Research on Path Planning Based on Q-learning

Yihui Qiu *

College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, 310024, China.

* Corresponding Author Email: dzhen@ldy.edu.rs

Abstract. With the advancement of technology, the path planning and navigation technology of robots is more mature than before. However, as the environment becomes increasingly complex, the limitations of traditional navigation methods are becoming more prominent. To overcome this challenge, reinforcement learning was introduced in the path planning problem. As a typical reinforcement learning algorithm, Q-learning does not rely on models. Among them, factors such as learning rate and discount coefficient have a significant impact on Q-learning results. Based on this, this article investigates the influence of Q-learning learning rate and discount factors on robot path planning in certain maps, explores the trend of the number of iterations required for robot training changing with learning rate and discount factors, and reflect the trend of the number of iterations changing with the learning rate and discount factor on the line graph, thus obtaining a universal conclusion, aiming to provide ideas for future research in Q-learning.

Keywords: Path planning; Reinforcement learning; Q-learning.

1. Introduction

Path planning is one of the key technologies for autonomous navigation of robots. Traditional path planning methods mostly require obtaining map information in advance before planning, making it difficult to handle unexpected situations and having poor robustness. Based on this, reinforcement learning has been introduced to overcome the impact of complex environments on robot path planning, which can be divided into two aspects: model-based reinforcement learning and reinforcement learning [1]. Among them, reinforcement learning does not rely on the environment and directly obtains empirical data through interaction with the learning environment, which is suitable for situations where the environment is unknown. Among them, the most classic reinforcement learning algorithm is Q-learning. Q-learning is widely used in daily life, and in many games, the behavior of AI opponents is obtained through the Q-learning algorithm, which trains AI by simulating the reward value of the game. In the auto drive system, Q-learning can be used to learn the driver's behavior, such as turning, lane changing, etc., and can solve the path planning problem in automatic driving.

Previously, many scholars have conducted research on Q-Learning, such as Lin et al. using Q-Learning for dynamic path planning of robots, enabling them to avoid obstacles and reach the endpoint in simulated environments [2]. Based on this, Mahmoud et al. explored an improved geometric Q-Learning algorithm that enables cars to avoid all obstacles and find the shortest path in a simulated environment [3]. However, in some complex environments, Q-learning algorithms can consume a significant amount of time and storage space when constructing Q-tables, which greatly reduces the efficiency of the algorithm. Based on this, Yun et al. proposed a method that combines Q-learning with artificial neural networks (ANN), which enables robots to autonomously learn in completely unknown situations and improve the efficiency of path planning [4]. Huang et al. proposed combining Deep Q-Network (DQN) with the Artificial Potential Field Method for path planning of Automated Guided Vehicles (AGVs), reducing network convergence time and improving AGV performance [5]. However, the DQN algorithm still faces the problem of overestimation. To address this issue, Jiang et al. proposed the Deep Double Q-Network (DDQN), which can more accurately evaluate the value function and perform better in complex environments [6].

However, current research has limited exploration of the parameters in Q-learning algorithms. During the Q-learning process, each parameter affects the navigation performance of the robot in the final iteration, with the most critical parameters being the learning rate and discount factors (α and γ).

Therefore, exploring the impact of learning rate and discount factors on the results is of great significance for robot path planning.

This article focuses on the impact of learning rate and discount factors on the navigation performance of robots in Q-learning. The article focuses on the influence of parameter changes on the number of iterations required for robot training, and proposes to explore this impact using equidistant value-added methods. The results are analyzed to explore the patterns involved.

2. MethodologySection Headings

2.1. Principle of Q-learning

Reinforcement learning is a type of machine learning method that learns through interactive feedback from the environment [7]. Observing the environmental state after taking an action in an unfamiliar environment and receiving feedback from the environment. As a classic method of reinforcement learning, Q-Learning is a value reinforcement learning algorithm that selects the next action by learning a Q-value function that measures the expected reward of taking an action in a given state. When exploring the environment in Q-Learning, it will receive positive or negative feedback. When learning positive behavior, it will receive rewards, and when learning negative behavior (such as hitting obstacles), it will be punished [8]. The relevant formulas for Q-Learning are as follows:

$$\mathrm{newQ_{S,A}} = (1-\alpha)\mathrm{Q_{S,A}} + \alpha\left(\mathrm{R_{S,A}} + \gamma * \mathrm{maxQ'(s', a')}\right) \quad (1)$$

Where α is the learning rate, and γ is the discount factor. This formula represents the new Q prediction value, represents the proportion of the old Q value in the new Q value, and represents the reward brought by this learning.

In this study, the map adopts a grid design. This can promote the establishment of a Q-table and improve the performance of the algorithm [9]. Consider the robot as a particle, possessing the ability to freely move within the four surrounding grids in a gridded map, namely up, down, left, and right, as shown in Fig. 1.

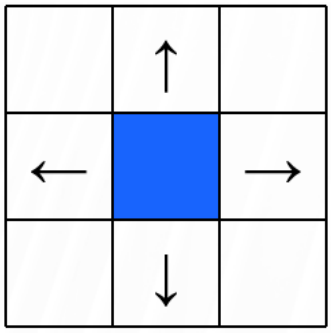


Fig 1. The action of a robot. (Picture credit: Original).

2.2. Assumption

The following assumptions can be made through formula 1:

Assumption 1: The larger the alpha value, the clearer the planned path result and the fewer iterations required.

Assumption 2: A lower gamma value will make it difficult for the path to continue to the endpoint.

2.3. Research Process

The core method of this study is the control variable method, and simulation is conducted through the Robotics playground extension package in Matlab. Firstly, create a map based on the actual environment (such as logistics storage stations, mazes, etc.), and then debug the learning rate and discount factor (fixed gamma value to study alpha value, fixed alpha value to study gamma value) on

the already created map. Record the number of iterations required for each training process, and record five iterations of data for each set of alpha and gamma values to reduce the impact of accidental errors. Draw a line chart to explore the effects of alpha and gamma values. In iteration, it is necessary to ensure that the maximum number of iterations is greater than the number of iterations required for each training process to avoid situations where the robot cannot reach the endpoint due to insufficient iterations. After each iteration, initialize the Q matrix to ensure that the results of each iteration are independent.

3. Results and Discussions

3.1. Map Analysis and Performance Analysis

Figs. 1 and 2 are maps created in Matlab based on the actual environment of the logistics warehouse station and the maze.

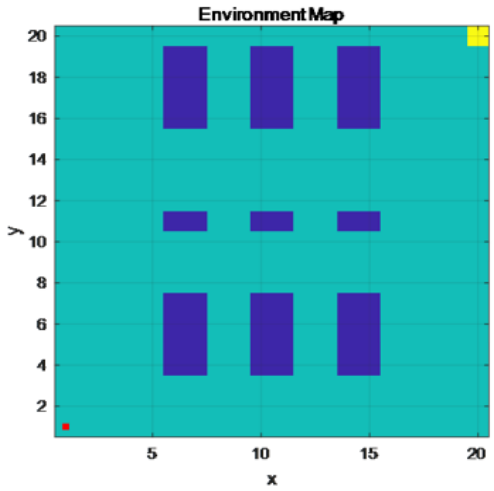
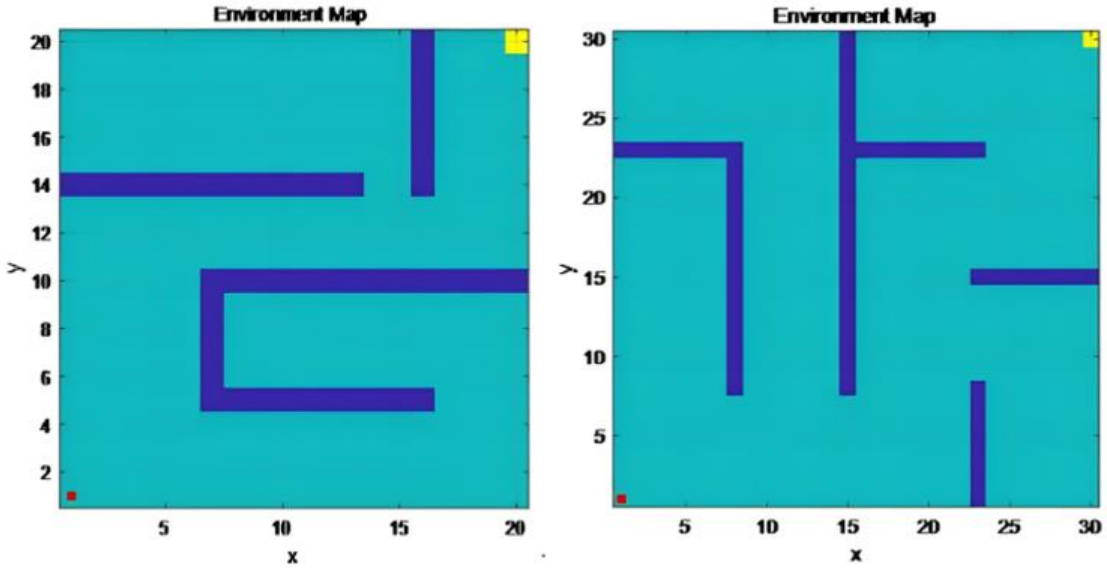


Fig 2. Shelves at logistics storage stations. (Picture credit: Original).



(a) (b)

Fig 3. Map of maze. (a) Maze 1; (b) Maze 2 (Picture credit: Original).

It can be observed that compared to the logistics storage station in Fig. 2, the two maps in Fig. 3 have blind spots that may prevent the robot from reaching the destination. Fig. 3 (b) expands the map from 20 * 20 to 30 * 30, further testing the performance of the robot.

Among them, the red dot in the lower left corner is the starting point, the yellow area in the upper right corner is the target point, and the following rewards are set on these maps:

$$\text{Rewards} = \begin{cases} \text{Goal:} & 1 \\ \text{Nothing:} & 0 \\ \text{Obstacles:} & -1 \end{cases} \quad (2)$$

Using Formula 2 as the reward function, encourage the robot to move towards the target point. When the robot reaches the target point, a reward is given. When the robot walks on obstacles or encounters map boundaries, a negative reward is given. If it does not reach the target, no reward is given.

When using Matlab for simulation, the alpha value is set to 0.6 and the gamma value is set to 0.9. The trained robot path is as follows:

As shown in Figs. 4 and 5, the time on the right represents the time when the trained robot reaches the endpoint, which depends only on the size and design of the map. Therefore, time cannot be used as a criterion for evaluating the performance of the trained robot. However, robot performance can be evaluated by the number of iterations required by the robot. A smaller number of iterations required by the robot means that it can react faster and develop strategies to reach the endpoint. In order to investigate the specific impact of different parameter changes on robot performance, the control variable method can be used for simulation. This method fixes all other variables and changes only one parameter to observe the specific impact of parameter changes on robot performance.

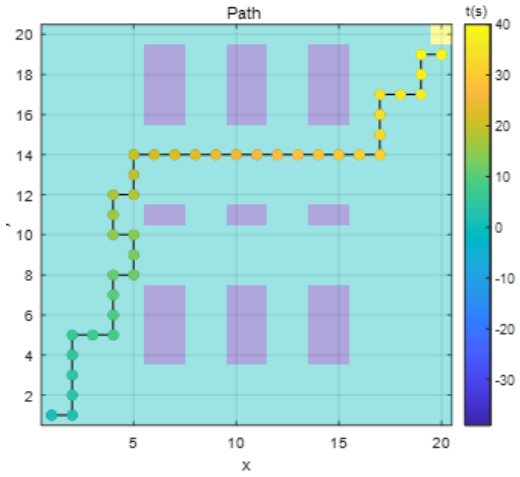
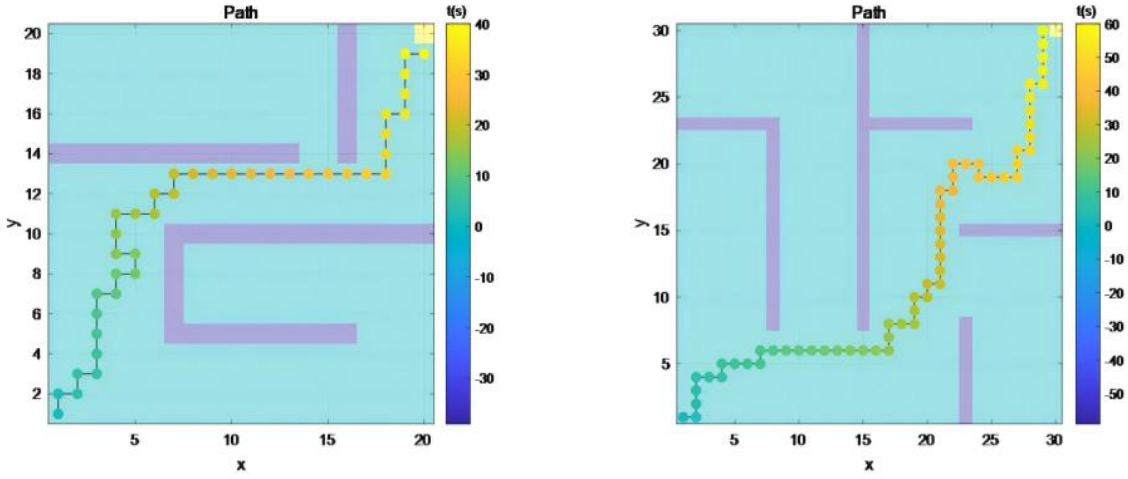


Fig 4. Path planning result of shelves. (Picture credit: Original).



(a) (b)

Fig 5. Path planning results of Maze 1 & 2 (Picture credit: Original).

3.2. Parameter Analysis

According to the analysis in section 3.1, in order to make the results more convincing, the equidistant incremental method can be used to record the number of iterations. The fixed gamma

value is 0.6, and then the starting value of alpha is set to 0.9, gradually decreasing in steps of -0.1 until the alpha value reaches 0.1. Fix the alpha value to 0.6, set the starting value of gamma to 0.9 and reduce it to 0.1 with a step size of -0.05, and record the number of iterations required for the robot to complete under different conditions.3.2.1 Alpha analysis

When studying the impact of alpha on robot performance, exploration can be conducted by analyzing the average number of iterations required during robot training for each set of data. This method can better explore the role and influence of alpha in the robot training process. By collecting and organizing relevant data, a line graph of the average number of iterations can be drawn, as shown in Fig. 6.

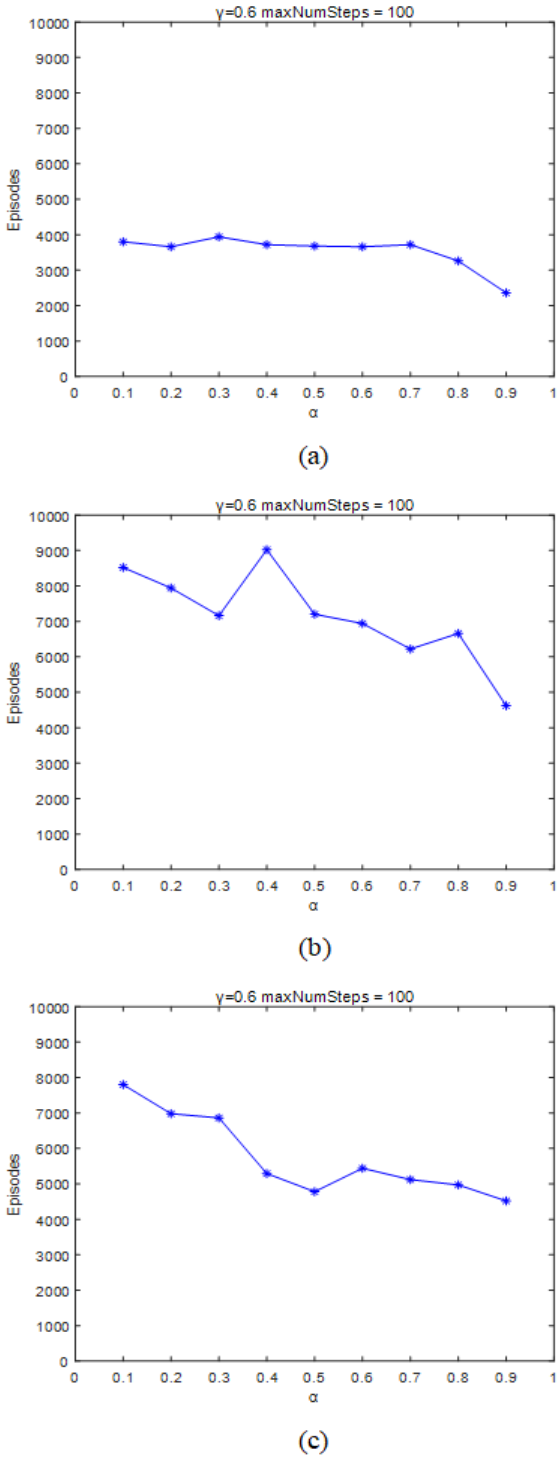
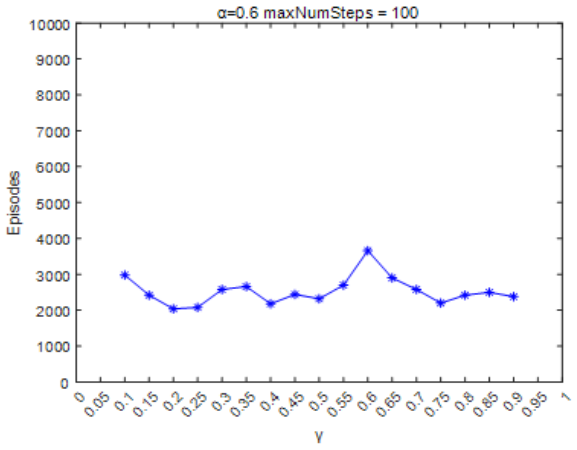


Fig 6. Alpha comparison chart. (a) Result of shelves.
(b) Result of maze 1. (c) Result of maze 2. (Picture credit: Original).

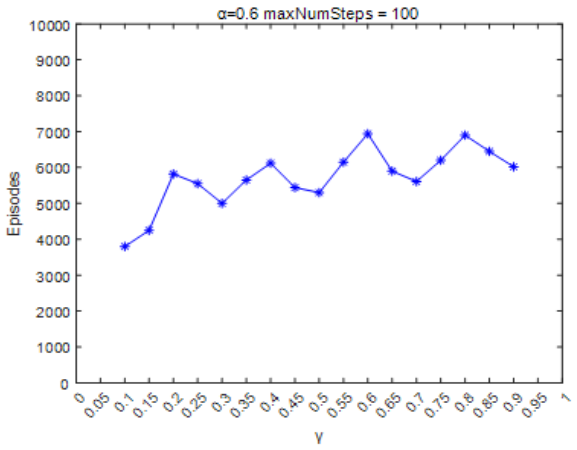
According to the path planning results of the three maps in Fig. 6, it can be seen that overall, as the alpha value increases, the number of iterations required for the robot to complete the training will significantly decrease. However, this reduction process is not entirely consistent across different maps.

3.3. Gamma Analysis

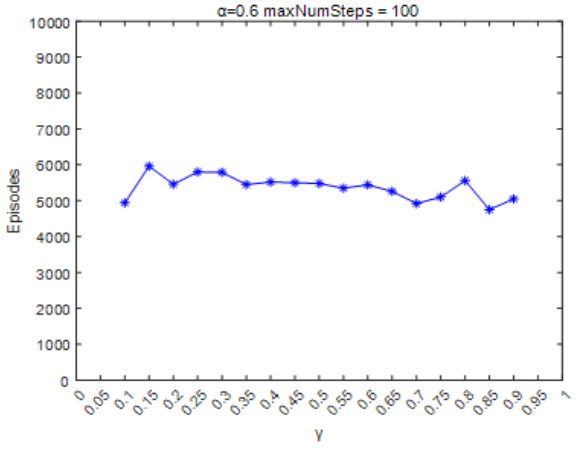
Similarly, for γ, the average number of iterations can also be compared using the method in 3.2.1. The line graph of average iteration times is shown in Fig. 7.



(a)



(b)



(c)

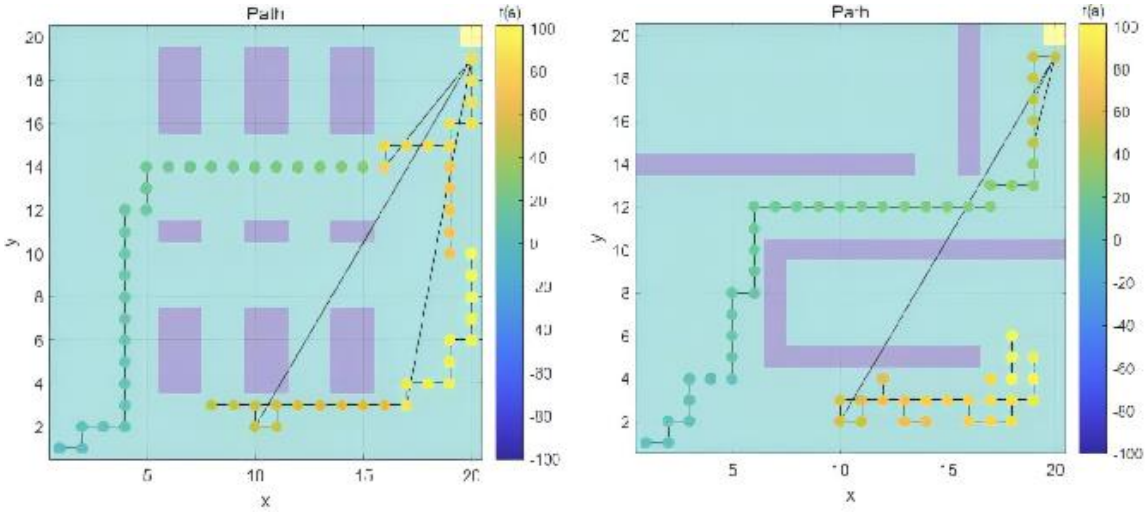
Fig 7. Gamma comparison chart. (a) Result of shelves.
(b) Result of Maze 1. (c) Result of Maze 2. (Picture credit: Original).

For gamma, its variation has an oscillatory effect on the number of iterations required for robot training. It can be observed in Fig. 7 (a) and Fig. 7 (c) that as the gamma value changes, the number of iterations required for robot training fluctuates around an average value, and the fluctuation amplitude of the data in Fig. 7 (c) is relatively small. Fig. 7 (b) shows an overall upward trend with significant fluctuations.

3.4. Results and Discussion

According to the formula of Q-Learning, the learning rate α represents the proportion of Q learned from the old Q value to the new Q value [10]. When the alpha value is 1, it indicates that new information is the most important, and when the alpha value is 0, it indicates that old information is the most important (meaning that nothing can be learned). From Fig. 6, it can be observed that as the alpha value increases, the overall performance of the robot improves, which means that the robot can react faster to reach the target position when facing a completely new environment. Therefore, assumption 1 in 2.2 is basically correct.

For the discount factor γ, the larger the value of γ, the greater the role played by formula 1 in the entire formula. In other words, the robot is more likely to act based on previous experience. In Maze 1, as the gamma value increases, the number of iterations required to complete the robot training increases. However, in Shelves and Maze 2, the number of iterations required for the completion of robot training oscillates with the variation of γ. So the change in gamma value has little effect on the number of iterations, but as the gamma value decreases, the trained robot will extend in an unplanned direction in the exploration map, as shown in Fig. 8.



(a) (b)

Fig 8. Results of path planning. (a) Shelves. (b) Maze 1. (Picture credit: Original).

Therefore, the selection of γ requires a balance between long-term and short-term rewards. A higher gamma value can enable robots to explore more areas in search of long-term rewards, while a lower gamma value can make robots pay more attention to past experiences for short-term rewards [11]. It can be seen that hypothesis 2 of 2.2 does not match the simulation results.

Overall, based on Q-Learning, robots will adopt different movement strategies according to different environments to avoid obstacles and reach their targets. Based on the path planning results in Figs. 4 and 5, it can be observed that despite changes in the number, location, and size of obstacles on the map, the robot is still able to avoid obstacles and reach the target through Q-Learning.

4. Conclusion

Regarding the impact of learning rate and discount factor on robot navigation performance in the Q-learning algorithm, the control variable method and equidistant value-added method were used to study the learning rate and discount factor respectively. A simple reward function was set up and

simulated in Matlab. In the simulation process, the path of the robot was first analyzed, and it was found that the robot exhibited a positive level of exploration in completely unknown environments, and was able to explore a relatively perfect path. Next, the average of the results obtained from multiple simulations was taken to draw a line graph of the number of iterations changing with the learning rate and discount factor. These line graphs can be used to verify the correctness of hypothesis 1 and the error of hypothesis 2 in the article. In this study, it has been shown that: (i) the higher the value of α: the faster the learning rate, which means that fewer iterations are required for the robot to complete training. (ii) The selection of gamma value depends on the map environment and requires a balance between long-term and short-term rewards. However, the method of taking the average of five simulation results when calculating the number of iterations still has some randomness and cannot effectively eliminate accidental errors. Therefore, in future research, more data should be collected and processed using algorithms. Secondly, when studying the impact of changes in discount factors on the number of iterations, this study did not find significant patterns. Therefore, in future research, maps of different sizes and complexities can be studied to discover universal patterns.

References

[1] Pareigis, S. Algorithmic Foundations of Reinforcement Learning. Lecture Notes in Networks and Systems, 1009 Lecture Notes in Networks and Systems, 2024, 1–27.

[2] Lin, J.-K., Ho, S.-L., Chou, K.-Y., & Chen, Y.-P., Q-learning based Collision-free and Optimal Path Planning for Mobile Robot in Dynamic Environment, 2022 IEEE International Conference on Consumer Electronics - Taiwan, Taipei, Taiwan, 2022, pp. 427-428.

[3] Elfoly, M. Ibrahim, H. Hendy and A. T. Hafez, Optimized area coverage path planning in a complex environment using geometric Q learning algorithm, 2024 14th International Conference on Electrical Engineering (ICEENG), Cairo, Egypt, 2024, pp. 60-65.

[4] Yun, S. C., Parasuraman, S., Ganapathy, V., & Joe, H. K. Neural Q-Learning based mobile robot navigation. Advanced Materials Research, 2012, 433–440: 721–726.

[5] Huang, Y., Yao, X., Jing, X., & Hu, X. DQN-based AGV path planning for situations with multi-starts and multi-targets. Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, 2023, 29(8), 2550–2562.

[6] Jiang, J., Zeng, X., Guzzetti, D., & You, Y. Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures. Acta Astronautica, 2020, 171, 265–279.

[7] Gök, M., Tekerek, M., & Aydemir, H. Reinforcement learning based local path planning for a mobile robot, arXiv preprint arXiv: 2403. 12463.

[8] Nourian, A. and Sadedel, M. "Investigating the Performance and Reliability, of the Q-Learning Algorithm in Various Unknown Environments", 2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM), 2023, pp. 21-28.

[9] Zhou, Q., Lian, Y., Wu, J., Zhu, M., Wang, H., & Cao, J.. An optimized Q-Learning algorithm for mobile robot local path planning. Knowledge-Based Systems, 2024, 286.

[10] Yu, Z., Mao, J., & Qi, W., Path planning algorithm of mobile robot based on improved Q-learning, 2024 7th International Conference on Advanced Algorithms and Control Engineering, Shanghai, China, 2024, pp. 1450-1453.

[11] Sonny, A., Yeduri, S. R., & Cenkeramaddi, L. R. Q-learning-based unmanned aerial vehicle path planning with dynamic obstacle avoidance. Applied Soft Computing, 2023, 147.