Path Planning Method of Mobile Robot Based on Q-learning

Qi Jiang*

1Southwest JiaoTong University, Chengdu, Sichuan Province, China, 610000
*Corresponding author’s e-mail: jianguuqi@my.swjtu.edu.cn.

Abstract. With the progress of science and technology, mobile robots gradually play an increasingly important role in the industry, military, science and technology and other fields. Aiming at the core problem of path planning in the path planning of mobile robots, this paper studies and designs a path planning method based on a Q-learning algorithm. Q-learning is widely used in robot path planning, as it only needs the interaction between the current state and the environment to make rewards and punishments for robot actions, to make decisions on the next action. Aiming at the problems of low efficiency and slow convergence in the original Q-learning algorithm, this paper improved the algorithm to enable the robot to quickly complete the planning and get the optimal and shortest path. The grid method was used to establish the environment running program to visualize the convergence process and obtain data. Finally, software simulation is used to establish the environment and code the robot to simulate the real environment, which proves the practical value of the algorithm.

1. Introduction
Path planning is one of the popular researches of mobile robots, which is very important for the implement of autonomous navigation of robots [1]. At present, there are many achievements in the research of path planning algorithm, including: Genetic Algorithm, particle swarm optimization, Artificial field potential method, A*(A-star) algorithm, Dijkstra algorithm [2], and Multi-dimensional fusion algorithm based on neural network or reinforcement learning [3,4], etc.

Q-learning algorithm is a mode of reinforcement learning and a Markov decision process [5]. In this process, the mobile robot does not have any prior knowledge. Based on strategy and learning, it takes different actions in the experiment to get different rewards, and to find the next action that can get the biggest reward. Through the interaction with the environment, the mobile robot can obtain feedback information from the environment, and then optimize the decision, and finally obtain the overall optimal strategy [6]. In this paper, the shortest and optimal path from the starting point to the end point in the unknown environment will be obtained. In the following article, the basic principle and algorithm logic of Q-learning algorithm will be introduced first, and then the mobile robot and the environment will be modeled and simulated. Finally, the optimal path from Q-learning algorithm will be obtained. Results will be checked, compared and analyzed.

2. Reinforcement learning and Q-learning algorithm
Reinforcement learning is the science of making optimal decisions. It imitates human interaction with the outside world to obtain different feedback [7], and the calculation method of learning from actions is as follows:
Similarly, the Q-learning algorithm is based on interactive learning logic, through the composition of "state", "action" and "reward", the ultimate goal is to achieve the maximum overall return:

Formula 1: Q-value

\[ newQ_{s,A} = Q_{s,A} + \alpha \left( R_{s,A} + \gamma * \max Q'(s', a') - Q_{s,A} \right) \]  

Where \( s_t \) and \( s_{t+1} \) represent the state in the environment; \( A \) represents action; \( \alpha \) is learning rate, is the learning rate, which defines the proportion of the new Q that an old Q value learns from the new Q value. The larger the \( \alpha \) is, the faster Q value converges; \( \gamma \) represents a discount factor, which defines the importance of future rewards, and the greater the \( \gamma \), the greater the long-term rewards' impact. \( \max Q'(s', a') \) indicates that the selected action A 'reaches the new state S' so that the value of Q is the maximum; \( R(s, a) \) represents a reward based on action A and state S.

At the beginning of the algorithm, initialize the Q value table, define the learning times episode=1, set the upper limit of the input learning times max-episode, then entering the cycle: Confirming the state S in the environment, select the next action A according to the Q table, reaching the next state S', and obtaining the immediate environmental reward R; Updating the Q table using Formula 1; Update S, increase the learning times by 1, judge whether the end point is reached, whether the maximum path limit is reached. If not, continue the cycle "confirm s- select action A - to reach S' exploration process.

When the agent reaches the target state, the program jumps out of the loop to judge whether the learning times at this time are greater than the maximum learning times. If the judgment result is no, the latest Q value table will be used to replace the initial Q value table, and the program iterates again until the number of learning converges to the required maximum number, so that the Q value is optimal, as well as the path is optimal[8], then the program ends. The algorithm logic can be represented by the following flow chart:
3. Path planning simulation experiment

3.1. Modeling of environment
Before the simulation of a mobile robot, it is necessary to provide an environment for the robot to move. In this paper, the environment space is established based on MATLAB, and a 20*20 grid environment is established. The starting point is (20, 1), and the end point is (20, 20). The black part is an obstacle that cannot be passed, as shown in the figure:
3.2. Setting of values

3.2.1. Action value setting
This experiment abstracts the mobile robot as a particle, the robot moves according to the table $A=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, where each value represents the robot's up, down, left, right, left, left, right, up, down, static.

3.2.2. State value setting
The state $S$ is directly represented by the coordinates of the mobile robot in the environment, and its value range is all coordinates except the obstacle.

3.2.3. Reward value setting
The reward value $R$ is shown in the figure, in which the mobile robot will get the reward value of -0.3 when moving to the obstacle, 1 when moving to the end point, and 0 when moving to all other positions to avoid falling into the local optimal, explore the map more comprehensively, and finally obtain the optimal path.

$$
\begin{array}{c}
0 & \text{path} \\
1 & \text{destination} \\
-0.3 & \text{obstacle}
\end{array}
$$

(2)

3.2.4. Parameter Value Setting
After several tests, in order to obtain the best convergence speed and optimal path, the parameter learning rate and discount factor in Q-learning algorithm are set as $\alpha=0.6$ and $\gamma=0.9$ respectively, and the number of iterations is set as $n=80$.

3.3. Simulation results and analysis
First, at the beginning of the simulation, the mobile robot gropes from the starting point to the end, an $S$ value is updated for each action $A$, obtains the reward value $R$ according to the position information, and updates the Q value table with feedback. According to the Q value, the next step is decided, and the cycle is repeated. After several iterations, the maximum Q value and the optimal path are obtained. After judging that the robot reaches the end point each time, the number of learning increases by 1 from 1. After judging the learning episode $>80$, iterative learning is stopped to obtain the optimal path.

Second, the optimal path finally learned and the path length change of each iteration learning is also shown in the figures:
3.4. Discussion

As shown in the figure, when the discount factor is set to $\gamma=0.95$, the final path is reached after 10 iterations, which can be observed that the final path is very close. However, there are still many deviations compared with the optimal path. It can be seen that with the increase of the discount factor, the algorithm can get the final path when the number of iterations is low. There is still a distance of 3-4 meters from the optimal solution, because the discount factor is greater than a certain threshold value, and the algorithm excessively pursues the convergence speed of reaching the end point. After adjusting the parameters, the accuracy is improved obviously.
4. Conclusion
This paper proposes, describes and uses a path planning method based on reinforcement learning. Q-learning algorithm is used to explore and learn the path, and raster method is used to establish a MATLAB simulation environment, so that the mobile robot moves in a complex environment with multiple obstacles, and the Q-value table is updated once for each step, and the optimal path is obtained after several iterations. The Q-learning method in this paper has many advantages compared with other path planning methods, such as genetic algorithm and A* heuristic intelligent search algorithm. Genetic algorithm carries out iterative elimination strategy by generating random initial feasible solutions, which has the disadvantages of too many iterations and slow calculation speed[9,10], A* algorithm has shortcomings of insufficient global optimization and path close to obstacles[11]. Q-learning algorithm has the advantages of independent planning in an unfamiliar environment, moderate planning time and optimal planning path. The interactive reward and punishment mode make the mobile robot more adaptable to the path planning in the static environment of complex obstacles. It is suitable for the obstacles with volume and height in the actual environment, and the autonomous navigation of the robot has a higher use value.

Acknowledgments
First, I want to express my gratitude to my program coordinator Fumiya. It is he who provided me the opportunity to get to understand and get into the field of Robotics and AI with his lectures. Fumiya also offered splendid discussions to us which enhanced my comprehension of new knowledge.

Second, I extremely appreciate the help from my friend Min Han who had brainstorming with me and offered a lot of excellent ideas.

Finally, I want to thank my senior Jiahui, who shared her experience in experiment to me and helped me in difficult process of analysis.

References
[1] Contreras-Cruz MA, Ayala-Ramire V, Hernandez-Belmonte UH. (2015) Mobile robot path planning using artificial bee colony and evolutionary programming. Applied Soft Computing; 30:319-328.
[2] Lin H.X, Xiang D, Ou YD, Lan XD. (2021) Review of Path Planning Algorithms for Mobile Robots. Computer Engineering and Applications. 11:1-13.

[3] WU Q, CHEN Z, WANG L. (2020) Real-Time Dynamic Path Planning of Mobile Robots: A Novel Hybrid Heuristic Optimization Algorithm. Sensors. 20(1):188-188.

[4] WANG J, CHI W, LI C. (2020) Neural RRT: Learning-Based Optimal Path Planning. IEEE Transactions on Automation Science and Engineering; 17(4):1748-1758.

[5] SOONG LE, PAULINE O, CHUN CK. (2019) Solving the optimal path planning of a mobile robot using improved Q-learning. Robotics and Autonomous Systems. 115:143-161.

[6] Yu HZ, Bertsekas DP. (2013) Q-learning and policy iteration algorithms for stochastic shortest path problems. Annals of Operations Research. 208: 95–132.

[7] BAE H, KIM G, KIM J, Qian D, Lee S. (2019) Multi-Robot Path Planning Method Using Reinforcement Learning. Applied Sciences; 9(15):3057.

[8] Yu NG, Wang C, Mo FF, Cai JX. (2017) Dynamic Environment Path Planning Based on Q-Learning Algorithm and Genetic Algorithm. Journal of Beijing University of Technology. 43(7):1009-1016.

[9] Qu H, Xing K, Alexander T. (2013) An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing; 120:509-517.

[10] Chen WJ, Jhong BG, Chen MY. (2016) Design of Path Planning and Obstacle Avoidance for a Wheeled Mobile Robot. International Journal of Fuzzy Systems; 18(6):1080-1091.

[11] Zhang XY, Zou YS. (2021) Collision-free path planning of automated guided vehicles based on improved A* algorithm. Systems Engineering Theory and Practice. 41(01):240-246.