Path Planning Algorithms for Mobile Robots Based on Deep Reinforcement Learning

Zihan Zhang

Zhenhai High school of Ningbo, Ningbo, China

jinwen@ldy.edu.rs

Abstract. This paper gives an introduction of path planning algorithms for mobile robots based on deep reinforcement learning (DRL). Firstly, the traditional path planning algorithms are compared with the deep reinforcement learning path planning algorithms. One key advantage of DRL-based algorithms is their ability to meet real-time interactions due to lower calculation requirements. Then, some basic path planning algorithms based on deep learning used in unknown and dynamic environment are introduced. After that, the efforts on the improvement of the basic DRL-based path planning algorithms are discussed. Their training results are compared with the traditional path planning algorithms and the DRL-based path planning algorithms without improvements. Finally, the advantages and disadvantages of DRL-based path planning algorithms are analyzed. Although these algorithms have already a relative fast convergence speed and high success rate, they still do not meet the accuracy requirement for sophisticated work. Besides, the train period is also expected to be shorten further. The potential improvements in the future are also pointed.

Keywords: Deep reinforcement learning, path planning algorithms, mobile robots.

1. Introduction

Path planning algorithms have been playing a significant role in controlling mobile robots. With the development of mobile robots, the path planning algorithms used are also changing and updating.

Before the wide application of Deep Reinforcement Learning (DRL), path planning algorithms aimed to search the map in mathematical ways. Depth First Search (DFS) and Breadth First Search (BFS) are two basic path planning algorithms. The Dijkstra algorithm is also a path planning algorithm. It can find the shortest path between the initial point and the goal. Although they can successfully find the path in a simple environment, they are not able to solve complex map due to the extremely long period will be taken. Compared with the algorithms mentioned above, A* algorithm is better because it uses heuristic to explore the map [1]. In addition, there are also other algorithms such as Dynamic Window Approach (DWA) and Genetic algorithm [2].

However, those algorithms have some disadvantages in common. The numerous calculations they required make them difficult to meet the real-time requirements [3]. To enable the robots to adapt a more complex environment, the algorithms based on Deep Reinforcement Learning are invented. The general ideas are using the information gained during the exploration to update the neural network with reward values [3]. Although the models need to be trained before used, they can adapt the specific environment very well.

In this article, several algorithms based on Deep Reinforcement Learning will be introduced. Firstly, the application of path-planning algorithms in unknown and dynamic environment will be introduced. The Proximal Policy Optimization (PPO), Double Deep Q-Network (DDQN), Deep Deterministic Policy Gradient (DDPG) algorithms and the combination of Q-learning and AKL model have been regard as the solution of these complex environments [4-6]. After that, some innovative improvements on the basic path planning algorithms based on DRL are introduced, including the combination of PRM and TD3, TD3-KDHER algorithm, RLPSO algorithm and a use of time-sensitive reward [7-10]. The improved algorithms are compared with the traditional and original unimproved path planning algorithms. These improvements can substantially increase the quality of the samples, as well as the convergence and success rates of the algorithms.

Besides the introduction of the algorithms, the analysis of the advantages and disadvantages of different algorithms are done in this article. Finally, the potential trends and space for further improvement are discussed.

2. In different environments

2.1. In unknown environment

Although a few of path-planning algorithms such as A* algorithm and probabilistic roadmap has been introduced to mobile robots, they cannot be used when the global map is unavailable. Hao Qin et al. contributed a DRL-based algorithm dealing with the unknown environment [4].

They used Proximal Policy Optimization (PPO) algorithm to train the robot [4]. Because the signal from Lidar can be used as the input of learning work directly, this algorithm doesn't combine the motion controller of robot except in the end [4]. This can significantly shorten the time for training the robot. Besides, their assumptions simulate the unknown environment well [4]. The Lidar can gain a view of 360°, but it can only detect the objects in 15m from it, so the robot can only get the local map of the environment [4]. In addition, they used PID control to control the robot to move from one point to the next point with an error less than 0.02m [4].

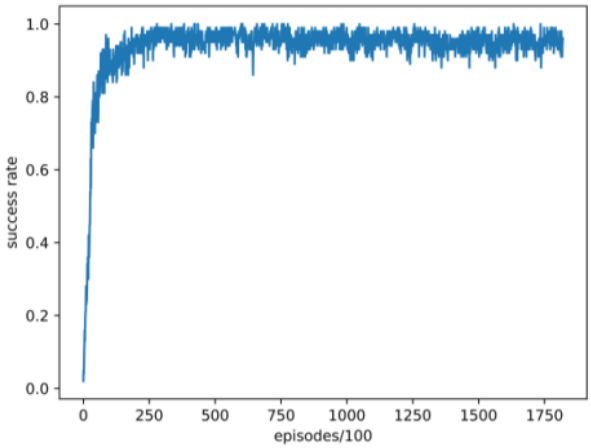

Figure 1. The success rate against the episodes in the training [4]

Their research has won great success. As in Fig. 1, the success rate rises very quickly. The success rate soon reaches 90% in 500 episodes. Besides, the average reward reaches 1000 in 1000 epochs. According to their experiment, the whole process for training only takes around 1 hour, substantially saving the time for training [4].

2.2. In dynamic environment

The core of Zhixiang Cheng et al's research is using a robot based on ROS system and constructing the environmental map through the Simultaneous Localization and Mapping (SLAM) algorithm. They used the Cartographer algorithm, one kind of SLAM algorithm. This algorithm is efficient in memory and many computing resources, since it's based on graph optimization. His algorithm can also perform better in an environment with a large area, compared with particle filter algorithm. Besides, because of its loop detection, the errors produced in detection can be eliminated. Thus, these errors will not accumulate to result an inaccurate construction of surroundings. In addition, a simple DWA algorithm failed to solve the dynamic path planning, so they improved the DWA algorithm through the Bug2 algorithm, which enabled the robot to avoid the obstacles successfully [5].

Despite these improvements, some important factors in real life were ignored in this research. The 2D Lidar can only detect a plane, so the robot may fail to avoid the short obstacles and cannot perform well in a rough environment. More importantly, the Lidar cannot detect transparent objects, so the robot may ignore the glass-made obstacles [5].

Irene - Maria Tabakis and Minas Dasygenis' research also provides solutions for dynamic environment. They emphasized the limitations of traditional path-planning algorithms such as Dijkstra in the complex heterogeneous environment. Then, they compared the effectiveness of Double Deep Q-Network (DDQN) and Deep Deterministic Policy Gradient (DDPG) algorithms as solutions of DRL algorithms for the complex environment [5].

Firstly, DDQN is used because of its higher stability compared to DQN, making it better suited for discrete changing environments. Additionally, DDPG is chosen for its suitability in continuous action spaces [5].

Table 1. The performance of DDQN and DDPG in different environment [5]

Scenario type	DDQN Performance	DDPG performance
Static	High Success Rate: 95% Path Efficiency: 105%	Moderate Success Rate: 90% Path Efficiency: 107%
Dynamic	Moderate Success Rate: 88% Path Efficiency: 115%	High Success Rate: 94% Path Efficiency: 110%
Mixed	Good Success Rate: 91%	Excellent Success Rate: 92%

According to the experiment, DDQN and DDPG have different advantages in different environments as shown in Table 1. DDQN is more suitable for a static environment with a success rate of 95%, since it is more stable and predictable. On the other hand, in a dynamic environment DDPG is preferable since it has a higher adaptability [5].

However, these results showed that there is still not an algorithm suitable for both static and dynamic environment. The needs to combine the advantages of different algorithms still exist.

In Yanming Hu et al's research, Q-learning and the adaptive kernel linear (AKL) model are combined for incremental learning, in order to improve the performance of mobile robots in changing environments [6].

In the varying environment, the data cannot be detected in advance, so the offline learning cannot be adopted. Although Incremental hierarchical discriminant regression (IHDR) and developmental network (DN) can be used to solve this problem, the cost of them is quite high due to their needs for high-quality training materials. However, in their research, the adjustment of the parameters for the AKL model is done by L2-norm kernel recursive least squares (L2-KRLS). Innovatively, saving the costs. Besides, to short the time for the convergence of Q-learning, local ε-greedy policy is adopted and this can select the samples with higher quality [6].

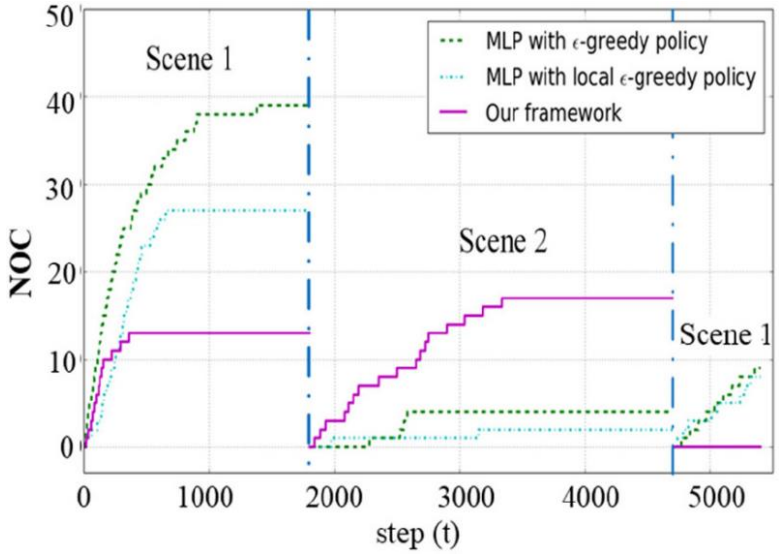

Figure 2. The convergence results for different methods [5]

They trained their model in two scenes, a wide corridor Scene 1 and a wide corridor with obstacles Scene 2. Fig. 2 shows the success of the use of local ε-greedy policy, with the blue line (the second method) having a higher convergence rate than the green line (the first method) in both scenes. More importantly, when the model is used for Scene 1 again, their framework (the red line) is the only one which can still plan the path successfully [6]. This emphasizes the ability to have a memory about the scenes. However, it also highlights the need for improvement in convergence rate across different scenes.

3. Improvements on basic DRL algorithms

3.1. A combination of PRM and TD3

In most cases, DRL-based path planning needs a strong support of computation resources. The long training period of the model limits its popularization among real mobile robots due to high computational requirements. To address these challenges, Junli Gao et al. adopted a series of measures in order to solve this severe problem [7].

Firstly, differing from traditional way, they did not train the DRL model in a 3D simulation environment with physical engine. Instead, they constructed a 2D environment without a physical engine, successfully saving much time and valuable resources for computation. Executing the same task, it took 26 hours to train the model in the 3D Gazebo, while in 2D Gym environment it only took 5 hours, executing the same task. Besides, the same interface for Gazebo and Gym enables that the best solution from Gym can be transferred into Gazebo environment [7] (Fig. 3).

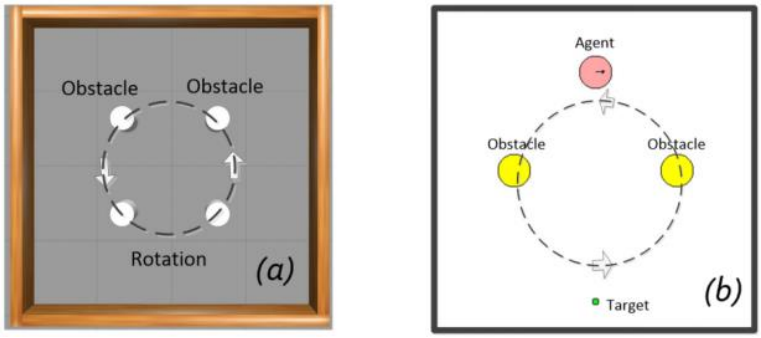

Figure 3. Simulation environment for the model. (a) The 3D Gazebo environment. (b) The 2D Gym environment [7]

Secondly, they combined Probabilistic Roadmap (PRM) with Twin Delayed Deep Deterministic Policy Gradients (TD3). In large scale environment, this method can find a shorter time for single steps compared with A*+DWA and A*+TEB. As shown in Table 2, the time taken for one simple step by PRM+TD3 is less than one tenth of other algorithms [7].

Table 2. Planning index of different algorithm [7]

Numble	Algorithm	lpath (m)	Tstep (s)	Tpath (s)
1	A*+DWA	14.54	0.06~0.3	74.50
2	A*+TEB	13.64	0.05~0.4	72.10
3	PRM+TD3	15.60	0.001~0.01	80.24

However, the drawback of their strategy is obvious. The length of total path and the total time acquired are longer than other algorithms. The reason for this non-optimal solution gained may be caused by the sampling nature of PRM. In addition, this algorithm cannot get stable solutions [7].

3.2. A novel DRL algorithm, TD3-KDHER

To address the possibly frequent collisions and low efficiency, Jin Zhang and Hongqiang Zhao combined TD3 with Hindsight Experience Replay (HER) to form TD3-KDHER [8].

Firstly, the reason for using HER is that this can improve the utilization efficiency of the training samples. This algorithm uses Kernel Density Estimation (KDE) to improve the quality of samples in complex environment, thus showing a higher convergence speed. Furthermore, the use of adaptive action noise and action masking strategies can lead to a better exploration strategies, so the efficiency of training can be improved further [8].

According to the experiments, TD3-KDHER has a higher success rate and shorter convergence time than other algorithms, as shown in Fig. 4 Jin Zhang et al.'s experiments also showed that the average time and length of path needed are shorter than the other 2 algorithms. These results suggest the TD3-KDHER is more suitable for the path planning in a complex environment than single TD3. It is expected to be combined with other algorithms and widely used in mobile robot to explore complex environment [8].

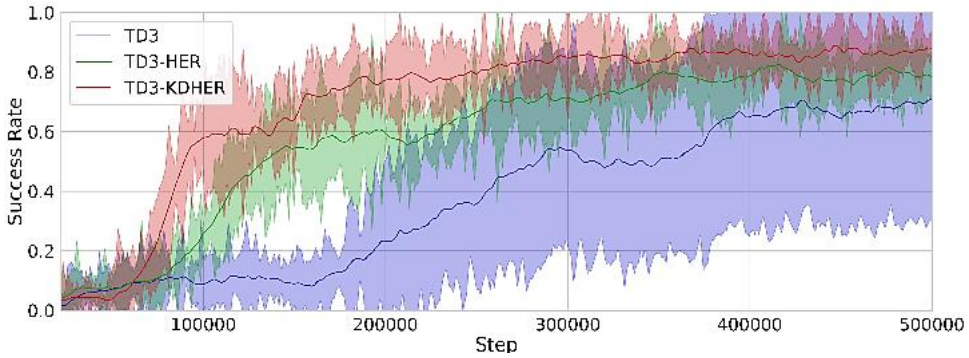

Figure 4. The training result of different algorithms [8]

3.3. RLPSO for improvements on Q-learning

The problem of low convergence rate is one of the biggest problems of deep reinforcement learning algorithms. Yang Yang and Linhao Zhang proposed an reinforcement learning particle swarm algorithm (RLPSO) based on Q-learning and the particle swarm algorithm [9].

Traditional Q-learning algorithm needs a long period for exploration and the length of the planning path can be rather long, but the learning rate and discount factor of Q-learning can be improved by the particle swarm algorithm. In addition, particle swarm algorithm can adjust the parameters of the Q-learning, gaining a stronger ability to find a shorter path [9].

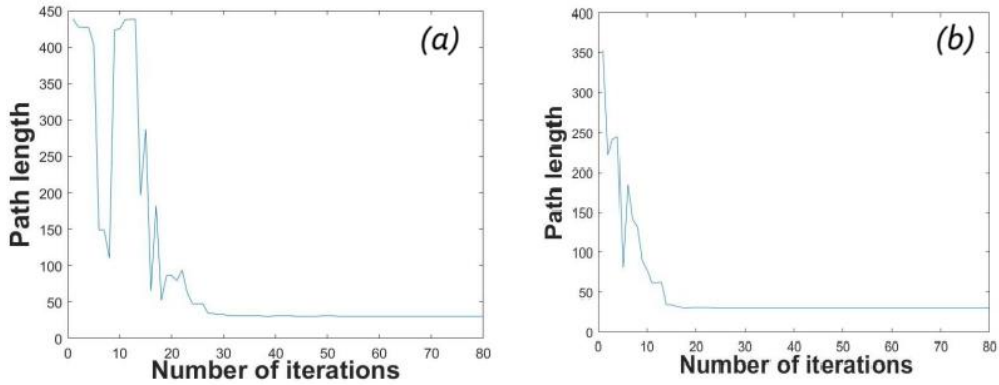

Figure 5. The length of the planning path for (a) unoptimized (b) optimized Q-learning [9]

As shown in Fig. 5 the stability of Q-learning has been improved after the adoption of particle swarm algorithm, since the path length keeps go down. Besides, the convergence rate has almost been doubled, proving the success of RLPSO.

3.4. A use of time-sensitive reward

In a map with a large size, the convergence rate of existing DRL algorithms tend to be low due to sparse rewards. Zhao Ruqing et al. designed a novel time-sensitive rewards for the agent. To avoid

gaining sparse rewards during the exploration, collision information is integrated into the final reward. To solve the problem brought by changing target due to Q-functions, the freezing target networks method is used to fix the target network. Then, the Evaluation Q-network replaces the Target Q-network, ensuring stability during training [10].

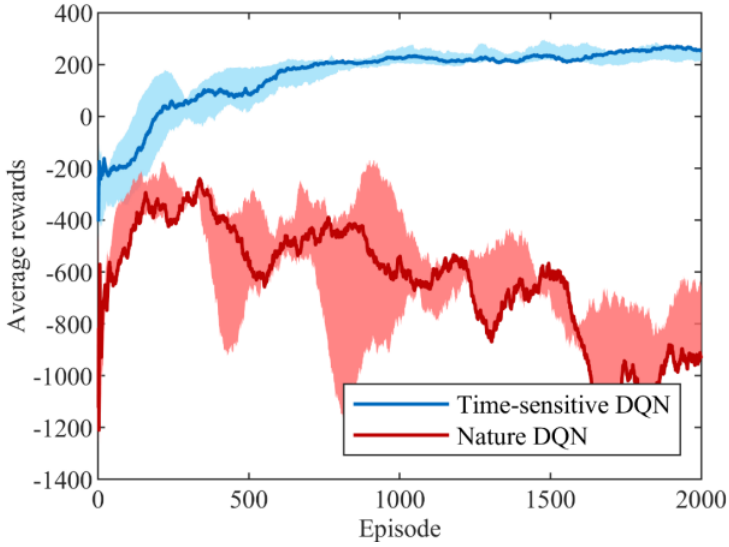

Figure 6. The performance of Time-sensitive DQN and Nature DQN in learning [10]

Fig. 6 compares the performances of Time-sensitive DQN with Nature DQN. The map used in the experiment is a sparse grid which the mobile robot needs to have a strong exploration ability. From Fig. 6 the Nature DQN is unable to find the correct path in that problem, as the average rewards become lower and lower, while the Time-sensitive DQN can explore the map efficiently. However, there are still some improvements that can be made. For instance, a time-sensitive reward system that can automatically adjust parameters is needed to enable the robot to explore environments of varying sizes [10].

4. Conclusion

In this paper, some path planning algorithms based on Deep Reinforcement Learning are discussed. Each of them has its own advantages and disadvantages. Firstly, the algorithms are discussed in the context of different environments. In an unknown environment, PPO algorithm can be used to train the robot and has a high success rate. For a changing environment, DDPG can keep a relative high success rate. Besides, Q-learning and AKL can be combined, with the ε-greedy policy, to make the robot has a memory of the environments which it explored before.

Secondly, improvements on basic DRL-based path planning algorithms are also discussed. For example, the 3D training environment can be turned into 2D to significantly shorten the training time. Then, the combination of PRM and TD3 can shorten the time for single step during training. TD3 can also be combined with HER to form TD3-KDHER so the training efficiency and success rate can be improved. Besides, the RLPSO based on Q-learning and particle swarm algorithm can reduce the long period, improve the stability and find a shorter path. In addition, a time-sensitive rewards can enable the mobile robot to explore the map in a more efficient way.

However, these algorithms still have their drawbacks which can be studied by future research. For example, the algorithms analyzed in this paper are only responsible for the path planning. In real environment, how to convey the path to the motors and how to correct the deviation of the path due to the motor are the problems need to be considered. Secondly, most of these algorithms still need a relative long period for training to adapt a specific environment, which means they may not able to meet the emergencies like an environment suddenly changed. Besides, the success rate can be kept around 90% for most DRL-based algorithms mentioned in this paper. This means these algorithms can still not solve the missions which have high requirement on success rate such as automatic driving.

References

[1] Karur K, Sharma N, Dharmatti C, Siegel JE. A Survey of Path Planning Algorithms for Mobile Robots. Vehicles, 2021, 3 (3): 448-468.

[2] Qin H, Shao S, Wang T, Yu X, Jiang Y, Cao Z. Review of Autonomous Path Planning Algorithms for Mobile Robots. Drones, 2023, 7 (3): 211.

[3] Han H, Wang J, Kuang L, Han X, Xue H. Improved Robot Path Planning Method Based on Deep Reinforcement Learning. Sensors, 2023, 23 (12): 5622.

[4] Qin H, Qiao B, Wu W, Deng Y. A Path Planning Algorithm Based on Deep Reinforcement Learning for Mobile Robots in Unknown Environment. In Proceedings of the 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 2022: 1661-1666.

[5] Cheng Z, Li B, Liu B. Research on Path Planning of Mobile Robot Based on Dynamic Environment. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, Guangxi, China, 2022: 140-145.

[6] Hu Y, Li D, He Y, Han J. Incremental Learning Framework for Autonomous Robots Based on Q-Learning and the Adaptive Kernel Linear Model. IEEE Transactions on Cognitive and Developmental Systems, 2022, 14 (1): 64-74.

[7] Gao J, et al. Deep reinforcement learning for indoor mobile robot path planning. Sensors, 2020, 20 (19): 5493.

[8] Zhang J, Zhao H. Mobile Robot Path Planning Based on Improved Deep Reinforcement Learning Algorithm. In Proceedings of the 2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE), Guangzhou, China, 2024: 1758-1761.

[9] Yang Y, Zhang L, Guo H. The Path Planning Research for Mobile Robot Based on Reinforcement Learning Particle Swarm Algorithm. In Proceedings of the 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE), Shanghai, China, 2024: 1577-1580.

[10] Ruqing Z, Xin L, Shubin L, Jihuai Z, Fusheng L. Deep Reinforcement Learning Based Path Planning for Mobile Robots Using Time-Sensitive Reward. In Proceedings of the 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2022: 1-4.