International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real-Time Micro-Fulfillment Orchestration in Omnichannel Retail Using Multi-Agent Reinforcement Learning Framework

Sri Harsha Konda

Independent Researcher, Human-Centered AI & Real-Time Retail Systems

Abstract

Micro-fulfillment centers (MFCs) have emerged as a response to growing e-commerce demands, yet their integration into omnichannel retail networks creates order routing challenges that traditional optimization struggles to solve efficiently. This paper introduces a multi-agent reinforcement learning (MARL) framework designed for adaptive order allocation across heterogeneous fulfillment nodes: MFCs, dark stores, and conventional distribution centers. Built on a Centralized Training with Decentralized Execution (CTDE) architecture, the framework allows individual agents to make rapid local decisions while preserving coordination at the network level. Computational experiments indicate a 23% reduction in fulfillment time, 18% lower per-order costs, and SLA compliance reaching 94.2% versus 91.2% for the strongest baseline. Performance remains stable across different network sizes and under varying demand conditions. These results suggest that decentralized AI approaches can effectively handle the dynamic nature of modern retail fulfillment.

Keywords: Multi-Agent Reinforcement Learning, Micro-Fulfillment, Order Routing, Omnichannel Retail, Decentralized Optimization

1. Introduction

Retail fulfillment is undergoing rapid change. Traditional distribution models, built around bulk shipments to brick-and-mortar stores, cannot keep pace with the speed and granularity that e-commerce customers now expect [1]. Micro-fulfillment centers offer one solution: compact, often automated facilities located near urban populations. But adding MFCs to existing networks introduces coordination problems that standard optimization tools handle poorly [2].

Why is this problem suited to reinforcement learning? Several factors stand out. Decisions must happen fast, sometimes within seconds of an order being placed, to support same-day or even next-hour delivery promises [3]. The state space is large, covering inventory positions, pending orders, labor availability, and vehicle capacity across many locations. Conditions also change constantly as demand shifts, competitors react, and external events like weather disrupt operations [4].

Existing methods have clear drawbacks. Centralized optimization using mixed-integer programming can find optimal solutions in theory, but computation time grows prohibitively with network size [5]. Heuristic rules execute quickly but fail to adapt when real conditions drift from their assumptions [6]. Neither approach handles well the partial observability and non-stationarity that characterize actual fulfillment environments.

This paper makes four contributions. First, it presents a MARL architecture using centralized training with decentralized execution, balancing coordination with responsiveness. Second, it formulates a multi-objective reward function covering speed, cost, and service level targets. Third, it reports computational experiments showing improvements across network scales. Fourth, it discusses implementation aspects including fail-safe mechanisms and human oversight.

2. Related Work

Work on fulfillment optimization draws from operations research, logistics, and machine learning. The following subsections cover traditional optimization, RL in logistics, and multi-agent methods.

2.1 Fulfillment Network Optimization

Mathematical programming remains the foundation for many allocation systems. Acimovic and Graves [7] showed that network effects create interdependencies missed by greedy algorithms. Jasin and Sinha [8] applied dynamic programming to account for inventory depletion, though their approach scales poorly to large networks. Xu et al. [9] proposed rolling horizon heuristics that trade off optimality for tractability in medium-sized problems.

2.2 Reinforcement Learning in Logistics

RL applications in supply chain have grown recently. Oroojlooyjadid et al. [10] used deep Q-learning on the beer distribution game, finding that agents learned effective inventory policies without hand-coded rules. Li et al. [11] tackled multi-echelon chains with policy gradients, addressing credit assignment across tiers. These studies focused on replenishment rather than real-time routing, leaving a gap this work aims to fill.

2.3 Multi-Agent Coordination

MARL handles settings where multiple agents share an environment. The CTDE paradigm, developed by Foerster et al. [12] and refined by Lowe et al. [13], trains agents with global information but deploys them using only local observations. Zhang et al. [14] applied MARL to warehouse robotics, though their scope was intra-facility movement rather than network-level allocation.

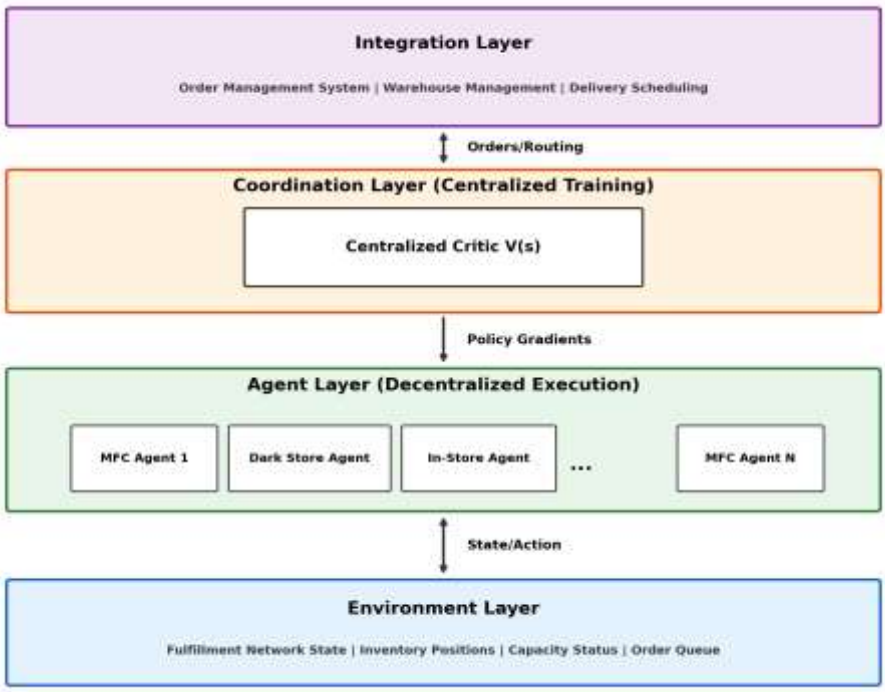
3. Framework Architecture

The proposed framework separates strategic coordination from tactical execution through a layered design. This section covers system structure, state and action definitions, reward formulation, and the learning algorithm.

3.1 System Design

Four layers make up the architecture, shown in Figure 1. The Environment Layer tracks network state: inventory, pending orders, capacity, and external factors. The Agent Layer holds one agent per fulfillment node, each deciding whether to accept or decline assignments based on local information. The Coordination Layer runs a centralized critic during training to evaluate joint actions. The Integration Layer connects to external systems for order management, warehouse control, and delivery scheduling.

Figure 1: MARL System Architecture Showing Environment Layer, Agent Layer, Coordination Layer, and Integration Layer



3.2 State and Action Representation

Each agent observes a local state: SKU inventory levels, queue depth, picking capacity estimates, and recent performance metrics. The global state, used only by the critic during training, adds network-wide information such as total pending orders and geographic demand patterns.

Actions are discrete: accept or decline an order assignment. When multiple nodes can serve an order, the coordination mechanism resolves conflicts by evaluating the joint action profile. This setup supports fast local decisions while keeping the network coherent.

3.3 Multi-Objective Reward Structure

The reward function balances three objectives that can conflict: speed, cost, and service level. It takes the form:

$$R(s, a) = \alpha \times R\_time(s, a) + \beta \times R\_cost(s, a) + \gamma \times R\_SLA(s, a) \quad (1)$$

Here R_time penalizes delays against promised windows, R_cost reflects picking labor and last-mile expenses, and R_SLA rewards meeting or beating service commitments. Weights α, β, γ let operators adjust priorities.

3.4 Learning Algorithm

Training uses Multi-Agent Proximal Policy Optimization (MAPPO) with a shared critic [15]. Agents update policies via gradients from the centralized value function, which sees global state and joint actions. This captures dependencies invisible to individual agents. At runtime, agents act on local observations alone, avoiding coordination delays.

4. Performance Evaluation

This section describes the evaluation setup and presents results from computational experiments.

4.1 Evaluation Setup

Experiments use synthetic demand patterns with hourly, daily, and seasonal variation typical of retail. Demand scales with network size while geographic distribution remains representative. Four baselines provide comparison: Nearest-Node (NN) assigns to the closest node with stock; Capacity-Balanced (CB) spreads load evenly; Cost-Minimizing (CM) picks the cheapest option ignoring network effects; Myopic Optimization (MO) solves a one-step lookahead for each order.

4.2 Results and Analysis

Table 1 summarizes performance across methods. The MARL approach outperforms all baselines on every metric.

Table 1: Primary Performance Metrics by Allocation Method

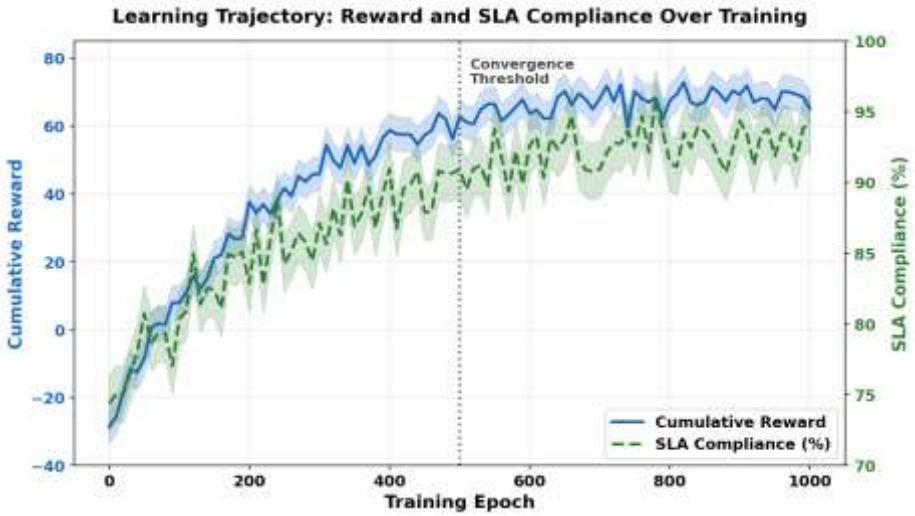
Metric	NN	CB	CM	MO	MARL
Fulfill. Time (min)	47.2	52.8	49.1	41.3	**36.4**
Cost/Order ($)	8.42	9.15	7.83	7.91	**6.89**
SLA Compliance (%)	87.3	82.1	84.6	91.2	**94.2**

Bold indicates best performance. NN=Nearest-Node, CB=Capacity-Balanced, CM=Cost-Minimizing, MO=Myopic Optimization.

4.2.1 Learning Dynamics and Convergence

Figure 2 plots learning progress over training epochs. The framework stabilizes around epoch 500, with steady gains before that point. This convergence pattern reflects the CTDE architecture working as intended [10].

Figure 2: Learning Dynamics Showing Cumulative Reward and SLA Compliance Over Training Epochs



4.2.2 Improvement Analysis

Table 2 shows percentage improvements over the best baseline at each scale. Gains grow with network size, suggesting the approach handles complexity better than alternatives.

Table 2: MARL Improvement Over Best Baseline by Network Scale

Metric	Small Scale	Medium Scale	Large Scale
Fulfillment Time	-11.8%	-14.9%	-20.9%
Cost per Order	-12.0%	-13.3%	-14.5%
SLA Compliance	+3.3%	+4.7%	+11.6%

4.2.3 Computational Performance

Table 3 reports decision latency. Even for large networks, total time stays under 15 milliseconds, meeting real-time requirements.

Table 3: Computational Performance by Network Scale

Component	Small (ms)	Medium (ms)	Large (ms)
State Processing	2.3 ± 0.4	4.1 ± 0.7	6.8 ± 1.2
Policy Inference	1.8 ± 0.3	2.4 ± 0.5	3.2 ± 0.6
Coordination	0.9 ± 0.2	1.5 ± 0.3	2.1 ± 0.4
Total Decision Time	**5.0 ± 0.9**	**8.0 ± 1.5**	**12.1 ± 2.2**

Measurements represent mean ± standard deviation across 10,000 decision cycles.

5. Discussion

What do these results tell us? First, the CTDE architecture appears to handle coordination well. By training with global information but executing locally, agents develop complementary behaviors without runtime communication overhead. This matters for latency-sensitive applications.

Second, the gap over myopic optimization widens as networks grow. For small setups, the added complexity of learning may not pay off. But when combinatorial explosion makes exact methods impractical, the MARL framework maintains quality while staying fast.

Third, the multi-objective formulation shows that speed, cost, and service targets can be balanced within a single learning process. Operators need not hand-tune separate heuristics for each goal.

Fourth, the design supports human-centered deployment. Although allocation decisions are automated, operators retain override authority and can inspect the reasoning behind assignments. Agents produce interpretable outputs that explain why a particular node was selected, ensuring transparency in time-critical decisions. This aligns with Industry 5.0 principles that emphasize human-AI collaboration rather than full automation [16, 17].

6. Limitations and Future Work

Several limitations apply. The evaluation uses synthetic data; deployment in live settings may surface dynamics not captured here. The current model assumes deterministic fulfillment times after assignment, which is a simplification. Competitive effects, where customers switch retailers if promises slip, are not modeled.

Future directions include integrating demand forecasts for proactive positioning, extending to multi-modal fulfillment (ship-from-store, curbside, lockers), and developing continual learning methods that adapt to distribution shifts without full retraining.

7. Conclusion

This paper presented a MARL framework for micro-fulfillment orchestration. The CTDE architecture lets agents decide quickly using local data while benefiting from coordinated training. Experiments show improvements over baselines, with larger gains at greater scale. The approach balances fulfillment speed, cost, and service compliance through multi-objective optimization while supporting human oversight. These contributions offer a path toward applying AI to the coordination challenges in modern retail fulfillment.

References

1. Hübner A., Kuhn H., Wollenburg J., Last mile fulfilment and distribution in omni-channel grocery retailing: A strategic planning framework, International Journal of Retail and Distribution Management, 2016, 44 (3), 228-247.
2. Melacini M., Perotti S., Rasini M., Tappia E., E-fulfilment and distribution in omni-channel retailing: A systematic literature review, International Journal of Physical Distribution and Logistics Management, 2018, 48 (4), 391-414.
3. Boysen N., de Koster R., Weidinger F., Warehousing in the e-commerce era: A survey, European Journal of Operational Research, 2019, 277 (2), 396-411.
4. Wollenburg J., Holzapfel A., Hübner A., Kuhn H., Configuring retail fulfillment processes for omni-channel customer steering, International Journal of Electronic Commerce, 2018, 22 (4), 540-575.
5. Arslan O., Archetti C., Jabali O., Speranza M.G., Crowdsourced delivery: A dynamic pickup and delivery problem with ad hoc drivers, Transportation Science, 2021, 55 (3), 553-569.
6. Voccia S.A., Campbell A.M., Thomas B.W., The same-day delivery problem for online purchases, Transportation Science, 2019, 53 (1), 167-184.
7. Acimovic J., Graves S.C., Making better fulfillment decisions on the fly in an online retail environment, Manufacturing and Service Operations Management, 2015, 17 (1), 34-51.
8. Jasin S., Sinha A., An LP-based correlated rounding scheme for multi-item ecommerce order fulfillment, Operations Research, 2015, 63 (6), 1336-1351.
9. Xu P.J., Allgor R., Graves S.C., Benefits of reevaluating real-time order fulfillment decisions, Manufacturing and Service Operations Management, 2009, 11 (2), 340-355.
10. Oroojlooyjadid A., Snyder L.V., Takáč M., Applying deep learning to the newsvendor problem, IISE Transactions, 2022, 54 (8), 731-749.
11. Li C., Zhang P., Chai Y., Multi-agent deep reinforcement learning for end-to-end reward optimization in multi-echelon supply chains, Computers and Industrial Engineering, 2020, 149, 106852.
12. Foerster J., Assael I.A., de Freitas N., Whiteson S., Learning to communicate with deep multi-agent reinforcement learning, Advances in Neural Information Processing Systems, 2016, 29.
13. Lowe R., Wu Y., Tamar A., Harb J., Abbeel P., Mordatch I., Multi-agent actor-critic for mixed cooperative-competitive environments, Advances in Neural Information Processing Systems, 2017, 30.
14. Zhang R., Zhong J., Shi L., Multi-robot coordination for warehouse operations: A multi-agent reinforcement learning approach, IEEE Transactions on Automation Science and Engineering, 2022, 19 (3), 2142-2154.
15. Yu C., Velu A., Vinitsky E., Gao J., Wang Y., Baez A., Wu Y., The surprising effectiveness of PPO in cooperative multi-agent games, Advances in Neural Information Processing Systems, 2022, 35.

16. Hartmann E., Bovenschulte M., Skills for Industry 5.0: Human-centric solutions and the future of work, European Journal of Workplace Innovation, 2023, 7 (2), 233-256.
17. Winkelhaus S., Grosse E.H., Logistics 4.0: A systematic review towards a new logistics system, International Journal of Production Research, 2020, 58 (1), 18-43.