Research on a Dynamic Algorithm of Simulated Warehouse Environment Based on Q-learning

Jiaming Lu *

Brunel London School, North China University of Technology, Beijing, 100144, China

* Corresponding Author Email: 2282119@brunel.ac.uk

Abstract. This paper uses Q-learning and warehouse simulation environment path planning algorithm training and research. The application environment is a dynamic and complex environment for a real warehouse. However, traditional research is more focused on exploring the Q-learning algorithm itself, focusing on the algorithm itself rather than the application, which lacks practicability. Therefore, this study will take the real map of the warehouse as a sample can combine Q-learning to simulate the algorithm closer to reality. The algorithm simulation will be carried out in a unique three-stage form to fit the influence of dynamic and changeable on the algorithm training in the actual situation. After algorithm training and simulation, the visual path result of Q-learning in the warehouse map is obtained. Based on the result, the advantages and disadvantages of Q-learning in warehouse path planning are discussed, and suggestions for future algorithm improvement. Q-learning algorithm is a promising algorithm. Through its research, more mature path planning results can be obtained, and it can be used to optimize cargo transportation efficiency and warehouse management mode in industrial production, logistics, and other fields, to obtain actual benefits.

Keywords: Q-learning; Path planning; intelligent storage; Machine learning.

1. Introduction

With the rapid development of the modern logistics industry and industrial processing industry, the position of warehouses in the supply chain is increasingly prominent. The efficiency of warehouse management directly affects the operating cost and distribution speed of enterprises, especially when the complexity of warehousing operations increases significantly. The path planning and dynamic scheduling of robots and other equipment in automated warehouses have become one of the key issues. By optimizing the task scheduling and path selection of these devices, the dynamic response capability and resource utilization of warehouses can be effectively improved. Therefore, how to realize intelligent scheduling in a complex dynamic warehouse environment is an important direction of current logistics intelligence research [1].

In recent years, the application of reinforcement learning algorithms, especially Q-Learning, in dynamic warehouse environments has been widely discussed. As a model-free reinforcement learning algorithm based on value functions, Q-Learning is suitable for solving path-planning problems in discrete state Spaces and optimizing strategies through continuous interaction with the environment [2]. Although the algorithm has a wide application prospect in theory, it still faces many challenges in practical application.

To improve the efficiency and adaptability of Q-learning, many researchers have improved it. For example, by introducing a multi-agent cooperation mechanism, researchers designed a multi-agent Q-Learning algorithm, which enables task coordination among different AGVs in the warehouse, reduces conflicts, and improves the overall scheduling efficiency [3]. In addition, some studies combined deep Learning technology to propose a deep Q network (DQN), which uses neural networks to estimate value functions and effectively expands the application of Q-learning in complex environments [4]. Although DQN performs well in many scenarios, it still has significant shortcomings in terms of training time and resource consumption. In addition, Q-Learning's performance in dealing with complex dynamic problems such as sudden tasks and demand fluctuations is still limited [5].

Although the current research has realized the application and promotion of reinforcement learning in theory, in the actual warehouse environment, complex factors such as dynamic change and task burst put forward higher requirements for the adaptability of the existing algorithms [6]. For example, when the traditional Q-Learning algorithm is faced with a large-scale warehouse environment, the computational complexity increases sharply with the expansion of the state space, resulting in a slow convergence speed, and it cannot timely respond to changes in the demand for goods in the warehouse [7]. In addition, the optimization difficulty of multi-task scheduling and path planning increases exponentially with the increase of the type and number of tasks, which also poses new challenges to the scalability of the algorithm [8].

Based on the Q-Learning algorithm, this paper presents a three-stage algorithm training strategy. To cope with the dynamic and complex warehouse environment. The warehouse map used for simulation is based on the real warehouse map, and the state of the robot equipped with the algorithm can be restored as much as possible during actual operation. It can provide a high reference value for the application of this technology in reality.

The focus of this paper is to provide an efficient and flexible solution for intelligent warehouse path planning and task scheduling. It also verifies the advantages highlighted by Q-Learning when running this scheme and collects the problems exposed in the simulation process for further in-depth research and improvement.

2. Methodology

2.1. Principle of Q-learning Algorithm

Q-learning is a model-free reinforcement learning technique that allows an agent to learn policies that tell it what actions to take in what situations, without the need for a model of its environment. It is a form of deep machine learning and relies on the use of an artificial neural network that processes the input data of the environment through sensors making it more intelligent every time [9].

Reinforcement learning is a machine learning paradigm in which agents interact with their environment and learn how to take actions in different states to maximize long-term returns. RL problems can often be modeled as Markov decision processes. A reinforcement learning system is a system in which the agent learns action strategy from the mapping of the environment to behaviors to maximize the value of the reward. Rewards provided by the environment in reinforcement learning systems are evaluations of the quality of actions. Reinforcement learning systems gain knowledge in an action-evaluation environment and improve their action strategy to adapt to the environment [10].

The core of Q-learning is the Q function, denoted as Q (s, a), which represents the expected cumulative reward that can be obtained by taking action a understate s. The goal is to find the optimal strategy π* by learning the Q function, that is, to choose the action that maximizes the value of Q in each state.

Q-learning is based on the principle of Bellman optimality, and the Q-value function satisfies the following Bellman equation:

$$Q(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} Q(s',a') \mid s,a\right] \tag{1}$$

Where R_{t+1} is an immediate reward for taking action a, s' is the next state after taking action a, γ is the discount factor, $\max_{a'} Q(s',a')$ represents the maximum Q value in the next states'.

Q-learning approximates the optimal Q by iteratively updating the Q value. The update rules are as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{2}$$

Where α is the learning rate that controls the pace of updates, r is an immediate reward, s' is the next state reached after taking action a.

Under appropriate conditions, Q-learning is guaranteed to converge to the optimal Q-valued function, namely Q*. This means that the agent is ultimately able to find the optimal strategy that maximizes long-term cumulative rewards.

2.2. Comparison between Q-Learning and Other Algorithms

Q-learning is a modeless reinforcement Learning algorithm, which ADAPTS to the unknown environment by updating the Q value gradually through interaction with the environment. Other algorithms, such as the A* algorithm, are heuristic search algorithms that need to know the current environment and find the optimal path by estimating the cost function. Q-Learning focuses on long-term rewards and exploration and is suitable for dynamic environments, while A* focuses on shortest path search in static environments. The application scenarios of the two are different, so this paper chooses the Q-Learning algorithm to study.

2.3. Three-stage Algorithm Design Logic in a Simulated Warehouse Environment

2.3.1. Map construction

In MATLAB, the warehouse mapping model is established with the actual warehouse as the sample to improve the feasibility of the final scheme in reality. In addition, the three-stage path simulation of the Q-Learning algorithm is preliminarily carried out. The warehouse map model is shown in Fig. 1.

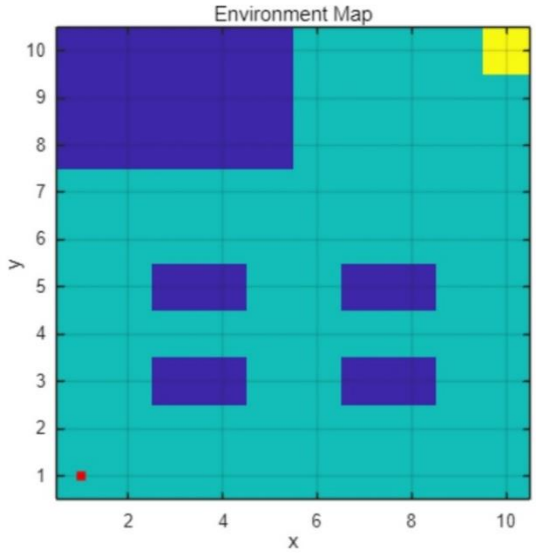


Fig 1. Warehouse map model (Photo/Picture credit: Original).

2.3.2. Three-stage path planning

The focus of this paper is to realize the proposed three-stage path planning scheme based on the Q-Learning algorithm, aiming to solve the problem that the traditional algorithm cannot realize the optimal path planning, or even cannot run in the dynamic environment and possible obstacle position changes. Combine Q-Learning with a variety of training to secure or optimize solutions.

The three-stage path planning and design idea is adopted in this study, because the robot may encounter obstacles in the path to move or other robots in the process of completing the task. This paper considers that only one path planning training cannot meet the research objectives of warehouse dynamic path planning. Therefore, the path is divided into three segments and independent algorithm training is carried out. Under the premise of this training mode, the Q-Learning algorithm will show its characteristics and advantages as much as possible, and carry out better dynamic path planning. The results presented should also be in line with the actual situation, with high persuasiveness and realistic feasibility. It also provides a good reference value for the follow-up research and analysis of this paper. The algorithm of MATLAB core part expressed by logic flow chart is shown in Fig. 2.

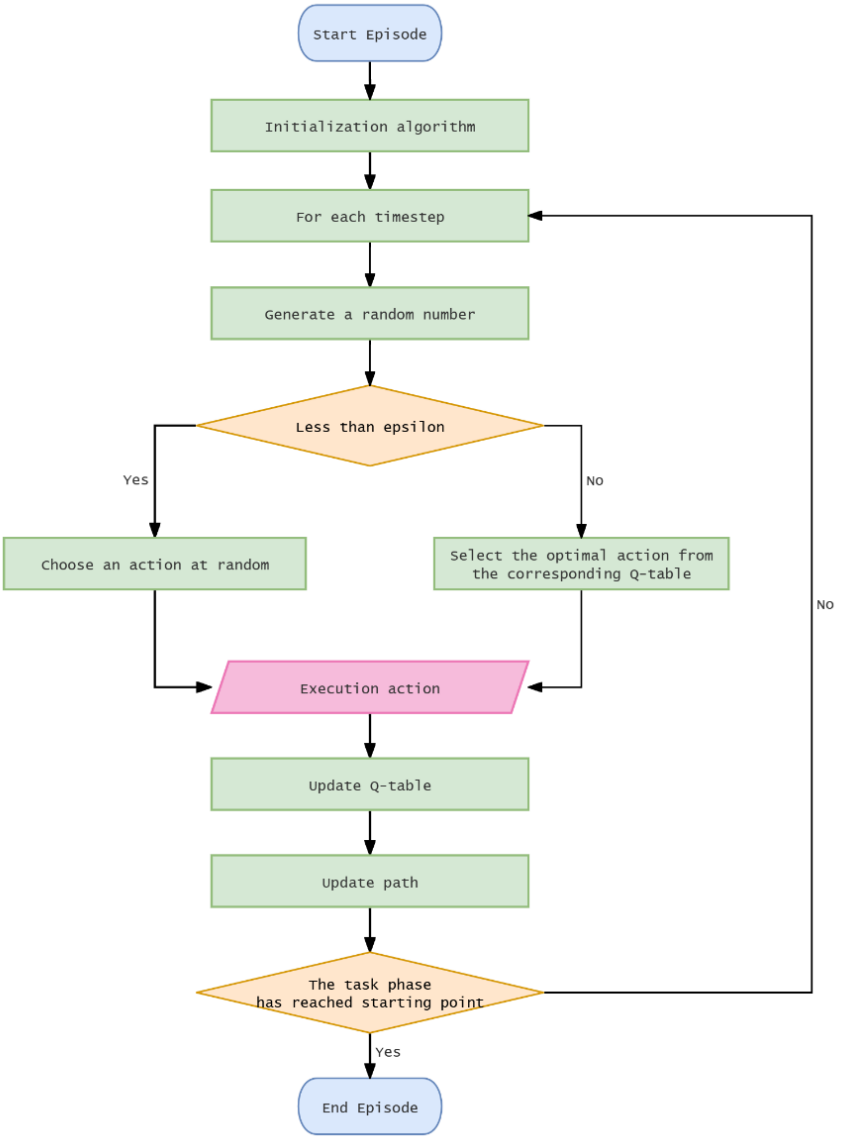


Fig 2. The logical flow chart of the algorithm (Photo/Picture credit: Original).

2.3.3. Scheme step

a) The robot will complete the movement from the starting point to the coordinates of the goods.
b) The goods will be transported to the assigned shipping port.
c) The robot returns to its starting point.

3. Results and Discussions

3.1. Results Based on Q-learning

In MATLAB algorithm simulation, Q-Learning algorithm learning iteration is used to generate the optimal three-stage path planning scheme.

The three paths are simulated respectively, the distance of the cargo handling robot from the fixed robot parking point to the target shelf, from the shelf to the unloading point, and back to the robot parking point. Since the three distances are run by robots at different times, and the environment is dynamic, there may be other robots active in the warehouse, and there will be goods that suddenly appear on the path. Therefore, the map condition of the previous period cannot fully reflect the situation of the next period, so it is necessary to conduct algorithm training and path simulation independently.

3.1.1. First path

The significance of the first path is to enable the robot simulated in the algorithm to generate the optimal route from the starting point to the simulated shelf. The specific results are shown in Fig. 3.

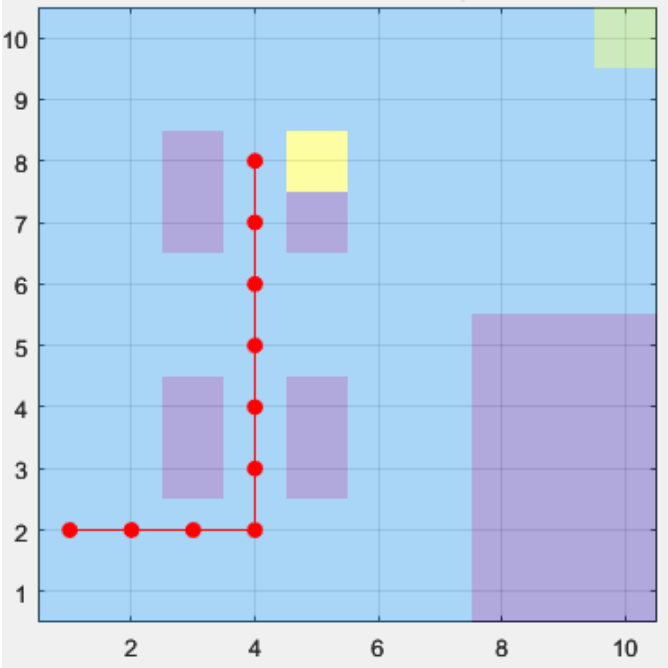


Fig 3. First path (Photo/Picture credit: Original).

3.1.2. Second path

The meaning of the second path is to make the robot from the shelf to the unloading point (the end point), where the optimal path scheme is generated. The specific results are shown in Fig. 4.

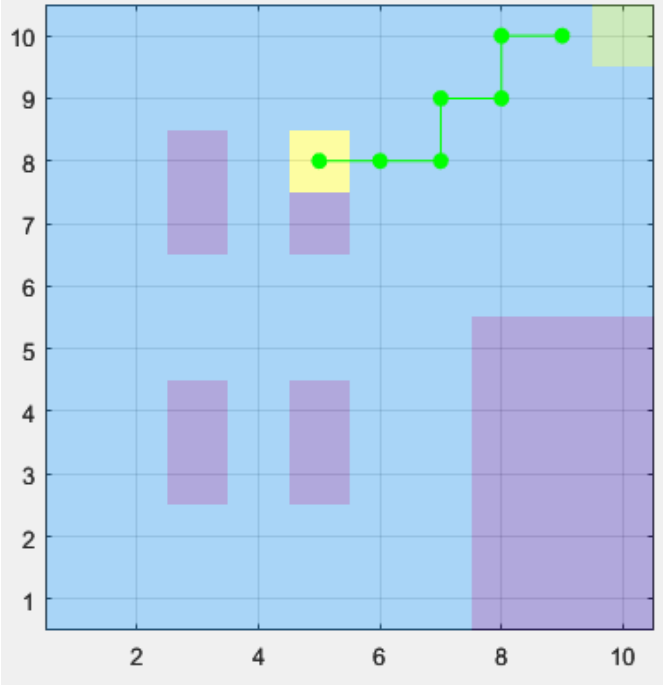


Fig 4. Second path (Photo/Picture credit: Original).

3.1.3. Third path

The significance of the third section of the path is to make the robot return to the starting point from the unloading point to complete the closed loop of the entire robot running route. The specific results are shown in Fig. 5.

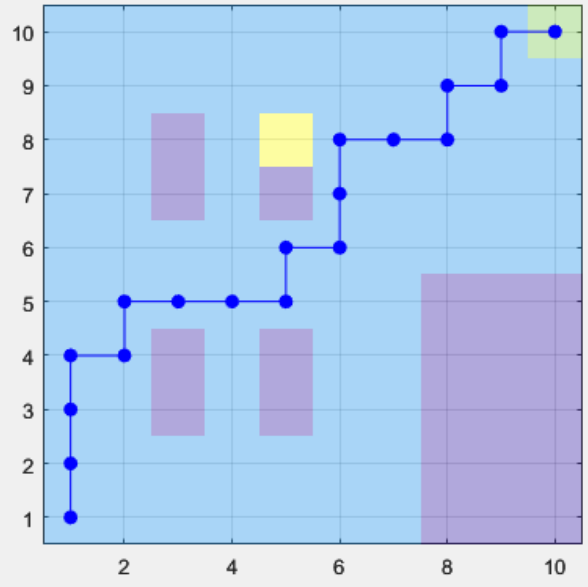


Fig 5. Third path (Photo/Picture credit: Original).

This completes a simplified version of path planning in a dynamic warehouse environment. The general route map after consolidation is shown in Fig. 6.

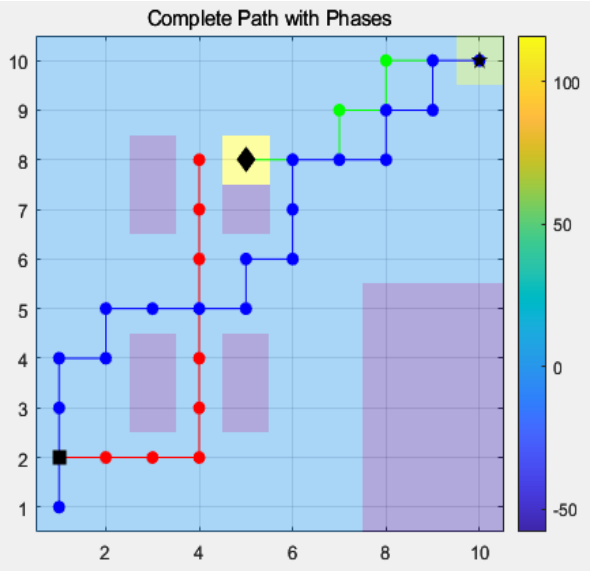


Fig 6. Path summary diagram (Photo/Picture credit: Original).

3.2. Advantages and Disadvantages of Q-learning Algorithm in Warehouse Environment

In the warehouse simulation environment, the advantages and disadvantages of using the Q-learning algorithm can be analyzed through different dimensions:

3.2.1. Q-Learning highlights the advantages of research

From the algorithm point of view, Q-learning is a model-free algorithm, which does not need to know the layout, path, or inventory of the warehouse in advance. Through continuous interaction with the environment, it can learn the optimal handling path and other strategies to adapt to dynamic environmental changes, such as changes in the position of goods/obstacles or path obstruction caused by other robots.

During the exploration phase, Q-learning is able to explore various potential optimal paths and strategies in the warehouse environment by constantly trying different actions. Especially with the support of the ε-greedy strategy, the algorithm can effectively balance exploring new paths and utilizing existing optimal paths.

In practical applications, once Q-learning converges, it can adapt itself to environmental changes and achieve dynamic adjustment. For example, when the location of the goods or the channel changes, Q-learning can adjust the strategy according to the new state in time, without having to redesign the entire system.

3.2.2. The unsolved problems and shortcomings

Due to the lack of prior knowledge, Q-learning has a low learning efficiency in the early stage and requires a lot of exploration to converge to the optimal strategy. In a large warehouse environment, the state and action space can be very large, resulting in a long convergence time. Its overall efficiency still has a lot of room for improvement. In the early exploration phase, the algorithm may choose a sub-optimal path, resulting in higher operating costs, such as increased time or energy consumption. In addition, over-exploration may make it difficult for the algorithm to converge to local optimality.

For complex warehouse layouts, Q-learning may face the problem of dimensional explosion of state-action pairs, resulting in excessive Q-table and low efficiency of update and storage. At this point, methods such as Deep Q networks (DQN) need to be combined to solve large-scale problems.

Q-learning has flexibility and adaptability in a warehouse simulation environment, which is suitable for dealing with dynamic changes, but its learning efficiency and convergence speed face challenges when the state space is large.

3.3. Actual Performance of Q-learning Algorithm in Complex Warehouse Dynamic Environments

This paper starts from the strategy of Q-learning three-stage path planning, and through MATLAB simulation, the results show the characteristics of Q-Learning, as well as the adaptability and superiority in the complex warehouse dynamic environment. Advancements in Warehouse Management Systems can be classified into different trends. On the one hand, some advancements aim to enhance specific processes of the ware-houses, improving the effectiveness and efficiency of the whole activity. On the other hand, other advances focus on improving more general processes, using Machine Learning methods such as Reinforcement Learning [11].

In the dynamic environment of a complex warehouse, the actual performance of the Q-Learning algorithm is affected by the complexity and dynamic factors of the environment, showing the following characteristics:

3.3.1. Strong adaptability

By interacting with the environment in a model-free way, Q-Learning can adapt to changing factors in the warehouse, such as shelf position changes, path blocking, temporary task insertion, etc. It updates policies based on real-time feedback and progressively optimizes routing and resource scheduling. In a complex and dynamic environment, Q-learning can cope with uncertainty through continuous learning and find the optimal path or strategy.

3.3.2. Strong ability to handle delayed feedback

Warehouse tasks often involve multi-step decisions, such as item picking/path planning/task scheduling, etc. Q-Learning can handle long-term returns through time-difference learning, and gradually adjust the current decision to optimize the entire process. Q-learning deals well with long-term feedback issues and is particularly suited for complex tasks that require multiple actions to work together.

3.3.3. Flexible response in a dynamic environment

In a dynamic warehouse environment, Q-Learning can adjust behavior through real-time environmental interaction. When encountering environmental changes (such as robot failure, shelf rearrangement, etc.), the algorithm can gradually adapt to changes through new learning rounds and find new optimal solutions. Compared with static programming algorithms, Q-Learning has better adaptability and flexibility to the dynamic changing environment.

3.3.4. Low early efficiency

In the early stage of learning, Q-Learning requires a lot of exploration, and there may be inefficient handling paths or invalid actions in the warehouse, resulting in increased task completion time or waste of resources. It must be mentioned that when the state space of the warehouse is large, the algorithm needs to traverse more state-action combinations, and the learning process is slow. This leads to low initial efficiency, which is already reflected in MATLAB simulations, even in relatively simple environments. Considering the actual situation, in a complex environment, it takes a long time to train to get close to the optimal strategy.

3.3.5. Optimization space of machine learning

With the increase in warehouse sizes, state space, and action space, such as multiple handling robots, multiple alternative paths, and multiple task objectives. The size of Q tables has grown exponentially, resulting in a significant increase in storage and computing requirements. In addition, the frequently changing environment makes the Q value update complicated and easy to cause slow or unstable convergence. In order to deal with this problem, methods such as deep Q networks are usually combined with neural networks to replace Q tables to deal with high-dimensional state space and improve efficiency.

In general, in the dynamic environment of complex warehouses, the Q-Learning algorithm shows strong adaptability and long-term optimization ability, and its potential is immeasurable. However, low efficiency in the early stage, dimensional disaster in large state space, and slow convergence speed are the main bottlenecks. By combining deep learning technology, Q-Learning can improve its performance in complex scenarios.

4. Conclusion

Aiming at the path planning problem in a complex dynamic warehouse environment, this paper adopts a Q-Learning algorithm to build a simulated warehouse environment and uses MATLAB for algorithm training and simulation. This paper proposes an innovative three-stage path planning model, which divides the robot path into three independent stages: from the robot resident point to the shelf, from the shelf to the unloading point, and back to the resident point. Each path is trained independently, which makes the algorithm more adaptable to the dynamic change of obstacle position with time. This method improves the flexibility of path planning in practical warehouse applications and the efficiency of task completion.

The results show that the proposed algorithm performs well in improving the ability of automation systems to cope with dynamic environment changes. Through the way of segmented training, the algorithm can deal with the real-time change of obstacle position more effectively in real situations, which is a common problem in the warehouse environment. The simulation results show that robots can complete the task and adapt to the environment change in the dynamic environment. The value of this research is that it provides a more flexible and efficient solution for path planning and task scheduling in complex warehouse environments, which helps to improve the overall efficiency of automated logistics systems.

However, the research also reveals some limitations of this method, especially in the training process. With the increase of environmental complexity and task quantity, the computational complexity of the Q-Learning algorithm increases significantly, resulting in a significant slowing down of convergence speed and a decrease in training efficiency. This problem shows that there is still room for further improvement in algorithm optimization. For example, future research could consider adopting hybrid reinforcement learning methods in combination with other algorithms or heuristic techniques to accelerate convergence. In addition, the performance of the algorithm in larger scale and more complex dynamic warehouse environments needs to be further verified.

Going forward, research should focus on addressing the computational complexity exposed in this study, especially in terms of algorithmic scalability and efficiency gains in large-scale environments.

Q-Learning's development space is infinite. Through continuous algorithm iteration and the combination of multiple algorithms, the ability of an intelligent warehouse management system will be further improved, bringing more benefits to logistics solutions in practical applications and transportation efficiency in automated production.

References

[1] Li, W., & Smith, J. Warehouse management systems and logistics optimization. International Journal of Logistics Management, 2017, 28(2), 23-39.

[2] Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction. MIT Press, 2018.

[3] Wang, X., & Lu, J. Multi-agent Q-learning approach for warehouse task coordination. IEEE Transactions on Automation Science and Engineering, 2019, 16(3), 728-736.

[4] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540), 529-533.

[5] Li, X., & Chen, Y. Dynamic Q-learning algorithm for uncertain warehouse environments. IEEE Access, 2020, 8, 75235-75245.

[6] Van Hasselt, H., Guez, A., & Silver, D. Deep reinforcement learning with double Q-learning. AAAI Conference on Artificial Intelligence, 2016.

[7] Chen, G., & Zhang, H. Reinforcement learning challenges in dynamic environments: A survey. ACM Computing Surveys, 2021, 53(4), 1-35.

[8] Kumar, A., & Verma, R. Task scheduling optimization using reinforcement learning in large-scale warehouse systems. International Journal of Advanced Robotic Systems, 2021, 18(1), 22-31.

[9] 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.

[10] Jiang L, Huang H, Ding Z. Path planning for intelligent robots based on deep Q-learning with experience replay and heuristic knowledge. IEEE/CAA Journal of Automatica Sinica, 2019, 7(4): 1179-1189.

[11] 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.