Reinforcement Learning Lab

Module: Finite-Horizon Inventory Optimization for Seasonal Demand

Author: Charalampos (Babis) Giannakis
Affiliation: MSc Business Analytics – Computational Intelligence, VU Amsterdam
Version: v1.0 **Project ID:** RL-2025-01
Date: November 30, 2025
Contact: github.com/CharalamposGiannakis

We start from structure. Complexity follows.

1 Introduction

This assignment concerns solving a finite-horizon dynamic programming problem and verifying the resulting optimal policy through simulation. The objective is to compute the value function and extract the optimal action in each state and time step, and then validate the policy empirically. The implementation was carried out in Python, and the report was written in LaTeX. All figures and results presented here are directly generated from the code, ensuring full reproducibility.

2 Problem

Horizon $T = 150$. Initial stock $x_1 = 5$.
In each period t, demand $D_t \in \{0, 1\}$ with $\mathbb{P}(D_t = 1) = p_t = \frac{t}{150}$

Action $a_t \in \{0, 1\}$ (order one item or not). Delivery is *immediate before demand* and succeeds with probability $\mathbb{P}(Y_t = 1 \mid a_t) = 0.5\, a_t$, where $Y_t \in \{0, 1\}$ is the arrival. Profit per sale is 1. Holding cost is 0.1 per unit of end-of-period inventory. No backorders are allowed, and any inventory remaining at the end of period T has no salvage value.

State and action. The state variable is the inventory level at the start of period t, denoted x_t. Since the initial stock is 5: $x_t \in \{0, 1, \ldots, x_{\max}\}$

In principle, x_t can grow as high as $5 + 150$ if an item arrives in every period. However, the *expected* maximum total number of successful deliveries remaining at time t is $0.5(T - t + 1)$. Thus a conservative, time-dependent upper bound is: $\bar{x}(t) = 5 + 0.5(T - t + 1)$

The largest such value occurs at $t = 1$, giving $\bar{x}(1) = 80$.
Therefore, we restrict the state space to: $x \in \mathcal{X} = \{0, 1, \ldots, 80\}$

This reduces computation while retaining all states the optimal policy would realistically enter. We later verify in simulation that inventory rarely exceeds 5–6 units, confirming the cap is more than sufficient.

The decision each period is whether to attempt to order one unit or not. Because only one unit may be ordered at a time and delivery is uncertain, the action set is binary:

$$\mathcal{A} = \{0, 1\},$$

where $a_t = 1$ means placing an order, and $a_t = 0$ means not ordering.

Within-period dynamics. Given x_t and $a_t \in \{0, 1\}$, a replenishment arrives *before* demand with probability 0.5 if $a_t = 1$ (otherwise never). Let $x'_t = x_t + \mathbf{1}\{\text{arrival}\}$ (capped at 80). Demand occurs with probability $p_t = t/150$. Sales are $S_t = \min\{x'_t, \mathbf{1}\{\text{demand}\}\}$ and $x_{t+1} = x'_t - S_t$. The one-period reward is $r_t = S_t - 0.1\, x_{t+1}$.

3 Method (finite-horizon DP)

Let $V_t(x)$ be the maximal expected cumulative reward from periods $t, \ldots, T$ when starting with x units at t. Terminal condition:

$$V_{T+1}(x) = 0, \quad \forall x \geq 0 \quad \text{because leftover inventory has no salvage value}$$

$$V_t(x) = \max_{a \in \{0,1\}} \mathbb{E}[r_t + V_{t+1}(x_{t+1})], \quad t = 1, \ldots, T.$$

For implementation, we enumerate the four events $(Y_t, D_t) \in \{0, 1\}^2$. For $x \geq 1$:

$$x'_t = x + Y_t, \;\; S_t = \begin{cases} 1, & D_t = 1 \\ 0, & D_t = 0 \end{cases} \quad \Rightarrow \quad x_{t+1} = \begin{cases} x + Y_t - 1, & D_t = 1 \\ x + Y_t, & D_t = 0 \end{cases}.$$

For $x = 0$ the sale occurs only if $x'_t = 1$. This yields a closed-form expected value per (t, x, a) by summing the four probabilities:

$$\mathbb{P}(Y_t = 1) = 0.5\, a, \qquad \mathbb{P}(D_t = 1) = p_t = \tfrac{t}{150}.$$

We compute $V_t(x)$ backward for $t = T, \ldots, 1$ and store the optimal action

$$\pi_t(x) \in \arg\max_{a \in \{0,1\}} \mathbb{E}\big[r_t + V_{t+1}(x_{t+1})\big].$$

Ties in max are broken in favor of $a = 0$.

Complexity is $O(T\, x_{\max}\, |\mathcal{A}|)$; here $|\mathcal{A}| = 2$.

4 Results

The value function was computed backward from $t = T$ to $t = 1$. The optimal expected cumulative reward starting with $x_1 = 5$ units is:

$$V_1(5) = 33.62.$$

Figure 1 shows the optimal policy $\pi(t, x)$, where black corresponds to ordering and white to not ordering. We observe that ordering is only optimal when the stock is low and demand probability is high.

The optimal policy shows a clear structure:

- Early in the horizon (low demand probability), it is optimal not to order, as expected demand is low and holding costs dominate.
- Late in the horizon (high demand probability), it becomes optimal to order only when the inventory is low.
- Inventory stays in a narrow band near $x \approx 1$–5.

5 Simulation Validation

To validate the computed policy, we simulate the inventory process for 10,000 independent sample paths. The empirical mean total reward is:

$$\hat{\mu} = 33.64, \qquad 95\%\ \text{CI} = [33.51,\ 33.77].$$

Since $V_1(5) = 33.62$ lies well within this interval, the simulation confirms that the dynamic programming solution is correctly implemented.

Furthermore, the simulated inventory trajectories almost never exceed $x = 6$, which empirically validates the chosen state-space truncation $x_{\max} = 80$. The cap is therefore computationally efficient yet does not alter optimal decisions.

Reward distribution. Figure 2 shows the histogram where the distribution of total rewards across simulations is approximately symmetric and bell-shaped, centered near 33.6. This illustrates that the optimal policy behaves consistently under stochastic arrivals and demand, with variability arising only from randomness and not from policy instability.

6 Conclusion

The optimal policy keeps inventory low and only orders when demand is likely and stock is low. The system never builds large inventory because holding cost punishes it, and demand uncertainty early makes ordering unattractive.

Appendix: Plots

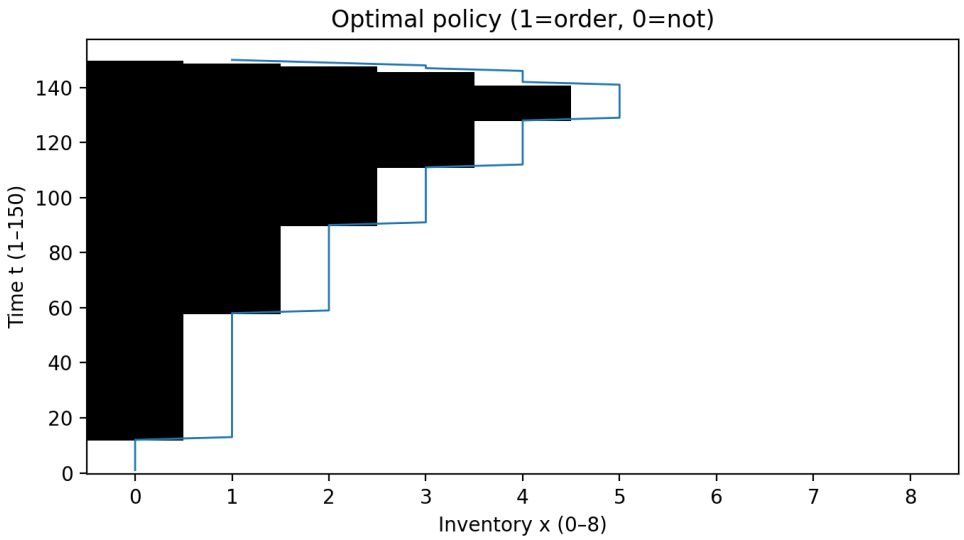

Figure 1: Optimal policy $\pi_t(x)$ (black = order, white = no order).

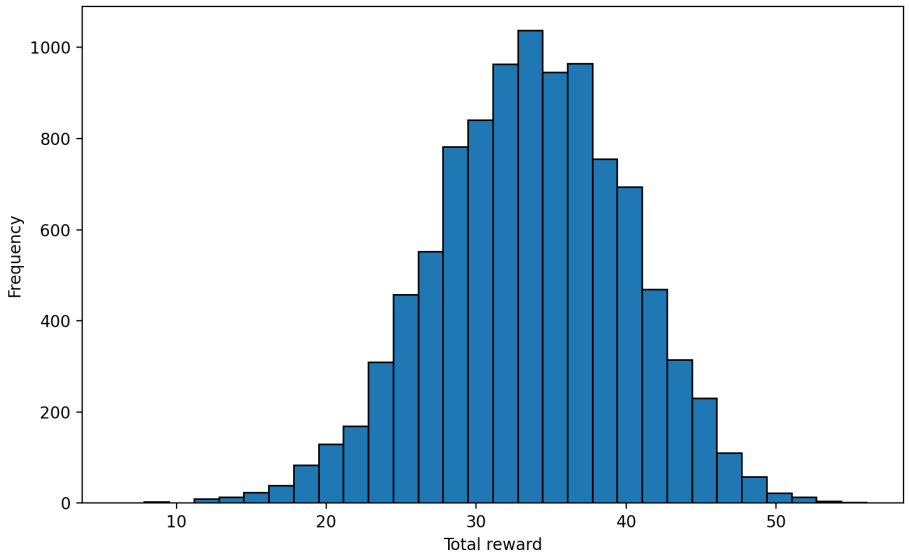

Figure 2: Distribution of total rewards over 10,000 simulation runs under the optimal policy.