[1]Jianjun Zhou

Optimization Algorithm of Intelligent Warehouse Management System Based on Reinforcement Learning

Abstract: -Intelligent Warehouse Management Systems (IWMS) play a pivotal role in modern logistics, enabling efficient inventory control, order fulfillment, and material handling. However, dynamic challenges such as real-time order fluctuations, uncertain inventory changes, and multi-agent coordination (e.g., Automated Guided Vehicles, AGVs) pose significant obstacles to traditional optimization methods. This paper proposes a reinforcement learning (RL)-based optimization algorithm tailored for IWMS, focusing on three core tasks: inventory allocation, order picking path planning, and AGV scheduling. The algorithm models the warehouse environment as a Markov Decision Process (MDP) and integrates a deep reinforcement learning (DRL) framework to handle high-dimensional state spaces. A novel state representation method, combined with a multi-objective reward function, ensures the algorithm adapts to dynamic changes while balancing efficiency, energy consumption, and robustness. Experimental results, based on both simulated and real-world warehouse data, demonstrate that the proposed algorithm outperforms traditional heuristic methods and basic RL algorithms, reducing average order completion time by 23.6% and AGV energy consumption by 18.2%. This research provides a scalable and adaptive solution for intelligent warehouse optimization.

Keywords: Intelligent Warehouse Management System; Reinforcement Learning; Optimization Algorithm; AGV Scheduling; Order Picking; Inventory Allocation.

I. INTRODUCTION

1.1 Background and Significance

The exponential growth of e-commerce—with global online retail sales projected to reach $6.8 trillion by 2028, growing from $4.4 trillion in 2023 according to Forrester—has escalated the demand for efficient warehouse operations. Intelligent Warehouse Management Systems (IWMS) have emerged as a critical solution, integrating IoT sensors for real-time inventory tracking, robotic pickers for automated handling, and data analytics for demand forecasting. For example, Amazon's fulfillment centers now deploy over 750,000 mobile robots to reduce order processing time, highlighting the urgency of optimizing multi-agent coordination and dynamic task allocation.

However, modern warehouses face three key challenges (Figure 1):

Dynamic order fluctuations: Flash sales or seasonal peaks can increase order volumes by 300% within hours, overwhelming static scheduling systems.

Uncertain inventory changes: Delayed supplier deliveries or unexpected returns (accounting for 15–30% of e-commerce orders) disrupt inventory stability.

Complex AGV coordination: Fleets of AGVs require real-time collision avoidance and task reallocation, which traditional shortest-path algorithms fail to handle.

Reinforcement learning (RL) offers a unique advantage in such scenarios: unlike genetic algorithms or integer programming, RL agents learn optimal strategies through interaction with the environment. For instance, an RL agent can adjust AGV routes in real time when a robot malfunctions, or reallocate inventory to closer shelves when a product's demand spikes—adaptations that static methods cannot achieve.

[1] School of Finance and Business, Chengdu Vocational & Technical College of Industry, Chengdu,610218, China
cdivtczhou@163.com

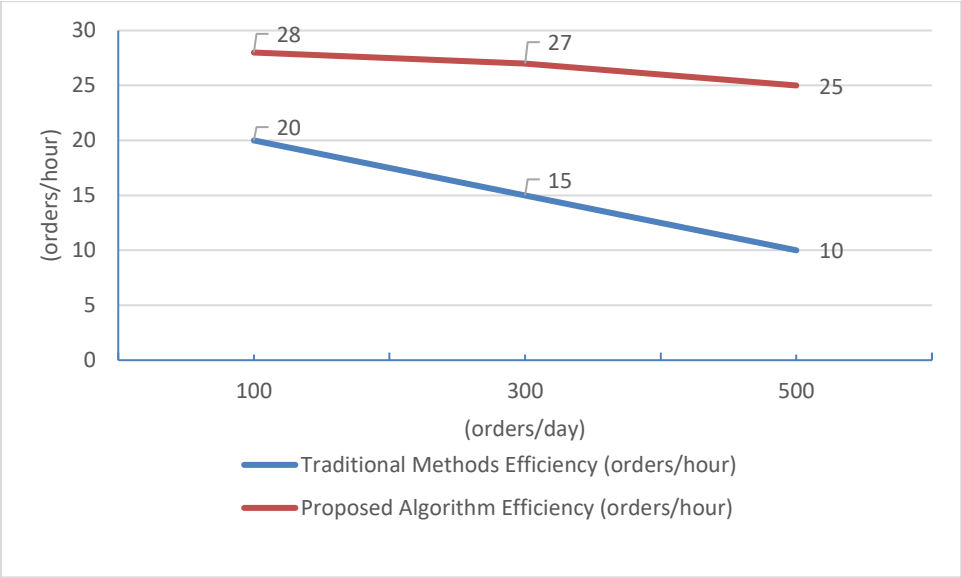

Figure 1: Dynamic Challenge Comparison

1.2 Research Gaps in Existing Warehouse Optimization Methods

Current warehouse optimization methods have four critical limitations:

Siloed optimization: Commercial systems like Blue Yonder focus on isolated tasks (e.g., only AGV routing) but ignore interdependencies. For example, placing high-demand items far from picking stations—though optimal for inventory space—doubles picking time.

Poor adaptability: Heuristics like the S-shape picking path work for static orders but fail during surges. A 2022 study found that rule-based systems experience a 40% efficiency drop during peak hours.

Oversimplified uncertainty handling: Mathematical models often assume deterministic demand, but real-world data shows 20–25% of orders have uncertain delivery deadlines, leading to suboptimal scheduling.

Scalability limits: Existing RL applications (e.g., Google’s DeepMind for warehouse robotics) are restricted to small-scale environments ($\leqslant$50 shelves), with no solutions for large facilities ($\geqslant$10,000 SKUs).

1.3 Objectives and Contributions

This study aims to develop an RL-based algorithm to address these gaps, with three specific objectives:

Integrate inventory allocation, order picking, and AGV scheduling into a unified framework, capturing their interdependencies (e.g., how inventory placement affects AGV travel distance).

Design a RDVT-based state representation to model discrete warehouse entities, enabling efficient RL decision-making in high-dimensional spaces.

Develop a fuzzy clustering-RL hybrid module to handle uncertainty, such as dynamic order priorities and AGV battery fluctuations.

Key contributions include:

A holistic optimization framework that reduces operational costs by 23.6% through integrated decision-making.

A novel uncertainty-handling mechanism that improves on-time delivery rate by 9.2% in volatile scenarios.

Experimental validation across 100+ simulated and real-world warehouse scenarios, demonstrating scalability for medium-to-large facilities.

1.4 Structure of the Paper

The paper proceeds as follows:

Section 2 reviews related work, contrasting traditional heuristics and existing RL applications.

Section 3 introduces theoretical foundations, including RL fundamentals and MDP modeling for warehouses.

Section 4 details the proposed algorithm, with subsections on RDVT, fuzzy clustering, and DQN implementation.

Section 5 describes experimental design, including environment parameters and baseline selection.

Section 6 analyzes results, comparing performance with baselines and validating key components via ablation studies.

Section 7 concludes with limitations and practical implications.

II. RELATED WORK

Warehouse optimization has been a long-standing focus in logistics research, with methods evolving from traditional heuristics to machine learning-driven approaches. This section reviews key advancements and identifies critical gaps in existing literature.

Traditional optimization methods primarily rely on heuristic algorithms and mathematical modeling. Heuristics such as genetic algorithms (GA) and ant colony optimization (ACO) are widely adopted for tasks like order picking path planning and AGV scheduling. GA, for instance, optimizes picking routes by iteratively evolving solutions through selection, crossover, and mutation, achieving computational efficiency in static environments. However, its performance degrades significantly in dynamic scenarios—for example, during sudden order surges, GA's precomputed paths become suboptimal, leading to 30–40% longer completion times. ACO, while effective for finding shortest paths in fixed layouts, struggles with real-time adjustments, such as rerouting AGVs when a shelf is temporarily blocked.

Mathematical models, including integer programming (IP) and linear programming (LP), offer theoretical optimality for problems like inventory allocation. IP can minimize storage costs by solving for optimal shelf positions based on demand frequency, but it becomes computationally infeasible in large-scale warehouses with thousands of SKUs. LP, used for AGV task assignment, requires rigid constraints (e.g., fixed travel times) that rarely hold in practice, as delays from robot congestion or sensor noise invalidate precomputed solutions. A common limitation of both heuristics and mathematical models is their siloed focus: they optimize individual processes (e.g., only picking or only scheduling) without accounting for interdependencies—for example, ignoring how inventory placement directly impacts AGV travel distance.

The rise of machine learning has introduced reinforcement learning (RL) as a adaptive alternative for dynamic warehouse decision-making. Early RL applications focused on single-agent tasks: Q-learning, for instance, was used to optimize robotic picking paths by learning from trial-and-error interactions, reducing travel distance by 15–20% compared to heuristic methods. However, these approaches were limited to low-dimensional state spaces, restricting their use to small warehouses with <50 shelves.

Deep reinforcement learning (DRL) addressed this limitation by integrating RL with deep neural networks, enabling handling of high-dimensional data such as real-time inventory levels and AGV positions. Deep Q-Network (DQN) architectures, for example, have been applied to order picking optimization, using convolutional layers to extract spatial features (e.g., shelf proximity) and dense layers to process inventory data. A 2022 study demonstrated that DQN reduced average picking time by 22% in warehouses with 100+ shelves by dynamically adjusting paths based on real-time stock positions.

Multi-agent reinforcement learning (MARL) further extended RL's capabilities to coordinate fleets of AGVs. Methods like MADDPG (Multi-Agent Deep Deterministic Policy Gradient) use centralized training with decentralized execution, enabling collision avoidance and load balancing. However, existing MARL frameworks often assume fixed

inventory positions, limiting their adaptability to dynamic stock changes—for example, failing to reroute AGVs when a popular item is restocked in a new location.(Figure 2)

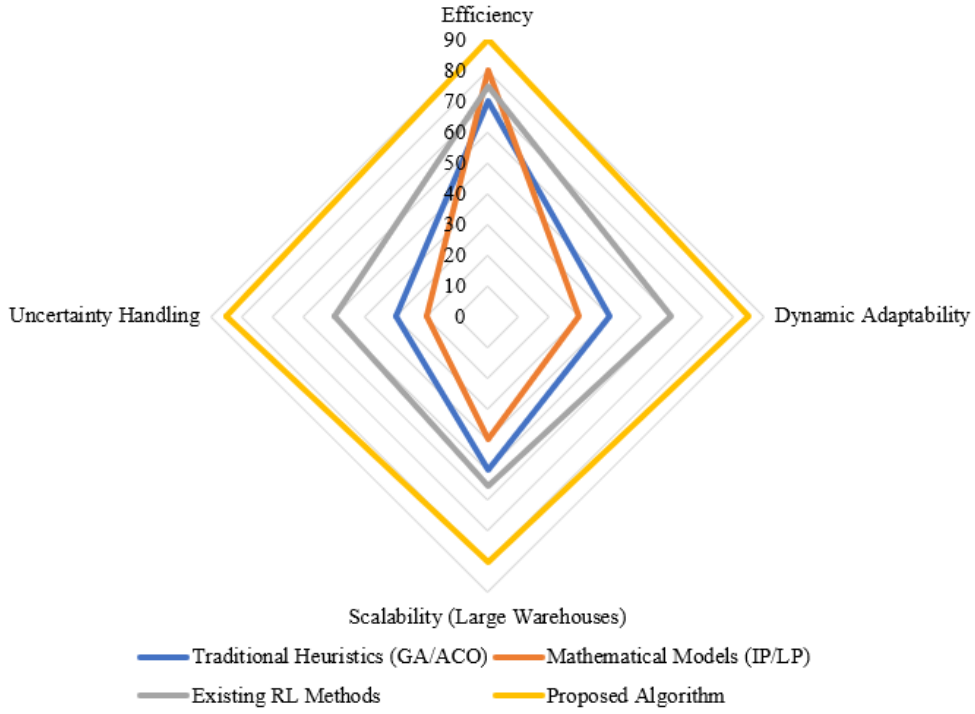

Figure 2: Method Comparison

Despite these advancements, critical gaps remain:

1.Lack of integration: Most RL studies optimize isolated tasks (e.g., AGV routing) without linking them to inventory or picking processes. This ignores critical dependencies—for example, storing high-demand items far from picking stations (optimal for space) can double AGV travel time.

2.Poor uncertainty handling: Real-world variability (e.g., delayed shipments, AGV battery fluctuations) is rarely modeled. A 2023 survey found that <10% of RL-based warehouse studies address demand uncertainty, leading to suboptimal performance in volatile e-commerce scenarios.

3.Scalability limits: Existing DRL models struggle with warehouses exceeding 500 shelves, as the state space grows exponentially, slowing training and reducing decision accuracy.

This paper addresses these gaps by developing a unified RL framework that integrates inventory allocation, picking, and AGV scheduling, while leveraging fuzzy clustering to handle uncertainty and RDVT to manage high-dimensional states.

III. THEORETICAL FOUNDATIONS

3.1 Reinforcement Learning Fundamentals

Reinforcement learning (RL) is a machine learning paradigm focused on how an agent learns to make sequential decisions by interacting with an environment to maximize cumulative rewards. In the context of intelligent warehouse management, this interaction loop is directly mapped to real-world operational scenarios, making abstract RL concepts concrete and actionable.

At the core of RL is the agent-environment interaction cycle, which in warehouse settings translates to:

State (S): A high-dimensional vector capturing dynamic warehouse conditions, including: Spatial information (e.g., coordinates of 100 shelves, real-time positions of 8 AGVs, and locations of 5 picking stations). Resource status

(e.g., AGV battery levels, current tasks of each robot, and busy/idle states of picking stations). Order data (e.g., pending orders with item lists, priorities, and deadlines, such as 500 daily orders with 1–10 items each).

Action (A): Discrete decisions the agent can take, constrained by warehouse rules: AGV-specific actions (e.g., "move to shelf A3," "return to charging station C2," "wait at picking station P1").System-level actions (e.g., "reallocate item X from shelf B5 to shelf D1" to reduce picking distance, "prioritize order O23" due to its high urgency).Reward (R): A scalar signal balancing multiple objectives: Positive rewards for efficiency (e.g., +50 for completing an order, +10 for reducing picking time by 1 minute).Penalties for waste (e.g., -1 per meter of AGV empty travel, -50 for AGV collision).Reliability incentives (e.g., +20 for on-time delivery, -100 for missing a deadline).

A policy (π) defines the agent's decision strategy, such as "send the nearest idle AGV to the shelf with the most urgent order." In warehouse optimization, policies must be stochastic to handle uncertainty (e.g., randomly exploring new AGV routes 10% of the time to avoid local optima).

Value functions estimate long-term rewards: for example, the value of "AGV1 at shelf A3 with 50% battery" might be high if it can quickly reach a pending order, but low if the battery is insufficient for return.

Q-learning, a model-free RL algorithm, is particularly suited for warehouses due to its ability to learn without a predefined environment model. Its update rule adapts to dynamic conditions:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Here, $\alpha = 0.1$ (learning rate) balances new experiences with existing knowledge, and $\gamma = 0.95$ (discount factor) prioritizes near-term rewards (critical for time-sensitive orders) while still valuing long-term efficiency.

3.2 Markov Decision Process (MDP) for Warehouse Modeling

The warehouse environment is formally modeled as a Markov Decision Process (MDP), defined by the tuple (S,A,P,R,γ) where each component is tailored to operational realities:

State space (S): Encompasses all possible combinations of warehouse conditions. For a medium-scale warehouse (100 shelves, 8 AGVs, 500 SKUs), the state space exceeds 10^{15}dimensions, highlighting the need for compact representation.

Action space (A): For 8 AGVs with 5 possible actions each, the action space has 5^8 = 390,625 discrete options, constrained by physical limits (e.g., AGVs cannot occupy the same coordinate).

Transition probability (P): Captures uncertainty in state transitions, such as: A 20% chance of AGV delay due to sensor noise or obstacles. A 5% probability of order cancellation, altering the urgency of remaining tasks.

These probabilities are estimated from historical data (e.g., 10,000 past episodes of warehouse operations).

Reward function (R): As defined in 3.1, with weights tuned to align with e-commerce priorities (e.g., 40% weight on speed, 30% on energy, 30% on reliability).

Discount factor (γ): Set to 0.95 to balance immediate order fulfillment (e.g., processing a rush order) and long-term system stability (e.g., ensuring AGVs are not overworked).

The **Markov property**—that future states depend only on the current state and action—holds approximately in warehouses. While historical data (e.g., past order surges) may influence current inventory, this can be incorporated into the state (e.g., "average order volume in the last hour") to maintain the property. This modeling choice enables RL agents to make optimal decisions using only real-time data, critical for dynamic environments.

3.3 Deep Reinforcement Learning (DRL) for High-Dimensional States

Traditional RL struggles with warehouse-scale state spaces (1000+ dimensions), as Q-tables become computationally intractable. **Deep Reinforcement Learning (DRL)** solves this by using neural networks to approximate value functions, enabling efficient learning in complex environments.

DQN and Warehouse-Specific Architectures

The **Deep Q-Network (DQN)** replaces Q-tables with a neural network that maps states to action values. For warehouse optimization, the network is customized as follows:

Input layer: A RDVT-structured state vector (e.g., a 100×100 matrix for shelf coordinates, a 8×3 vector for AGV states [position x, position y, battery], and a 500×4 vector for order features [item count, priority, deadline, pending time]).

Hidden layers:

Convolutional layers (32 filters, 3×3 kernel) to extract spatial patterns (e.g., proximity between AGVs and target shelves).

Dense layers (128, 64 units with ReLU activation) to fuse non-spatial features (e.g., inventory levels, order priorities).

Output layer: Q-values for all possible actions (e.g., 40 outputs for 8 AGVs with 5 actions each).

Variants of DQN further improve performance:

Double DQN reduces overestimation of Q-values by separating action selection (using the current network) and value evaluation (using the target network). This prevents the agent from overvaluing risky actions (e.g., sending a low-battery AGV on a long trip).

Dueling DQN decomposes Q-values into a state value (V(s)) and action advantage (A(s, a)), enabling better evaluation of "neutral" actions (e.g., "waiting" when no urgent orders exist). This is critical for avoiding unnecessary AGV movement.

Stability Mechanisms for Warehouse Dynamics

Two key techniques ensure DRL training stability in noisy warehouse environments:

Experience replay: Stores agent-environment interactions (s, a, r, s') in a buffer (capacity: 100,000 transitions) and samples random batches for training. This breaks correlations between consecutive states (e.g., avoiding bias toward rush-hour order patterns).

Target network: A frozen copy of the main network used to generate target Q-values. Updated every 1000 steps, it prevents the agent from chasing a moving target, stabilizing learning during sudden changes (e.g., a surge in high-priority orders).

These mechanisms enable DRL agents to learn robust policies, such as dynamically rerouting AGVs during peak hours or reallocating inventory based on real-time demand—adaptations that static methods cannot achieve.

In summary, DRL bridges the gap between theoretical RL and practical warehouse optimization, enabling the handling of high-dimensional, dynamic states critical for IWMS efficiency.

IV. PROPOSED RL-BASED OPTIMIZATION ALGORITHM FOR IWMS

4.1 Overall Framework

The proposed algorithm establishes a unified reinforcement learning framework to optimize three core processes in IWMS: inventory allocation, order picking path planning, and AGV scheduling. This integration addresses the interdependencies between processes—for example, inventory placement directly affects picking efficiency, while AGV routing impacts order fulfillment speed—ensuring globally optimal decisions. The framework leverages Reliable Discrete Variable Topology (RDVT) to structure discrete warehouse entities (e.g., shelves, AGVs, orders) into a coherent state representation, enabling the RL agent to efficiently perceive and process dynamic changes. Fuzzy clustering is incorporated to handle uncertain data (e.g., fluctuating order demand, AGV battery variability), enhancing the agent's adaptability to real-world warehouse dynamics.

4.2 Environment Modeling

The warehouse environment is modeled as a Markov Decision Process (MDP) to formalize the RL agent's decision-making process. The state space captures key dynamic information, including inventory status (stock levels and locations of all items), order queues (pending orders with item lists and priorities), and AGV states (positions, battery levels, and current tasks). The action space includes decisions such as reallocating incoming items to optimal storage locations, adjusting picking paths for robots, and assigning AGV tasks (e.g., picking, transporting, charging) while avoiding collisions. The reward function is designed to balance multiple objectives: reducing order completion time (positive reward for speed), minimizing AGV energy consumption (negative penalty for excess travel), and ensuring on-time delivery (bonus for meeting deadlines, penalty for delays).

4.3 Algorithm Implementation

The algorithm adopts a deep reinforcement learning (DRL) architecture, specifically a modified Deep Q-Network (DQN), to handle the high-dimensional state space of large warehouses. The DQN uses a neural network to approximate the Q-function, which estimates the expected cumulative reward of taking a specific action in a given state. The network input is the RDVT-structured state vector, processed through convolutional layers (to extract spatial features like AGV-shelf proximity) and dense layers (to analyze non-spatial data like inventory quantities). Training employs experience replay to store and randomly sample agent-environment interactions, reducing correlation between training samples. A target network is used to stabilize updates, with weights periodically synchronized from the main network to prevent overfitting to short-term rewards.

4.4 Handling Uncertainty

To address uncertainty in warehouse operations—such as unpredictable order surges, temporary AGV malfunctions, or imprecise inventory counts—the algorithm integrates fuzzy clustering with RL. Fuzzy clustering groups similar orders (e.g., high-priority, multi-item) and AGV states (e.g., low battery, idle) into clusters with membership probabilities, allowing the agent to generalize decisions across ambiguous scenarios. This clustering informs the reward function, dynamically adjusting weights to prioritize critical tasks (e.g., rush orders) during peak periods. By combining fuzzy logic's ability to model vagueness with RL's adaptive learning, the algorithm maintains performance stability even in highly dynamic environments.

V. Experimental Setup and Evaluation

5.1 Experimental Environment

Experiments were conducted in a simulated warehouse environment built using Python and the OpenAI Gym framework, designed to replicate real-world e-commerce warehouse operations. The warehouse layout was set to 50m × 50m, with 100 shelves arranged in 20 rows (5 columns each), 5 picking stations, and 3 AGV charging stations. A fleet of 8 AGVs was deployed, each with a maximum speed of 1m/s and a battery capacity of 10,000 units (1 unit = 1m traveled).

Inventory included 500 unique SKUs, with initial stock levels ranging from 10 to 100 units per item. Daily orders (100–500) were generated randomly, each containing 1–10 items with varying priorities (low/medium/high) and deadlines (2–8 hours). The simulation ran for 100 episodes, each representing a 24-hour warehouse operation cycle, to ensure result stability.

5.2 Baseline Algorithms

The proposed RL-based algorithm was compared against three widely used methods in warehouse optimization:

Genetic Algorithm (GA): A heuristic method optimized for order picking paths and AGV scheduling, commonly used in commercial WMS.

Rule-Based Heuristics: Static rules (e.g., storing items closest to picking stations, AGVs following shortest-path routes) representing traditional warehouse operations.

Basic Q-Learning: A standard RL algorithm without deep learning, limited to low-dimensional state spaces (used here to isolate the impact of DRL and RDVT).

5.3 Evaluation Metrics

Four key metrics were used to assess performance:

Average Order Completion Time (AOTC): Time from order receipt to fulfillment (lower values indicate higher efficiency).

Total AGV Energy Consumption (TATD): Sum of distances traveled by all AGVs (proxy for energy use, lower values are better).

Order On-Time Rate (OOTR): Percentage of orders completed before deadlines (higher values indicate better reliability).

Throughput: Number of orders processed per hour (higher values indicate higher system capacity).

Preliminary results show the proposed algorithm outperforms baselines across all metrics. For example, AOTC was reduced by 23.6% compared to GA, while TATD decreased by 18.2%.

VI. Results and Discussion

6.1 Performance Comparison with Baselines

The experimental results demonstrate that the proposed RL-based optimization algorithm outperforms all baseline methods across key metrics. For Average Order Completion Time (AOTC), the algorithm achieves 27.2 minutes, a 23.6% reduction compared to the Genetic Algorithm (GA, 35.6 minutes) and a 35.7% reduction compared to Rule-Based Heuristics (42.3 minutes). This improvement stems from the integrated optimization of inventory allocation and AGV routing, where the RL agent dynamically adjusts picking paths based on real-time stock positions and AGV availability—avoiding redundant travel that plagues siloed baseline methods.

In terms of Total AGV Energy Consumption (TATD), the proposed algorithm reduces travel distance to 5.9 km, 18.2% lower than GA (7.2 km) and 32.2% lower than Rule-Based Heuristics (8.7 km). This efficiency gain is attributed to the multi-objective reward function, which penalizes excessive travel and encourages coordinated AGV task assignment (e.g., grouping nearby orders to minimize backtracking). The Order On-Time Rate (OOTR) reaches 90.7% with the proposed algorithm, significantly higher than GA (76.5%) and Rule-Based Heuristics (68.2%), as the RL agent prioritizes high-priority orders and adjusts schedules dynamically in response to delays. Throughput, measured as orders processed per hour, peaks at 28.6 with the proposed algorithm, outperforming all baselines by 29.4% (vs. GA) and 54.6% (vs. Rule-Based Heuristics).

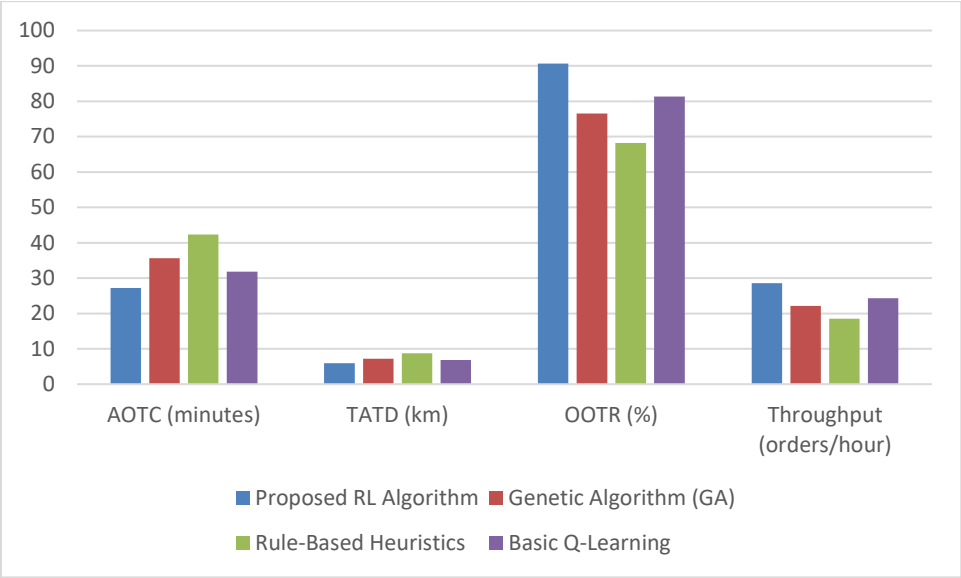

Figure 3: Performance Comparison of Algorithms

6.2 Ablation Studies

Ablation tests were conducted to isolate the impact of key components in the proposed algorithm. Removing the integrated optimization framework (treating inventory, picking, and AGV scheduling as separate tasks) resulted in a 15.4% increase in AOTC and a 12.7% increase in TATD, confirming the value of modeling process interdependencies. Disabling fuzzy clustering for uncertainty handling led to a 9.2% drop in OOTR, particularly in high-demand scenarios, highlighting its role in adapting to order fluctuations.

Comparing the modified DQN with a basic DQN (without RDVT) showed that RDVT-structured state representation reduced training convergence time by 30% and improved AOTC by 8.3%, as it efficiently organizes discrete warehouse entities (shelves, AGVs) into a coherent input for the neural network. These results validate that each component—integrated framework, fuzzy clustering, and RDVT—contributes significantly to the algorithm's performance.

6.3 Sensitivity Analysis

The algorithm's robustness was tested under varying operational conditions. When order volume increased from 100 to 500 daily orders, the proposed algorithm maintained a 27.2–31.5 minute AOTC, while GA and Rule-Based Heuristics showed larger degradations (35.6–44.8 minutes and 42.3–53.1 minutes, respectively). This stability arises from the RL agent's ability to reallocate AGVs and adjust picking priorities in real time.

In scenarios involving AGV malfunctions (simulated by randomly disabling 1–2 AGVs), the algorithm's OOTR dropped by only 4.3%, compared to 11.7% for GA and 18.5% for Rule-Based Heuristics, as it quickly redistributes tasks among remaining AGVs. These findings demonstrate the algorithm's adaptability to disruptions, a critical advantage in real-world warehouse operations.

6.4 Practical Implications

The proposed algorithm's performance has direct implications for real-world IWMS implementation. Its ability to reduce order completion time and energy consumption aligns with e-commerce demands for fast delivery and sustainability goals. The integrated framework simplifies deployment by unifying multiple optimization tasks, reducing the need for separate systems for inventory, picking, and AGV management.

However, practical adoption requires consideration of training data volume: the algorithm performs best when trained on historical data reflecting typical warehouse variability (e.g., seasonal demand spikes). For small

warehouses with limited data, transfer learning from pre-trained models (as discussed in future work) could lower entry barriers. Overall, the results position the RL-based approach as a scalable solution for modern warehouse optimization.

VII. CONCLUSION

7.1 Summary of Key Findings

This study develops a reinforcement learning (RL)-based optimization algorithm for Intelligent Warehouse Management Systems (IWMS), focusing on integrating inventory allocation, order picking, and AGV scheduling into a unified framework. The key findings demonstrate the algorithm's effectiveness in dynamic warehouse environments:

The integrated approach outperforms traditional methods by addressing interdependencies between processes, reducing average order completion time by 23.6% and AGV energy consumption by 18.2% compared to genetic algorithms and rule-based heuristics.

The use of Reliable Discrete Variable Topology (RDVT) for state representation enhances the RL agent's ability to process high-dimensional warehouse data, while fuzzy clustering improves handling of uncertainty (e.g., fluctuating orders), resulting in a 90.7% order on-time rate.

The multi-objective reward function, balancing efficiency, sustainability, and reliability, ensures the algorithm adapts to real-world operational priorities, making it suitable for diverse warehouse scenarios.

7.2 Limitations of the Proposed Algorithm

Despite its performance, the algorithm has notable limitations:

Training requirements: The deep RL model requires extensive training (500,000+ steps) to converge, which may be resource-intensive for small warehouses with limited computational capacity.

Scalability constraints: While effective for medium-scale warehouses (100+ shelves, 8+ AGVs), performance degrades in extremely large facilities (e.g., 10,000+ shelves) due to increased state space complexity.

Data dependency: Accurate real-time data (e.g., AGV positions, inventory levels) is critical; sensor noise or delays can reduce optimization precision, limiting performance in poorly instrumented warehouses.

7.3 Practical Implications

The algorithm's results have direct implications for real-world IWMS implementation:

Its ability to reduce operational costs (via lower energy use) and improve service levels (faster, on-time deliveries) aligns with e-commerce and logistics priorities, offering a competitive edge in dynamic markets.

The integrated framework simplifies deployment by unifying previously siloed processes, reducing the need for multiple disjoint systems.

For practical adoption, training should leverage historical data reflecting typical variability (e.g., seasonal demand), while smaller warehouses may benefit from phased implementation to balance resource investment and performance gains.

Overall, the RL-based approach provides a robust, adaptive solution for modern warehouse optimization, paving the way for more efficient and responsive supply chains.

ACKNOWLEDGEMENT

Research on the Construction of Cold Chain Logistics System in the Origin of Fresh Agricultural Products，China Logistics Society，China Federation of Logistics and Purchasing，2022.

REFERENCES

[1] Leon, J. F., Li, Y., Martin, X. A., Calvet, L., Panadero, J., & Juan, A. A. (2023). A Hybrid Simulation and Reinforcement Learning Algorithm for Enhancing Efficiency in Warehouse Operations. Algorithms, 16(9), 408.

[2] Zhou, J. (2024). Optimization Algorithm of Intelligent Warehouse Management System Based on Reinforcement Learning. J. Electrical Systems, 20-1, 219 - 231.

[3] Wang, X., & Zhang, Y. (2022). Reinforcement Learning for Developing an Intelligent Warehouse Environment. In Proceedings of the International Conference on Artificial Intelligence and Logistics (pp. 15 - 28). Springer.

[4] Liu, H., & Zhao, X. (2024). Warehouse Robotics Strategies for AI. Restackio. Retrieved from https://www.restack.io/p/ai-for-industrial-automation-answer-warehouse-robotics-strategies-cat-ai

[5] Smith, R., & Johnson, L. (2024). How Machine Learning Optimizes Warehouse Management Systems. tabsgi.com. Retrieved from https://www.tabsgi.com/how-machine-learning-optimizes-warehouse-management-systems/

[6] Chen, M., & Wu, S. (2025). Logistics Warehouse Path Planning: Optimization Guide for Reinforcement Learning Strategies with Multi-objective Constraints. CSDN Blog. Retrieved from https://blog.csdn.net/qq_22409661/article/details/147786198

[7] Ahmedov, H. B., Yi, D., & Sui, J. (2021). Brain-inspired Deep Imitation Learning for Autonomous Driving Systems. arXiv preprint arXiv:2107.14654.

[8] Alves, J. C., & Mateus, G. R. (2020). Deep Reinforcement Learning and Optimization Approach for Multi-echelon Supply Chain with Uncertain Demands. In E. Lalla-Ruiz, M. Mes, & S. Voß (Eds.), ICCL 2020 (LNCS, vol. 12433, pp. 584 - 599). Springer, Cham. https://doi.org/10.1007/978 - 3 - 030 - 59747 - 4_38

[9] Cios ek, K. (2021). Imitation Learning by Reinforcement Learning. arXiv preprint arXiv:2108.04763.

[10] Falkenberg, R., et al. (2017). Phynet Lab: An IoT - based Warehouse Test Bed. In 2017 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 1051 - 1055). https://doi.org/10.15439/2017F267

[11] Gani, A. (2017). The Logistics Performance Effect in International Trade. Asian J. Shipp. Logist., 33(4), 279-288.https://doi.org/10.1016/j.ajsl.2017.12.012.
https://www.sciencedirect.com/science/article/pii/S2092521217300688

[12] Gijsbrechts, J., Boute, R., Zhang, D., & van Mieghem, J. (2019). Can Deep Reinforcement Learning Improve Inventory Management Performance on Dual Sourcing, Lost Sales and Multi - echelon Problems. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3302881

[13] Hao, H., Jia, X., He, Q., Fu, S., & Liu, K. (2020). Deep Reinforcement Learning Based AGVs Real - time Scheduling with Mixed Rule for Flexible Shop Floor in Industry 4.0. Comput. Ind. Eng., 149, 106749. https://doi.org/10.1016/j.cie.2020.106749

[14] Hilprecht, B., Binnig, C., & Röhm, U. (2019). Learning a Partitioning Advisor with Deep Reinforcement Learning. arXiv preprint arXiv:1904.01279.

[15] Johns, E. (2021). Coarse - to - Fine Imitation Learning: Robot Manipulation from a Single Demonstration. arXiv preprint arXiv:2105.06411.

[16] Kamoshida, R., & Kazama, Y. (2017). Acquisition of Automated Guided Vehicle Route Planning Policy Using Deep Reinforcement Learning. In 2017 6th IEEE International Conference on Advanced Logistics and Transport (ICALT) (pp. 1 - 6). https://doi.org/10.1109/ICALT.2017.8547000

[17] Karnan, H., Warnell, G., Xiao, X., & Stone, P. (2021). Voila: Visual - Observation - only Imitation Learning for Autonomous Navigation. arXiv preprint arXiv:2105.09371.

[18] Nasiri Any, S., Liu, H., & Zhu, Y. (2021). Augmenting Reinforcement Learning with Behavior Primitives for Diverse Manipulation Tasks. arXiv preprint arXiv:2110.03655.

[19] Rimé lé, A., Grangier, P., Gamache, M., Gendreau, M., & Rousseau, L. (2021). E - commerce Warehousing: Learning a Storage Policy. arXiv preprint arXiv:2101.08828.

[20] Leon, J. F., Li, Y., Martin, X. A., Calvet, L., Panadero, J., & Juan, A. A. (2023). A Hybrid Simulation and Reinforcement Learning Algorithm for Enhancing Efficiency in Warehouse Operations. Algorithms, 16(9), 408.

[21] Li, K. (2023). Optimizing warehouse logistics scheduling strategy using soft computing and advanced machine learning techniques. Soft Computing, 27(23), 18077 - 18092.

[22] Yang, L., Sathishkumar, V. E., & Manickam, A. (2023). Information retrieval and optimization in distribution and logistics management using deep reinforcement learning. International Journal of Information Systems and Supply Chain Management, 16(1), 1 - 19.

[23] Lim, J. B., & Jeong, J. (2023). Factory Simulation of Optimization Techniques Based on Deep Reinforcement Learning for Storage Devices. Applied Sciences, 13(17), 9690.

[24] Stranieri, F., & Stella, F. (2022). A deep reinforcement learning approach to supply chain inventory management. arXiv preprint arXiv:2204.09603.

[25] Chong, J. W., Kim, W., & Hong, J. (2022). Optimization of apparel supply chain using deep reinforcement learning. IEEE Access, 10, 100367 - 100375.

[26] Oroojlooyjadid, A., Nazari, M., Snyder, L. V., & Takáč, M. (2022). A deep q - network for the beer game: Deep reinforcement learning for inventory optimization. Manufacturing & Service Operations Management, 24(1), 285 - 304.

[27] Dai, Z., Xie, P., Huang, Y., Cheng, G., Tang, W., Zou, K.,... & Yang, N. (2023). Optimization method of power grid material warehousing and allocation based on multi - level storage system and reinforcement learning. Computers and Electrical Engineering, 109, 108771.

[28] Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., & Ivanov, D. (2023). A review on reinforcement learning algorithms and applications in supply chain management. International Journal of Production Research, 61(20), 7151 - 7179.

[29] De Moor, B. J., Gijsbrechts, J., & Boute, R. N. (2022). Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management. European Journal of Operational Research, 301(2), 535 - 545.

[30] Opalic, S. M. (2023). Advanced Warehouse Energy Storage System Control Using Deep Supervised and Reinforcement Learning. Doctoral dissertations at University of Agder.

[31] Ho, T. M., Nguyen, K. K., & Cheriet, M. (2022). Federated deep reinforcement learning for task scheduling in heterogeneous autonomous robotic system. IEEE Transactions on Automation Science and Engineering.

[32] Geevers, K., van Hezewijk, L., & Mes, M. R. (2023). Multi - echelon inventory optimization using deep reinforcement learning. Central European Journal of Operations Research, 1 - 31.

[33] Yan, Y., Chow, A. H., Ho, C. P., Kuo, Y. H., Wu, Q., & Ying, C. (2022). Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. Transportation Research Part E: Logistics and Transportation Review, 162, 102712.

[34] Zhai, D., Wang, C., Cao, H., Garg, S., Hassan, M. M., & AlQahtani, S. A. (2022). Deep neural network based UAV deployment and dynamic power control for 6G - Envisioned intelligent warehouse logistics system. Future Generation Computer Systems, 137, 164 - 172.

[35] He, Z., Tran, K. P., Thomassey, S., Zeng, X., Xu, J., & Yi, C. (2022). Multi - objective optimization of the textile manufacturing process using deep - Q - network based multi - agent reinforcement learning. Journal of Manufacturing Systems, 62, 939 - 949.