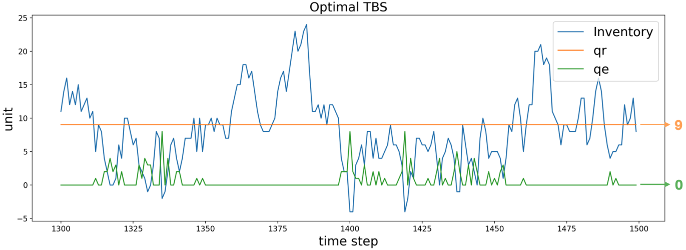
Table: $\lambda = 5$.

$(\Delta L, b)$	$(2, 19)$	$(10, 19)$
TBS	-1016.55 ± 7.30	-1019.55 ± 6.73
initial NN	-1095.06 ± 67.06	-1116.21 ± 60.64
NN	-1047.78 ± 4.92	-1040.43 ± 5.24

Table: $\lambda = 10$.

Output NN Policy vs. Optimal TBS Policy $(\lambda, \Delta L, b) = (10, 10, 19)$





Learning Curve $(\lambda, \Delta L, b) = (10, 10, 19)$