*Applied Mathematics, Modeling and Computer Simulation*
*C. Chen (Ed.)*


*doi:10.3233/ATDE220046*

# Robot Path Planning Based on Improved Reinforcement Learning

Jian FANG[a,b,1] , Kuang YIN[b], Hongbin WANG[b], Wenxiong MO[b] and Fan YANG[b]
[a]*Power Test Research Institute of Guangzhou Power Supply Bureau of Guangdong Power Grid Co., Ltd, Guangzhou 510410, China*
[b]*State Key Laboratory of Power Transmission Equipment & System Security and New Technology, Chongqing University, 400000, China*

**Abstract.** With the continuous progress and development of robotics, mobile robots have been widely used in many different fields. In the power grid, mobile robots are used to inspect electrical equipment, which greatly reduce the investment of manpower and material resources. However, in many cases, mobile robots need to work in a constantly changing and complex environment. Because they cannot obtain environmental information in time, it is often difficult to make path planning. In response to this problem, this paper proposes a path planning method for mobile robots based on improved reinforcement learning. This method establishes a grid environment model, defines the return value through the number of steps of the robot, and then proposes a changing action selection strategy for the balance between the robot's exploration and utilization of the environment in reinforcement learning, so that the exploration factor dynamically change with the increase of the robot's exploration degree of the environment, thus speeding up the convergence speed of the learning algorithm. Simulation results show that this method can realize autonomous navigation and path planning of mobile robots in complex environments. Compared with traditional algorithms, it greatly reduces the number of iterations.

**Keywords.** Path planning, grid environment, reinforcement learning, exploration factor.

## 1. Introduction

Mobile robot path planning[1] means that in an environment with obstacles, the mobile robot finds an smooth and collision-free path[2] from the start point to the end point according to a given task or condition. The main goal of path planning is to find the best route based on performance indicators such as distance, time and energy when the robot is in an obstacle environment. Based on the understanding of the environment, path planning is divided into two research directions: global path planning based on environmental a priori complete information and local path planning based on unknown environment. In practical applications, mobile robots need to have the ability to adapt to unknown environments, therefore, solving the problem of robot path planning in

[1] Corresponding Author, Jian FANG, Power Test Research Institute of Guangzhou Power Supply Bureau of Guangdong Power Grid Co., Ltd, Guangzhou 510410, China; State Key Laboratory of Power Transmission Equipment & System Security and New Technology, Chongqing University, 400000, China. Email: zhangyingyi@bucea.edu.cn.

unknown environments has great significance to the application and popularization of robot technology, which is the premise and foundation of various application researches of mobile robots.

At present, domestic and foreign researchers have conducted a lot of research and achieved many corresponding results for some robot path planning problems with unknown environmental information. Literature[3] proposed a new path planning method of artificial potential field method, which adopts the method of setting intermediate target points to give the robot an external force point to stop or hover at a small point, overcoming the shortcomings of the traditional artificial potential field method in finding the optimal path, such as too slow convergence speed and easy to fall into the local optimum. Literature[4] proposes a path planning method for mobile robots based on hybrid genetic algorithm, which combines genetic algorithm[5] and simulated annealing algorithm[6], using grid method to build a model of the environment, at the same time, adding insertion operators and deletion operators to optimize the path in genetic operator, compared with the basic genetic algorithm, it significantly improved convergence speed and search quality. These methods overcome the problem of insufficient environmental prior information to a certain extent, and at the same time accelerate the convergence speed when solving the path planning problem, then plan a better path. However, due to the poor interaction between the algorithm itself and the environment, it is difficult to fully obtain information from the environment. When faced with a more complex environment, it spends a lot of time on unnecessary searches, which leads to slower convergence speed and prone to local optimal situations.

With the rapid development of machine learning[7-10], using reinforcement learning-based methods to solve path planning problems has attracted more and more attention. Reinforcement learning can realize online learning without a tutor, so it fully meets the needs of mobile robot path planning. This article uses the grid method[11] to establish a model, uses the model to automatically generate a state transition matrix, and defines the reward function to update the reward value through the interaction between the agent and the environment, so that the agent does not move blindly when exploring the environment. It is helpful to find the optimal path and improve the accuracy of the path solution. At the same time, improving the action selection strategy by guiding the action selection of the agent according to the exploration situation of the environment, further reduce the invalid exploration of the environment in the later planning stage, then improve the algorithm convergence speed.

## 2. Principles of Reinforcement Learning Algorithms

Q-learning is a reinforcement learning algorithm based on value iteration to learn action strategies[12]. The algorithm uses the Q function[13] to find the optimal action-selection strategy, and its core is to continuously update a table (Q matrix) which composed by state, action and reward. The Q value in the Q matrix is used as a standard for the quality of actions taken in a certain state. We use the Q value measurement method to find the best action that should be taken in each state. With the continuous iteration of the program, the Q matrix will eventually tend to converge.

The algorithm uses the formula $Q(s,a)$, that is the value of the state-action pair is used as a function, and its learning process is to continuously iterate to make neighboring Q value converge. Its basic form is as follows:

$$Q^*(s,a) = R(s,a) + \gamma \sum T(s,a,s_t) \max Q^*(s_t,a_t) \tag{1}$$

In the formula: $Q^*(s,a)$ refers to the sum of the optimal reward values obtained by taking action $a$ in the state $s$, and then defined $V^*(s)$ as the optimal value function in the state , then:

$$V^*(s) = \max Q^*(s,a) \tag{2}$$

Introduce the Bellman equation, and use the Bellman equation to solve the optimal decision sequence in the algorithm decision process. The Bellman equation is as follows:

$$V^{\pi}(s) = E_{i\sim\pi(s)}[r(s,\pi(s)) + \gamma V^{\pi}(s')] \tag{3}$$

$V^{\pi}(s)$ represents the reward obtained by using strategy $\pi$ to select an action in state $s$.

Consider the selection of action $a$ by the ε-greedy strategy in the state , transfer to the state $s_t$, and get the reward $r$, $\alpha$ is the learning rate, $\gamma$ is the discount, the iterative formula of Q-learning algorithm and the basic expression of the ε-greedy strategy are as follows:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max Q(s_t,a_t) - Q(s,a)) \tag{4}$$

$$prob(a_i) = \begin{cases} 1-\varepsilon & if \ \ a = \arg\max Q(s,a_i) \\ \varepsilon & others \end{cases} \tag{5}$$

The Q-learning algorithm initializes the Q value in the Q matrix, and then the agent selects actions in different states according to the ε-greedy strategy, the magnitude of ε determines the action selected by the agent. After the agent execution action reaches the new state, the actual iterative value is obtained through the Q-learning iteration formula and the Q matrix is updated. When the target state is reached, the iteration process finishes, and the agent automatically selects the next sample and continues to iterate from the initial state until finishes the entire learning process.

## 3. Reinforcement Learning Path Planning Based on ε-Decreasing

### *3.1. Action-selection Strategy Improvement*

In reinforcement learning, the agent influences the distribution of training samples through the action sequence. On the one hand, the agent needs to try as many different actions as possible to explore environmental information. On the other hand, it is also necessary to consider selecting the action with the largest value function to bring greater returns. The question of the balance between exploring and using the environment is

essentially a question of how to choose actions in each decision of the agent. Extensive exploration prolongs the training time of the agent, while over-utilize the environment causes the agent to converge prematurely and miss the optimal path. Therefore, it is necessary to improve the action-selection strategy so that the Q-learning algorithm can reach balance between exploring and using the environment, which not only prevents too many meaningless explorations from reducing the performance of the learning algorithm, but also avoids blindly choosing the most advantageous action and falling into the local optimum. The specific implementation method is as follows:

At the beginning, since the agent is facing an unfamiliar environment, it is necessary to explore the environment through random actions as much as possible. As the degree of mastery of environmental information continues to increase, the agent gradually reduces the exploration of the environment. Instead, the agent choose the most beneficial action to bring greater returns. Therefore, it is hoped that ε will maintain a downward trend along with the understanding of the environment throughout the operation of the algorithm, which is called ε-decreasing:

$$\varepsilon = e^{-\left|\frac{Q(S_t,a_b)-Q(S_t,a_t)}{n}\right|} \tag{6}$$

Among them, $Q(s_t, a_b)$ represents the maximum value of the corresponding action selected in current state $S$, $Q(s_t, a_t)$ represents the value of the next random action selected in the current state $S$, and $n$ represents a value that continuously decreases as the number of iterations increases. The calculation method is as follows:

$$n = \frac{M-m}{M} * n \tag{7}$$

Among them, $M$ represents the maximum number of iterations, $m$ represents the current number of iterations. From equation (6), it can be seen that when the agent knows nothing about the environment, the value of $Q(s_t, a_b)$ and $Q(s_t, a_t)$ is equal, so the ε factor is equal to 1, which means that all the actions selected by the agent are exploring the environment at this time. As the number of iterations increases and the agent is familiar with environmental information, the $n$ decreases while the difference between $Q(s_t, a_b)$ and $Q(s_t, a_t)$ increases, and the ε keeps getting smaller, which means that the agent gradually begins to choose the most favorable action to obtain the optimal path.

### *3.2. Dynamic Learning Rate Selection*

In formula (4), $\alpha$ represents the learning rate, which determines how much information the agent learns from the environment after each action. However, during the operation of the algorithm, the learning rate of every action must not be the same. When exploring the environment, it is necessary to learn more about the environment, so set a larger learning rate, while when using the environment, set a larger learning rate to prevent falling into the local optimum, so the $\alpha$ is set as follows:

$$\alpha = \begin{cases} 0.7 & \text{if } \varepsilon_0 < \varepsilon \\ 0.1 & others \end{cases} \tag{8}$$

*3.3. Action-selection Strategy Improvement*

(1) Initialize the grid of the current environment and determine the starting point and target point coordinates;
(2) Establish the Q matrix according to the rasterized environment, and initialize the Q matrix, the robot will explore the environment from the starting point;
(3) According to the state $s$ of the robot, use the improved action selection strategy described above to select action $a$ that the robot needs to perform in the state $s$;
(4) After executing the action $a$, the robot moves to the state $s'$, and update the corresponding value of the Q matrix according to the iterative formula;
(5) Determine whether the current position is the target point coordinate or reach the maximum number of steps of the mobile robot, if not, return to step(3), otherwise go to step(6);
(6) Determine whether the Q matrix has converged or reached the maximum number of iterations. If yes, finish the entire learning process and output the optimal path; otherwise, return to step(2) to execute the next learning attempt.

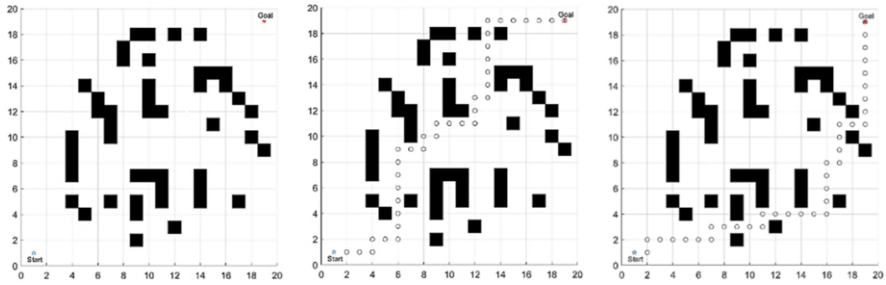
## 4. Reinforcement Learning Path Planning Based on ε-Decreasing

In order to verify the feasibility of the path planning method based on reinforcement learning proposed in this paper, MATLAB is used to simulate it. Table 1 shows the parameters needed for the basic Q-learning algorithm.

**Table 1.** Basic Q-learning algorithm parameters.

| Parameter | Value |
|---|---|
| Map size | 20*20 |
| Reward value | 100 |
| Penalty value | -100 |
| Learning rate | 0.7 |
| Exploration factor | 1 |
| Discount factor | 0.9 |
| Maximum step size | 2000 |
| The maximum number of iterations | 1000 |

Transforming the learning rate and exploration factor in table 1 according to formula (6) and formula (8) is improved Q-learning algorithm. The simulation environment settings are shown in figure 1(a), and the paths planned using the basic Q-learning algorithm and the method proposed in this article are shown in figures 1(b) and (c):

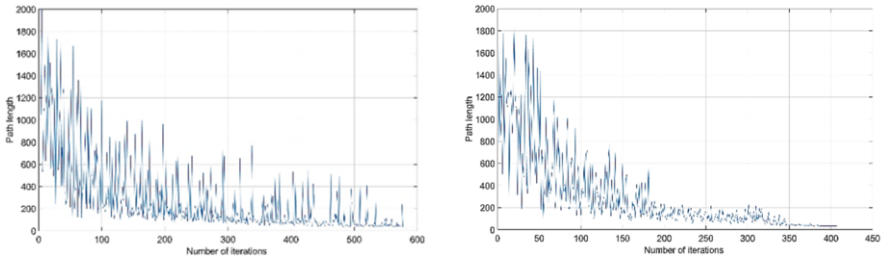



(a) (b) (c)

**Figure 1.** (a)simulation environment; (b) basic Q-learning algorithm path planning; (c) improved Q-learning algorithm path planning.

As can be seen from figure 1(b) and (c), both methods can successfully plan robot path in an unknown environment. Although the planned routes are different, the path length is 36, which shows that both methods can find the optimal path in this environment.

In order to further compare the performance of the two methods, figure 2(a) and (b) show the step size convergence of the two methods. It can be seen that the path length jitter of the two reinforcement learning methods is large at the beginning of the iteration. As the number of iterations increases, the path length shows a downward trend, but the basic Q-learning algorithm still has more frequent and severe jitter in the later stage of convergence, while the improved Q-learning algorithm converges gently. The reason for this phenomenon is that in the initial stage, the agent needs to explore the environment as much as possible to fully obtain environmental information. Therefore, the two Q-learning algorithms have dramatic path length changes at the beginning of the iteration. As the number of iterations increases, the agents corresponding to the two algorithms gradually deepen their understanding of the environment, and "learning experience" gradually guides path planning, so the path length shows a downward trend. In the later stage, although the basic Q-learning algorithm has been iterated enough times, the value of ε in the adopted ε-greedy strategy remains unchanged, and the environment is still explored randomly, so there is still severe jitter in the later stage, finally converged after 596 iterations. The ε-decreasing strategy of the improved Q-learning algorithm dynamically reduces the value of ε as the number of iterations increases, so that the agent's actions to explore the environment are continuously converted into actions to use the environment, so the fluctuation of the path length at the end of the iteration gradually decreases and eventually it tends to be flat, and it reaches the state of convergence after only 405 iterations.




(a) (b)

**Figure 2.** (a) basic Q-learning algorithm path planning; (b) improved Q-learning algorithm path planning.

Two algorithms were tested for 20 times in this environment, the path length and the number of iterations were recorded, and the average path length and the average number of iterations were calculated. The results are shown in table 2:

**Table 2.** Basic Q-learning algorithm parameters.

| Method | Average path length | Average number of iterations |
|---|---|---|
| Basic Q-learning algorithm | 36 | 593 |
| Method of this article | 36 | 407 |

## 5. Conclusion

Aiming at the problem of mobile robot path planning in an unknown environment, this paper proposes a dynamic action selection strategy based on the traditional Q-learning algorithm. This strategy is guided by the degree of mastery of environmental information, adaptively choose to explore the environment or use the environment, and dynamically set the learning rate according to the state of the agent, effectively improve the situation where the Q-learning algorithm is easy to fall into the local optimum. Compared with the traditional Q-learning algorithm, the method proposed in this paper has a faster convergence speed and shows better path planning performance under the premise of ensuring the optimal path.

## Acknowledgments

This work is supported by China Southern Power Grid (project number: GZHKJXM20180069).

## References

[1] Fethi M, Boumedyen B, Brahim M and Mohamed N A 2020 Contribution to the path planning of a multi-robot system: centralized architecture Intell. Serv. Robot. 13 147-158.
[2] Wan Y, Wang M et al 2016 A Feature Selection Method Based on Modified Binary Coded Ant Colony Optimization Algorithm Appl. Soft Comput. 49 248-258.
[3] Ren Y and Zhao H 2020 Improved Robotic Path Planning Base on Artificial Potential Field Method Computer Simulation 37 365-369.
[4] Liang Y and Xu L 2009 Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm *Genetic and Evolutionary Computation* (New York :Association for Computing Machinery) p 303–308.

[5] Hao K, Zhao J, Yu K et al 2020 Path Planning of Mobile Robots Based on a Multi-Population Migration Genetic Algorithm Sensors 20 5873-5873.
[6] Miao H and Tian Y 2013 Dynamic robot path planning using an enhanced simulated annealing approach Appl. Math. Comput. 39 17-24.
[7] Wai R J and Prasetia A S 2019 Adaptive Neural Network Control and Optimal Path Planning of UAV Surveillance System with Energy Consumption Prediction IEEE Access PP 1-1.
[8] Mortaza Z et al 2014 Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms Eng Appl Artif Intell 29 163-177.
[9] Chen H, Ji Y, Niu L 2020 Reinforcement learning path planning algorithm based on obstacle area expansion strategy Intell. Serv. Robot. 13 1-9.
[10] Woo M et al 2018 A study on the optimal route design considering time of mobile robot using recurrent neural network and reinforcement learning J. Mech. Sci. Technol 32 4933-4939.
[11] Jung J H and Kim D H 2020 Local Path Planning of a Mobile Robot Using a Novel Grid-Based Potential Method Int. J. Fuzzy Log. Intell. Syst. 20 26-34.
[12] Chen C, Chen X, Ma F et al 2019 A knowledge-free path planning approach for smart ships based on reinforcement learning Ocean Eng. 189 106299.
[13] Michael L L 2001 Value-function reinforcement learning in Markov games Cogn. Syst. Res. 2 55-66.