Research on Path Planning of Warehouse Robots Based on Q-learning

Yuzeyuan Song *

Department of Ocean Engineering, Harbin Institute of Technology, Weihai, 264200, China

* Corresponding Author Email: yuzeyuan.song.2022@uni.strath.ac.uk

Abstract. The rapid growth of e-commerce has intensified the demand for efficient warehouse management and automation. This research focuses on applying Q-learning, a reinforcement learning algorithm, to the path planning of warehouse robots. Path planning is a critical aspect of warehouse robotics, particularly in dynamic environments where efficient navigation can significantly enhance operational productivity. This study aims to develop a Q-learning-based algorithm that enables warehouse robots to navigate their workspace effectively, minimizing the time taken to reach designated goals while avoiding obstacles. The approach involves constructing a grid world model to simulate the warehouse environment, where the robot learns to make optimal decisions based on rewards and penalties associated with reaching goals and colliding with obstacles, respectively. By modeling the warehouse as a discrete grid with obstacles and defining the robot's task cycle—from starting point to cargo pickup to delivery, and return—this study investigates how varying the learning rate (α) and discount factor (γ) affects the robot's learning efficiency and path optimization. A full-factor experiment testing 81 combinations of α and γ revealed that higher values of γ, especially when paired with appropriate α values, significantly reduce the number of steps required for task completion and decrease path-planning failures. The findings underscore the importance of parameter tuning in Q-learning algorithms to enhance the efficiency and reliability of autonomous warehouse robots.

Keywords: Q-learning; Path planning; Full-factor experiment.

1. Introduction

The exponential rise of e-commerce has fundamentally revolutionized consumer purchasing behaviors, leading to an unprecedented increase in online transactions worldwide. This surge has shifted consumer expectations, emphasizing the need for quick and reliable service, and has placed immense pressure on supply chain and logistics operations. As online retailers strive to meet these heightened expectations, efficient warehouse operations have become critical to handle the vast array of products and expedite delivery processes. Traditional manual warehouse management systems, reliant on human labor for inventory handling and order fulfillment, struggle to keep pace with these demands. Such systems are often plagued by inefficiencies, higher operational costs, and increased susceptibility to human error, which can lead to delayed shipments and customer dissatisfaction. These challenges have prompted a significant shift towards automation and the integration of intelligent robotics within warehouse environments to enhance operational efficiency, accuracy, and scalability. Many warehouses have applied robotic systems on robots [1].

According to Wu, autonomous mobile robots have emerged as a pivotal technology in modern warehouses, offering flexibility and scalability in inventory management and order fulfillment [2]. As the demand for automation in logistics continues to grow, the ability of robots to maneuver through complex layouts while avoiding obstacles becomes increasingly important.

Effective navigation and path planning are critical for these robots to operate efficiently in dynamic and obstacle-rich environments. The appropriate algorithm helps warehouse robots operate autonomously [3]. Reinforcement learning algorithms, particularly Q-learning, have shown promise in enabling robots to learn optimal paths through interaction with their environment without explicit programming of movement strategies, particularly in dynamic environments such as warehouses where efficient navigation can significantly enhance operational productivity.

The concept of Q-learning came up with Christopher Watkins [4]. Tan et al. have demonstrated the potential of Q-learning in robotics for navigation and task execution [5]. However, challenges remain in optimizing the learning parameters to ensure efficiency and stability, especially in complex environments like warehouses where obstacles and operational dynamics can vary significantly. This study aims to address these challenges by systematically analyzing the impact of the learning rate (α) and discount factor (γ) on the performance of a Q-learning algorithm applied to warehouse robot path planning.

By simulating a warehouse environment using a grid world model, this study provides a controlled setting to investigate how different parameter configurations affect the robot's ability to learn and execute efficient paths. The grid world model abstracts the warehouse space into a grid of discrete states, simplifying the complex real-world environment. A full-factor experiment is employed to conduct a comprehensive exploration of the parameter space, enabling the identification of optimal settings that minimize path-planning failures and reduce the total number of steps required for task completion. This methodical approach ensures that the findings are robust and generalizable.

This research contributes to the field of autonomous warehouse robotics by offering insights into parameter optimization for reinforcement learning algorithms. By understanding the influence of key parameters on learning performance, more effective algorithms that enhance the intelligence and efficiency of robotic systems in warehouse operations can be developed. Ultimately, this work aims to improve the operational efficiency of warehouses in the burgeoning e-commerce landscape, where the ability to rapidly and accurately fulfill orders is critical to maintaining competitive advantage and achieving customer satisfaction.

2. Methodology

2.1. System Design

The system was designed to simulate a warehouse environment using a grid world model, which provided a simplified yet effective representation of the spatial dynamics within a warehouse. The grid world had a 3:5 ratio, creating a structured layout where each cell corresponded to a discrete state that the robot could occupy. Obstacles representing warehouse shelves were strategically placed within the grid to mimic real-world conditions, and the outbound area was designated as the goal location at coordinates (10, 10). The robot's starting position was set, and its task was to operate from the starting point to the location of the target cargo, then continue to transport the cargo to the exit, and finally return to the starting point while avoiding obstacles.

This design allowed for a controlled environment where the robot's learning and decision-making processes could be monitored and analyzed. The sampling strategy involved enabling the robot to explore all possible actions from each state—moving left, right, up, or down—thus allowing it to learn optimal paths through interaction with the environment. By discretizing the workspace, the complexity of continuous navigation was reduced, facilitating the application of reinforcement learning algorithms.

2.2. Data Collection Methods

Data collection was conducted through simulated experiments where the robot interacted with the grid world environment. Each experiment consisted of multiple episodes, during which the robot attempted to reach the goal state from its starting position. Data collected included the state transitions, actions taken, rewards received, and the number of steps taken in each episode.

The experiments varied the learning parameters—learning rate (α) and discount factor (γ)—across a range of values to observe their effects on the robot's learning efficiency and path-planning capabilities. A full-factor method was employed, testing 81 different combinations of α and γ (each ranging from 0.1 to 0.9 in increments of 0.1). This systematic approach ensured comprehensive coverage of the parameter space and facilitated the identification of trends and optimal settings.

2.3. Instrumentation

The primary algorithm Q-learning is implemented as follows:

Initialization: The Q-table, which records the expected rewards for each action in each state, is initialized to zeros. The learning parameters, namely the learning rate (α) and the discount factor (γ), are set at 0.1 and 0.9, respectively. These parameters are critical for determining how quickly the agent learns and how much it values future rewards.

State Transition and Reward Calculation: The agent transitions between states based on the chosen action, which can result in moving left, right, up, or down. The rewards are defined as follows [6]:
A positive reward is awarded for reaching the goal state.

A negative reward is incurred for hitting an obstacle or moving out of bounds.
No reward is given for actions that do not result in a goal being reached.
Q-Value Update: After each action, the Q-value is updated according to the formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \ (R + \gamma \max Q(s',a') - Q(s,a)) \tag{1}$$

Where s is the current state, a is the action taken, R is the reward received, and s′ is the new state. This update rule allows the agent to refine its policy based on the cumulative rewards received.

A greedy algorithm was also implemented to enhance exploration and prevent the agent from getting trapped in local optima. At each time step, the agent selects actions based on the current state using an epsilon-greedy strategy. This method balances exploration and exploitation by allowing the agent to choose the action with the highest Q-value with a probability of (1−ϵ) (1−ϵ) and a random action with probability ϵ. This approach ensures that the agent explores the environment sufficiently while also exploiting known optimal actions. This algorithm allowed the robot to select actions leading to states with the highest Q-values, thus optimizing the path by considering immediate rewards. The combination of Q-learning and the greedy approach aimed to improve both learning efficiency and path optimality.

Simulation software was used to model the grid world environment and implement the algorithms. Data visualization tools were employed to create figures and graphs that illustrated the results, such as the total number of steps taken under different parameter settings.

2.4. Procedures

In a typical small warehouse, the robot follows the command to operate from the starting point to the location of the target cargo, then continues to transport the cargo to the exit, and finally returns to the starting point to stand by, thus completing a full work cycle. The experimental procedures began with initializing the grid world environment and the Q-learning algorithm's parameters. The Q-learning approach with a greedy algorithm is applied to determine the path for this circle. Then the different parameter combinations of alpha and gamma are used to run the algorithm repeatedly. For each combination of α and γ, the robot underwent a series of episodes where it interacted with the environment.

The environment map depicted in Fig. 1(a) serves as a grid-based representation of a simulated warehouse layout utilized in the Q-learning algorithm. Fig. 1 (b) illustrates the path of the full work cycle in the simulated warehouse environment.

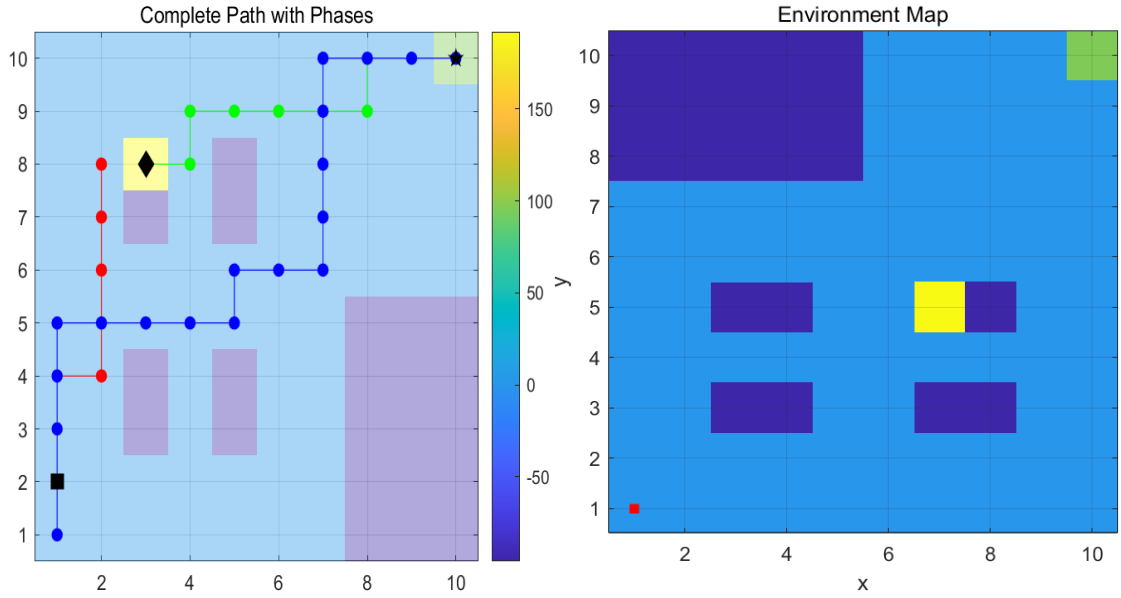


Fig 1. The path of a robot operation cycle.
(a) The environment map used in the Q-learning simulation for warehouse operations; (b) final path found by Q-learning. (Photo/Picture credit: Original).

The full-factor experimental method is employed to evaluate two factors: alpha and gamma. Each factor has nine levels, ranging from 0.1 to 0.9 at intervals of 0.1, resulting in a Two-Factor Nine-Level Design with a total of 81 parameter combinations. The outputs of the experiment are the total step number, which is the sum of all paths, and stability, which refers to the variation in the outcomes of optimization. The experiment aims to find the optimal parameter combinations that yield the most stable and efficient path-planning results.

Data cleaning involved averaging the results of over five experiments for each parameter combination to mitigate the effects of any anomalies or outliers. Instances of path planning failure—where the robot failed to reach the goal—were noted, and their occurrence was analyzed concerning the parameter settings. The collected data was then used to assess the influence of the learning parameters on the efficiency and reliability of the path-planning algorithm.

3. Results and Discussions

3.1. Results

As shown in Fig. 2, the experiments yielded comprehensive data illustrating how different combinations of learning rate (α) and discount factor (γ) affected the robot's path-planning performance.

Higher values of γ generally led to a decreased average number of episodes required for the robot to find the optimal path. This effect was more pronounced when paired with higher values of α, indicating enhanced learning efficiency. Specifically, the robot learned to prioritize future rewards more effectively, resulting in more strategic path planning.

The total number of steps required for the robot to complete the operation cycle decreased as the value of γ increased. The minimum number of steps observed for an effective path was 40 steps. Parameter combinations such as (α=0.1, γ=0.4) and (α=0.1, γ=0.5) resulted in the minimum step counts, highlighting the impact of γ on path efficiency.

Lower values of α and γ were associated with a higher incidence of path-planning failures. For example, combinations like (α=0.1, γ=0.1) and (α=0.1, γ=0.2) resulted in failures, evidenced by a step count of zero. This indicates that insufficient learning rates and discount factors hinder the robot's ability to learn effective strategies.

Variation in the number of steps taken, even with the same parameter combinations, was observed. This suggests that other factors, possibly inherent randomness in the epsilon-greedy strategy or specific environmental interactions, influenced the outcomes.

Figs. 2 and 3 in the study visualized these findings, showing trends in total steps and highlighting parameter combinations leading to failures or optimal performance.

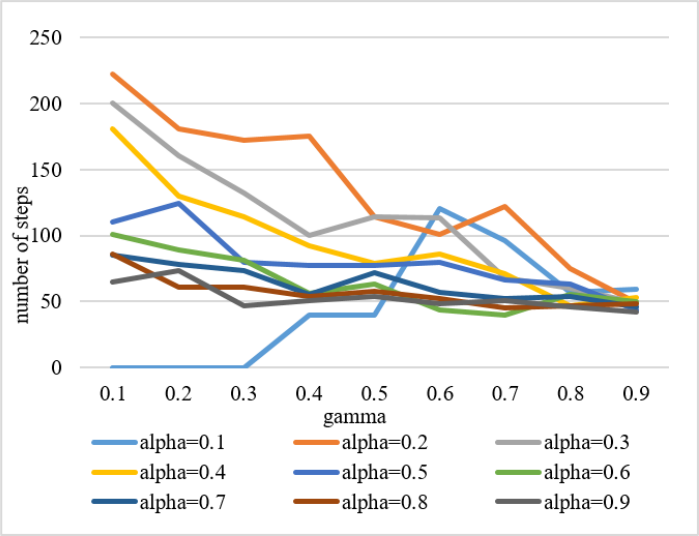


Fig 2. Total steps via different α and γ (Photo/Picture credit: Original).

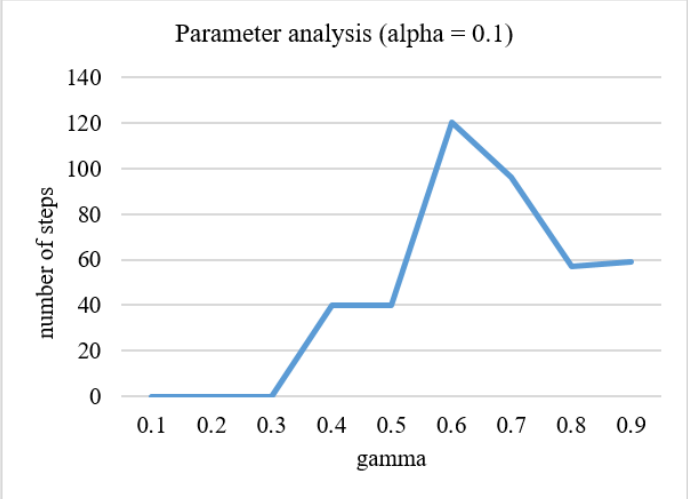


Fig 3. Parameter analysis at alpha = 0.1 (Photo/Picture credit: Original).

3.2. Discussion

3.2.1. Influence of parameters

The results demonstrate the significant influence of the learning parameters on the Q-learning algorithm's effectiveness in path planning for warehouse robots. The learning rate (α) controls how much newly acquired information overrides old information. Higher values of α allowed the robot to adapt more quickly to the environment, reducing the number of steps and failures. However, excessively high learning rates can lead to instability by overreacting to recent experiences. The discount factor (γ) determines the importance of future rewards. Higher γ values encouraged the robot to consider long-term benefits, leading to more efficient paths and decreased episodes of convergence.

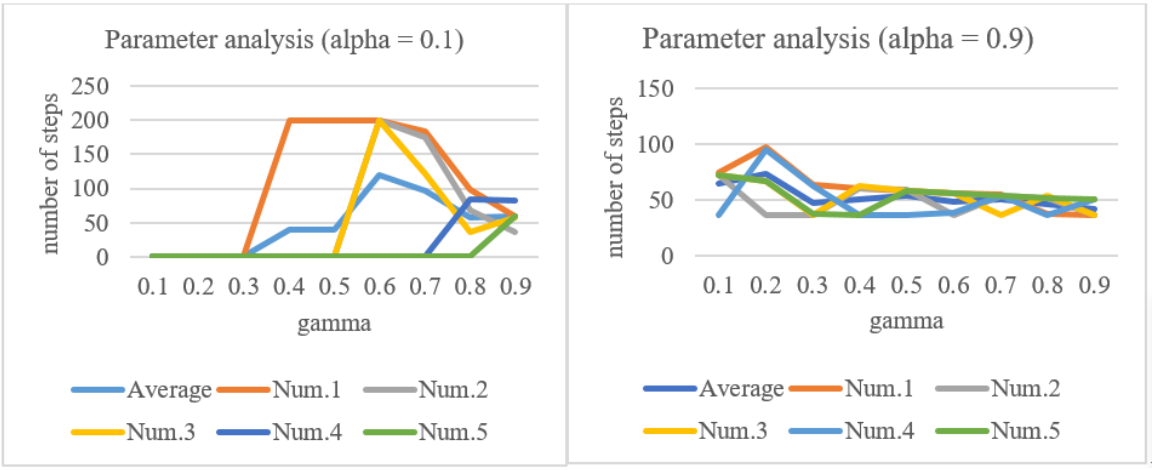
Together, these findings underscore the importance of carefully tuning both α and γ to improve the agent's learning efficiency and stability. This analysis aligns with the earlier observations of the agent's final path and training process, highlighting how adaptive learning through reinforcement

techniques can effectively address the complexities inherent in modern warehouse operations within the e-commerce landscape.

Furthermore, the minimum number of steps from three stages of path planning within an operation cycle, essential for measuring path planning efficiency, is calculated as an average from five experiments using identical parameters. This method ensures a dependable performance measure. The document notes a decrease in the total steps needed as the gamma value increases, indicating its significant impact on efficiency.

3.2.2. Path planning failure

There is a general decrease in the total number of steps required across the three stages of the planning process as shown in Fig. 4. Fig. 4(a) illustrates the number of steps for the work cycle when alpha equals 0.1 while Fig. 4(b) illustrates the circumference when alpha equals 0.9. A step count of zero signifies path planning failure, with 40 steps being the minimum effective threshold. Notably, the minimum step count occurs at parameter combinations (0.1, 0.4) and (0.1, 0.5), suggesting these settings pose challenges for the algorithm. Subsequently, as shown in Fig. 4, it indicates that unreasonable combinations of parameters, such as those at (0.1, 0.1) and (0.1, 0.2), can result in path planning failure. Specifically, a step count of zero is a clear indicator of such a failure, suggesting that these parameter values do not yield a viable path.



(a) (b)

Fig 4. The variation in the path planning.
(a) Alpha= 0.1; (b) alpha = 0.9. (Photo/Picture credit: Original).

Failures in path planning were primarily associated with low values of α and γ. When α was too low, the robot updated its Q-values minimally, resulting in slow learning and an inability to adapt to the environment. Similarly, low γ values caused the robot to focus excessively on immediate rewards, neglecting the benefits of long-term planning. These factors led to the robot failing to find a viable path to the goal within a reasonable number of steps.

The occurrence of failures highlights the sensitivity of the Q-learning algorithm to its parameters. This observation underscores the necessity of careful parameter tuning to ensure the effectiveness of the path-planning process.

3.2.3. Optimization Methods

The full-factor method in the abovementioned experiments is comprehensive as it covers all possible combinations of two factors and nine levels to fully explore alpha and gamma relationships without omitting relevant ones. It is also accurate because it tests all combinations to determine the influence of each factor and their interactions precisely, providing a solid basis for understanding path planning. Meanwhile, it is simple for small-scale experiments, being easy to design and implement with fewer factors and levels and allowing direct observation and analysis without complex statistics, and the results are clear, facilitating the identification of main effects and interactions to help make accurate decisions regarding process optimization.

Further optimization could involve adaptive methods where α and γ are adjusted dynamically based on the learning progress. Techniques such as meta-parameter optimization, grid search, or even applying machine learning methods to tune these parameters could enhance performance. Additionally, integrating other reinforcement learning strategies or algorithms might mitigate some of the observed limitations.

3.3. Limitations and Prospects

3.3.1. Comparison to existing research

The findings of this study align with existing research on the application of reinforcement learning to robotic path planning. Similar to Lemos et al., this study demonstrated that higher discount factors improve learning efficiency by emphasizing future rewards [7]. The reduction in path planning failures with higher learning rates supports the observations made by Maoudj et al. regarding the importance of parameter tuning for adaptation in dynamic environments [8].

However, while Peyas et al. incorporated deep Q-learning and neural networks to handle more complex environments, this study relied on a tabular Q-learning approach within a discrete grid world [9]. This simplification allowed for a clearer analysis of the effects of α and γ but may limit generalizability to real-world applications.

The inclusion of a greedy algorithm to enhance exploration differentiates this study from others, providing an additional method to avoid local optima. This approach complemented the findings of Cestero et al., who emphasized the importance of learning from interactions within the environment to improve navigation capabilities [10].

Overall, the study contributes to the body of knowledge by reinforcing the importance of parameter optimization in reinforcement learning for robotic path planning and by demonstrating the practical implications of these parameters on learning efficiency and stability.

The results and discussions highlight the critical role of learning parameters in the performance of the Q-learning algorithm for warehouse robot path planning. The study's findings provide insights into optimizing these parameters to enhance efficiency and reliability, contributing valuable knowledge to the field of autonomous robotics in warehouse management.

3.3.2. Limitations and prospects of this study

The grid world model, while useful for simulation, does not capture the full complexity of a real warehouse environment. Factors such as dynamic obstacles, varying floor conditions, and sensor noise are not represented.

Besides, the Q-learning algorithm with a discrete state-action space may not scale well to larger, continuous spaces without modifications.

Variations in results, even with identical parameter settings, suggest that additional factors influence performance. This could be due to the stochastic nature of the epsilon-greedy strategy or the random initialization of the Q-table.

4. Conclusion

This study investigated the application of Q-learning for path planning in warehouse robots within a simulated grid world environment, focusing on the impact of the learning rate (α) and discount factor (γ) on the algorithm's performance. The systematic exploration using the full-factor experimental method revealed that higher values of γ significantly improve learning efficiency by encouraging the robot to prioritize future rewards, leading to more strategic and efficient path planning.

The results demonstrated that optimal combinations of α and γ reduce the total number of steps required for the robot to complete its operational cycle and minimize the incidence of path-planning failures. Specifically, higher γ values paired with appropriate α levels enabled the robot to learn

effective strategies more quickly and reliably. These findings underscore the critical importance of parameter tuning in reinforcement learning applications for real-world scenarios.

Despite the promising outcomes, the study acknowledges limitations due to the simplified grid world model, which may not capture all the complexities of a real warehouse environment, such as dynamic obstacles and sensor noise. Future research could address these limitations by implementing more sophisticated models, incorporating deep reinforcement learning techniques, and exploring adaptive parameter optimization methods to enhance scalability and applicability.

In conclusion, the study makes a valuable contribution to the field by providing insights into optimizing Q-learning parameters for autonomous warehouse robots. By demonstrating the significant impact of parameter tuning on learning efficiency and performance, offers a step toward developing more efficient and reliable warehouse automation solutions. This advancement is particularly relevant in response to the growing demands of the e-commerce industry, where efficient logistics and rapid order fulfillment are increasingly important. The findings suggest that with appropriate optimization of reinforcement learning parameters, warehouse robots can achieve higher levels of autonomy and efficiency, ultimately contributing to more effective and competitive warehouse operations.

References

[1] Lee Ckm, Lv Y, Ng Kkh, Ho W, Choy Kl, Design and application of Internet of things-based warehouse management system for smart logistics. International Journal of Production Research, 2018, 56(8): 2753-2768.

[2] Wu J P. Application analysis and the prospect of Kiva robot in Amazon warehouse. Logistics Technology and Applications, 2015, 20(10): 159-164.

[3] Kumar N V, Kumar C S. Development of collision free path planning algorithm for warehouse mobile robot. Procedia computer science, 2018, 133: 456-463.

[4] Watkins C J C H, Dayan P. Q-learning. Machine learning, 1992, 8: 279-292.

[5] Tan L, Zhang H, Liu Y, et al. An adaptive Q-learning based particle swarm optimization for multi-UAV path planning. Soft Computing, 2024, 28(13): 7931-7946.

[6] Wen X, Zhang H, Li H, et al. Fusion q-learning algorithm for open shop scheduling problem with AGVs. Mathematics, 2024, 12(3): 452.

[7] Lemos M R, de Souza A V R, de Lira R S, et al. Robot training and navigation through the deep Q-Learning algorithm. 2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2021: 1-6.

[8] Maoudj A, Christensen A L. Q-learning-based navigation for mobile robots in continuous and dynamic environments. 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE). IEEE, 2021: 1338-1345.

[9] Peyas I S, Hasan Z, Tushar M R R, et al. Autonomous warehouse robot using deep q-learning. TENCON 2021-2021 IEEE Region 10 Conference (TENCON). IEEE, 2021: 857-862.

[10] Cestero J, Quartulli M, Metelli A M, et al. Storehouse: A reinforcement learning environment for optimizing warehouse management. 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022: 1-9.