Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/14/20230788

Research on path planning techniques based on deep reinforcement learning

Tianyu Dong

School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, China

dty1210@mail.sdufe.edu.cn

Abstract. Path planning problem stands for a sort of problem that find a secure path from the departure position to the destination in the unique application area without conflicts with other objects under the premise of minimum time or distance cost. Path planning has many applications in real life. This paper summarizes and classifies the current main research results on path planning, which are classified according to conventional techniques, single-agent path planning techniques and multi-agent path planning techniques. Conventional techniques are classified into three different algorithms: Traditional techniques, Graphics techniques and Intelligent bionics techniques. For the single-agent path planning techniques, it is primarily broken down into two categories: value-based and strategy-based. For multi-agent techniques, according to the improvement techniques, it is split in three categories: expert demonstration type, improved communication type and task decomposition type. Based on above classification, the mechanisms, advantages, and disadvantages of the existing algorithms are summarized and compared, and the future work in this field is prospected.

Keywords: path planning, reinforcement learning, deep reinforcement learning, machine learning

1. Introduction

By developing information technology, artificial intelligence and automation, the degree on machine intelligence continues to improve and as an important indicator to evaluate machine intelligence, path planning problem has received more and more attention. The multi-agent path planning (MAPP). MAPP is to take multi-agent as the target object, the target object to minimize the cost of time, distance, cost, and other conditions, in the specified range area to search a most effective path between the point of departure and the target position, the key constraint is to ensure that the target objects will not conflict with each other under the premise of reaching the target position and it needs to maintain speed and quality during driving [1].

MAPP plays an increasingly important role in many application scenarios, for example, from typical Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP) and other operational research problems to urban traffic network, games, warehouse management, etc., or in the high-tech fields of artificial robots, UAV, unmanned ships and so on. In particular, the connection of deep reinforcement learning and MAPP has become a burning theme within the domain of artificial intelligence [2].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/14/20230788

In recent years, due to the successive evolution in various algorithm models and the exploding expansion in big data, more and more experimental teams have conducted research on MAPE and have made unprecedented breakthroughs. However, there is no systematic classification and summary of various excellent algorithms for MAPP. However, at present, only the classic MAPP algorithms have been classified and sorted out at home and abroad, but the MAPP algorithms which are a new force in artificial intelligence in recent years has not been systematically summarized and classified [3]. In this paper, the MAPP problem is described in detail, the path planning techniques of common methods are sorted out and the advantages and disadvantages of the algorithms are compared. Then the associate concepts of DRL are described, single agent reinforcement learning algorithms are outlined and contrasted in detail, and the principle and characteristics of the current mainstream multiple agent reinforcement learning algorithms are analysed. On the basis, the future development of the path planning algorithms built on reinforcement deep learning are prospected from practical problems.

2. Main body

2.1. Based on the conventional method of path planning problem

2.1.1. Path planning problem. Path planning problem stands for a sort of problem that find a secure path from the departure position to the destination in the unique application area without conflicts with other objects under the premise of minimum time or distance cost [1].

2.1.2. Conventional methods. Traditional path planning techniques are often divided into three categories: traditional techniques, graphics techniques, and intelligent bionics techniques.

1)Traditional techniques mainly cover the Simulated Annealing Algorithm (SAA), Artificial Potential Field Algorithm (APFA) and the Fuzzy Algorithm (FA). These algorithms have the main characteristics of simple implementation and easy description, so they were first used to solve the path planning problem, but their defects are also obvious, they cannot fully use the previous information and global information, and they often trap into the problem of the local optimal solution or unachievable goal in the way of solving practical problems [4].

2) Graphics techniques mainly include A-Star Algorithm (A*) and Grid Algorithm (GA). These algorithms generally provide modelling methods to solve the problem that cannot be modelled in traditional algorithms. However, due to their cumbersome modelling process and relatively low search efficiency, it is difficult to be widely used in practical scenarios [5].

3) Intelligent bionics techniques mainly include Genetic Algorithm (GA), artificial Neural Network algorithm (ANN), Ant Colony Algorithm (ACA) and Particle Swarm Optimization Algorithm (PSO). The basic principles of these algorithms are very close to the nature or ecological mechanism of organisms in nature, such as imitating the law of biological genetics and evolution, and many common biological behaviors such as ant colony foraging and migration, so the technology is called intelligent bionic technology. Due to the bionic characteristics of the technology, this kind of algorithm is more intelligent and more efficient than the first two types of algorithms, but in the actual application process, there are still many problems existing high possibility to trap into the partial optimum solution, slow convergence speed of results and so forth.

In order to compare various conventional techniques and algorithms more intuitively, Table 1 summarizes the advantages and disadvantages of various conventional algorithms.

DOI: 10.54254/2755-2721/14/20230788

Table 1. The advantages and disadvantages of various conventional algorithms.

Conventional Techniques	Algorithms	Advantages	Disadvantages
Traditional techniques	SAA	It has simple implementation and easy description	It is easy to fall into local optimal solution or the target is unreachable
	APFA		
	FA		
Graphics techniques	A*	It provides modelling methods to tackle issues	Low search efficiency
	GA (Grid Algorithm)		
Intelligent bionics techniques	GA (Genetic Algorithm)	It has the characteristics of bionics and is more intelligent and efficient	It is easy to fall into the local optimal solution and the convergence speed is slow
	ANN		
	ACA		
	PSO		

2.2. Reinforcement learning based on deep learning

2.2.1. Reinforcement learning. RL refers to the continuous error correction learning of the agent under the stimulus of reward or punishment feedback in a specific environment, and continuously adjust the strategy according to the feedback, and finally achieve a specific goal or maximize the reward [5]. Reinforcement learning methods mainly include four elements: state, strategy, action, and reward. In state s_t, the agent selects an activity at related policy π and moves from state s_t to a new stage s_{t+1}, and receiving an award r from the environment feedback. Based on the obtained reward r, the agent obtains the optimal policy π^*.

$$\pi^* = argmax_\pi E_\pi[\sum_{t=0}^{\infty} \gamma^t r(S_t, a_t)|S_0 = S]\ \gamma\epsilon(0,1) \quad (1)$$

When mobile robots carry out path planning in complex and unknown environments, the blindness of the initial exploration strategy leads to the slow convergence of reinforcement learning, and it consumes massive of time in robot training process. Moreover, with the increase of environment complexity and system state dimension, the amount of parameters to be formed increases exponentially, so it will cost lots of training time or storage space, eventually lead to the curse of dimensionality. In addition, reinforcement learning has poor portability and generality, and the trained robot cannot directly move in a new environment according to the desired plan.

2.3. Single agent reinforcement learning algorithms

DRL connects the decision-making ability of RL and the representation advantages of DL, and has achieved significant success in AI fields. This section briefly reviews single agent algorithms for value function-based DRL and policy gradient-based DRL.

DOI: 10.54254/2755-2721/14/20230788

2.3.1. DRL algorithm based on value function. Value-based methods are mainly applicable to discrete action Spaces, in which the target is to obtain optimal policy by maximizing value function for each state. Value function is used to measure the goodness of the robotic selection strategy at the current stage. According to the discernable independent variables, value function can be divided into the state-value function $V\ (s)$ and the state-action pair value function $Q(s,a)$, as shown in Equations (2) and (3).

$$V^{\pi}(\mathrm{s}) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^{t} r(S_t, a_t) | S_0 = S] \tag{2}$$

$$Q^{\pi}(s_t, a_t) = r(s_t, a_t) + \gamma V^{\pi}(s_{t+1}) \tag{3}$$

From Equations (2) and (3), it can be known that the state value function is the reward feedback value under a certain state, and the state-action pair value function is the award feedback value under the state-action pair. Therefore, the final award can be maximized by only maximizing the value function.

The main algorithm based on value function is deep Q-network (DQN), DQN applies neural network to estimate action value function, and adopts neural network to replace Q-table of Q-learning. DQN uses experience replay buffer to decrease the connection between data, and stores. The data $D = e_1, e_2, \dots, e_N \quad e_n \epsilon (s_n, a_n, r_n, s_{n+1}), n \epsilon [1, N]$ connecting with the environment in the experience replay pool. In each iteration, loss function based on TD is used as shown in Equation (4).

$$L_i(\theta_i) = E_{(s,a,r,s')\sim D}[(r + \gamma max_{a'} Q(s', a', \theta_i^{-}) - Q(s, a, \theta_i))^2] \tag{4}$$

In Equation (4), θ_i^{-} and θ_i represent the parameters of the target network and Q network separately. The main drawback of DQN stands for that it can only deal with low-dimensional dispersed action spaces. Since then, the DQN algorithm has evolved into many improved variants, such as Double DQN with two different neural networks, Dueling DQN with action value advantage function, priority experience replay algorithm, NoisyNet DQN that adds network noise to improve search, and Rainbow DQN that combines the above algorithms.

2.3.2. DRL algorithm based on policy gradient. Built on the depth of the value function of reinforcement learning algorithm in discrete space has been widely applied control tasks, and demonstrate the superior performance. However, due to the limitation of the discrete output of the value function, the above algorithms are often powerless in the face of reinforcement learning tasks in continuous action Spaces. Continuous control tasks are exactly where policy gradient-based deep reinforcement learning algorithms come into play [6].

For better learning, Deep Deterministic Policy Gradient (DDPG) connects DQN and AC methods. DDPG includes four neural networks: current Actor network, target Actor network, current Critic network, target Actor network [7-9]. The target of DDPG is to domain Actor functions that map states to actions and to learn Critic functions that estimate state action values.

After DDPG, the classical policy-based deep reinforcement learning algorithms developed rapidly. Schulman proposed the Trust Region Policy Optimization algorithm (TRPO) to solve the oscillation problem of evaluator training, which ensures the stable improvement of policy optimization process. Mnih proposed asynchronous advantage actor-critic (A3C), which is based on AC framework, multi-thread operation and asynchronous sampling training. It greatly enhances training speed and performance of different algorithms, and significantly decreases the need of hardware. Based on TRPO algorithm, Schulman proposed proximal policy optimization (PPO), which optimizes the parameter adjustment process of the policy gradient algorithm. The algorithm only uses a first-order optimization algorithm, and uses a multi-step update algorithm for the policy, which simplifies the implementation process and improves the performance of the algorithm while ensuring the stability and reliability of the algorithm. Haarnoja proposed soft Actor-critic algorithm (SAC), which introduced the concept of entropy into the objective function, and encouraged agents to explore by maximizing entropy, which improved the robustness of the algorithm while avoiding the convergence of agents to suboptimal strategies [1-6]. The main algorithms of single-agent reinforcement learning are summarized in Table 2.

DOI: 10.54254/2755-2721/14/20230788

Table 2. Comparison of single agent reinforcement learning algorithms.

Classification	Algorithm	Mechanism	Characteristics
DRL algorithm based on value function	DQN	Neural network Q-Network and experience replay pool are introduced based on Q-learning. Q-network fits Q value, which effectively solves the problem that Q-table cannot be stored under high-dimensional state and action. Experience replay breaks the interrelationship between data and enhances sample utilization.	The issue of high-dimensional status input and low-dimensional status output is solved, but Q value is overestimated and the training is unstable.
	Double DQN	Two Q-networks are introduced, one network estimates the activity with the maximal value, and the other network estimates Q-value of this action.	The problem of overestimation of Q value is avoided.
	Dueling DQN	The output of DQN neural network is optimized by decomposing $Q(s,a)$ into the summation of the status value function $V(s)$ and Advantage function $Advantage(s,a)$.	The update speed of Q value is greatly improved.
	NoisyNet DQN	Gaussian noise is put to the last layer of the network, and the parameters are updated by the training of the weight network.	To be able to use less computational cost, achieve a better result.
	Rainbow DQN	Six improved techniques of DQN are integrated.	Wider applicability
DRL algorithm based on policy gradient	DDPG	Based on the AC framework, four different neural networks are used: Actor network, Critic network, Actor target network and Critic target network.	Solve problems of continuous motion, the output is a direct action and convergence in small tasks quickly.
	TRPO	Solve the problem of strategy parameters change too much, to guarantee the monotone increasing every step of the new strategy.	Faster convergence.
	A3C	Borrow the idea of an experiential playback pool and refine it, using multiple threads to interact with the environment and update asynchronously.	The probability of each action is output and the training is stable and robust.
	PPO	Based on the AC framework, importance sampling technique and N-step update are used.	The correlation between data is reduced and convergence is faster.
	SAC	The idea of entropy is introduced into the objective function, and the agent encourages exploration by maximizing the entropy.	The robustness of the algorithm is improved and agent convergence to the suboptimal strategy is avoided.

DOI: 10.54254/2755-2721/14/20230788

2.4. Multi-agent deep reinforcement learning algorithms

2.4.1. Multi-agent path planning definition. A standard multi-agent path planning (MAPP) may be set to a quad $< Q, k, P, T, >$. In this quad, $Q =< V, E >$ is a non-directed graph, in which a node $v \in V$ is a position that an agent may occupy, and an edge $e = (v_j, v_q) \in E$ is a line between v_j and v_q, stating that an officer may change from v_j to v_q.k stands for the number of agents of the problem, namely agents $\{a_1, a_2, \ldots, a_k\}$. Each agent's starting position is unique $s_j \in P \in V$ and a single goal post $g_j \in T \in V$. S is the set of original positions of all agents and T is the set of target positions of all agents.

In MAPP, time is generally discretized in increments of time. At any time stage, an agent can only take one action, which is usually split into standby and on-the-go. In MAPP, there are two kinds of conflicts between the two parties, which are collision conflict and exchange conflict [3]. As shown in the figures. Figure 1 represents the collision conflict and Figure 2 represents the exchange conflict.

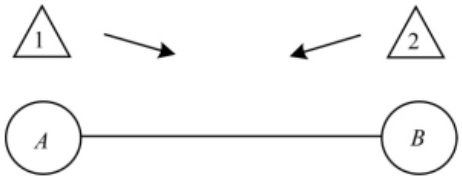

Figure 1. Crash collision

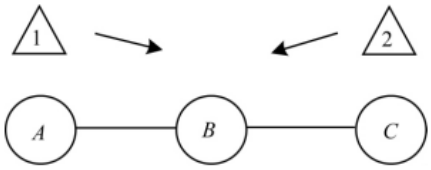

Figure 2. Swap collision.

The sequence of activities undertaken by the agent a_j to move from s_j to g_j constitutes a path p_j, then one possible solution to this issue is the set of k paths $P = \{p_1, p_2, \ldots p_k\}$, where the agent a_j corresponds to the path p_j and there is no collision between two paths p_j and p_q in the path-set. Therefore, multi-agent path planning problems usually need to minimize some global cumulative cost function. The commonly used cost functions are as follows:

$$\max_{1 \le i \le k} t_i \tag{5}$$

$$\max_{1 \le i \le k} l(p_i) \tag{6}$$

$$\sum_{i=1}^{k} t_i \tag{7}$$

$$\sum_{i=1}^{k} l(p_i) \tag{8}$$

The Cost function (5) represents the time taken by the agent arriving at the final position at the latest, the cost function (6) Stands for the length of the longest path in the defined path, the cost function (7) represents the total of the time taken by all agents to reach their respective targeted positions from their respective starting positions. The cost function equation (8) stands for the total length of all paths in the path-set.

The state space of MAPP shows exponential growth in proportion to the increase in the number of agents. Therefore, the optimal MAPP has good practical value only when there is a relatively low number of agents. Sacrificing the optimal performance appropriately can significantly enhance the efficiency of the algorithm, which is the central idea of boundary MAPP.

Using RL methods to solve MAPP problems faces many challenges, such as sparse environmental rewards and complex environmental dynamics. Any kind of reinforcement learning algorithm directly

DOI: 10.54254/2755-2721/14/20230788

applied to MAPF problems will result in slow learning speed and low learning quality. To solve the above problems, researchers have adopted various combination technologies to improve the MAPP method based on RL, so that MAPF method of RL can be extended to the environment of thousands of agents, and the quality and efficiency of solution have been greatly improved [4]. According to the characteristics of the improved technology, it may be roughly broken into three kinds: expert demonstration algorithms, improved communication algorithms and task decomposition algorithms.

1) Expert demonstration algorithms mainly adopt the combination of reinforcement learning and imitation learning (IL) [8]. The trained strategies can be extended to large-scale agent environments with fast learning speed, but the centralized MAPP planner is often used to generate expert presentations, resulting in time-consuming computation.

2) In MAPP with high density of agents, the system needs multiple agents to be related to each other when the environment is not completely known, which requires the communication between agents. Therefore, the improved communication algorithms come into being. The improved communication algorithm has high success rate, low average step size and low communication overhead, but it cannot be applied in large scale because of its long training time and slow learning speed [10-12].

3) MAPP for large dynamical environments is a very complex issue, as agents not only need to effectively reach their goals, but also avoid conflicts with other agents or dynamic objects. An important way to solve this challenge is to decompose tasks, so the task decomposition algorithms were born [7].

The characteristics of the three different classes of algorithms are summarized in Table 3.

Table 3. The characteristics of the three different classes of algorithms.

Classification	Algorithm	Mechanism	Characteristics
Expert Demonstration Algorithms	PRIMAL	Reinforcement learning and imitation learning are combined to train fully decentralized strategies, where agents plan paths in a partially observable environment while exhibiting implicit coordination between agents	The learned policies can be scaled to 1000 agent-scale environments
	MAPPER	To improve the performance of agents in large-scale environments, a long task is decomposed into multiple simple tasks under the guidance of a central planner	It can achieve better results with lower computational cost
	GLAS	Combined with avoid local minimum of the advantages of centralized planning and the advantages of scalable distributed execution	It has a high success rate under various robot and obstacle densities
Improved Communication Algorithms	PICO	Incorporating the implicit planning priority into the decentralized MARL framework, the default priority learning module may be used to form a dynamic communication topology, thus establishing an effective conflict avoidance mechanism	Higher success rate and faster learning
	DCC	Agents choose nearby agents that can change their own strategies for communication	Improve the communication efficiency and reduce the communication bandwidth

DOI: 10.54254/2755-2721/14/20230788

Table 3. (continued).

	DHC	Combining communication with deep Q-learning	Higher success rate, average step size smaller
	MAGAT	Based on a similar one-key query mechanism which can determine information about neighboring agents	Close to the expert performance of centralized planning, it is very effective at different agent densities and different communication bandwidths
Task Decomposition Algorithms	HPL	The MAPP problem is decomposed into two subtasks of reaching the goal and avoiding collisions. In order to complete each task, different reinforcement learning algorithms are used to design the mapping of the agent's observations to actions. Finally, the learned target reaching policy and collision avoidance policy are mixed into a single policy.	Hybrid strategies are significantly better than independent reinforcement learning methods
	G2RL	It combines global planning and local reinforcement learning-based planning to facilitate the learning of end-to-end policies in dynamic environments and it is proposed a new reward structure which offers dense rewards without forcing the agent to strictly follow the overall plan every step of the way, so as to motivate the agent to explore more potential paths	It maintains good performance in different types of maps and different scale obstacle environments, and has good generalization
	VRL	Deep learning techniques are used to extract high-dimensional characteristics from raw visual findings and compress them into effective representations, and graphic neural networks are used to learn how to share and aggregate information between neighboring robots for efficient coordination of local movements	It has good scalability in large robot networks and large environments

3. Problems and challenges

By summarizing, comparing, analyzing, and thinking about the path planning problem solved by various algorithm frameworks of traditional, single-agent algorithm and multi-agent algorithm, this paper summarizes the future research directions as follows:

1) New algorithms are emerging gradually, but there are few studies on the difference between objective functions, and most studies only focus on one of them for verification of algorithm performance.

2) Research on the variant of the classical multi-agent path planning problem is receiving more and more attention. The classical model has weak modeling ability for the problems in real life. In real life, the path planning problem will contain many practical factors, so that the classical algorithm is no longer applicable. Therefore, in order to better apply MAPP algorithms into practical problems, mainstream research cannot be limited to classical models only.

DOI: 10.54254/2755-2721/14/20230788

3) Similar to model fusion in machine learning, considering the differences in solving performance of various multi-agent path planning algorithms for problems with different characteristics, the fusion of various algorithms can be studied to develop comprehensive algorithms to solve MAPP problems more efficiently.

4. Conclusion

This paper first briefly describes the path planning problem, and then summarizes three popular path planning techniques based on traditional methods: Traditional techniques, Graphics techniques and Intelligent bionics techniques are compared and analyzed. Secondly, the related concepts of reinforcement learning and deep learning and the definition of multi-agent path planning are introduced. Moreover, two kinds of single-agent path planning technologies are summarized: DRL algorithm based on value function and DRL algorithm based on policy gradient, and their specific algorithms, mechanisms and characteristics are summarized. After that, the multi-agent deep learning algorithm is introduced, it is broken into three kinds: Expert Demonstration Algorithms, Improved Communication Algorithms and Task Decomposition Algorithms. Mechanism and characteristics of these algorithms are summarized. Finally, prospect of path planning based on DRL is put forward.

References

[1] Liu Q Z, Wu F. Research progress of multi-agent path planning. *Comput. Engineer*. 2020,46(4):1-10.

[2] Yan J J, Zhang Q S, Hu X P. Review of path planning techniques based on reinforcement learning. *Comput. Engineer*., 2021, 47(10):16-25.

[3] Sternr, Sturtevantnr, et al. Multi-agent pathfinding: definitions, variants, and benchmarks, Twelfth Annual Symposium on Combinatorial Search, 2019: 121-132.

[4] Sharon G, Stern R, Goldenberg M, et al. The increasing cost tree search for optimal multi-agent pathfinding, *Arti. Intel*., 2013, 195: 470-495.

[5] Liu Q, Zhai J W, et al. A survey on deep reinforcement learning. *Chinese J. Comput.*, 2018,41(1):1-27.

[6] Ryan M. Constraint-based multi-robot path planning, *Inter. Conf. Robot. Auto.* 2010:38-54.

[7] Shu X X. Research on optimal path planning of multi robots. *China Compu. Comm.*, 2017(15):62-64.

[8] Sartoretti G, Kerr J, Shi Y, et al. Primal: path finding via reinforcement and imitation multi-agent learning. *IEEE Robot. Auto.* Let., 2019, 4(3):2378-2385.

[9] Liu Z, et al. Mapper: multi-agent path planning with evolutionary reinforcement learning inmixed dynamic environments, *Inter. Conf. Intel. Robot. Sys.,* 2020: 11748-11754.

[10] Lin W, Liu Z, et al. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robot. Auto. Let.,* 2021,6(3):5533-5540.

[11] Niu P F, Wang X F. Survey on Vehicle Reinforcement Learning in Routing Problem. *North Minzu Univ.*, 2022,58(01):41-55.

[12] Kool W, Van Hoof H, et al. Deep policy dynamic programming for vehicle routing problems. arXiv:2102.11756, 2021.