Journal of Computers Vol. 35 No. 6, December 2024, pp. 137-151
doi: 10.53106/199115992024123506011

AGV Path Planning Method and Intelligent Obstacle Avoidance Strategy for Intelligent Manufacturing Workshops

Jun-Kai Zhang[1,2,3], Xiao-Yang Zhang[1], Mei Han[1*],
Ning Ren[1], and Ya-Jin Hou[1]

[1] Department of mechanical engineering, Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China
{junkai976, xiaoyan56782024, meigui9992024, ningning7865, yajin6666}@163.com

[2] Hebei Province Electromechanical Equipment Intelligent Perception and Advanced Control
Technology Innovation Center, Xingtai City 054000, Hebei Province, China

[3] Xingtai City Electromechanical Equipment Intelligent Perception and Intelligent Control Technology Innovation Center,
Xingtai City 054000, Hebei Province, China

Received 22 November 2024; Revised 30 November 2024; Accepted 2 December 2024

Abstract. The research content of this article is aimed at the intelligent manufacturing workshop of new energy vehicle batteries. Regarding the connection of production processes in the workshop, that is, AGV can achieve path planning and obstacle avoidance strategies in the face of dynamic and complex scenes during material handling, firstly, based on the production layout in the workshop, a neighborhood weighted grid modeling method is used to reduce the path offset error caused by the large speed difference between adjacent grid units in the grid model, and construct a more reasonable grid map. Then, on the basis of the grid map, the deep Q learning algorithm is used to achieve AGV path planning. The traditional algorithm introduces the method of updating the data in the experience pool and improving the direction reward function to solve the dimensional disaster and low learning efficiency of the deep Q learning algorithm in unknown environments. The, Due to issues such as poor robustness of planning strategies, a reasonable AGV path was obtained through simulation experiments, while improving the calculation speed and accuracy of the algorithm.

Keywords: AGV, grid map method, path planning, deep Q-learning

1 Introduction

China is a manufacturing powerhouse, with traditional manufacturing industries and a leading number of manufacturing enterprises in the world. With the continuous development of intelligent manufacturing technology, countries are successively proposing and formulating development strategies that are suitable for their own industrial status [1]. At present, China's traditional manufacturing industry is also in the process of industrial upgrading and transformation in the new era. China's development strategy in the manufacturing industry is to complete the transformation from a manufacturing power to a manufacturing powerhouse and stand out in the globalized manufacturing industry. China's manufacturing industry has chosen the main direction of intelligent manufacturing development [2].

Traditional small and medium-sized manufacturing enterprises often lack relevant large-scale transformation cases in the tide of digital transformation, and their level of production intelligence is greatly challenged in the increasingly demanding market. In recent years, with the rise and progress of intelligent manufacturing technology, it has brought new development paths for traditional manufacturing as well as production management in manufacturing workshops. Big data, the Internet, Industrial Internet of Things (IIoT) and Radio Frequency Identification (RFID) can realize the acquisition of real-time data in the workshop production process, and realize the visualization and manageable intelligence of production management. In addition, the production simulation of virtual models can optimize workshop scheduling, improve the efficiency of production information transmission, and improve production efficiency [3].

* Corresponding Author

In the process of intelligent manufacturing, mobile logistics robots play an important role in transporting goods and connecting process flows. AGV (Automated Guided Vehicle) running in the intelligent manufacturing workshop is a typical production handling mobile robot, generally a wheeled mobile robot. After receiving instructions from the control center, AGV relies on various sensors installed on itself to perceive the environment, and at the same time, relies on the path planning algorithm running in the controller to plan and avoid obstacles, and then carries out cargo handling along the planned optimal path, which has safety, reliability, and efficiency performance. Generally, AGV's power comes from rechargeable batteries [4].

The research on AGV planning algorithms has high practical value, and path planning and automatic obstacle avoidance technology are key technologies of AGV systems. AGV determines its position, surrounding environment, and the direction of the target endpoint through its various sensor devices, and then finds a route to reach the final position through path planning. The purpose of path planning is to find the best feasible route from the starting position to the final position. In the process of intelligent manufacturing material transportation, a high-performance path planning algorithm can make the planned path more optimal, improve the work efficiency of AGV, enhance production efficiency, reduce costs, and enhance the competitiveness of enterprises.

This article mainly focuses on the assembly and production process of new energy vehicle power batteries in the intelligent manufacturing workshop of new energy vehicle power batteries. Based on the production layout of existing workshop production processes, the path planning and obstacle avoidance strategy research of AGV vehicles are completed. The work done is as follows: Summary as follows:

1) Firstly, the construction of the workshop environment map was completed, and the grid map method was used in the map construction process. At the same time, an improved grid method was used to address the directional deviation that occurred in traditional grid map construction, enhancing the guiding role of the grid map;

2) In terms of path planning, AGV path planning is completed based on deep Q-learning algorithm, and an improved algorithm is proposed to address the problems of dimensionality disaster and poor planning robustness of traditional algorithms;

3) We constructed a simulation experiment section and verified the algorithm's improvement in path planning performance through experiments. At the same time, we completed path planning simulation experiments using existing workshop grid maps.

2 Related Work

There are many research achievements in AGV path planning, which can be roughly divided into two categories: one mainly focuses on path planning, and the other focuses on algorithm improvement.

In terms of path planning, Jianhua Wang from Nanjing Forestry University designed an autonomous navigation system based on optimized HectorSLAM (simultaneous localization and mapping, real-time positioning and mapping) algorithm to address the problems of inaccurate positioning, low mapping accuracy, and error accumulation of robots in complex terrain environments. This system aims to enable robots to accurately reach target points while avoiding obstacles in a timely manner. The average relative error of the optimized autonomous navigation system in mapping is about 0.44%, with a minimum mapping error of 0.236 m, a decrease of 0.041 m compared to before optimization, effectively solving the problem of unclear mapping caused by error accumulation and motion distortion [5].

Jingze Lv from Southwest University proposed an improved method for expanding grid modeling based on the characteristics of discrete manufacturing and dynamic assembly lines, as well as the limitations of single target motion in conventional grid modeling environments for automatic guided vehicles (AGVs). The method was optimized and applied using ant colony algorithm. By simulating the optimization and correction effects of segmented paths and obstacle grid units with different proportions, the closer the transfer station is to the obstacle grid, the greater the impact of the optimization algorithm. The higher the proportion of obstacle grids within the controllable range of the path, the more ideal the optimization algorithm's effect [6].

Yu Zhou from Shanghai Ship Equipment Research Institute used an improved A^* algorithm to generate an initial path for each AGV, and then used time window detection method to detect and resolve conflicts for each AGV, obtaining a set of feasible and conflict free paths. The effectiveness of the proposed method was verified through simulation experiments using Python, and compared and analyzed with other methods. The results showed that the proposed method has certain advantages in solving the problem of multi AGV collaborative path planning, which can enable multiple AGVs to work better together and achieve efficient and safe path planning [7].

Yansong Huang from South China University of Technology proposes a global path planning solution model for AGV based on Deep Q-Network (DQN) to address the problem of poor generalization performance in the current deep reinforcement learning algorithm research, which often considers path planning from a single starting point to a single endpoint and involves multiple starting points and endpoints. Through comparative experiments with A^* algorithm and fast extended random tree algorithm in different scale environments, as well as model scalability experiments, the path planning ability of this method in multi endpoint situations was verified [8].

In terms of improving path planning algorithms, Yihu Chen from Guangxi Normal University has designed an improved JPS algorithm that combines diagonal distance and direction information to address the problems of low search efficiency, node redundancy, and low route safety in traditional algorithms [9].

Weidong Zhao from Anhui University of Technology proposes an improved path planning algorithm based on the fusion of A^* and third-order Bezier curves to address the problem of A^* algorithm easily crossing obstacles and having inflection points in path planning. On the basis of traditional A^* algorithm, redundant points on the path are removed through the principle of vector parallelism to improve the efficiency of path search; Improve the traditional eight neighborhood search to skip the points to be extended that are close to obstacles when searching for a path, reducing the possibility of the path passing through obstacles; Introducing a third-order Bezier curve to optimize and improve the path inflection point planning algorithm; Conduct simulation experiments on map environments with two different obstacle rates in the C++ programming environment to verify the effectiveness of the improved algorithm [10].

Wan Xu from Hubei University of Technology proposed an improved artificial potential field method based on concave obstacle filling for local path planning of mobile robots in environments with complex obstacles, addressing the issues of local minima and unreachable targets that arise when using the artificial potential field method [11].

Through the above analysis and research, this article has achieved AGV path planning and obstacle avoidance for intelligent manufacturing workshops in existing research results. The composition of this article is as follows: Chapter 2 introduces some research results in AGV path planning, laying the foundation for the research direction of this article. Chapter 3 completes the construction of environmental maps and uses an improved grid method. Chapter 4 mainly uses deep Q-learning algorithm to complete path planning and proposes improvement methods for the planning algorithm. Chapter 5 is the experimental simulation process, where an experimental simulation environment was constructed to complete path planning experiments and verify the performance of the algorithm.

3 Modeling of AGV Operating Environment

According to the analysis in Chapter 2, the main functions of AGV cars in battery production workshops are summarized as follows:

1) Automated material handling: AGV forklifts complete the automatic handling of battery raw materials, components, and finished products, achieving efficient material transportation.

2) Battery and module handling: In the production of new energy vehicle batteries, AGVs are specifically designed for handling semi-finished battery modules to ensure their safe transportation.

3) Warehousing and Logistics Center: In the warehousing and logistics center, AGV accurately identifies the location and size of goods, automatically grasps and places goods, and can efficiently perform cargo loading and unloading operations, improving the operational efficiency of goods.

4) Quality inspection and assembly: AGV forklifts can assist in the quality inspection and assembly of new energy vehicles, automatically transporting the parts or vehicles to be inspected to the quality inspection area, and after the inspection is completed, transporting qualified parts or vehicles to the assembly area, achieving automated production processes.

In summary, AGV is required to complete material transfer in the entire new energy battery production line. Therefore, for specific production scenarios, it is necessary to construct an accurate environmental map before determining the path planning and obstacle avoidance problems of AGV in the intelligent workshop. Therefore, based on the performance parameters of a certain actual vehicle model's power battery, this chapter first divides the existing workshop layout into grids, and then uses the grid map method to construct an environmental map, providing an accurate environmental model for further AGV path planning [12].

3.1 Analysis of Production Workshop Layout

According to the technical specifications, the production process flow of the target power battery in this article is shown in Fig. 1.

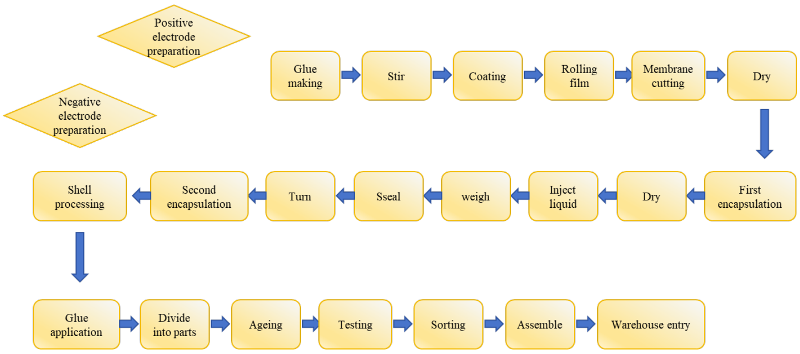


Fig. 1. Production process flowchart of power battery

The overall lithium battery manufacturing process is divided into three parts, which are manufactured in three workshops: the front-end workshop, which mainly completes the manufacturing of electrode plates, including the preparation of positive and negative electrodes, glue making, mixing, coating, film rolling, die-cutting, weighing, drying and other processes; the back-end workshop, which mainly completes the synthesis of battery cells, including stacking, shell punching, sealing, testing, weighing, drying, liquid injection, secondary weighing, sealing and other processes; and the back-end workshop, which mainly completes the packaging process including chemical conversion, sealing, shell processing, adhesive pasting, volume separation, detection, aging, sorting, module assembly, system assembly, and warehousing. Due to the high requirements for technology and safety performance of lithium batteries, many processes require specific equipment. The production goal of the front-end workshop is to manufacture the basic unit of the battery - the electrode sheet. Glue making and stirring are the processes of preparing raw materials. By putting positive and negative electrode materials and other materials into the machine, the colloids used in the subsequent coating process can be produced. The coating process is to coat the colloids obtained in the front-end process on the substrate. As the most important and valuable process in the front-end workshop, it has a great impact on the safety and durability of the battery products. The middle workshop mainly manufactures battery cells, involving equipment such as laminating machines, liquid injection machines, packaging equipment, etc. Stacking is the process of stacking the completed pole pieces into the required battery cells for subsequent processes. One sealing is to place the battery cells into an aluminum-plastic shell, position the pole ears, and seal the aluminum-plastic film into a bag with a side opening. Drying is to further remove the moisture in the battery cells by vacuum baking, and liquid injection is to inject electrolyte into the battery cells. After vacuum standing, the electrolyte is fully absorbed in the diaphragm and pole pieces, and then sealed. The subsequent processes include chemical conversion, secondary sealing, shell processing, adhesive bonding, volume separation, testing, aging, sorting, module assembly, system assembly, and warehousing. The equipment required mainly includes volume separation cabinets and testing equipment [13].

Based on the above process flow, the functional arrangement and workshop layout of the front workshop, middle workshop, and rear workshop are shown in Fig. 2.

In the picture, the outer rectangular frame represents the entire workshop, where the production equipment and corresponding operators used in the production line work. The inside of the outer frame is the processing area for each process, and the small rectangle can be regarded as the basic unit of layout. The black block represents the obstacle area, and the white area is the area where the AGV can move. A transition area is designed between the black and white areas. Firstly, based on actual operation considerations, in the workshop, there are often additional objects around the obstacles, such as workers who appear during the work process and splashing materials during processing. The purpose of setting the transition area is to enable the AGV to avoid obstacles to the maximum extent possible when planning the path, without any collision intersection.

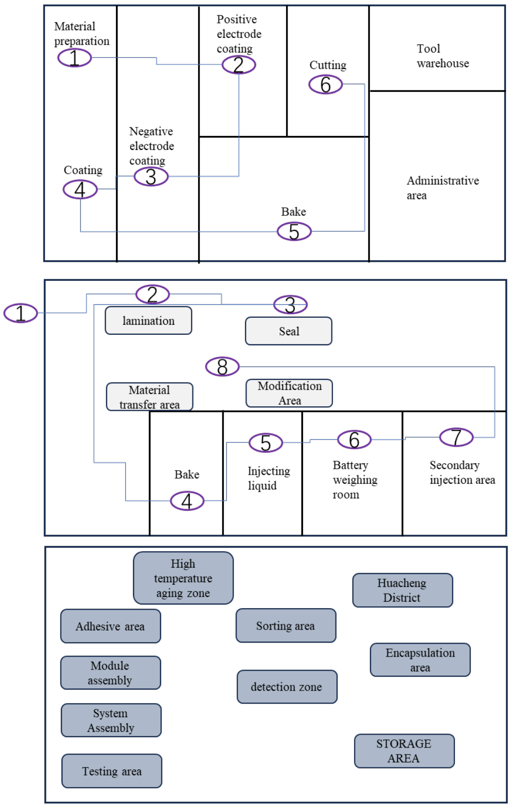


Fig. 2. Workshop layout diagram

3.2 Grid Based Environmental Modeling

In traditional grid models, the traffic speed of grid cells depends on the traffic speed of the road segments included in the grid cell area. In grid network models, there is still a large gradient in the speed of adjacent grid cells, which can lead to significant differences in travel paths between the grid network model and the network model [14]. In order to reduce the path offset error caused by the large speed difference between adjacent grid cells in the grid model, a neighborhood weighted grid modeling method is proposed [15]. After constructing the grid network, the initial traffic speed attribute values of each grid cell are obtained, and then each grid cell is traversed sequentially. For each grid cell, the weighted sum of the initial traffic speed of adjacent grid cells and its own initial traffic speed is used to correct its own traffic speed, so that the difference in traffic speed between adjacent grid cells tends to flatten. The velocity calculation formula for the corrected grid cells is:

$$V_{A,traffic-speed} = V_{A,start-speed} * \lambda_{start} + \frac{\sum V_{A^{'},traffic-speed}}{N_{A^{'}}}\left(1-\lambda_{start}\right) \quad (1)$$

In the formula, $V_{A,traffic-speed}$ represents the traffic speed of the modified grid cell A, $V_{A,start-speed}$ represents the traffic speed of the initial grid cell A, λ_{start} represents the weight of the initial traffic speed, $\sum V_{A^{'},traffic-speed}$ represents the traffic speed of adjacent grid cells, and $N_{A'}$ represents the number of adjacent grid cells.

In the grid model, the links between adjacent grid cells are constructed based on four point links, eight point links, sixteen point links, and even more link rules. These links are constructed in a way that simulates possible road directions in reality, but also creates road directions that do not exist in real roads. In order to compensate for the shortcomings of accessibility evaluation in the grid model mentioned above, when selecting a grid cell for calculation and analysis, the grid cell and the eight grids surrounding it are selected as a local area, and the road network included in this local area is called a local subnet. Therefore, this article focuses on the process of constructing a grid for the power battery workshop as follows:

Step 1: Based on the previous grid model, obtain the grid cells and their attribute data in the grid model.

Step 2: Select a grid cell A in sequence, obtain the road network data of the local subnet of grid cell A_i , and calculate the number of connected nodes N_i of the local subnet.

Step 3: Divide the grid cell into N_{ri} grids r_{ij} based on the number N_{ri} of connected clusters to which the road network nodes in grid cell A belong, where $N_{ri} \leq N_i$ represents the number of connected clusters in the local subnet to which j belongs. As shown in the figure, the local subnet contains three connected clusters, and the grid cells contain road network nodes belonging to connected clusters 2 and 3.

Step 4: Each grid block r_{ij} of the grid unit inherits the original grid unit attribute values, including latitude, longitude, and traffic speed, and records the nodes in the network model of the roads contained in grid block r_{ij} through vector V_{ij} .

Step 5: Determine the grid nodes for improving the grid network model. Mark the sequence numbers of all grid cells' grid blocks, and replace the center node of the grid block with the center node of the grid cell in the grid network model as the node in the improved model.

Step 6: Establish links in the improved grid network model based on local subnets and long lines. After partitioning all grid cells based on local subnet connected clusters, the grid block nodes are traversed sequentially. The grid blocks of the grid cell are linked to the grid blocks of neighboring grid cells in eight directions belonging to the same local subnet connected cluster. The neighboring grid blocks that do not belong to a connected cluster are not linked.

Step 7: Build a local subnet to improve the weights of links in the grid model. After establishing the links in the improved grid network model based on local subnets and long lines, the weights of the links between grid blocks inherit the properties of the original grid cells.

After the above process, this article divides the working area of the mobile robot into grids based on its specific situation during task execution. The numbers $A_1 - A_n$ from the top left corner to the bottom right corner, from left to right, and from top to bottom respectively, specify that the robot moves on the center point of the grid, and the grid coordinates are represented by the center point. The movable area is marked in white, and the robot can pass through it. Black is the restricted area, indicating the presence of obstacles on the path. When the robot moves to a certain grid in the picture, it can move in any direction near its current position by removing its previous movement direction. The grid is shown in Fig. 3.

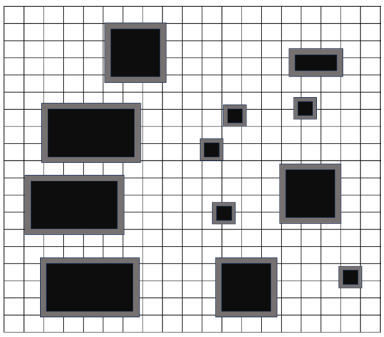


Fig. 3. Environmental map model constructed by grid map method

The grids in the map need to be numbered, and the numbering rule is as follows:

$$\begin{cases} x = \text{mod}(i, H) - 0.5 \\ y = H + 0.5 - ceil(i / H) \end{cases} \quad (2)$$

In the formula, H is the number of rows and columns in the grid map, and i is the sequence number. In the map, the meanings of each area are shown in Fig. 4.

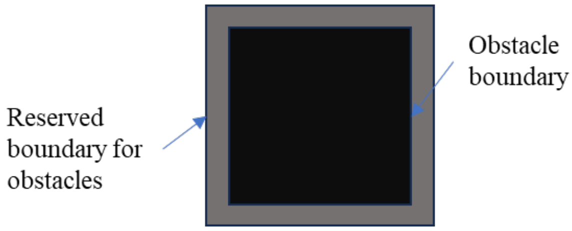


Fig. 4. Schematic diagram of the meanings of each color and region in the graphic

In the manufacturing workshop of new energy vehicle power batteries, raw materials, components, etc. can be divided into discrete units for production, which usually need to be combined, assembled, processed, or handled during the manufacturing process. In order to verify the superiority of subsequent intelligent algorithms over traditional algorithms, this paper designs a simulated two-dimensional map environment for AGV path planning in a discrete manufacturing system using the grid method based on Python language. The eight blocks on the periphery of the map are equivalent to different workstations in reality. The grid pixels in the simulation environment are 20×20, and the size of the grid environment is 19×19. In the initialization settings, set the upper left corner of the grid map environment as the origin of the coordinates, the horizontal direction as the x-axis, and the vertical direction as the y-axis. Define the starting position as (0,0), the target workstation position as (17,17), the non shaded area as the movable area, and the shaded area as the obstacle or congested module.

4 AGV Path Planning Algorithm

This article uses deep Q-learning [16] to implement path planning, but deep Q-learning has inherent flaws in the path planning process:

1) Deep learning algorithms typically require a large amount of datasets and computing resources to support their continuous learning and adaptation, and the lack of sufficient data and computing resources can reduce the effectiveness of deep learning methods.

2) The theoretical framework structure is not strong enough: there are many unresolved issues in its theoretical framework, such as exchangeability, normativity, and interpretability.

3) Relatively weak interpretability: The complexity and black box nature of deep learning algorithms result in relatively weak interpretability, making it difficult to understand how the algorithm works internally.

4) Overfitting problem: Deep learning models are prone to overfitting on training data, resulting in poor performance on unseen data. For example, when AGV encounters dynamic interference in path planning, regularization and other techniques are needed to alleviate it.

5) The training process is complex: the algorithm has relatively low efficiency, requires a large amount of computing resources and time, and is prone to getting stuck in local optima.

6) Sensitive to noisy data, high-quality data preprocessing and cleaning work is required.

7) High hardware requirements: Deep learning algorithms typically require high-performance hardware support, especially GPU accelerated hardware environments, to handle the training process of complex models, and higher hardware requirements bring increased manufacturing costs to enterprises.

Therefore, regarding the environmental data presented in this article, the following issues need to be addressed in actual AGV path planning:

1) In response to the curse of dimensionality and low learning efficiency in unknown environments, as well as poor robustness of planning strategies in deep Q-learning, this paper proposes a heuristic deep Q-learning approach [17]. Heuristic deep Q-learning introduces trajectory pool to update data in the experience pool based on the traditional DQN algorithm, promoting the selection of effective actions in the trajectory pool by the DQN algorithm.

2) In order to solve the problems of low learning efficiency and poor robustness of planning strategies in unknown environments, this paper improves the directional reward function [18]. During the process of AGV from the starting point to the target workstation, it will experience many different states. If the environmental feedback obtained from the state closer to the endpoint and the state farther away from the endpoint are the same, it is difficult to guide the AGV intelligent agent to learn the optimal strategy. This article first enlarges the overall

algorithm structure improved by the rewards obtained by the AGV intelligent agent in different states. Then, in order to express the position relationship between the AGV and the obstacles in the environment and the target workstation. Introducing a heuristic factor, the setting of the heuristic factor reflects the Euclidean distance between the current state of the AGV and the target workstation, as well as the total Euclidean distance between the current state of the AGV and all obstacles in the map environment, and the proximity of the penalty state [19]. The overall algorithm structure after improvement is shown in Fig. 5.

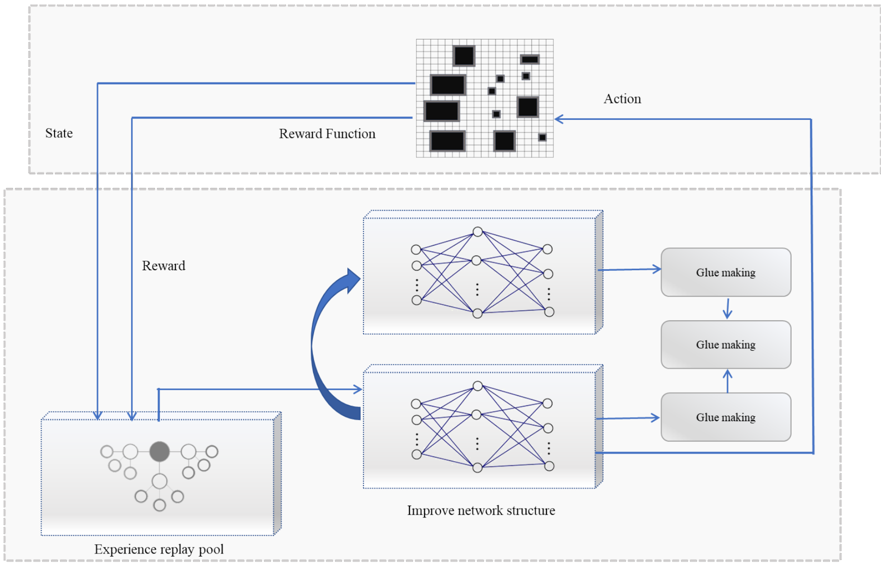


Fig. 5. Improved deep Q-learning algorithm framework

4.1 Improvement of Network Structure

Learn from the DQN algorithm proposed by the DeepMind team, which uses Q-values to train neural networks, and then uses the trained neural network to replace the Q-value table. Using neural networks to approximate the Q-value table and complete the calculation of Q-values. The Q network with parameter w is defined as:

$$Q\left[s,\left(a_1,a_2,\cdots,a_n\right)\middle|w\right] \approx Q^{\pi}\left[s,\left(a_1,a_2,\cdots,a_n\right)\middle|w\right] \tag{3}$$

In the formula, a_i represents the i-th action selected by the robot, and s represents the state. If the current state s is taken as the input value, the state space of the robot is represented as:

$$S=\left[p_1,p_2,\cdots,p_n,r_1,r_2,\cdots,r_m\right] \tag{4}$$

In the formula, p_i, $i\in[1,n]$ represents the position coordinates of the robot, including the current coordinate information of the robot in the grid, and r_j, $j\in[1,m]$ represents the completion status of the robot at task point j. The representation method is as follows:

$$r_j=\begin{cases}0 & \text{Unfinished task}\\ 1 & \text{Complete the task}\end{cases} \tag{5}$$

State s is a $n+m$-dimensional state space, where the output vector is the Q value of all joint actions in that state. The mapping structure diagram is shown in Fig. 6.

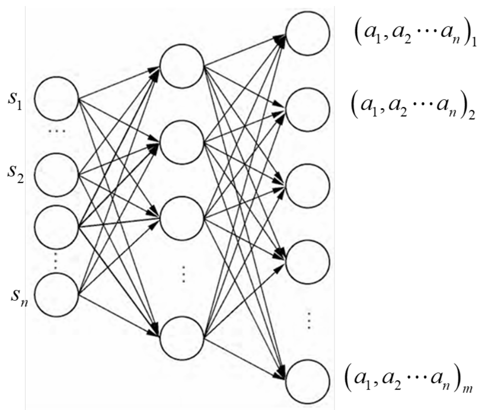


Fig. 6. Network mapping structure diagram

In this article, two neural networks with the same structure but different parameters, Q_{train} and Q_{copy}, are added to the original algorithm network. The Q_{train} network is used to participate in each algorithm training and update the parameters. The Q_{copy} network is used to directly copy the parameters of the $Q_{\_eval}$ network to the Q_{train} network at regular intervals. The improved algorithm training flowchart is shown in Fig. 7.

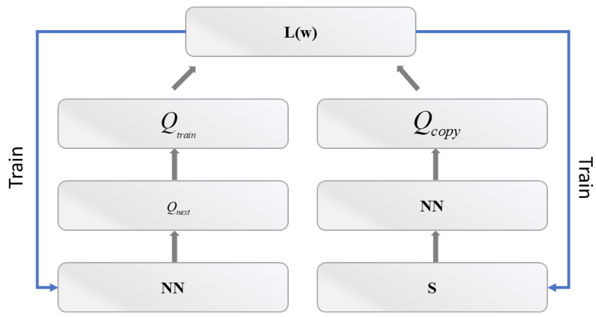


Fig. 7. Improved algorithm training flowchart

4.2 Design of Directional Reward Function

How to safely and reasonably avoid obstacles when AGVs work in unknown environments with obstacles is a crucial issue. In practical work, the relative motion directions of AGV and obstacles/targets overlap closely, which increases the difficulty of obstacle avoidance. Therefore, it is necessary to set a reasonable direction selection strategy for AGV. This article refers to Coulomb's law to model the directional reward function. The attraction and repulsion between charges have a good fit with AGV trajectory planning tasks in obstacle environments. The relationship between obstacles and AGVs can be expressed as mutual repulsion between the same type of charges, while the relationship between target points and AGVs can be seen as mutual attraction between different types of charges. The principle diagram of the reward function is shown in Fig. 8. AGV, $T^{'}$ is the attraction vector of the target point, and AGV, $O^{'}$ is the repulsion vector of the obstacle. The mathematical expressions for the two are as follows:

$$\begin{cases} T^{'} = \dfrac{Q_{e,AGV} * Q_{e,tar}}{l_1^2} \dfrac{T}{|T|} \\ O^{'} = \dfrac{Q_{e,AGV} * Q_{e,obj}}{l_2^2} \dfrac{O}{|O|} \end{cases} \tag{6}$$

In the formula, l_1 is the relative distance from the end of the AGV to the target point, l_2 is the relative distance from the AGV to the obstacle, $Q_{e,AGV}$ is the "charge amount" of the AGV, $Q_{e,obj}$ is the "charge amount" of the obstacle, and $Q_{e,tar}$ is the "charge amount" of the target. In practical work, the attraction effect of the target point on the AGV should be greater than the repulsion effect of obstacles, otherwise it may cause the AGV to be unable to reach the target point in order to avoid obstacles. Setting $Q_{e,tar}$ to twice $Q_{e,obj}$ ensures that the AGV can both avoid obstacles and complete tasks. AGV, B represents the expected relative motion direction, AGV, C is the actual motion vector at the end of the robotic arm, α is the angle between AGV, B and AGV, C, used to measure the degree of fit between the current motion vector and the motion vector planned by the agent. The smaller α, the higher the fit. The expression is as follows:

$$\alpha = \arccos \frac{\left(AGV,O^{'} + GV,T^{'}\right) * AGV,C}{\left|AGV,O^{'} + GV,T^{'}\right|\left|AGV,C\right|} \tag{7}$$

Therefore, the directional reward function is as follows:

$$J(\alpha) = \beta - \frac{\alpha \cdot \pi}{180} \tag{8}$$

β is the forward compensation parameter, obtained through experimental experience, and the value of β is set to 0.815.

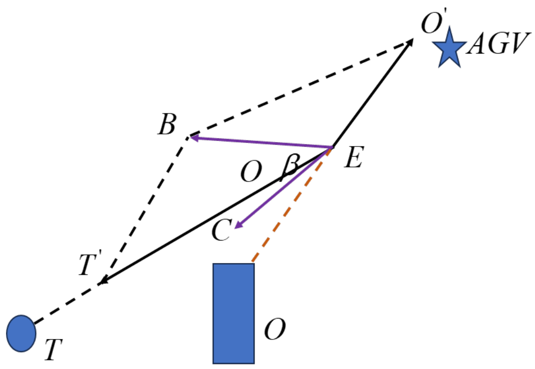


Fig. 8. Schematic diagram of directional reward function

After improvement, this section has achieved structural improvements for the deep Q-learning algorithm, and the improved network structure has advantages in parameter update speed and efficiency. In order to correctly guide AGV to the predetermined target, this paper improves the reward function. The principle of interaction between charges is used as the design basis for the reward function in this section, and the modeling method is based on Coulomb's law. The attraction and repulsion between charges have a good fit with the AGV trajectory planning task in obstacle environments. The relationship between obstacles and AGV can be expressed as mutual repulsion between the same type of charges, while the relationship between the target point and AGV can be regarded as mutual attraction between different types of charges. Finally, the overall improvement of deep Q-learning is completed. The improved algorithm needs to be verified in real simulation cases, so the next step is to design simulation experiments.

```
Pseudo code for improving deep Q-learning
// (1) Initialize the experience replay buffer D with size K
Initialize experience replay buffer D, size K
// (2) Initialize the predictive Q-network Q(θ) and the target Q-network Q'(θ'), set θ' = θ = initial
parameters
Initialize predictive Q-network Q(θ)
Initialize target Q-network Q'(θ')
Set θ' = θ = initial parameters
// (3) Train for M episodes
For episode from 1 to M:
  // (4) Train for T steps within each episode
  For t from 1 to T:
    // (5) For a given state S_t, execute action a_t based on the network Q, using a given action
selection policy
    // Use Q-values to train the neural network and replace the Q-table with the trained neural net-
work
    State S_t = getCurrentState()
// Update the network using r_t and S_{t+1}, and perform parameter updates
    Update Q-network parameters θ (using r_t, S_{t+1}, and other necessary information)
      // (7) Store (S_t, a_t, r_t, S_{t+1}) in the experience replay buffer D
    Add (S_t, a_t, r_t, S_{t+1}) to D
      // (8) Randomly sample a batch of size d (S_j, a_j, r_j, S_{j+1}) from D
    // Set the target value y_j = r_j + γ * max_a Q'(S_{j+1}, a; θ')
    Batch samples = randomly sample d samples from D
    For each sample (S_j, a_j, r_j, S_{j+1}) in the batch:
      Compute target value y_j = r_j + γ * max_a Q'(S_{j+1}, a; θ')
        // (10) Perform gradient descent on the parameters θ in (y_j - Q(S_j, a_j; θ)) to approximate
Q to y_j
    Execute gradient descent to update θ to minimize the loss function L = average of (y_j - Q(S_j,
a_j; θ))^2
     // (11) Reset θ' = θ every N steps
  If episode % N == 0:
    θ' = θ
// (13) End for loops
```

5 Simulation Experiment and Result Analysis

The power battery workshop is manufactured in separate and discrete units. In order to verify the comparison between the improved deep *Q*-learning algorithm and the original algorithm, this paper is based on Python language. In the grid map constructed in Chapter 3, the target position is set as (17,17).

Build the network structure of the improved deep *Q*-learning algorithm proposed in this article using TensorFlow and train it. The configuration of the training environment is shown in Table 1.

Table 1. System hardware configuration and algorithm parameter settings

Parameter	Parameter values
CPU	R7
CPU architecture	Zen 5
Number of CPU cores	8
CPU thread	16
Operating System	Windows 11
System Memory	16GB
GPU	RTX4060T
Python version	3.11.1
Learning rate	0.01
Reward decay rate	0.85
Explore factors	0.1
Experience pool size	6000
Batch processing quantity	200
Priority probability offset rate	0.7
Probability offset rate of priority weight	0.5

Two algorithms were used to conduct 5000 rounds of AGV path planning experiments in the same simulation environment. The results are shown in Fig. 9. From the comparison in the figure, it can be intuitively seen that the improved algorithm has better smoothness in the planned path and can use AGV more efficiently and energy-saving in manufacturing systems. The path planned by traditional DQN algorithm is "too careful" to turn or change lanes in advance when approaching obstacles, resulting in excessive path angles. This is a manifestation of the overestimation problem of traditional DQN algorithm: after encountering obstacles and being punished during training, once there is an action that can be avoided, it is easy to overestimate its Q value and not explore other effective paths.

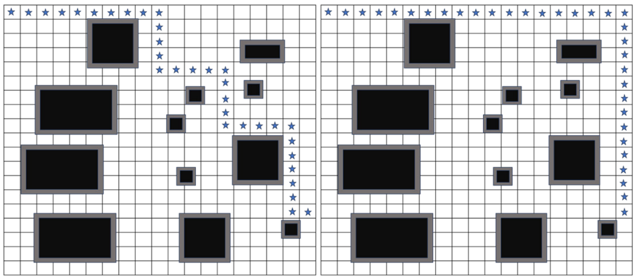
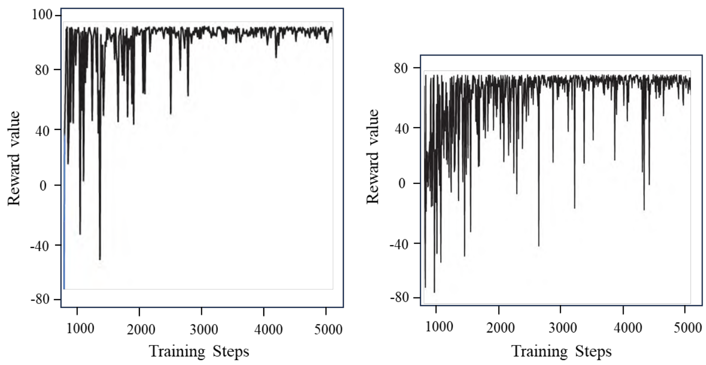


Fig. 9. Comparison effect of planned paths

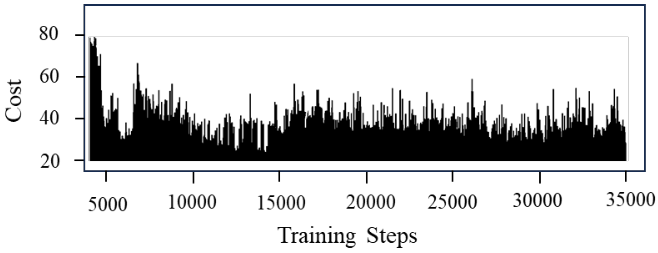
The performance of deep Q-learning algorithms is generally measured by the reward value of the algorithm, as the core idea of the algorithm is to continuously obtain rewards or punishments from the environment through AGV agents to train behavioral strategies. The improved algorithm obtains a higher cumulative reward value, and the improved deep Q-learning algorithm can obtain the maximum cumulative reward value in the environment after interacting for about 1500 rounds, while the original algorithm only starts to receive the maximum reward after about 4000 rounds. The comparison results of reward values before and after algorithm improvement are shown in Fig. 10.



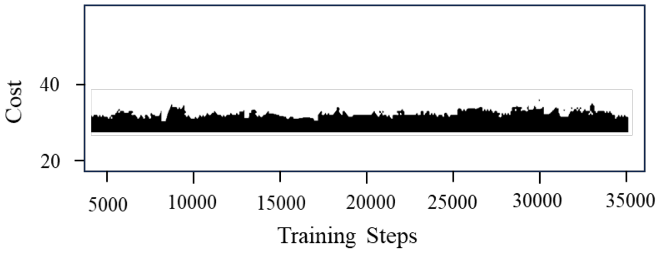
(a) Improved (b) Before Improved

Fig. 10. Trend chart of reward value changes

The judgment of the output results of deep Q-learning algorithm is the same as other algorithms that use neural network backpropagation, that is, the lower the cost of the algorithm, the closer the network output results are to the target value, and the smaller the oscillation amplitude of the cost curve, the more stable the performance. From Fig. 11, it can be seen that compared to the original deep Q-learning algorithm, the cost value of the improved deep Q-learning algorithm is significantly reduced, and the amplitude of oscillation is very small, indicating that the output results of the improved algorithm are closer to the target value, with good numerical performance and stable algorithm performance.



(a) Before Improved



(b) Improved

Fig. 11. Comparison results of algorithm cost and training steps

On the other hand, since the improved algorithm no longer randomly samples training data, but obtains more data that needs to be trained through priority weight sampling, it can be seen from the change in the number of steps required for algorithm training shown in Fig. 12 that the average number of steps for the improved algorithm has been reduced by about 40%, indicating an improvement in the convergence speed of the improved algorithm. The convergence effect of the algorithm before and after improvement is shown in Fig. 12.

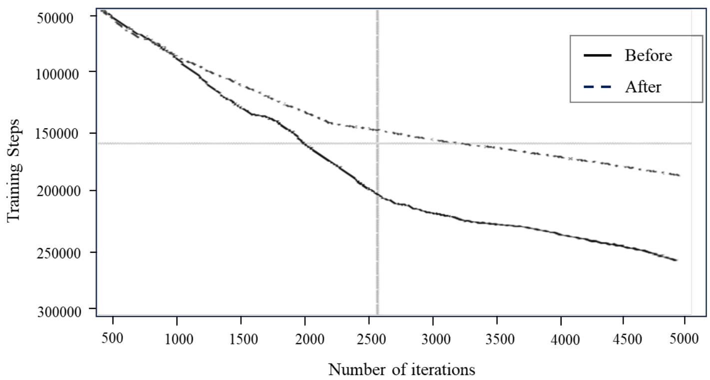


Fig. 12. Comparison chart of algorithm convergence effect

From the graph, it can be seen that both before and after the improvement of the deep Q-learning algorithm, as the number of training epochs increases, the algorithm is in a convergence state. However, after the improvement, the performance of the algorithm begins to show differences when the number of training epochs reaches 900. As the number of training epochs increases, the convergence speed and effect of the algorithm before and after the improvement are the same before 900 epochs. After 900 epochs, the convergence speed of the improved algorithm is significantly accelerated, and the convergence of the improved algorithm reaches the expected effect after about 4000 epochs of training. The algorithm before improvement has not yet achieved good convergence within 5000 training epochs.

In summary, by simulating the path planning, a better AGV path was obtained. The improved path enables the AGV to reach the target position at the fastest speed, saving production costs and reducing carbon consumption. At the same time, the performance parameters, reward values, costs, and training rounds of the algorithm itself were compared to demonstrate the improvement in algorithm performance before and after the improvement.

6 Conclusion

This article mainly studies the path planning of AGV vehicles in the existing production workshop layout. Firstly, the production process of new energy vehicle power batteries is analyzed, and then the layout inside the workshop is determined. The grid method is used to construct an environmental map, and the shortcomings of the grid method are analyzed. The neighboring weighting strategy is used to improve the grid map. During the path planning, the shortcomings of the existing deep Q-learning algorithm are analyzed. Due to the dynamic changes in the layout of the new energy production process, employee walking, and other dynamic environmental changes of logistics vehicles and transportation equipment, the curse of deep Q-learning dimensions and the low learning efficiency in unknown environments, as well as the poor robustness of planning strategies, this article proposes a heuristic deep Q-learning method, Heuristic deep Q-learning introduces trajectory pool to update the data in the experience pool based on the traditional DQN algorithm, which promotes the selection of effective actions in the trajectory pool by the DQN algorithm. In order to solve the problems of low learning efficiency and poor planning strategy robustness in unknown environments, this paper improves the direction reward function. The specific research results of this paper are as follows:

1) The construction of the workshop environment map has been completed, and the grid map method was used in the map construction process. At the same time, an improved grid method was used to address the directional deviation that occurs in traditional grid map construction, enhancing the guiding role of the grid map;

2) In terms of path planning, AGV path planning is completed based on deep Q-learning algorithm, and an improved algorithm is proposed to address the problems of dimensionality disaster and poor planning robustness of traditional algorithms;

3) We constructed a simulation experiment section and verified the algorithm's improvement in path planning performance through experiments. At the same time, we completed path planning simulation experiments using existing workshop grid maps.

This article studies the AGV path planning problem using deep reinforcement learning, and the methods and research results also demonstrate the superiority of the algorithm. However, there are still shortcomings, mainly reflected in the following aspects:

1) The research work in this article is based on a static environment. In the environment where AGVs are located, the size and position of obstacles are fixed. However, in the real world, the environment where mobile robots are located is mostly a dynamically variable environment. Therefore, for path planning, dynamic and variable environments should be further studied.

2) The problem of multi robot collaborative path planning, with the popularization of AGVs in various industries, will involve multiple AGVs working together in the production process of power batteries. Therefore, how to cooperate and cooperate among AGVs to achieve obstacle avoidance, and how to allocate tasks among AGVs, all require deeper research.

3) The experiments conducted in the simulation environment have proven the effectiveness of the improved algorithm both theoretically and experimentally. However, this paper lacks real running experiments. Further research should not only make corresponding improvements at the algorithm level, but also conduct experimental verification in real environments.

7 Acknowledgement

Project of Hebei Provincial Department of Education: Research on Path Planning of Mobile Robots Based on Deep Q-Learning (ZC2021222).

References

[1] B.-D. Sun, D. Shu, H.-D. Fu, D.-H. Wang, L.-M. Peng, X.-Y. Wang, Y.-Y. Zhu, H.-M. Wang, W.-J. Yu, J.-X. Xie, Intelligent Manufacturing for High-End New Materials: Opportunities and Directions, Strategic Study of CAE 25(3) (2023) 152-160.

[2] K. Xie, P. Lu, J.-Y. Sheng, J.-H. Xiao, H.-B. Sun, Artificial Intelligence, Product Innovation and Adaptive Upgrading of Manufacturing Industry, Journal of Beijing Jiaotong University(Social Sciences Edition) 23(1)(2024) 84-95.

[3] X. Tao, J.-H. Bao, H.-R. Li, Structural Design and Research of AGV Vehicle for Transmission in Cylinder Head Assembly Station, Mechanical Engineering & Automation (2)(2022) 86-89.

[4] Y.-Q. Liao, Z.-Y. Xie, Design and application of separated AGV system in solar cell workshop, Modern Machinery (3) (2022) 63-66.

[5] J.-H. Wang, L. Huang, Y.-T. Shi, X.-Q. Zhang, L.-J. Qi, Design of robot autonomous navigation system based on optimized Hector-SLAM algorithm, Chinese Journal of Engineering Design 30(6)(2023) 678-686.

[6] J.-Z. Lv, W.-J. Qiao, S. Liu, Research on the Method of AGV Path Planning With Multiple Target Station Based on Expansion Grid, Machine Design & Research 39(6)(2023) 6-10+21.

[7] Y. Zhou, L.-Z. Wong, X.-W. Quan, Multi-AGV Path Planning Method Based on Improved A* Algorithm and Time Window Detection Method, Ship Engineering 46(1)(2024) 121-126.

[8] Y.-S. Huang, X.-F. Yao, X. Jing, X.-Y. Hu, DQN-based AGV path planning for situations with multi-starts and multi-targets, Computer Integrated Manufacturing Systems 29(8)(2023) 2550-2562.

[9] Y.-H. Chen, M. Su, J.-J. Bi, Y. Liang, T.-H. Li, Path planning algorithm of mobile robot based on improved JPS algorithm, Electronic Measurement Technology 46(7)(2023) 88-93.

[10] W.-D. Zhao, D.-C. Zhou, An Improved Path Planning Algorithm Based on the Fusion of A* and Third-order Bessel Curves, Journal of Anhui University of Technology(Natural Science) 40(3)(2023) 333-338.

[11] W. Xu, Z. Cheng, L. Zhu, Y.-H. Zhang, A local path planning algorithm based on improved artificial potential field method 45(19)(2022) 83-88.

[12] X.-J. Zhao, H. Ye, W. Jia, Z.-X. Sun, Survey on AGV Path Planning and Obstacle Avoidance Algorithms, Journal of Chinese Computer Systems 45(3)(2024) 529-541.

[13] F. Liu, K.-Q. Yu, Y.-M. Xu, The key production processes and control points of Li-ion battery, Battery Bimonthly 50(4) (2020) 376-379.

[14] S.-J. Liu, Z.-H. Li, Z.-X. Liu, K. Zou, Research on spatial modeling method for excavation tunnels based on laser scanning and 3D grid map, Chinese Journal of Engineering Design 30(3)(2023) 306-314.

[15] J.-J. Cai, X.-L. Zhang, A Correlation Tracking Method Based on Weight Maximum Close Distance, Optics & Optoelectronic Technology 14(3)(2016) 105-108.

[16] S.-Y. Li, Y.-M. Ma, J. Liu, Smart shop floor scheduling method for equipment load stabilization based on double deep Q-learning network, Computer Integrated Manufacturing Systems 29(1)(2023) 91-99.

[17] Z.-Y. Zhang, Y.-F. Chen, Y.-H. Wang, G.-S. Feng, Multi-robot task allocation algorithm b Multirobot task allocation algorithm based on heuristically accelerated deep Q network, Journal of Harbin Engineering University 43(6)(2022) 857-864.

[18] W. Ran, H. Chen, J.-X. Yang, Y. Nishimura, C.-P. Guo, Y.-Y. Yin, Design Method of Motion Planning Reward Function Based on Utility Theory, Automotive Engineering 45(8)(2023) 1373-1382.

[19] D.-J. Fang, B.-B. Zhou, C.-M. Zhao, R. Schmidt, Research on AGV Path Planning Based on Deep Q Learning under Discrete Manufacturing System Environment 42(6)(2023) 45-51.