Comparing Deep Reinforcement Learning Algorithms in Two-Echelon Supply Chains

Francesco Stranieri[1,2] [0000−0002−5366−8499] and
Fabio Stella[1] [0000−0002−1394−0507]

[1] University of Milan-Bicocca, Milan MI 20125, Italy
[2] Polytechnic of Turin, Turin TO 10129, Italy
francesco.stranieri@polito.it

Abstract. In this study, we analyze and compare the performance of state-of-the-art deep reinforcement learning algorithms for solving the supply chain inventory management problem. This complex sequential decision-making problem consists of determining the optimal quantity of products to be produced and shipped across different warehouses over a given time horizon. In particular, we present a mathematical formulation of a two-echelon supply chain environment with stochastic and seasonal demand, which allows managing an arbitrary number of warehouses and product types. Through a rich set of numerical experiments, we compare the performance of different deep reinforcement learning algorithms under various supply chain structures, topologies, demands, capacities, and costs. The results of the experimental plan indicate that deep reinforcement learning algorithms outperform traditional inventory management strategies, such as the static (s, Q)-policy. Furthermore, this study provides detailed insight into the design and development of an open-source software library that provides a customizable environment for solving the supply chain inventory management problem using a wide range of data-driven approaches.

Keywords: artificial intelligence · deep learning · reinforcement learning · smart manufacturing · inventory management.

Supplementary Material

A Experimental Plan

Table 1. Experiments concerning the three scenarios considered: (a) for the 1P1W scenario, when two values are present, the first one refers to the factory, while the second one refers to the first (and only) distribution warehouse.; (b) for the 1P3W scenario, when there are four values, the first relates to the factory, while the remaining to the first, the second, and the third distribution warehouse, respectively; (c) referring to the round brackets of the 2P2W scenario, the first value denotes the first product type, whereas the second value indicates the second product type.

(a)

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Max Demand Value	10	5	5	10	5
Max Demand Variation	2	2	2	1	3
Sale Price	15	20	15	20	15
Production Cost	5	5	10	5	5
Storage Capacities	5, 10	5, 10	5, 10	10, 15	5, 10
Storage Costs	2, 1	2, 1	2, 1	4, 2	1, 2
Transportation Cost	0.25	0.05	1	0.25	0.25
Penalty Coefficient	1.5	0.1	2	1.5	0.1

(b)

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Max Demand Value	7	5	5	7	5
Max Demand Variation	2	2	2	1	3
Sale Price	15	20	15	20	15
Production Cost	5	5	10	5	5
Storage Capacities	3, 6, 9, 12	3, 6, 9, 12	3, 6, 9, 12	4, 8, 12, 16	4, 8, 12, 16
Storage Costs	4, 3, 2, 1	4, 3, 2, 1	4, 3, 2, 1	8, 6, 4, 2	4, 3, 2, 1
Transportation Costs	0.3, 0.6, 0.9	0.03, 0.06, 0.09	3, 2, 1	0.3, 0.6, 0.9	0.3, 0.6, 0.9
Penalty Coefficient	1.5	0.1	2	1.5	0.1

(c)

	Exp 1	Exp 2	Exp 3
Max Demand Values	3, 6	3, 6	4, 2
Max Demand Variations	2, 1	2, 1	2, 2
Sale Prices	20, 10	10, 15	20, 10
Production Costs	2, 1	2, 1	2, 1
Storage Capacities	(3, 4), (6, 8), (9, 12)	(3, 4), (6, 8), (9, 12)	(9, 4), (6, 8), (3, 12)
Storage Costs	(6, 3), (4, 2), (2, 1)	(0.5, 0.3), (1.0, 0.6), (1.5, 0.9)	(1, 3), (2, 2), (3, 1)
Transportation Costs	(0.1, 0.3), (0.2, 0.6)	(0.01, 0.025), (0.02, 0.050)	(0.1, 0.3), (0.2, 0.6)
Penalty Coefficient	0.5	1.5	0.5

B Hyperparameters Tuning

Table 2. The hyperparameters of DRL algorithms selected for tuning. Through a grid search, we instantiated 880 DRL algorithm instances for the 1P1W scenario, 880 for the 1P3W, and 528 for the 2P2W, for a total of 2 288 instances. Each instance is trained for a given number of episodes: 15 000 episodes for the 1P1W scenario and 50 000 for the 1P3W and 2P2W scenarios.

	A3C	PPO	VPG
Hidden Layers	{(64, 64), (128, 128)}	{(64, 64), (128, 128)}	{(64, 64), (128, 128)}
Learning Rate	{1e-4, 1e-3}	{5e-4, 5e-3}	{4e-4, 4e-3}
Rollout Fragment Length	{10, 100}	{20, 200}	{10, 100}
Train Batch Size	{200, 2000}	{400, 4000}	{200, 2000}
Grad Clip	{20, 40}	{0, 20}	-
SGD Mini-Batch Size	-	{128, 256}	-
SGD Iterations	-	{15, 30}	-