A path planning algorithm based on RRT and SARSA (λ) in unknown and complex conditions

ZOU Qijie[1,2], ZHANG Yue[1], LIU Shihui[1]

1. Information Engineering Faculty, Dalian University, Dalian 116000
E-mail: 1641211737@qq.com

2. National Institute of Innovation, National Defense Science and Technology Research Center for Unmanned Systems, Changsha, 410000, China
E-mail: jessie_zou_zou@163.com

Abstract: The path planning problem of wheeled mobile robot is different from the traditional method in the environment of outer space which is completely unknown. If robots are needed to explore unknown environments while adapting to terrain changes, it is a more challenging problem. In this paper, a Rapidly-exploring Random Tree(RRT) path planning algorithm based on reinforcement learning SARSA(λ) optimization(RL-RRT) is proposed to solve the local path planning problem in unknown and complex conditions. RL-RRT method improves the performance of RRT algorithm by adding optimization for selection of extension points, introducing the idea of biased goal, using task return function, target distance function and angle restriction, ensuring the randomness of RRT, reducing the number of invalid nodes and optimizing the planning performance. A simulation platform is built under Robot Operating System(ROS) and MATLAB to test the role of multi-objective optimization in path planning. The simulation results show that the RL-RRT method enables the wheeled mobile robot to reach the target node smoothly and steadily in complex and unknown environments without collision with obstacles, which verifies the reliability and effectiveness of the method.

Key Words: local path planning, RRT, SARSA (λ), wheeled mobile robot, unknown and complex environments.

1 INTRODUCTION

Over the past few decades, the ways and means of human entering, exploring and utilizing outer space have become more and more abundant, especially the exploration of the moon[1-2]. With the increasing demand of human mission to explore the moon, the exploration of the moon requires not only launching orbiters to survey the moon, but also more lunar activities and detailed investigation of the moon. Because the road surface condition of the moon is unknown and complicated, the wheeled lunar rover is an important tool for lunar surface detection. So the related autonomous path planning problem of wheeled mobile robots has also become a research hotspot. Path planning is a collision free optimal or near optimal path in a workspace for robot to search from a starting point to a target state. Local path planning is the solution to solve locally unknown and complex environments planning problem such as the moon, Saturn and Mars. It can obtain the current environmental information according to sensors and plan the optimal collision-free path from the current road node to the target road node.

According to the different search methods, the previous local planning methods can be divided into three categories: traditional methods, intelligent methods and machine learning methods.

This work was supported by the Liaoning Provincial Natural Science Foundation project, No.: 2019-zd-0578; general funded project of National Natural Science Foundation of China (61673084) : Research on the Mechanism of Abnormal Driving Behavior Recognition and Cooperative Control for Intelligent Driving Systems.

The first class of local path planning methods make use of traditional methods to deal with the optimal solution of the process[3-4]. Yang Long[5] clearly demonstrated a better initial population quality, which can get shorter, safer non-collision paths. Rapidly-exploring Random Tree(RRT) algorithm was employed since 1998, it was proposed as a random sampling tree structure planning algorithm, which was particularly useful in studying of path planning under high dimensional position[6-7].

The second class is local path planning approaches based on intelligent search algorithm, such as establishing neural network based on environmental information[8-9]. Ni J[10] revealed an improved dynamic bioinspired neural network and effectively located in the real-time path planning problem of robot.

The third class of local path planning methods employ machine learning algorithm. As an important class of machine learning methods, reinforcement learning(RL) can solve sequential decision making problems[11-14].

The typical algorithm, such as based chart search algorithm, is based on a fully known environmental model[15-16]. Unknown and complex environment planning is difficult, faster computing and higher security is a challenging and important task. Some of the collisions in some robots only affect performance, not security issues, but some areas require robots to arrive at a completely non-collision and comfortable arrival destination, such as lunar rover, wheels can neither skid nor roll over[17].

The main contributions of this article can be summarized as follows.

978-1-7281-5855-6/20/$31.00 ©2020 IEEE

- The RL-RRT method is proposed, which combines RRT with reinforcement learning SARSA (λ). Considering the optimization of RRT extended nodes, the proportion of effective nodes becomes larger and the planning efficiency is improved;
- We considered multi-objective constraints of path planning: smoothness, collision avoidance and travel efficiency. By reducing the number of times the wheeled mobile robot repeatedly adjusts its direction and choosing the appropriate rotation angle, it can meet the requirements of some fields that the robot arrives at its destination completely without collision and comfortably.
- We also established the Robot Operating System(ROS) and MATLAB simulation platform, and comprehensively tested the role of multi-objective in path planning.

The rest of this article is organized as follows. Section 2 introduces the RRT algorithm based on reinforcement learning(RL-RRT). In Section 3, the simulation environment is introduced and the simulation results are given. Finally, Section 4 concludes with the conclusion and the future work.

2 RL-RRT Method

The proposed algorithm in this paper is a local path planning algorithm based on reinforcement learning SARSA(λ)[18-20], which strives to reduce the blindness of random sampling of RRT algorithm and improve the real-time and reliability of robot local path planning under unknown and complex environment. Therefore, the RL-RRT method can be applied to certain areas of the environment path planning that requires the robot to be completely non-collision and comfortably reach the destination.

2.1 Framework of RL-RRT

Reinforcement learning plays an important role in solving the inertial order decision making problem, and is good at self-learning through interaction with the environment. Combining it with RRT can optimize the choice of RRT nodes.

In this paper, the mathematical model of robot local path planning is established as follows. Suppose that $X_{init}(x_i, y_i, \theta_i)$ the initial position is the starting position of RL-RRT the extension tree, and that $X_{goal}(x_g, y_g, \theta_g)$ represents the target position. Bias target guarantees that the speed of reaching the target point will be increased even if the distance from the obstacle is very close. Safe obstacle avoidance can ensure the safety of path planning. Rotation angle restriction can ensure the smoothness of wheeled mobile robot in the process of moving. The formula for $P(\Phi)$ calculating the path cost function of local path reprogramming proposed in this paper is as follows:

$$X_{init}(x_i, y_i, \theta_i) \xrightarrow{P(\Phi)} X_{goal}(x_g, y_g, \theta_g) \quad (1)$$

$$P(\Phi) = P_{goal}(x) + P(safe) + P_{smooth}(\theta) \quad (2)$$

Among them, $P_{goal}(x)$ is to expand the tree towards the target, $P(safe)$ is to secure the trajectory from the initial point to the target point, and $P_{smooth}(\theta)$ is to limit the smooth value of rotation angle.

$$P(\Phi) = \begin{cases} P(smooth) & |\theta| \le \theta_{\max} \\ P(goal) & x_{rand} = x_{goal} \\ P(safe) & P(\Phi) \cap obstacle = \varnothing \end{cases} \quad (3)$$

The flow chart of the RL-RRT method is shown in Fig.1. The dotted line contains part of the SARSA(λ)-based extended point selection process. In this paper, for the local path planning problem, the position of the mobile robot is regarded as the state, and different action decisions are taken according to the state, so that the robot has the action to be taken at any position, and the corresponding return is obtained, and then transferred to the new position, the process continues to circulate until it reaches the end state, as shown in Fig. 2.

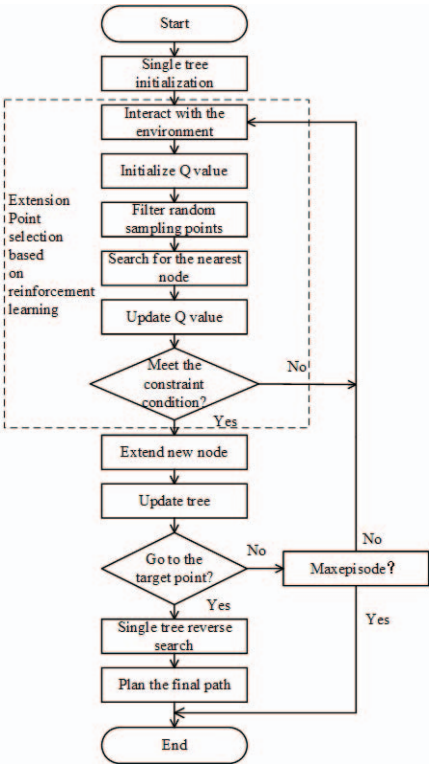

Fig. 1 Flow chart of RL-RRT method

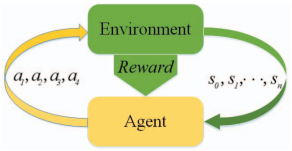

Fig.2 RL-RRT reinforcement learning process

2.2 Screening of Random Nodes

The core idea of random node selection is to select a new node in the random tree according to the introduction of

biased target idea and task return function $R(x)$. Biased target is to make $x_{rand} = x_{goal}$ through a certain probability in the process of random tree expansion, so that the extended tree will tend to extend to the target area, even if it is very close to the obstacles, it will also improve the speed of reaching the target point. The task reward function $R(x)$ affects the selection of random nodes by calculating the return value of each node to the target and obstacles, according to the presence or absence of obstacles. Set two different weights. In one case, when there is an obstacle in the environment, the obstacle avoidance action function has the highest priority, that is, the safety is first ensured, and then the target is considered. In another case, when there is no obstacle in the environment, the obstacle avoidance action function is not considered, and only the target action function is set. This design can both bypass obstacles and grow toward the target point. $R(x)$ can be expressed as formula (4):

$$R(x)=kx_o+(1-k)x_g \tag{4}$$

Among them, x_o is the return value of the obstacle avoidance action, x_g is the return value of the target action, $k \in \{0, 0.3, 0.7, 1\}$ is the weight of the action return function set in the experiment, when avoiding obstacles k=0.7 , and there is no obstacle k=0 .

2.3 Extended Tree

The RRT tree is randomly expanded in space, so there is randomness of the relative position of the parent node and the child node. However, considering the stability of the mobile robot during the advancement process, the RL-RRT method introduces the threshold of the path corner.

The core idea of node selection for the extended tree is based on $E(x)$ extended function, $E(x)$ contains $R(x)$ task reward functions, $L(x)$ target distance functions, and rotation angle limit $|\theta| \le \theta_{\max}$, which can be expressed as:

$$E(x) = R(x) + L(x) + \theta \tag{5}$$

The target distance function $L(x)$ affects the selection of the new node by calculating the distance between the random extension point and the extension to the current target node. $L(x)$ can be expressed as equation (3):

$$L(x)=\rho \cdot \frac{x_{rand}-x}{\|x_{rand}-x\|} \tag{6}$$

Among them, ρ is the step size of the random tree, x_{rand} is the random expansion point, x is the nearest neighbor node x_{near} , and $\|x_{rand}-x\|$ is the distance between the two points defined by the Euler norm.

Substituting equations (4) and (6) into equation (5) gives:

$$E(x)=kx_o+(1-k)x_g+\rho \cdot \frac{x_{rand}-x}{\|x_{rand}-x\|}+\theta \tag{7}$$

According to formula (7), the formula for generating new node x_{new} after introducing the idea of $E(x)$ extension function is obtained:

$$x_{new}=x_{near}+kx_o+(1-k)x_g+\rho \cdot \frac{x_{rand}-x}{\|x_{rand}-x\|}+\theta \tag{8}$$

2.4 The Establishment of the Q Function

The Q value update of the node on the random tree is for the current node and the successor node. The Q value of the successor node changes according to the environment. If the node is in the feasible space, it will generate a positive return; if it is in the obstacle environment, it will generate a negative return. The successor node is no longer extended. The update formula for SARSA(λ) is:

$$Q(s,a) \leftarrow Q(s,a)+\eta[r_{t+1}+\gamma \cdot Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]e(s_t,a_t) \tag{9}$$

$\eta > 0$ is the learning rate, r_{t+1} is the instantaneous reward obtained when the mobile robot takes action a_t at $t+1$, $\gamma \in [0,1]$ is the forward discounting factor controlling the robot, and $e(s_t)$ is the time t state-action trace.

r_{t+1} as a momentary reward, the size is generated by the formula (4) task return function $R(x)$, when the mobile robot reaches the target point to set a positive reward return, and collides with the obstacle to set a negative reward return.

$e(s_t)$ as a state-action trace for recording the number of accesses to a state that contribute to reaching the target point during the mobile robot path planning process. Once the state is accessed, the state-action trace is increased, and vice versa, backpropagating to all previous states, speeding up the convergence until the target point is reached.

3 Simulation

The simulation verification is divided into two aspects: one is the ROS simulation experiment; on the other hand, the performance analysis based on MATLAB.

3.1 ROS Simulation Experiment

First, a brief introduction of the Stage and Rviz is used by the ROS simulation. Second, the simulation system is built and the simulation experiment is carried out. Final, the stability of the algorithm is analyzed and the results are analyzed.

Stage is a software of the Player/Stage project in ROS that simulates the virtual environment of 2D graphics, robots, sensors and actuators, including sonar, laser scanning rangefinder, speedometer, anti-collision device, and general robot base. Some degree of Stage simulation ability can achieve high simulation test effect, when the equipment requirements are met, there is almost no need to make any adjustment can apply to the real mobile robot control environment.

Rviz is a 3D visualization tool in ROS, which is shown in images, models, paths, etc., and the effects of the visual rendering are completed, and the effect of this article is finally observed in the Rviz and Stage environment. The

communication mechanism between the modules in this paper is shown in Fig. 3.

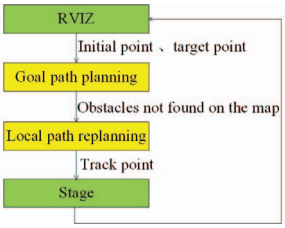

Fig.3 ROS communication

This paper focuses on solving path planning problems in complex and unknown environments, not in physics. Although the structure is not considered, this method is applicable to all kinds of robot programming. The simulation environment of the wheeled mobile robot Turtlebot2, the Stage robot simulation software and an Rviz instance are given in this paper, as shown in Fig. 4 and Fig. 5. Obstacles and goals in the environment are static, obstacles and environmental boundaries are gradually perceived by the robot.

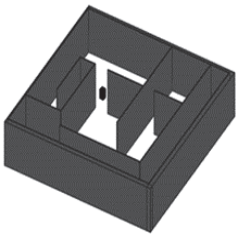

Fig.4 Stage robot simulation

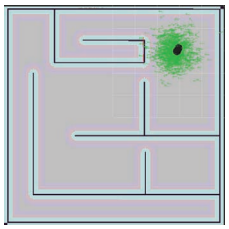

Fig.5 Example of Rviz

With a given environment map, a starting position, and a target location, the RL-RRT method makes the Turtlebot2 move from somewhere to the other. In Fig. 5, the grey represents an open area, black indicates an obstacle.

The global planner in navstack uses the path planning algorithm and the map to plan a shortest path from current location to target location. The shortest path is then passed into the local planner, and the local path tries to drive the mobile robot along this path. If the mobile robot is blocked by a number of obstacles that are not visible in the map, the local planner will use the sensor's information to bypass the obstacles in front of the mobile robot and recover from the error.

Although the global path is the path that the robot wants to follow, the actual path is actually generated by a local path plan. Therefore, local path planning is coordinated with the global path and the local obstacles detected by the sensor.

The simulation environment is a 3D simulation experiment with a length of 13m, and with a width of 13m, the height of the wall is 2.5m and the thickness is 0.15m. This is shown in Fig. 6. Turtlebot2 has a starting point of (100,100,0), and the target point is (110,30,0), the environment is surrounded by the wall, and the mobile robot needs a narrow channel to reach the target point. The diameter of Turtlebot2 is 351.55mm, with a height of 750mm, and the maximum speed is 0.65m/s. Fig. 7 shows the position of the moving robot to the target point.

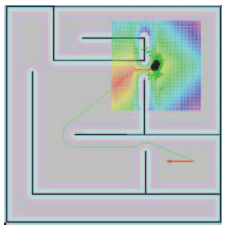

Fig.6 Initial position in Rviz

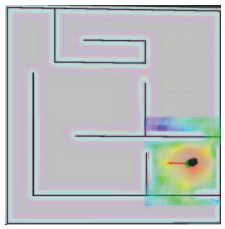

Fig.7 Target position in Rviz

Fig.8 Initial position in Stage

Fig.9 Target position in Stage

Further analysis showed that the robot has sonar, laser and other sensors, and the robot keeps detecting the location information of the obstacles in the environment during the movement, as shown in the green area of Fig. 8 and Fig. 9. Because of the limited range of detection, and it can be found in Fig. 6 and Fig. 7, Turtlebot2 is surrounded by a color square area, in the meanwhile, the square area also represents the scope of the local path planning. On the one hand, there is no obstacle in the range of Turtlebot2's sensor, Turtlebot2 will be passed quickly, as shown in Fig. 10. On the other hand, when Turtlebot2 is detected in a narrow channel, it will adjust to the security position in time, changing the right angle, the slower speed through the channel, and the dark grey color represents the trajectory of the adjustment of Turtlebot2, as shown in Fig. 11. Therefore, the RL-RRT method can be obtained in the local path planning process, and the robot will move smoothly, and could adjust the pos according to the change of the obstacle information, and adapt the change of the environment. As shown in Fig. 12, the red line represents the original RRT path, and the blue line represents the optimized path of the RL-RRT method in the three-dimensional environment.

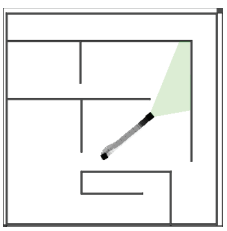

Fig.10 Open area

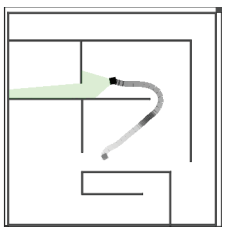

Fig.11 Narrow channel

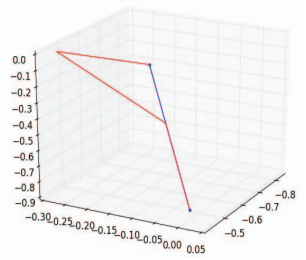

Fig.12 RL-RRT path

3.2 MATLAB Simulation

This paper uses MATLAB simulation platform and designs two different environments, the first is narrow channel, the second is the obstacle-heavy environment. In the experiment, black areas are obstacles, and white are viable areas. In the comparison of experimental analysis, RL-RRT method is compared with the RRT and RRT-Connect algorithm. The extension process of random generated trees in original RRT, RRT-Connect RL-RRT method are shown in Fig. 13-16. The blue line in the diagram is the generated tree of the starting position node, the green line in the diagram is the generated tree of the target location node, and the red line is the algorithm planning from the starting position to the target location path trajectory.

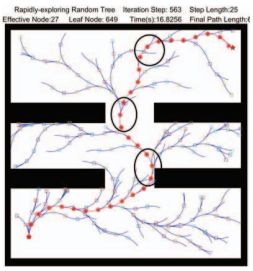

Fig.13 RRT

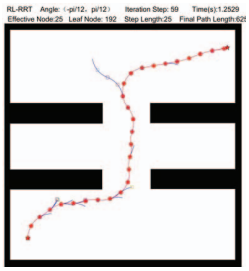

Fig.14 RL-RRT

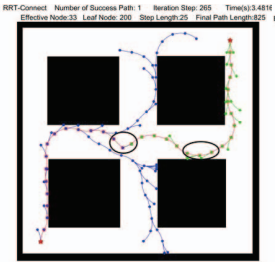

Fig.15 RRT-Connect

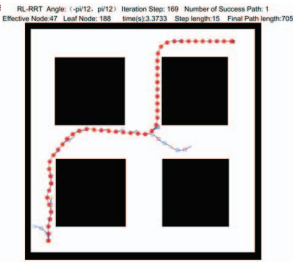

Fig.16 RL-RRT

As shown from Fig. 13, the original RRT algorithm extends several times around the channel in narrow channels, producing a large number of invalid points and wasting resources. And the circle, which is referring to the black arrow, can be seen that the angle of the position is large and the algorithm is less efficient. As shown in Fig.14 and table 1, the RL-RRT method reaches the target point with less extension, appropriate angle change and faster speed. As shown in Fig.15, Fig.16 and table 1, the RL-RRT method is compared with the RRT-Connect, and the effective node is higher, less time and less resources. In conclusion, the RL-RRT method is improved in the narrow channel and the obstacle-heavy environment.

Table 1 Comparison of different environmental performances of RRT and RL-RRT method

Environment	Algorithm	Effective Node Ratio	Iteration Number	Time	Effe-ctive Length
Narrow channel	RRT	4%	563	16.8	725
	RL-RRT	13%	59	1.2	625
Obstacle-heavy environment	RRT-Connect	16%	265	3.4	825
	RL-RRT	25%	169	3.3	705

4 Conclusions and Future Work

This paper discusses the problem of local path planning of the wheeled mobile robot in unknown and complex environments, adopts the planning idea of reinforcement learning, learns the avoidance strategy, effectively combines the prediction control with the feedback mechanism, and compares it with the other path planning algorithms. Under unknown and complex environments, the optimal path planning of RL-RRT method enables the wheeled mobile robot to efficiently complete the planning tasks, shorten the completion time and reduce energy consumption, and reasonable rotation angle setting also improves the stability of wheeled mobile robot. These experiments confirmed that RL-RRT method has the efficiency, effectiveness and adaptability of the path planning application in the unknown and complex environment.

This paper, from the improvement of path planning efficiency, reduces the blindness of RRT algorithm sampling to some extent, improves the efficiency of avoidance, and seeks a way to solve the local path planning method. However, the scope of this study is limited, and the generalization ability of the algorithm can be further improved, such as more complex environment, so that the wheeled mobile robot can adapt to dynamic and unstructured environment.

REFERENCES

[1] Cortright E M. Apollo Expeditions to the Moon: The NASA History 50th Anniversary Edition[M]. Courier Dover Publications, 2019.

[2] Kalita H, Thangavelautham J. Automated Multidisciplinary Design and Control of Hopping Robots for Exploration of Extreme Environments on the Moon and Mars[J]. arXiv preprint arXiv:1910.03827, 2019.

[3] Anlin Fang, Chen Tao, Cheng Aiguo, et al. Intelligent vehicle routing simulation based on artificial potential field algorithm [J]. Automobile Engineering, 2017, 39 (12): 1451-1456. DOI: 10.19562/j.chinasae.qcgc.2017.12.015.

[4] Bai Jianlong, Chen Yinning, Hu Yabao, et al. Ant colony algorithm based on negative feedback mechanism and its application in robot path planning [J]. Computer Integrated

Manufacturing System, 2019, 25 (7): 1767-1774. DOI: 10.13196/j.cims. 2019.07.017.

[5] Long Y, Su Y, Zhang H, et al. Application of Improved Genetic Algorithm to Unmanned Surface Vehicle Path Planning[C]//2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2018: 209-212.

[6] LaValle S M. Rapidly-exploring random trees: A new tool for path planning[J]. 1998.

[7] Chen yanjie, wang yaonan, tan jianhao, et al. Path planning of service robot with incremental sampling of local environment [J]. Chinese journal of instrumentation,2017,38(5):1093-1100.

[8] Zhuge Chengchen, Xu Jinsong, Tang Zhenmin. Local path planning algorithm based on support vector machine [J]. Journal of Harbin Engineering University, 2019, 40 (2): 323-330. DOI: 10.11990/jheu. 201708085.

[9] Ni J, Wu L, Shi P, et al. A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles[J]. Computational intelligence and neuroscience, 2017, 2017.

[10] Li Peng, Yang Caiyun, Wang Shuo. Local Path Autonomous Planning for Mobile Robots Oriented to Map Construction [J]. Control Theory and Applications, 2018, 35 (12):1765-1771. DOI: 10.7641/CTA. 2018.80486.

[11] Sutton R S, Barto A G. Introduction to reinforcement learning[M]. Cambridge: MIT press, 1998.

[12] Faust A, Chiang H T, Rackley N, et al. Avoiding moving obstacles with stochastic hybrid dynamics using PEARL: preference appraisal reinforcement learning[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 484-490.

[13] Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving[J]. arXiv preprint arXiv:1610.03295, 2016.

[14] Cao X, Zou X, Jia C, et al. RRT-based path planning for an intelligent litchi-picking manipulator[J]. Computers and electronics in agriculture, 2019, 156: 105-118.

[15] Song Z, Yuan L. Application of Improved A* algorithm in Mobile Robot Path Planning[C]//2019 3rd International Symposium on Autonomous Systems (ISAS). IEEE, 2019: 534-537.

[16] Huang H, Huang P, Zhong S, et al. Dynamic Path Planning Based on Improved D* Algorithms of Gaode Map[C]//2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2019: 1121-1124.

[17] Li B, Yue Z, Zhang J, et al. High‐Resolution Terrain Analysis for Lander Safety Landing and Rover Path Planning Based on Lunar Reconnaissance Orbiter Narrow Angle Camera Images: A Case Study of China's Chang'e‐4 Probe[J]. Earth and Space Science, 2019, 6(3): 398-410.

[18] Sutton R S. Learning to predict by the methods of temporal differences[J]. Machine learning, 1988, 3(1):9-44.

[19] Parr R, Russell S. Approximating optimal policies for partially observable stochastic domains[C]//IJCAI. 1995, 95: 1088-1094.

[20] Xu D, Fang Y, Zhang Z, et al. Path Planning Method Combining Depth Learning and Sarsa Algorithm[C]//2017 10th International Symposium on Computational Intelligence and Design (ISCID). IEEE, 2017, 2: 77-82.