The Shortest Path Planning Based on Reinforcement Learning

The Shortest Path Planning Based on Reinforcement Learning

Xiaoqi Wang*, Lina Jin and Haiping Wei

School of Computer and Communication Engineering, Liaoning Shihua University, Fushun, People's Republic of China

*Email: 1445946561@qq.com

Abstract: This paper proposes a shortest path planning of agent in an environment based on reinforcement learning. This method adopts the Q-learning algorithm, which has gained increasingly using in agent path planning recently. This algorithm can be fully designed to a reasonable environment model and applied to other professional fields. Yet, reinforcement learning in path planning still needs to improve the rate of convergence. The learning process takes several iterations and spends a long time to find the final path. A local optimal problem in search process deals with other explorations that provided shortest path for agent to take optimal actions. For the agent, the reward value of inclined movement is introduced to find the shortest path. The autonomous obstacle avoidance is utilized to obtain the optimal path in the process. Finally, the model of intelligent agent movement environment has established in matlab. The effectiveness of the algorithm has proved by simulation results.

1. Introduction

Reinforcement learning has gained increasingly popularity in autonomous mobile robot path planning recently. Reinforcement learning has got more and more interests by researchers due to its advantage, which can make a decision by its own [1]. As a machine learning technique, reinforcement learning is suitable for agents in an environment. Q-learning is different from other branches of machine learning, as a type of reinforcement learning, which is a famous algorithm related to the principle of reward and penalties [2]. The foundation of this learning method depends on the interaction of agents in the environment. Q-learning is a kind of mapping that represents the relationship between states and actions. Reinforcement learning is a goal-oriented learning method. The markov decision process provides an academic framework to support the rapid development of reinforcement learning. Reinforcement learning utilizes the markov decision process, as a theoretical basis [3].

Autonomous robot performs a behavior in an environment and receives an immediate reward or penalty for the action taken [4]. This behavior has been extensively spread its utilization in unsupervised missions which the robot will learn by its own ability and carry out a series of actions to attain the predefined target [5]. In order to accomplish the given mission, terrestrial exploration and path planning are important for mobile robots such that they know how to move from one place to another and how to perform the desired task. The solving method to path planning has proposed multitudinous ways [6]. The reinforcement learning was used in the disciplines of game theory and other researches at first. It has been adopted in the navigation of autonomous mobile robot in recent years [7]. Reinforcement learning can be used in navigating the unknown environment based on a set of rewards and penalties. The mobile robot which is investigated as the agent receives a reward for

collision-free action and receive a penalty when it collides with the obstacle. The data of Q-value table will be updated based on the directly received rewards or penalties and the state with the highest Q-value is regarded as a best path for the moving robot continuously. A large number of scholars have done a lot of researches on path planning. Application of some methods have also been proposed. The solution to path planning has been numerous methods, as mentioned in studies. The shortest path planning of mobile robot which has utilized the A~* algorithm for indoor mobile robots was proposed by Wang wei [8]. The planning of mobile robot in unlimited environment based on particle swarm optimization algorithm has also been used for path planning by Sun bo [9]. The robot obstacle avoidance of improved artificial potential field has been proposed by Zhang H [10]. Several algorithms are reported simply and quickly in literature, as observed in previous study. The aim of this paper is utilizing the Q-learning algorithm to generate a path planning of mobile robot in an environment.

2. Reinforcement learning

2.1. The framework of reinforcement learning

The general model structure of reinforcement learning involve state, action, reward, agent, and policy is shown in figure 1. A policy defines the learning agent's way of behaving at a given time. It is a mapping from perceived states of the environment to actions. A reward function defines the goal in a reinforcement learning problem. Without rewards there could be no values, and the only purpose of estimating values is to achieve more reward. Interaction of agent with around environment will obtain the definite value. The agent defines its location in environment whereas an agent chooses an action to make a movement from one state s_t to another state. The agent performs a random action in its movement environment. The action a_t update the original state of environment to change the agent approach a new state s_{t+1} on $t+1$. In the next new state, the reward from the environment makes a response r_{t+1}. The Agent based on state s_{t+1} generated a new feedback to implement action a_{t+1}. Now, the next state s_{t+1} is considered as an initial state. The step of the agent acting selection is receiving an immediate reward. Transforming the next state and the interaction of agent will be repeated forever.

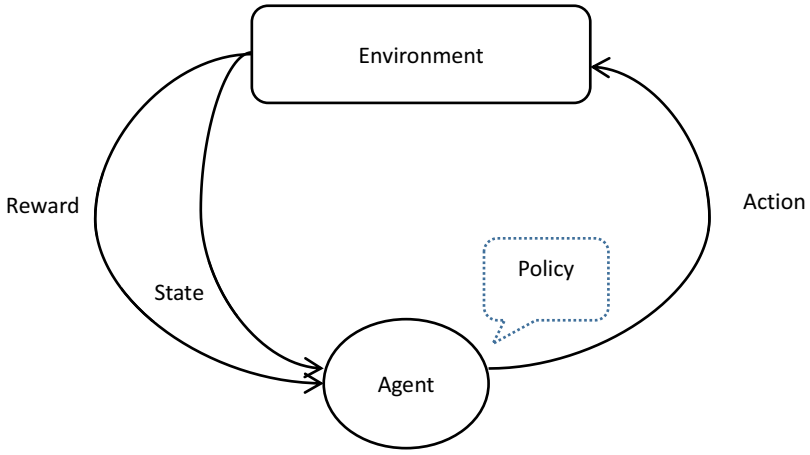

Figure 1. Reinforcement learning framework

2.2. Q-learning algorithm

Q-learning algorithm is an off-policy algorithm in reinforcement learning. Q-learning Algorithm is a typically model-free structure. Exploration is used at the beginning and exploitation is at the end of the learning process. Algorithm of Learning helps an agent perform better in similar situations. In

Q-learning, the agent selects next step from its current reward by Q-table. Q-table is the core of Q-learning algorithm. In the process of path planning, the agent selects an action from a limited table of actions and applies the action to interact with the environment. The environment accepts the action and transfers the state s_t to s_{t+1} and gets the reward value R. Q-learning adopt the all possible state-action like $Q(s,a)$ and obtain the optimal strategy. The Q-value of the Q-learning algorithm is updated using the expression of:

$$Q(s,a) = Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a' \in A} Q(s',a) - Q(s,a)\right) \tag{1}$$

After the first update:

$$\begin{aligned} Q(s,a) = {} & (1-\gamma)Q(s,a) \\ & + \alpha(r(s,a) + \gamma Q(s',a)) \end{aligned} \tag{2}$$

After the second update:

$$\begin{aligned} Q(s,a) = {} & (1-\alpha)^2 Q(s,a) \\ & + (1-\alpha)\alpha(r(s,a) + \gamma Q(s',a)) \\ & + \alpha(r(s,a) + \gamma Q(s',a)) \end{aligned} \tag{3}$$

After the third update:

$$\begin{aligned} Q(s,a) = {} & (1-\alpha)^3 Q(s,a) \\ & + (1-\alpha)^2\alpha(r(s,a) + \gamma Q(s',a)) \\ & + (1-\alpha)\alpha(r(s,a) + \gamma Q(s',a)) \\ & + \alpha(r(s,a) + \gamma Q(s',a)) \end{aligned} \tag{4}$$

After the nth update:

$$\begin{aligned} Q(s,a) = {} & (1-\alpha)^n Q(s,a) \\ & + (1-\alpha)^{n-1}\alpha(r(s,a) + \gamma Q(s',a)) \\ & + (1-\alpha)^{n-2}\alpha(r(s,a) \\ & + \gamma Q(s',a) + \cdots + \alpha(r(s,a) + \gamma Q(s',a)) \\ = {} & (1-\alpha)^n Q(s,a) + \alpha(r(s,a) \\ & + \gamma Q(s',a)) \cdot [(1-\alpha)^{n-1} + (1-\alpha)^{n-2} + \cdots + 1] \\ = {} & (1-\alpha)^n Q(s,a) + (r(s,a) + \gamma Q(s',a)[1-(1-\alpha)^n] \end{aligned} \tag{5}$$

We obtain the update of $Q(s,a)$, as $\alpha \in (0,1)$, $0 < 1\text{-}\alpha < 1$ When the number of updating increasing an high level, $(1-\alpha)^n \to 0$ and

$$Q(s,a) = r + \gamma Q(s',a) \tag{6}$$

The state s and state s' belong to set S in this formula. A serious of state are represented by S, and the next state is presented by s'. Action a is a member of A which present the action space. r represents the reward for getting action a in state s. The α represents the agent learning coefficient. The learning rate α determines the extent to which the newly acquired information will override the old information. A setting of α=0 make the agent stop learning, whereas α=1 would make the agent consider only the most recent information. The discount factor γ indicates the influence of further rewards on current action.

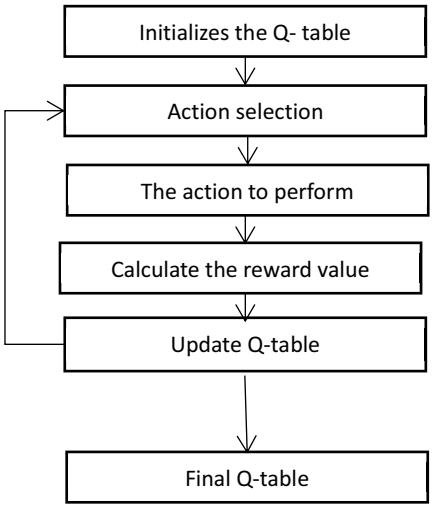

Figure 2. Flow chart of Q-learning algorithm

Firstly, initialized the Q-table with m states and n number of possible actions. Size of matrix is constructed by $n \times m$. Then the all number of Q value has to be defined zero and the current agent has many choices to make a decision. When moving from the current state to the next state, the agent has to select the action with the highest Q value among the n possible actions. This situation implies that the action is selected by the reward. The time of comparisons as (n-1) are required. When the number of iteration increases, the Q-table is constantly updated. Calculating the reward from the agent action. The action a is adopted and the state $Q(s,t)$ will be updated by the Bellman equation according to the current state and reward. Repeating the third step until obtain the final Q-table by the interaction between the agent and the environment. Finally, selecting the best path based on the value of Q-table. The update process of Q-learning is as shown in figure 2.

3. Result and performance of simulation analysis

The effectiveness of Q-learning algorithm for path planning exploration is verified by simulation experiments. The environment of agent movement must be constructed and consider a navigation with 20*20 to present the environment in matlab firstly. As the top-left point shown in the figure 3. The initial position is $SP(1,2)$, and the goal position is $EP(19,20)$. The blue area in the figure 3 predefined the obstacle in different color where the agent cannot pass through. The starting point and ending point of the agent can be input from the outside.

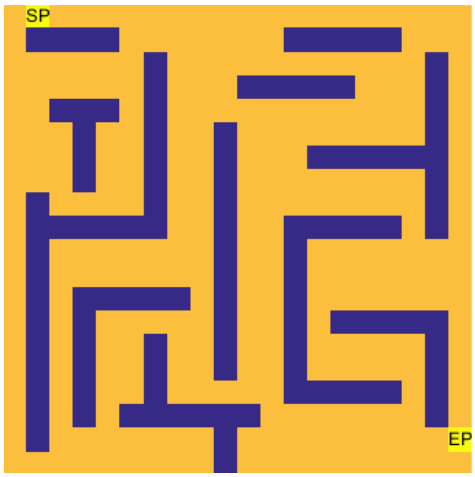

Figure 3. Agent movement environment

Q-learning algorithm is a greedy algorithm intuitively. There are many motivation behaviors in the possible action each time if the agent takes an action according to the maximum value. The agent could be able to fall into the local optimum, and the agent will not be able to adopt other behavior. In the same way, the agent unable to complete the shortest path planning. Therefore, the coefficient should be utilized so that the agent not only has a certain probability to select the optimal action, but also has the chance to select other actions. Keeping the track path in memory to avoid small loops. Let the reward value of oblique motion to "1". It could avoid the situation where the agent moves from upward to downward. Instead of moving two squares to the left, set the parameters of the algorithm in experiment condition which will affect the convergence speed directly. The initialization parameters are employed in this experiment. The parameters such as a=0.9 and b=0.6 are explored. The number of cycles is 100 times. The process of finding the shortest path in the process of path planning is shown in figure 4.

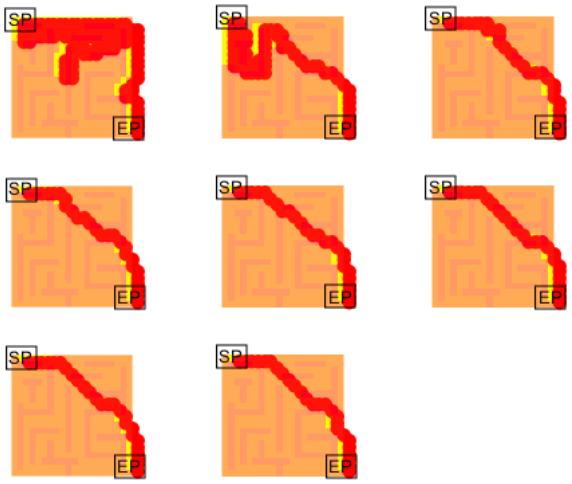

Figure 4. Procedure of the shortest path finding

Constructed the reward matrix by matlab, the possible motion states of the agent: upward motion, downward motion, left motion, right motion, right-lower motion, left-lower motion, right-upper motion and left-upper motion. The Q-table of *Q-learning* algorithm is obtained by iteration. the value table of Q-learning algorithm is updated by bellman formula. The final shortest path of agent is emerged in Figure 5.

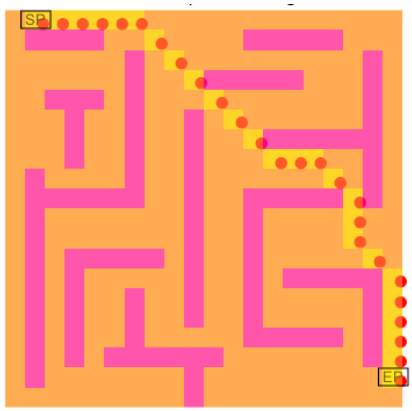

Figure 5. The Shortest path planning of agent

The variation trend of the search distance changed with search times. In the process of the agent path planning, utilized Q-learning algorithm to find the shortest path is shown in Figure 6. After 57 times of iterations, the shortest path planning of agent is 29.4.

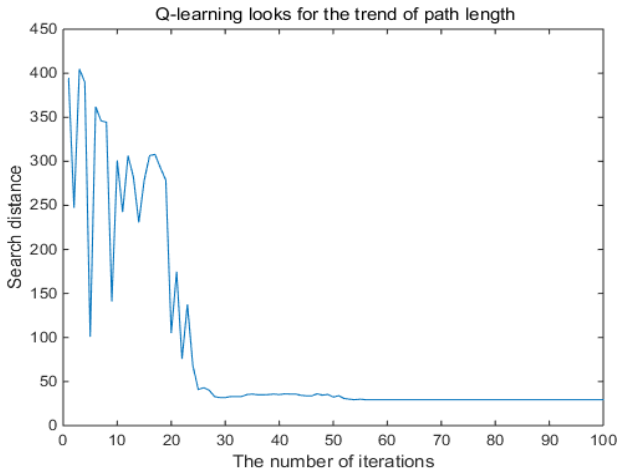

Figure 6. The convergence of Q-learning for path planning

4. Conclusion

In this paper, a path planning of mobile agent based on Q-learning has been proposed in a given environment with static obstacles, which marked with different color. Through the prior knowledge obtain from classical Q-learning, simulation results demonstrate that the Q-table serves a good exploration foundation to plan a shortest path. The advantages of Q-learning algorithm are fully proven in research about the shortest path planning of agent. The benefits of reinforcement learning are fully demonstrated in this paper. Reinforcement learning plays a critical role in path planning. It is advisable that set the parameters of oblique action to avoid the local optimal problem. In order to certificate the feasibility of the theory, the feasibility of the algorithm has verified finally.

Acknowledgments

This work is supported by the Scientific Research Fund of Liaoning Provincial Education Department L2019048, and Talent Scientific Research Fund of LSHU 2016XJJ-033 of China.

References

[1] Sutton R S, Barto A G 2018 *M Reinforcement learning: An introduction*
[2] Stone P, Sutton R S and Kuhlmann G 2005 *Reinforcement learning for robot-cup soccer keep away J. Adaptive Behaviour* **165** 188
[3] Sridhar M and Jonathan C.2015 *J Reinforcement learning of Automatic programming used by robot based on behaviour. Arti. Inte* **311** 365
[4] Carlucho I, De Paula M and Villar S A. 2017 *J Incremental Q-learning strategy for adaptive PID control of mobile robots. Expert Systems with Applications* **183** 199.
[5] Yan J, He H and Zhong X.2016 *J Q-learning-based vulnerability analysis of smart grid against sequential topology attacks. IEEE Transactions on Information Forensics and Security* **200** 210.
[6] Allab A E L, Abdou M and Perot E. 2017 *J Deep reinforcement learning framework for autonomous driving. Electronic Imaging* **70** 76
[7] Mnih V, Kavukcuoglu K and Silver D.2015 *J Human-level control through deep reinforcement learning.* Nature **529** 533
[8] Wei W, Dong P and Zhang F.2018 *J Mobile robotic path planning using improved A-Star algorithm. Journal of Computer Application* **307** 310
[9] Sun B. *Mobile robot path planning in unlimited environment based on particle swarm optimization algorithm. Control and Decision*
[10] Zhang H. *The robot obstacle avoidance of improved artificial potential field [J]. Technology Information* 2017, 015(004):100-103.