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

Jet Li, Yihan Shen, Jingwen Zhang

July 2023

Abstract

In this paper, we analyze the performance of Advantage Actor-Acritic(A2C) algorithm against the traditional Tailored Base-Surge(TBS) policy in the dual-sourcing problem through three experiments. First, we test the algorithm in a small-scale environment with different demand distributions. The compact state-action space enables us to effectively experiment different exploration strategies while controlling the learning rate. Next, we simulate a demand shock by altering the expected value and variance of the demand in the middle of an episode. We assess the algorithm's resilience to sudden changes and measure its adaptability by considering its time-to-convergence and average reward levels. Lastly, we apply A2C to a historical demand dataset from a global manufacturing company, evaluating its behavior in a real-market setting.

Our results reveal several interesting insights. In the first experiment, while both the A2C agents and TBS policy yield costs roughly double the theoretical optimum, A2C's performance declines with increased demand volatility. In the simulated shock, A2C demonstrates adaptability by quickly converging back to a reasonable stock level, while TBS struggles to recover. Real-world data shows that overall, A2C exhibits a riskier approach with quick adaptability, while TBS's conservative strategy ensures stability but often leads to overstocking. Notably, both policies perform comparably in average daily cost, even though TBS had a much better performance in volatile environments in our first experiment. This result illustrates potential weaknesses in evaluating the performance of A2C Agents solely based on simulated environments. Our findings offer valuable insights into applying reinforcement learning techniques in supply chain optimization and their adaptability to real-world scenarios.

1 Introduction

Supply chain management refers to the process of planning, implementing, and controlling the flow of goods from the point of origin to the point of consumption. One common practice in supply chain management is Dual Sourcing, which is also known as dual supply or dual vendor strategy. This strategy involves establishing relationships with two or more suppliers for a particular product, component, or raw material, thereby providing an alternative source of supply in case one supplier encounters disruptions or fails to meet the required business objectives. Considering the variability in both price and delivery time across different suppliers, the enterprise need to find an optimal strategy of managing the multiple suppliers in order to maximize their own profitability.

Under the dual sourcing problem, inventory can be replenished through two avenues: a slow and economical source, and a faster but more expensive source. This parallels the dual-mode problem, where inventory is restocked from a single supplier utilizing two complementary modes of transportation. In this paper, we call the former the regular supplier and the later express supplier, and their different capabilities are reflected in parameters regular/express lead time and regular/express costs.

We modeled the dual sourcing problem under conventional assumptions. The demand for the particular product is modeled by a sequence, which is either independently drawn from a Poisson distribution, or from the historical data of a global manufacturing company. The ordering costs and lead times are all deterministic. We held the inventory costs and back order costs constant as well. The former represents the expense of warehousing, handling, and depreciation associated with storing the goods, and the later corresponds to the penalty of failing to the meet the customers' demands due to insufficiency of inventory.

The unmet demand is fully backlogged.

Despite the conceptual simplicity of the dual sourcing problem, the search for optimal strategies remain challenging. Traditional heuristics, such as Tailored Base-Surge, have been shown to perform well under the classical framework with fixed parameters for the demaned sequence and lead times. As soon as more stochastic features are introduced or industry-specific constraints imposed, the performance of heuristic strategies can become compromised[1]. Other computational techniques such as dynamic programming also become intractable because of the famous curse of dimensions—the size of the state and action spaces increase exponentially with increase in dual sourcing parameters. Approximate Dynamic Programming(ADP) has recently shown promising results to solve the dual sourcing problem by estimating the state values, but it often requires problem-specific tailoring, such as reduction to particular convex optimization problems, and doesn't generalize well when facing complex, uncertain demands[2].

2 Literature Review

Models on characterizing the dual sourcing environment typically took on three forms: analytical models that involve mathematical formulations and optimizations as in [5], simulation-based models that explore dynamic and stochastic aspects of the game in [6], and game theory models that focus on analyzing the strategic interactions between suppliers and our buyer agent, such as [6].

In the past decade, Reinforcement Learning (RL), has emerged as a promising tool to solve complex control and decision-making problems including dual sourcing. RL algorithms learn from the interaction with the environment to make decisions, a setting that closely mimics the real-world conditions of dual sourcing. In particular, we deploy the Advantage Actor Critic(A2C) algorithm, which combines the strength of both value-based and policy-based RL approaches to guarantee an efficient learning process. A2C has been recently shown to perform well in multi-echelon inventory management problems and safety stock optimization for perishable goods[3][4].

In A2C, the algorithm consists of two primary components: the 'Actor' and the 'Critic'. The 'Actor' is responsible for making decisions. It determines how many orders to be placed to regular and express suppliers based on the current inventory levels, demand, and existing orders, aiming to maximize the expected long-term reward, represented by the negative of total cost in our dual-sourcing setting. The 'Critic', on the other hand, evaluates the quality of the actions taken by the 'Actor' and measures how profitable it is for the agent to be in a particular state under the current policy.

The key idea in A2C is that the 'Actor' updates the policy in a direction suggested by the 'Critic'. This allows the 'Actor' to learn more about the action-value function, not merely from the immediate reward, but also based on the potential future rewards estimated by the 'Critic'. The A2C algorithm specifically leverages the advantage function, which quantifies the relative benefit of taking a certain action compared to the average action in a given state. By doing so, it mitigates the high variance issue that often arises in policy gradient methods, making the learning process more stable.

The A2C model's ability to manage the exploration-exploitation trade-off and handle large action spaces makes it well-suited to address the intricate dynamics of dual sourcing problem. As such, our study seeks to analyze the performance of the A2C algorithm in a dual sourcing context. Specifically, we aim to investigate its potential to generate robust and adaptive ordering policies in response to both demand shock in the midst of simulation and under realistic demand shock circumstances.

3 Problem Formulation

In the dual-sourcing problem, the inventory can be replenished at unit cost c_r from a regular supplier R with lead time L_r or/and from an express source E with lead time L_e at premium unit cost c_e. In the beginning of any timestamp t, two order quantities, q_t^r and q_t^e, and must be decided after observing the last inventory level on hand, I_{t-1}, and outstanding receipts from regular and express suppliers,

$\mathbf{Q}^r_{t-1} = \left(q^r_{t-L_r}, q^r_{t-L_r+1}, \ldots, q^r_{t-1}\right)$ and $\mathbf{Q}^e_{t-1} = \left(q^e_{t-L_e}, q^e_{t-L_e+1}, \ldots, q^e_{t-1}\right)$.

After the order decision, orders $q^r_{t-L_r}$ and $q^e_{t-L_e}$ are received and added to the on-hand inventory. Then, the unknown demand D_t is realized and subtracted from the on-hand inventory. Excess demand is fully backlogged so that the inventory and outstanding receipts evolve as $I_{t+1} = I_t + q^r_{t-L_r} + q^e_{t-L_e} - D_t$. Finally, outstanding pipeline vectors are updated as $\mathbf{Q}^r_t = \left(q^r_{t-L_r+1}, q^r_{t-L_r+2}, \ldots, q^r_t\right)$; $\mathbf{Q}^e_t = \left(q^e_{t-L_e+1}, q^e_{t-L_e+2}, \ldots, q^e_t\right)$.

The dual-sourcing problem can be modeled as a Discrete Markov Decision Process with states represented by the on-hand inventory level and the pipeline vectors. $\mathbf{S}_t = (I_{t-1}, \mathbf{Q}^r_{t-1}, \mathbf{Q}^e_{t-1})$. The action taken in period t is now two dimensional: $\mathbf{a}_t = (q^r_t, q^e_t)$ consisting of the ordered quantities from the regular and expedited sources. To decide what new orders at t should be, we need a policy that maps states to actions, which is defined as π: $\{f^\pi_t, t \geq 0\}$, with $(q^r_t, q^e_t) = f^\pi_t(\mathbf{q}^r_t, \mathbf{q}^e_t, I_t)$.

Conventionally and most intuitively, a reward at time t given policy π is the negative cost: $R^\pi_t = -\left(c_r q^r_t + c_e q^e_t + h I^+_{t+1} + b I^-_{t+1}\right)$, where h, b represents the storing cost and back order costs, respectively. In our experiments, however, we modified the reward by introducing surplus and insufficiency penalty terms, which would encourage the A2C agent to hold an inventory that's reasonably close to 0. The former penalty is enacted when the agent makes excessive express orders when current inventory is already sufficient($I_t > L_r D^-_{t-1}$); while the latter penalty is dealt whenever agent orders insufficiently even when the inventory level is well below 0 ($I_t + q^r_{t-Lr} + q^e_{t-L_e} < 0$), which would result in consecutive rounds of out-of-stock. We defined the surplus penalty to be $p_1 \cdot h \cdot q^e_t$, and the insufficiency penalty to be $p_2 \cdot b \cdot (-I_t - q^e_t - q^r_t)$, where p_1, p_2 are hyper-parameters that were later fine-tuned.

In order to have a finite state space, we also kept a max inventory level I_{max} and any inventory overflow or back-logged inventory insufficiency are penalized. Similarly, to limit the dimension of state space and reflect the realistic supplier capacity, we set a maximum number of orders, O_{max}, that can be placed in each round.

Our goal is to identify policy $\pi^* = \{\mathbf{a_i}|\mathbf{S_i}\}_{\mathbf{i=1,2,3}\ldots}$ (i.e. a sequence of actions given current states) that maximizes the sum of total expected rewards after each episode of length L: $R(\pi) = \sum_{t=0}^{L-1} \gamma^t E[R^\pi_t]$. The rewards are discounted by a constant factor γ, which lies strictly between 0 and 1 to reflect the preference of receiving rewards sooner and the depreciation of currency. In our experiments, L is configured to fall in range of 4000 to 10,000 time stamps. After empirical testing, we established its default value to 2000 to promote efficient learning and ensure robust convergence of optimal policy.

Instead of maximizing the long-run average reward, we chose to optimize for the sum of rewards over a finite period for a few reasons. First, we were interested in investigating the influence of initial stock level(I_0) on the behavior of the agent. For example, we would like to inquiry if an initially stressful situation would push the agent to have a more volatile inventory levels, whereas the infinite-time mean expected reward is independent of the initial state. Secondly, we believe that the finite horizon simulation is more aligned with the short-term and mid-term business cycles, such as a fiscal quarter or a tax season. Therefore, optimization of rewards over such period might be more relevant and practical as a business objective. Lastly, the episodic simulation better models uncertainty inherent in the supply chain dynamics. As the supplier systems can constantly update their constraints and capacities, the demand, lead time, and delivery costs can all vary significantly from season to season.

4 Algorithms

4.1 Tailored Base-Surge 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 $S = L_e\mu + z\sqrt{L_e}\sigma$, where the μ and σ are approximated by the mean and standard deviation of the observed demand distribution, and z corresponds to z-score of the desired service level. The amount of order at time t for the two suppliers are:

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

Theoretically TBS is asymptotically optimal as the difference between L_r and L_e becomes large enough; empirically TBS often causes the inventory to fall below 0 with non-optimal choices of the policy parameters, which could be highly undesirable for products less frequently needed but of high stake, such as blood for transfusion. The simulated demand sequence drawn from a Poisson distribution with a low expected value would enable us to explore the optimal service level to balance between penalty on such shortage and compromise on the overall performance.

4.2 Advantage Actor-Critic (A2C) Algorithm

To search for the optimal policy, we need to characterize, for a given state, the 'advantage' of taking a specific action compared to performing the 'default' action. That surplus, $A^\pi(s,a)$, is the difference between $Q^\pi(s,a)$ and $V^\pi(s)$, where

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

The large state space lends itself to the scheme of function approximation. To learn π 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.

In our first experiment, we modeled the demand sequence as a Poisson process and simulated the dual sourcing process for 400 episodes, each episode with a length of 2000 time steps. In each episode, we get an estimate for the value function V_w, and we use the policy π_θ from the last episode to get a 2000-step discounted total return, and we get a trajectory in the form of $\{(s_t, a_t, r_t, s_{t+1})\}_{t=T}^{T+2000}$.

During each episode, we aim to minimize the combined loss from the *actor* and the *critic*

$$\begin{aligned}
L &:= L_{\text{actor}} + W \cdot L_{\text{critic}}, \text{where} \\
L_{\text{actor}}(\theta) &= -\sum_{t=0}^{2000}\log(\pi_\theta(s_{T+t}, a_{T+t}))\hat{A}(S_{T+t}, a_{T+t}) \\
&= -\sum_{t=0}^{2000}\log(\pi_\theta(s_{T+t}, a_{T+t}))\Big(\hat{Q}(S_{T+t}, a_{T+t}) - V_w(S_{T+t})\Big) \\
&= -\sum_{t=0}^{2000}\log(\pi_\theta(s_{T+t}, a_{T+t}))\Big(\sum_{i=t}^{2000}\gamma^{i-t}\cdot r_{T+i} - V_w(S_{T+t})\Big) \\
L_{\text{critic}}(w) &= \sum_{t=0}^{2000}\Big(\hat{Q}(S_{T+t}, a_{T+t}) - V_w(S_{T+t})\Big)^2 \\
&= \sum_{t=0}^{2000}\Big(\sum_{i=t}^{L-1}\gamma^{i-t}\cdot r_{T+i} - V_w(S_{T+t})\Big)^2
\end{aligned}$$

The *actor* explores the environment and approximates π_θ while the *critic* approximates V_w. We tune W to control their relative learning speed, and we minimize the combined loss L using an ADAM optimizer.

Algorithm 1 Advantage Actor-Critic (A2C) in Dual-Sourcing

Input: Differentiable policy and value-function parameterization: $\pi(\mathbf{a}|\mathbf{s}, \boldsymbol{\theta})$, $\hat{V}(\mathbf{s}, \mathbf{w})$
Dual Sourcing Environment $env = env(D, L_r, L_e, c_r, c_e, b, h)$
Parameters: Actor parameter $\boldsymbol{\theta}$, Critic parameter $\mathbf{w}$, learning rates $lr^{\boldsymbol{\theta}} > 0, lr^{\mathbf{w}} > 0$
discount factor γ, critic loss weight W, overstocking penalty p_1, insufficiency penalty p_2
Initialization: Initial State $\mathbf{s_0} = (I_0, \mathbf{Q}_0^r = \mathbf{0}^{L_r}, \mathbf{Q}_0^e = \mathbf{0}^{L_e})$, $T \leftarrow 1$, $t \leftarrow 1$, $T_{max} \leftarrow 400$, $L \leftarrow 2000$.

while $T \leq T_{max}$ **do**
  $t_{start} \leftarrow t$.
  while $t - t_{start} \leq L$ **do**
    draw action $\mathbf{a_t} = (q_t^r, q_t^e) \sim \pi(\cdot|\mathbf{s_t}, \boldsymbol{\theta})$.
    realize demand and observe next State $\mathbf{s_{t+1}} \sim P_{env}(\cdot|\mathbf{s_t}, \mathbf{a_t})$.
    incur cost $c_t = \left(c_r q_t^r + c_e q_t^e + h I_{t+1}^+ + b I_{t+1}^-\right)$
    compute penalty-adjusted reward $r_t \leftarrow -c_t$
    if $I_{t+1} > L_r \bar{D}_t$ **then** $r_t \leftarrow r_t - p_1 h q_t^e$
    end if
    if $I_{t+1} + q_{t-Lr+1}^r + q_{t-L_e+1}^e < 0$ **then** $r_t \leftarrow r_t + p_2 b (I_t + q_t^e + q_t^r)$
    end if
    $t \leftarrow t + 1$.
  end while
  $R \leftarrow \hat{V}(\mathbf{s_t}, \mathbf{w})$.
  $L_{\text{actor}} \leftarrow 0$. $L_{\text{critic}} \leftarrow 0$.
  for $i = t - 1, \ldots, t_{\text{start}}$ **do**
    $R \leftarrow \gamma R + r_i$.
    $L_{\text{actor}} \leftarrow L_{\text{actor}} - \log \pi(\mathbf{a_i}|\mathbf{s_i}, \boldsymbol{\theta})(R - \hat{V}(\mathbf{s_i}, \mathbf{w}))$.
    $L_{\text{critic}} \leftarrow L_{\text{critic}} + (R - \hat{V}(\mathbf{s_i}, \mathbf{w}))^2$.
  end for
  $L \leftarrow L_{\text{actor}} + W \cdot L_{\text{critic}}$.
  Gradient descent step to minimize L.
  $\theta \leftarrow \theta - lr^{\boldsymbol{\theta}} \nabla_\theta L$
  $\mathbf{w} \leftarrow \mathbf{w} - lr^{\mathbf{w}} \nabla_w L$
end while

5 Results

To comprehensively evaluate the performance of A2C Agent in Dual-Sourcing, we design three sets of experiments. Firstly, we test our algorithm in a small-scale setting with various demand distributions, each characterized by a unique Poisson parameter, λ. The reasonable size of the state-action space in such environment allows for quick exploration and easier control of the learning rate, which we experimented and analyzed against the the state-of-art heuristics. The optimality gap serves a key metric.

Secondly, we introduce a simulated demand shock into the dual sourcing environment, increasing its expected value by 25% and its variance by 50%. This experiment allows us to assess our algorithm's ability to handle unforeseen disruptions and make informed decisions in dynamic environments. The adaptability of the policies are measured by two factors, the time it takes for the agent to re-converge to a near-optimal strategy, and the change in average reward levels when compared to the baseline levels before the demand shock.

Lastly, to validate the applicability of our approach, we apply the A2C Agent to a real-life dataset available on Kaggle. This dataset records a global manufacturing company's demand sequence for Product-1938 from a particular warehouse, Warehouse J, from 2012-2016. It provides us with valuable insights into how well our algorithm performs in a real-world setting.

5.1 Different demand sequences across range of λ values

We work with simulated demand sequence whose entries come from a Poisson process with parameter λ: $D \sim 2 + Pois(\lambda)$. The Dual Sourcing Environment is set up as follows: Regular Lead Time=8, Express Lead Time=2, Regular Cost=5, Express Cost=6, Max Order=8, Max Inventory=30, Starting Inventory=15, Storing Cost=1.6, Back Cost=3.8, Discount Factor=0.95.

We test the performance of three policies:
i) ϵ-greedy A2C, which starts off with completely random policy to explore the environment, and ϵ gets decreased by 0.99 every time stamp.
ii) TBS-Exploratory A2C, which spends the first tenth of timesteps performing TBS-guided off-policy learning.
iii) Conventional TBS policy with service level=0.9

We carry out experiments with different values of λ and record the results as follows:

(Avg Cost, Avg Inv, Stdev Inv) Opt Avg Cost	λ=2 20	λ=4 30	λ=6 40
ϵ-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)

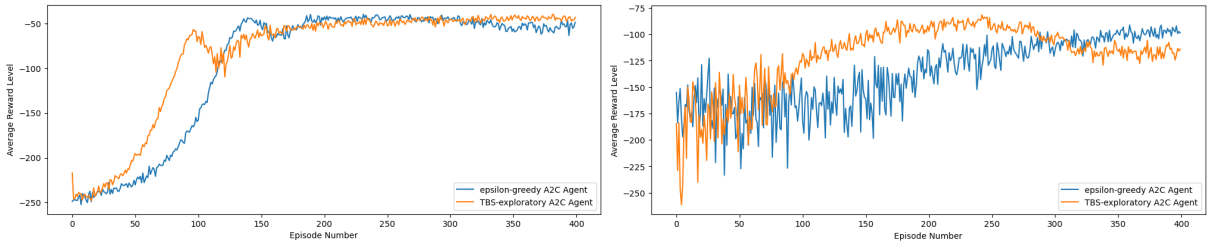
Table 1: Average Cost, Average Inventory, and Standard Deviation of Inventory of ϵ-greedy A2C, TBS-Exploratory A2C, and TBS Under Different Lambda Parameters

The overall performance of the ϵ-greedy A2C, TBS-Exploratory A2C, and TBS policies is evaluated in relation to the optimal average cost, defined as the product of average demand and regular order cost. This cost represents the theoretical minimum cost attainable, given perfect foresight of upcoming demand throughout the simulation.

Across all values of λ, both A2C agents and TBS policy yields costs that are approximately double the theoretical optimum(Table 1). Interestingly, the ϵ-greedy A2C demonstrates superior efficiency when demand remains relatively stable(i.e., λ=2). Here, the agent registers an average cost of 41.5, marginally outperforming the TBS strategy with its slightly higher cost of 46.7. However, as the demand becomes increasingly volatile, the performance of both A2C agents declines, a trend that persists with larger λ values as well.

The contrasting sourcing behaviors of the A2C agents and TBS policies become apparent as λ escalates. The A2C agents consistently display a declining average inventory level, dropping from 9.3 to 5.3 and then to 2.4, while the TBS policy shows an increment in average inventory necessary to uphold a larger safety stock to meet the same service level. As anticipated, both A2C agents show an increase in inventory variance as λ increases,, while TBS maintains a comparably steady inventory level in face of increasing demand volatility.

As expected, we also observe that larger λ increases the number of episodes required for both A2C agents to have their policies converge. When $\lambda = 2$, both A2C agents required approximately 200 episodes of learning before developing an optimal policy; when $\lambda = 6$, however, their policies didn't converge until around episode 360.



(a) λ=2 (b) λ=4

Figure 1: Learning of A2C Agents: Average Reward Level over Episodes

5.2 Simulated demand shock in the middle of each episode

In this experiment, we initially model the demand sequence with the same setup as before, superimposing a uniform sequence of 2 units with a Poisson process with an expected value of 2 units. In order to evaluate the adaptability of A2C Agents, we increase the length of each episode to 5000 time steps. In the middle of an episode (at the 2500-th time-step), we alter the Poisson part of the demand sequence so that it now has a mean of 3 units. This change emulates a surge in consumer interest or a shift in market trend. Correspondingly, we expect to observe a reflection in the demand level plot where both its mean and variance become amplified from the moment the shock is introduced.

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 suppliers. On the other hand, TBS has poor control over the inventory and is unable to recover the frequent negative balance. This is because the heuristic approach is deterministic on the part of the regular orders. Even with max-level orders from the express supplier we can only capture the local fluctuations and can't categorically adapt to the newly imposed trend due to the demand shock.

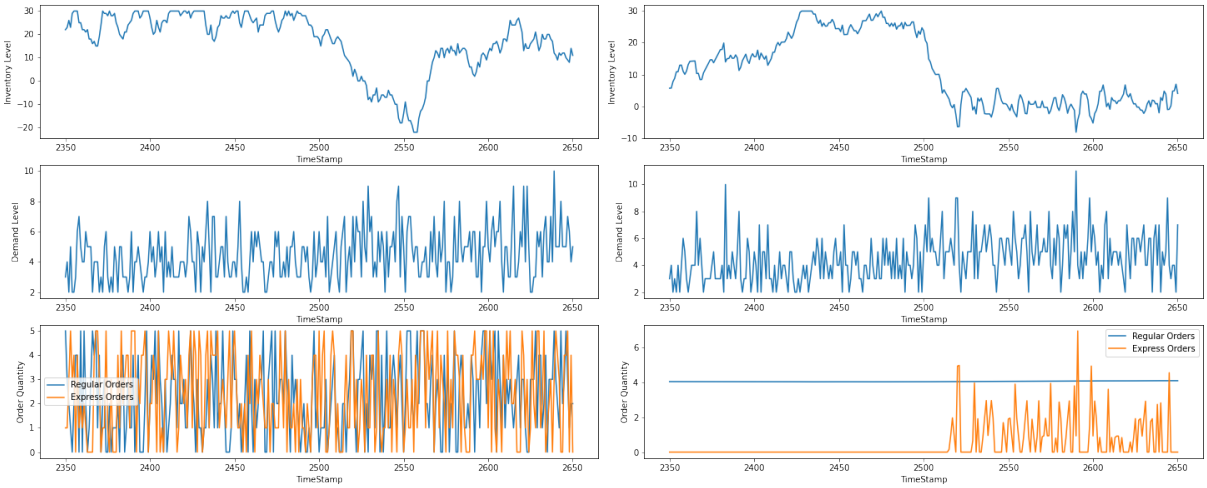


Figure 2: Response of A2C Agent(left) and TBS(right) to simulated demand shock

5.3 Real-Life Dataset

We aim to evaluate our model using historical demand data. We have two things in mind when searching for it: first, we need at least a few thousands time steps of record, for a single product, and from only a few, ideally two, suppliers; second, we need demand data, not sales data, as sales is always upper-bounded by the on-hand inventory while demand can exceed inventory. We settle on a manufacturing company[8], group the rows by the product type, and choose Product_1938 which has 929 rows of demand records from each of the two selected suppliers.

When the dual sourcing game is set up with historic demand data from a real-life global manufacturing company, the utilization of A2C Agent exhibits an erratic pattern of inventory level, consistently fluctuating around zero inventory. On the other hand, given the significant variance of the demand sequence, the TBS policy at 90% service level persistently sustains a safety stock substantially above the mean demand.

It is noteworthy that during periods of unpredictable spikes in the demand—for example, at timestamps 200-250 and 420-470—the A2C Agent exhibits volatile behavior(Fig 3). Its inventory levels ricochet between extremes, creating an unstable situation. Conversely, the 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.

Furthermore, when an abrupt surge in demand follows a period of constantly low demand, as can be observed around timestamp 620, the A2C Agent demonstrates a lack of resilience. It fails to recover, resulting in a drastic reduction in inventory that endures for 25 timestamps.In stark contrast, the TBS policy demonstrates a robust recovery, reestablishing its standard position at its regular pace.

The discrepancy between the A2C Agent and the TBS policy is further shown at timestamp 800. When faced with the dual challenge of the highest demand(40, 10 higher than second highest demand level) and largest demand increase, the A2C Agent's limitations become more apparent. Meanwhile, the TBS policy retains its capacity to recover, albeit at a slightly slower rate, highlighting the comparative stability of its policy.

Overall, although TBS deploys a more conservative strategy and A2C a riskier one, the performance of the two policies, as measured by average daily cost, is comparable, with the A2C Agent at 50.41 and the TBS policy at 47.38(Table 2). This result shows that TBS's buffering strategy archives stability at the expense of potential overstocking issues, whereas A2C's greater risk of stock-outs is mitigated by its quick adaptability under most circumstances.

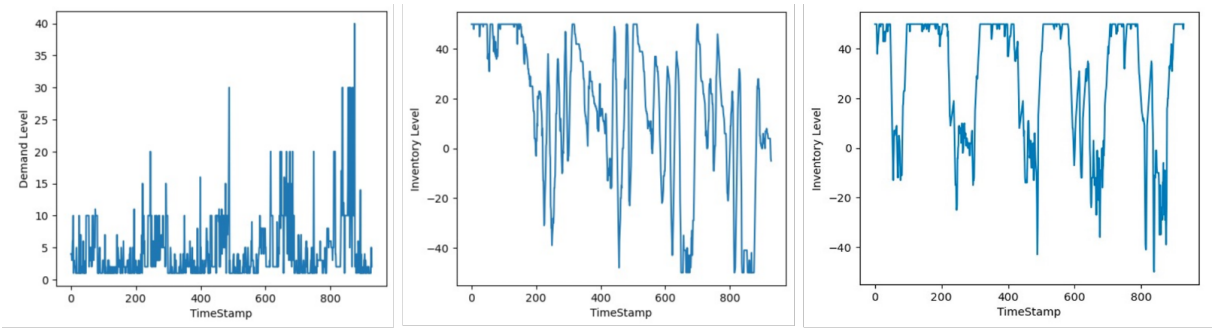


Figure 3: Demand Sequence(Left), Inventory Level of A2C (Middle), Inventory Level of TBS (Right)

	Avg Inv	Stdev Inv	Avg Inv Surplus	Avg Inv Deficit	Avg Cost
A2C	14.88	29.21	21.58	6.69	50.41
TBS	28.50	25.16	30.96	2.47	47.38

Table 2: Performance of A2C Agent and TBS Policy with Real-Life Demand for Manufacturing Company

A detailed examination of timestamps 650-700 in Figure 4 reveals a distinct divergence in the strategies employed by the A2C and TBS systems when confronted with a sudden surge in demand.

Initially, the A2C Agent attempts to restore inventory levels by uniformly augmenting both its regular and express orders by six units each—a tactic previously deployed, for instance, during a similar demand surge at timestamp 475. Intriguingly, it refrains from further increasing its orders for an additional 30 timestamps, even as demand intensifies, waiting instead until the demand begins to revert to its customary level.

Conversely, the TBS system responds more dynamically to the sudden demand increase. It promptly escalates its express order volume to a significant 15 units, while maintaining its regular orders unchanged. This swift and decisive action allows the TBS system to rapidly recover its safety stock, demonstrating agility in responding to demand fluctuations.

This analysis underscores the contrasting approaches of the A2C and TBS systems in managing unexpected demand shifts, with TBS demonstrating a more adaptive and robust response to such market challenges.

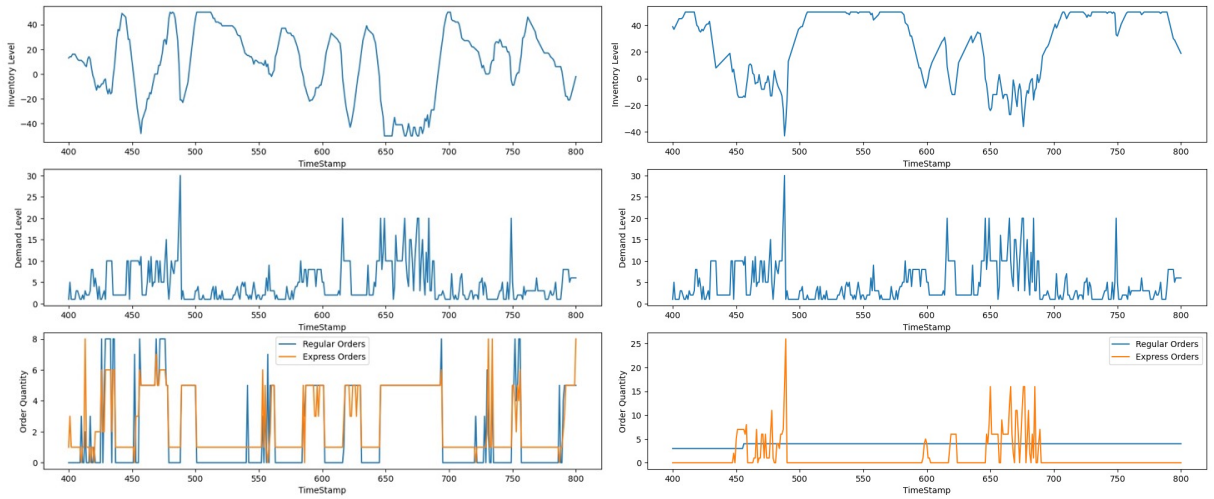


Figure 4: Response of A2C Agent(left) and TBS(right) to real-life demand surge

6 Future Work

In order to focus our study on the impact of simulated/real-world demand shocks on the behaviors and performance of the RL agent, we made a few assumptions and restrictions on the model which are worthy of further investigation. First, it will be interesting to explore the possibility of stochastic lead times, which can reflect the supplier reliability. In practical terms, the parameter can be measurement of our confidence in the delivery of orders from each supplier. Secondly, one can also further explore the problem of multi-echelon inventory management and see how different levels of the supply chain can encounter different challenges and adopt various strategies to combat uncertainty. Lastly, it would also yield important insights to consider the supplier's dynamic pricing strategies, as long-established relationship with one supplier might change its pricing, and the discount factors associated with bulk orders are also very important in real-world businesses.

7 References

1. Joren Gijsbrechts, Robert N. Boute, Jan A. Van Mieghem, Dennis J. Zhang (2022) Can Deep Reinforcement Learning Improve Inventory Management? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems. Manufacturing & Service Operations Management 24(3):1349-1368. https://doi.org/10.1287/msom.2021.1064

2. Chen, Wenbo, and Huixiao Yang. "A heuristic based on quadratic approximation for dual sourcing problem with general lead times and supply capacity uncertainty." IISE Transactions 51.9 (2019): 943-956.

3. Sultana, Nazneen N., et al. "Reinforcement learning for multi-product multi-node inventory management in supply chains." arXiv preprint arXiv:2006.04037 (2020).

4. Kosasih, Edward Elson, and Alexandra Brintrup. "Reinforcement learning provides a flexible approach for realistic supply chain safety stock optimisation." IFAC-PapersOnLine 55.10 (2022): 1539-1544.

5. Li, S., and Wu, Y. "Optimal Dual-Sourcing Strategy under Supply Disruption Risk: A Value-at-Risk Approach." *International Journal of Production Economics*, vol. 162, 2015, pp. 156-167.

6. Atashgahi, S., & Zanjirani Farahani, R. "A Simulation-Based Approach for Dual Sourcing in Supply Chain Risk Management." *International Journal of Production Economics*, vol. 211, 2019, p. 107476. doi: 10.1016/j.ijpe.2019.107476.

7. Tsao, Y. C., Liao, C. J., & Hsieh, M. H. "Coordination Mechanisms for Dual Sourcing with Capacity Constraints." *International Journal of Production Economics*, vol. 144, no. 1, 2013, pp. 153-163.

8. Felixzhao. (n.d.). Product Demand Forecasting. Kaggle. Retrieved July 27, 2023, from