Mathematics
BARNARD COLLEGE

# Reinforcement Learning in Dual-Sourcing Problem: A Focus on Real and Simulated Demand Shocks

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Jet Li[1], Yihan Shen[1], Jingwen Zhang[2]

[1]Mathematics Department, Columbia University

[2]Mathematics Department, Barnard College

## Background Introduction

Dual sourcing is a supply chain management strategy that involves establishing relationships with two suppliers for a particular product or component, and they are classified as the 'regular supplier' and the 'expedited supplier' based on their distinctive lead time and order cost.

- We model the problem under the Markov Decision Process framework and compare the Advantage Actor-Critic (A2C) algorithm against the traditional Tailored Base-Surge (TBS) policy.
- We evaluate their performance for different parameter values of the simulated demand sequence.
- We assess their adaptability to market volatility by simulating a demand shock and considering their time-to-convergence and average reward levels
- We investigate their sourcing strategies in real-life setting by applying the historical demand sequence from a global microchip manufacturing company.

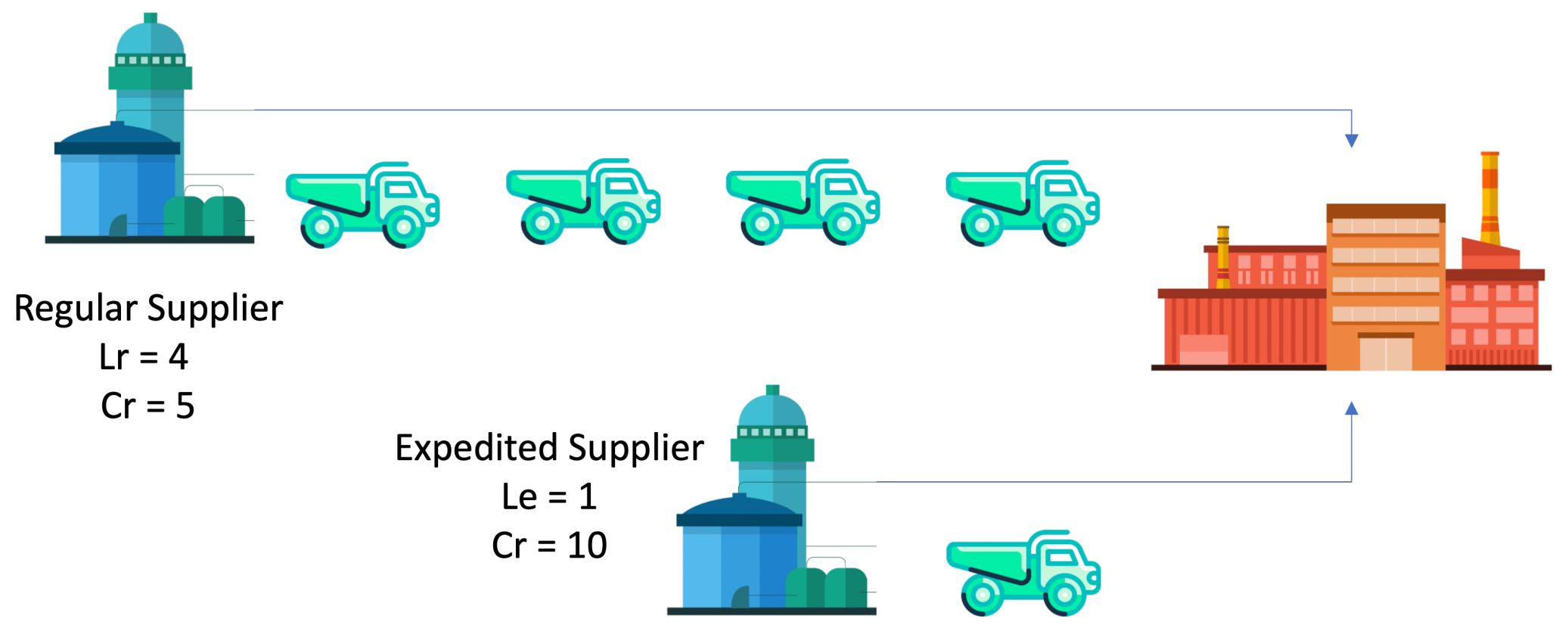

Figure 1. Illustration of Dual Sourcing

## Problem Formulation

- On-hand inventory as of time t: $I_t$
- Cost of holding too much: $h > 0$, or too little: $b > 0$
- Two suppliers, $R$ and $E$, with respective lead time & order cost: $L_r$, $c_r$, $L_e$, $c_e$, assuming $L_r > L_e$ and $c_r < c_e$
- Random demand sequence: $\{D_t, t \geq 0\}$.
- Order history, observing at time t:

$$\mathsf{q}_t^r = \left(q_{t-L_r}^r, q_{t-L_r+1}^r, \ldots, q_{t-1}^r, \right)$$
$$\mathsf{q}_t^e = \left(q_{t-L_e}^e, q_{t-L_e+1}^e, \ldots, q_{t-1}^e, \right)$$

- After placing the orders at time t, update the inventory and the order history:

$$I_{t+1} = I_t + q_{t-L_r}^r + q_{t-L_e}^e - D_t$$
$$\mathsf{q}_t^r = \left(q_{t-L_r}^r, q_{t-L_r+1}^r, \ldots, q_{t-1}^r, q_t^r\right)$$
$$\mathsf{q}_t^e = \left(q_{t-L_e}^e, q_{t-L_e+1}^e, \ldots, q_{t-1}^e, q_t^e\right)$$

- To decide what new orders at t should be, we need a policy that maps states to actions
- A policy $\pi$: $\{f_t^\pi, t \geq 0\}$, with $(q_t^r, q_t^e) = f_t^\pi(\mathsf{q}_t^r, \mathsf{q}_t^e, I_t)$
- Reward at time t, given policy $\pi$:

$$R_t^\pi = -\left(c_r q_t^r + c_e q_t^e + hI_{t+1}^+ + bI_{t+1}^-\right)$$

- We want to maximize the total expected reward over each episode of length L:

$$R(\pi) = \sum_{t=0}^{L-1} \gamma^t E[R_t^\pi]$$

## Algorithms

- Tailored Base-Surge (TBS) Policy: A TBS policy $\pi_{r,S}$ orders $r$ units from $R$ at each time step, and brings in additional products from $E$ only if inventory position $I_t$ fall below some safety stock level $S$:

$$q_t^r = r \qquad q_t^e = max(0, S - I_t)$$

- Theoretically TBS is asymptotically optimal as difference $L_r$ and $L_e$ is large enough; empirically TBS often causes the inventory to fall below 0, which could be highly undesirable for products less frequently needed but of high stake, such as blood for transfusion.
- Advantage Actor-Critic (A2C) Algorithm: To search for the optimal policy, we should characterize, for a given state, the 'advantage' of taking an action compared to doing nothing. That surplus, $A^\pi(s,a)$ is the difference between $Q^\pi(s,a)$ and $V^\pi(s)$, where

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \middle| s_0 = s, a_0 = a\right]$$
$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \middle| s_0 = s\right]$$

- To learn $\pi$ and $V^\pi(s)$ which are both unknown, we parameterize the policy with $\boldsymbol{\theta}$ and the value function with $\mathbf{w}$, and designate two neural networks to learn their parameters.
- During each episode, we aim to minimize the combined loss from the *actor* and the *critic*

$$L := L_{\text{actor}} + W \cdot L_{\text{critic}}, \text{ where}$$
$$L_{\text{actor}}(\theta) = -\sum_{t=0}^{2000} \log(\pi_\theta(s_{T+t}, a_{T+t}))\hat{A}(S_{T+t}, a_{T+t})$$
$$L_{\text{critic}}(w) = \sum_{t=0}^{2000} \Big(\hat{Q}(S_{T+t}, a_{T+t}) - V_w$$

## Experiments on Simulated Demand Distributions with Different Parameters

We work with simulated demand sequence whose entries come from a Poisson process with parameter $\lambda$: $D \sim 2 + Pois(\lambda)$. We carried out experiments with different values of $\lambda$ and recorded the results as follows:

| (Avg Cost, Avg Inv, Stdev Inv) | $\lambda$=2 | $\lambda$=4 | $\lambda$=6 |
|---|---|---|---|
| Opt Avg Cost | 20 | 30 | 40 |
| $\epsilon$-greedy A2C | (41.5, 9.3, 13.7) | (63.0, 5.2, 15.4) | (75.5, 2.4, 17.3) |
| TBS-Exploratory A2C | (43.4, 8.4, 14.2) | (67.1, 5.5, 17.7) | (79.8, 3.2, 17.6) |
| TBS | (46.7, 15.9, 8.5) | (59.2, 17.0, 8.7) | (67.8, 18.7, 8.4) |

Table 1. Average Cost, Average Inventory, and Standard Deviation of Inventory of $\epsilon$-greedy A2C, TBS-Exploratory A2C, and TBS Under Different Lambda Parameters

- Across all values of $\lambda$, both A2C agents and TBS policy yields costs that are approximately double the theoretical optimum.
- A2C manages to outperform TBS for small lambda values, but its performance declines when demand becomes more volatile, a trend that persists with larger $\lambda$ values.

## Simulated demand shock in the middle of each episode

we initially model the demand sequence with the same setup as before. In the middle of an episode, we increased the poisson parameter by 50%.

- A2C takes around 50 time-steps to pull the inventory away from the sub-zero region and converges it back the previous established stock level at around 20 units. It achieves so by ordering close to the max levels from both supplier
- On the other hand, TBS has poor control over the inventory and is unable to recover the frequent negative balance.

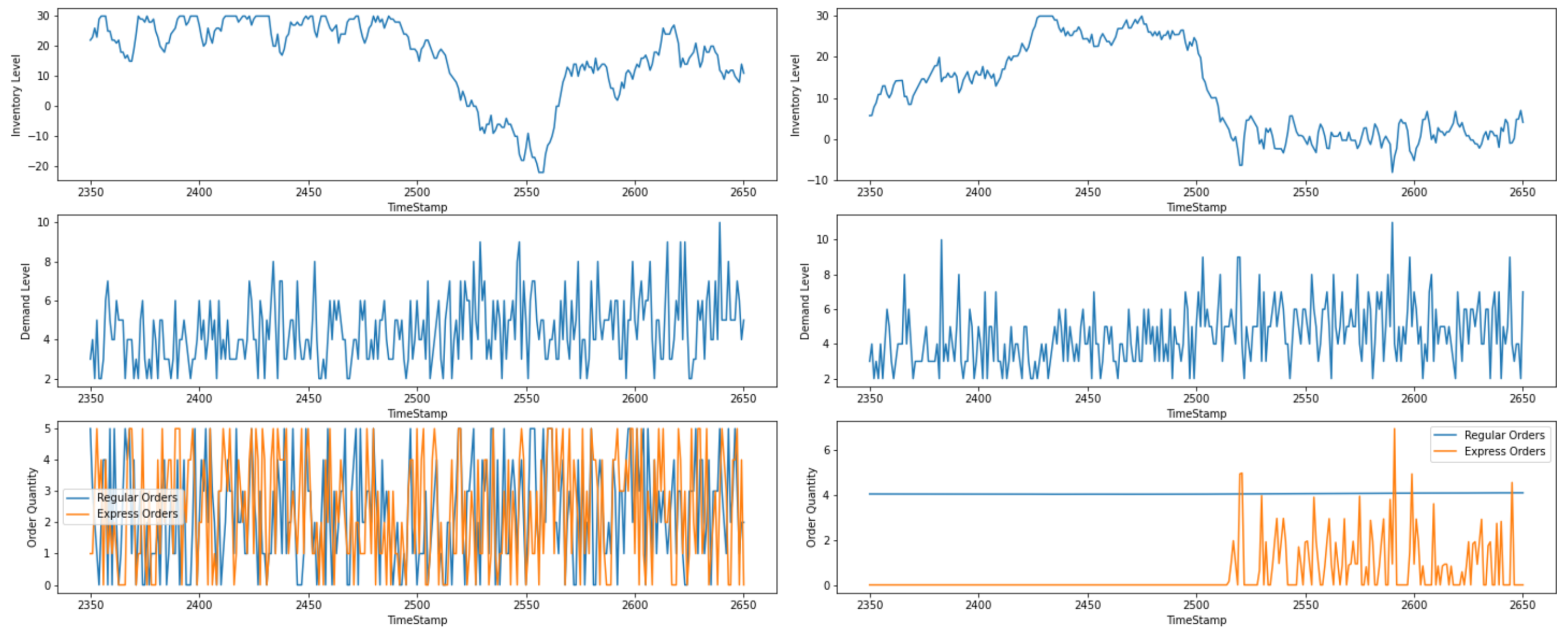

Figure 2. Response of A2C Agent(left) and TBS(right) to simulated demand shock

## Real-Life Demand Data

- During periods of unpredictable spikes(200-250 and 420-470), the A2C Agent ricochets between extremes, while TBS policy methodically works to replenish its safety stock, ultimately achieving this goal after a brief period in which the inventory hovers around zero level.
- Although A2C exhibits a riskier and more unstable approach while TBS a more conservative one, their overall performance, as measured by the daily cost, is comparable.

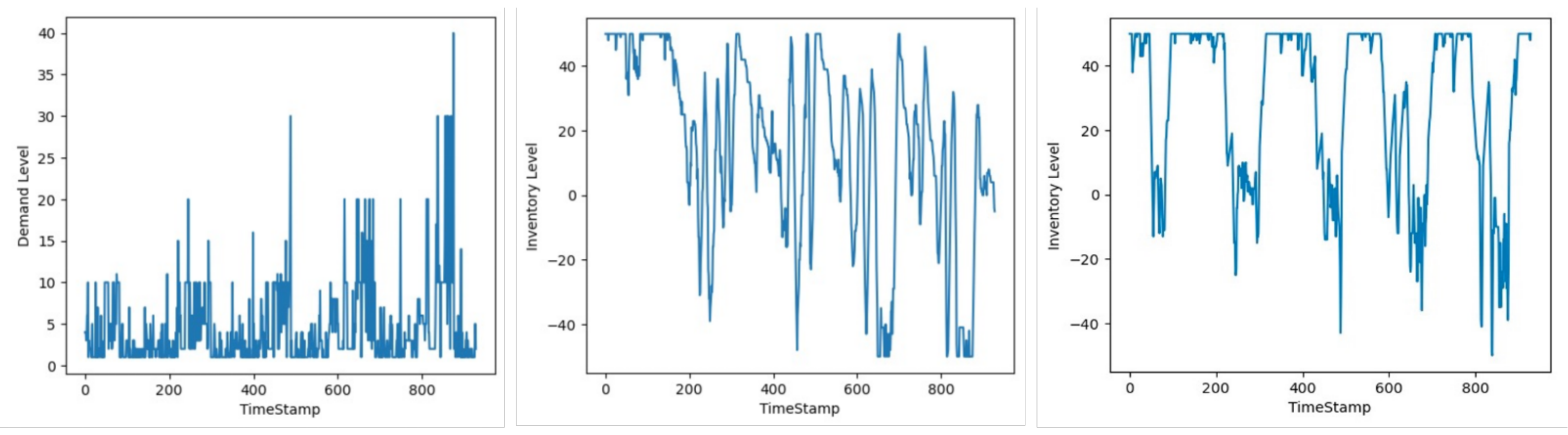

Figure 3. Response of A2C Agent(left) and TBS(right) to simulated demand shock

## References

[1] Ivars Dzalbs and Tatiana Kalganova. "Accelerating supply chains with Ant Colony Optimization across a range of hardware solutions". In: *Computers Industrial Engineering* 147 (2020), p. 106610. ISSN: 0360-8352. DOI: `https://doi.org/10.1016/j.cie.2020.106610`. URL: `https://www.sciencedirect.com/science/article/pii/S0360835220303442`.

[2] Joren Gijsbrechts et al. "Can Deep Reinforcement Learning Improve Inventory Management? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems". In: *Manufacturing & Service Operations Management* 24.3 (2022),