Performance of Deep Reinforcement Learning Algorithms in Two-Echelon Inventory Control Systems

Francesco Stranieri[†‡], Fabio Stella[†] and Chaaben Kouki[§]

[†]*Department of Informatics, Systems, and Communication, University of Milan-Bicocca, Italy*
[‡]*Department of Control and Computer Engineering, Polytechnic University of Turin, Italy*
[§]*Department of Operations Management and Decision Science, ESSCA School of Management, France*

Introduction

This study addresses a *supply chain network optimization problem*, focusing on a two-echelon inventory system subject to capacity constraints. The goal is to determine the optimal quantity of products to produce and ship, minimizing total costs. Obtaining an optimal solution is challenging due to the *curse of dimensionality*.

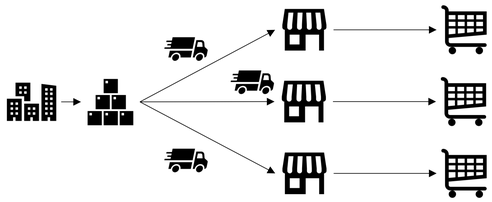

Fig. 1: A two-echelon inventory system composed of a central warehouse (first echelon) and three local warehouses (second echelon). Shopping carts represent customers' demands.

This study aims to evaluate the effectiveness of Deep Reinforcement Learning (DRL) algorithms by comparing their performance against traditional static inventory policies. This comparison will highlight the relative strengths of each approach, providing valuable insights to *decision-makers*.

Inventory Control System

For each product type $i \in \{1, \dots, I\}$, the factory determines its production level $a^i_{0,t}$, with a unit production cost of p^i_0. At each time step $t \in \{0, \dots, T\}$, $a^i_{j,t}$ units are shipped to local warehouse $j \in \{1, \dots, J\}$, which faces a stochastic demand d^ij, t. Products shipped to local warehouse j are received after a lead time L_j, incurring a unit transportation cost of zj^i. Each warehouse has a maximum capacity of c^i_j ($\sum_{i=1}^{I} c^i_0 = c_0$), a storage cost of h^i_j per unit, and a stock level at time t of q^ij, t.

Parameter	Explanation	Parameter	Explanation
I	Number of Product Types	p^i_0	Production Cost (per unit)
J	Number of Local Warehouses	z^i_j	Transportation Cost (per unit)
T	Episode Duration (time step)	$q^i_{j,t}$	Stock Level (units)
L_j	Transportation Lead Times (time step)	c^i_j	Storage Capacity (units)
$d^i_{j,t}$	Demand (units)	h^i_j	Storage Cost (per unit)
$a^i_{j,t}$	Production and Shipping Level (units)	b^i_j	Backorder Cost (per unit)

Tab. 1: The system parameters with their corresponding explanations (and units of measure). All these parameters are integrated and customizable into our open-source library, available on `https://github.com/frenkowski/SCIMAI-Gym`.

Products are non-perishable and supplied in discrete quantities. Any unsatisfied orders are back-ordered. Our study differentiates itself from existing literature [1] by incorporating *positive lead times* and considering *multiple product types*.

Main Contributions

We implement a continuous *action space* (i.e., the neural network directly generates the action value), denoted by $\mathbf{a}_t$, with an independent upper bound for each value, $0 \leq \sum_{i=1}^{I} a^i_{j,t} \leq c_j$. Additionally, our proposed balanced *allocation rule* ensures a fair distribution of products among local warehouses while maintaining non-negative stock at the central warehouse.

Algorithm 1 Our proposed allocation rule for continuous action spaces.

```
for all i ∈ {1, ..., I} do
    if Σ_{j=1}^{J} a^i_{j,t} > q^i_{j,t} then
        while Σ_{j=1}^{J} a^i_{j,t} > q^i_{j,t} do
            x ← U(1, J)
            if a^i_{x,t} > 0 then
                a^i_{x,t} ← a^i_{x,t} − 1
```

The *state vector* includes all current stocks (on-hand and ordered but not yet delivered) and the last τ demand values (excluding d_t) and is denoted as:

$$(\mathbf{s}_t, \mathbf{d}_{1,t}, \mathbf{d}_{1,t} \dots, \mathbf{d}_{J,t}) .$$

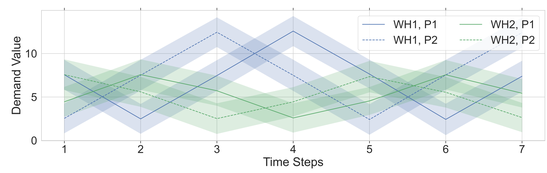

Fig. 2: An instance of the demand behavior considering two product types and two local warehouses.

Finally, we design the *reward function* for time step t as:

$$\text{(R)} \quad \max \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{i=1}^{I} -[h^i_j(q^i_{j,t-1} + x^i_{j,t-L_j} - d^i_{j,t})^+ + b^i_j(d^i_{j,t} - q^i_{j,t-1} - x^i_{j,t-L_j})^+ + z^i_j a^i_{j,t}] - \sum_{t=1}^{T} \sum_{i=1}^{I} [h^i_0 q^i_{0,t} + p^i_0 a^i_{0,t}]$$

The first term captures the inventory cost; the second term represents the backorder cost, while the third denotes the transportation cost; the final two terms account for the inventory and production costs at the central warehouse.

References

[1] Yimo Yan et al. "Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities". In: *Transportation Research Part E: Logistics and Transportation Review* 162 (2022), p. 102712.

[2] Francesco Stranieri and Fabio Stella. "A deep reinforcement learning approach to supply chain inventory management". In: *arXiv: 2204.09603* (2022).

Results

We conducted a set of numerical experiments to benchmark the performances of state-of-the-art *DRL algorithms* (i.e., PPO, PG, and A3C) against *static policies* (i.e., base-stock and (s, Q)-policies) [2]. We also implemented an *oracle* with perfect information (PI) on demand and an agent that knows the expected value of perfect information (EVPI) about demand. We performed 250 different simulations for each of the twelve experiments (by varying $I = \{1, 2\}$, $J = \{1, 2, 3\}$, and $L = \{1, 3\}$), evaluating seven demand realizations and reporting the *PI gap* achieved.

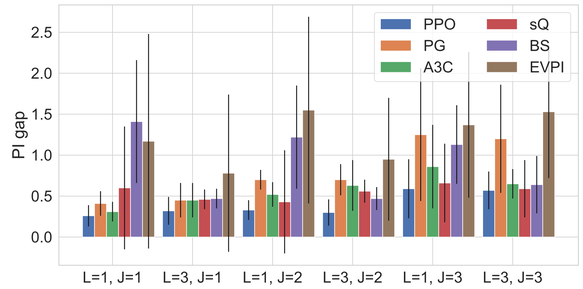

Fig. 3: With a single product type and lead times set to one, A3C outperforms PG, and the (s, Q)-policy demonstrates superior performance compared to the base-stock policy. When extending lead times to three, a degree of equilibrium is observed.

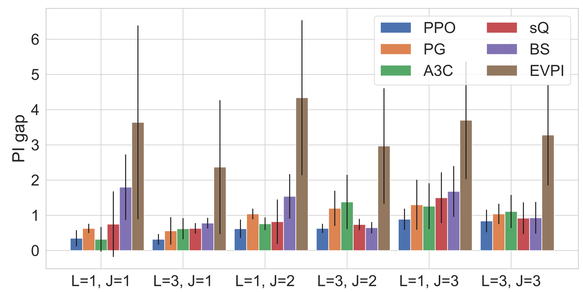

Fig. 4: With two product types and lead times set to one, DRL algorithms show slightly improved performance. When lead times are extended to three, static policies mildly outperform A3C and PG.

Our results indicate that *PPO consistently outperformed other algorithms* across all conducted experiments. The (s, Q)-policy performs commendably, emerging as the second-best choice. DRL algorithms also outperform EVPI, suggesting that complexities within the problem cannot be adequately addressed simply by considering the average demand value. Decision-makers should carefully consider these factors when selecting an *appropriate inventory management approach*.