Multi-Echelon Supply Chain Writeup

Sai Madhukiran Kompalli

1 Problem Description

This environment simulates a five-layer supply chain made up of one final *market*, one or more *retailers*, a set of *regional distributors*, *producers*, and *raw-material distributors*. During each discrete period $t = 1, \ldots, T$ the controller chooses a **continuous reorder quantity** $R_{(i,j),t}$ along every material route $(i \to j)$. Orders enter a deterministic pipeline, advance one slot per period, and are delivered to the downstream node after the route's lead time. Retailer inventory then meets stochastic market demand; unmet demand is stored as backlog. The environment updates all inventories and pipeline positions, realises revenues and costs, adds quadratic penalties for any bound violations, and returns the net profit as the scalar reward [1].

1.1 States S_t

At every decision epoch the state comprises the following elements:

1. **On-hand inventory** $I_{i,t}$—stock physically present at each inventory-holding node $i \in \{\text{retailers}, \text{distributors}, \text{producers}\}$.
2. **Pipeline inventory** $T^{(k)}_{(i,j),t}$—for every reordering route $(i \to j)$ and every lag position $k = 1, \ldots, \ell_{(i,j)}$ (slot k=1 is the most recent order).
3. **Sales and backlog** for each retailer-to-market link $(r \to 0)$:
   - sales realised this period $S_{(r,0),t}$;
   - backlog carried into next period $B_{(r,0),t}$.
4. **Demand** $D_{(r,0),t}$ realised at each retailer route.
5. **Time index** t.

Written compactly: $S_t = (I_t,\ T_t,\ S_t,\ B_t,\ D_t,\ t)$.

1.2 Parameters Information

- *Lead times* $\ell_{(i,j)}$ for every route.
- *Capacities* : inventory capacity inv_capacity_i and route capacity $\text{reordering\_route\_capacity}_{(i,j)}$.
- *Cost and price data* : inventory holding cost c_i^{hold}; pipeline holding cost $c_{(i,j)}^{\text{pipe}}$; operating cost c_i^{op} and yield y_i for producers; selling price $p_{(i,j)}$; backlog penalty $c_{(r,0)}^{\text{backlog}}$.

- *Demand process*: each retailer route draws $D_{(r,0),t}$ from a user-configured distribution (default: independent normal with mean, std, and seed supplied).
- Penalty constants P (large constant) and D (quadratic scaling), and a small numeric threshold ε for treating tiny orders as zero.

1.3 Actions A_t

The agent outputs a flat vector with one element per reordering route. Internally the environment

1. maps each raw number in $[-1, 1]$ to the physical interval $[0, \text{capacity}_{(i,j)}]$,
2. sets values with magnitude below ε to zero,
3. applies the bound-checking logic to see if actions are out of bounds

1.4 Action-Correction and Penalty Logic

- **Negative order** $(R < 0)$: set to 0 and add $P + D\,(|R|)^2$ to the penalty cost.
- **Over-capacity order** $(R > \text{cap})$: clip to capacity and add $P + D\,(R - \text{cap})^2$.
- The same quadratic-plus-constant formula is used for any state value that violates its observation bounds (e.g. inventory above storage capacity).

1.5 Transition Function $S_t \to S_{t+1}$

Given the corrected orders $R_{(i,j),t}$:

1. **Insert orders into pipeline.** Each $R_{(i,j),t}$ enters slot 1 of the pipeline array $T^{(1)}_{(i,j),t+1}$.
2. **Advance in-transit material.** All earlier slots shift one step; material in slot $\ell_{(i,j)}$ arrives at node j as usable inventory.
3. **Update on-hand inventory.** $I_{j,t+1} = I_{j,t} + \text{arrivals} - \text{outflow}$. Producer nodes also incur operating costs and yield factors.
4. **Realise and satisfy demand.** Draw $D_{(r,0),t}$, fulfil from $I_{r,t+1}$; record sales $S_{(r,0),t}$ and backlog $B_{(r,0),t+1}$.
5. **Compute period profit** (Sect. 1.6) and set reward.
6. **Build next observation**, clip-and-penalise if needed.
7. **Advance time** $t \leftarrow t + 1$; terminate when $t > T$.

1.6 Cost and Reward

Let holding$_t$, operating$_t$, pipeline$_t$, backlog$_t$, revenue$_t$ be the five monetary terms defined in the code. Quadratic penalties accumulated during the step form penalty$_t$.

$$\text{Profit}_t = \text{revenue}_t - \text{holding}_t - \text{operating}_t - \text{pipeline}_t - \text{backlog}_t - \text{penalty}_t, \qquad \text{Reward}_t = \text{Profit}_t.$$

1.7 7. Episode Termination

The episode ends when the period counter exceeds the horizon, i.e. when $t > T$. No additional early-stopping conditions are implemented in the current environment.

References

[1] Hector D Perez, Christian D Hubbs, Can Li, and Ignacio E Grossmann. Algorithmic approaches to inventory management optimization. *Processes*, 9(1):102, 2021.