Research of the Methods on Multi-Agent Path Finding

Ruining Zhou
School of Mathematics and Statistics, Wuhan University, Wuhan, China
2020302011111@whu.edu.cn

Abstract. Multi-Agent Path Finding is an essential real-life application of multi-intelligent systems for which scholars have solved many classical solutions. And with the evolution of theories related to machine learning, intelligent solutions are emerging daily. However, these algorithms have not been well summarized and concluded, and are comparatively difficult to consult. This essay's objective is to offer a thorough analysis of these issues. The paper will start with Path Finding and describe its three categories of classical algorithms. And then it will focus on the centralized planning algorithms based on the classical algorithms, and the distributed execution algorithms based on machine learning, and provide a theoretical explanation and comparison of their effectiveness. Finally, a prospection and outlook on the currently unresolved issues in the Multi-Agent Path Finding research area, including unification standards and directions for improvement, will be presented.

Keywords: Multi-Agent Path Finding; Classical Algorithms; Machine Learning.

1. Introduction

Reinforcement Learning (RL) is a subset of Machine Learning (ML) that is known alongside supervised and unsupervised learning. In RL, the key concern is how an agent may optimize its rewards in a challenging environment. The core idea is that the agent learns by trial-and-error through continuous interaction with the environment and explores optimal strategies based on the rewards of the environment.

The excellent performance of these techniques in single agent has inspired researchers to apply them to Multi-Agent Systems (MAS). Multi-Agent Path Finding (MAPF) is one of the earliest scenarios for MAS and has been figured out numerous breakthroughs. Finding the ideal set of pathways from the beginning positions to the goal positions for numerous agents is known as the MAPF issue, such as airfield towing, logistics storing, train dispatching, urban road planning and so on [1]. Compared to traditional path planning, MAPF requires the resolution of coordination and conflict between agents, as well as the huge joint action space constituted by respective action spaces.

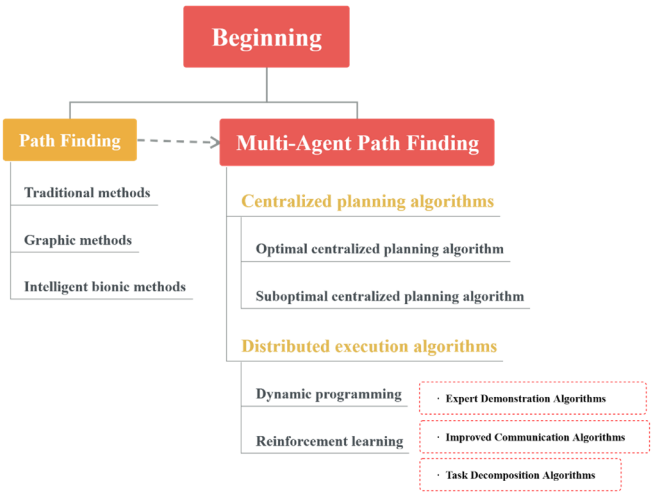

Fig 1. Overall structure of the paper

This paper will present a series of methods for MAPF from classical to RL algorithms. Each method will be evaluated in terms of solution method, solution efficiency and solution quality. On

this basis, the existing algorithms for MAPF will be summarized and the directions of further research will be envisaged. The general framework and main elements of the thesis are shown in the following Figure 1.

2. Multi-Agent Path Finding

This paper examines the notion of MAPF. It begins with an introduction to its predecessor Path Finding and its solution method. Then the classical MAPF is used as an example to illustrate the problem.

2.1 Path Finding

Path Finding is a problem about planning a safe path from the start to the end of a single object within a defined range. In most cases it needs to satisfy an optimization objective, such as minimizing time or cost. The conventional methods are divided into three categories: traditional methods, graphical methods, and intelligent bionic methods [2].

1. Traditional methods mainly include Simulate Anneal Arithmetic (SAA), Artificial Potential Field (APF) and Fuzzy Logic (FL). These approaches, which were first used for path planning, are an evolution of direct human reasoning and have the advantages of being simple to grasp and express. They do not, however, fully utilize past expertise and general knowledge.

2. Graphic methods mainly include A-Star Algorithm (A*) and Grid Method. This type of methods provides rules on how to build a mathematical model, but the search is inefficient. Among all of these, the A* also has an extended place in the MAPF.

3. Intelligent bionic methods mainly include Genetic Algorithm (GA), Artificial Neural Network (ANN), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and so on. These methods are mostly inspired by and imitate the activity pattern of living creatures in nature. Due to their bionic characteristics, these methods are more intelligent but converge slowly and operate intricately.

Table 1. Comparison of conventional methods applied to Path Finding

Conventional methods	Represented methods	Advantages	Disadvantages
Traditional methods	SAA APF FL	Being easy to understand and simple to describe	Being easily trapped in a local optimum solution
Graphic methods	A* Grid Method	Providing modelling methods	Search is inefficient
Intelligent bionic methods	GA ANN ACO PSO	Possessing bionic characteristics and being intelligent	Converging slowly and operating intricately

2.2 Definition of Multi-Agent Path Finding

The concept of MAPF emerges when multiple agents are placed together in one path-finding environment. The classical MAPF problem can be defined as a quadruple $\langle G, N, S, T \rangle$. $G = \langle V, E \rangle$ is an undirected graph. The vertices of the graph $v \in V$ represents the location where the agents can stay, and the edges of the graph $e = (v_i, v_j) \in E$ represents the routines where the agents can move from one side to another side. N represents the number of agents, equaling to the length of $\{a_1, a_2, \dots, a_N\}$. Each agent has its own start and end position, and they are respectively stored in $S \in V$ and $T \in V$. In other words, S is the set of all agents' start positions and T is the set of all agents' end positions.

Time is divided into discrete units termed segmental time steps in the abstract model above. In each time step, generally there are only two types of actions for one agent: stays where it is or follows a certain course, which are called simply as Stay or Go. In summary, Spatial construction and temporal segmentation together form this classic MAPF problem.

2.3 Conflict between Agents

As described in the introduction, the primary issue that MAPF addresses over the single-agent PF problem is how to deal with conflicts between agents and coordinate their actions. To this general purpose, the concept of collision is introduced. There are four common types of collision [3], as shown in Figure 2.

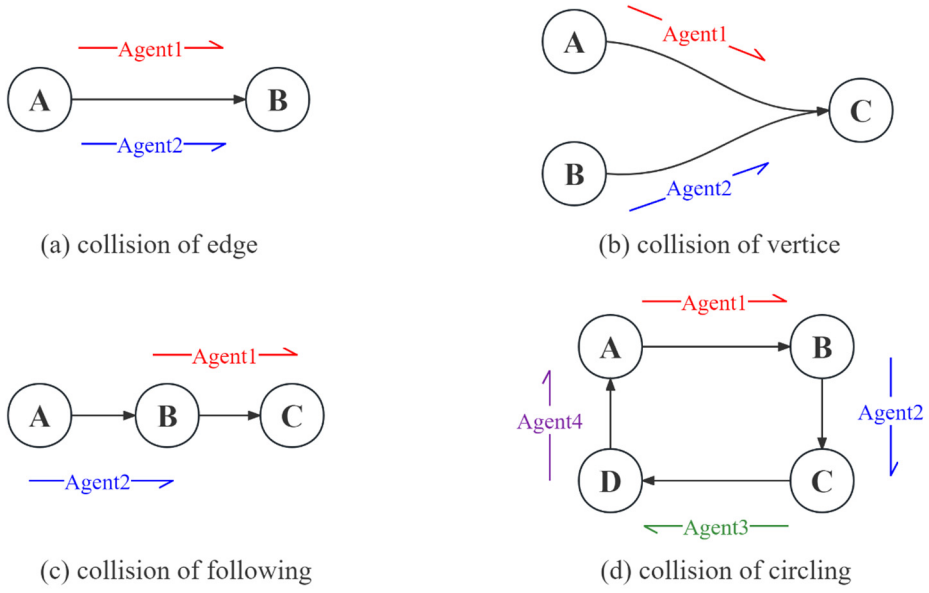

Fig 2. Classification of collisions

2.4 Objective Function for Evaluation

It is necessary to set an objective function for the MAPF problem that is worth optimizing. This is because in practical problems the aims for path finding are often to minimize economic and time costs, as well as conflict risks. In the classical MAPF methods, the three objective functions commonly used to evaluate the effectiveness of a policy π are shown below:

$$\max_{1\le i\le N}\left(t(\pi_i)\right) \tag{1}$$

$$\sum_{1\le i\le N}\left(t(\pi_i)\right) \tag{2}$$

$$\sum_{1\le i\le N}\left(l(\pi_i)\right) \tag{3}$$

The duration it takes for each agent to get at their destination is represented by objective function (1). The total amount of time it takes for all the agents to arrive at their destination locations is represented by objective function (2). The distance traveled by all agents to get to their destination locations is represented by objective function (3). All three of them portray the performance of a policy π in different dimensions.

3. Centralized Planning Algorithms

The centralized planning algorithm is an important type of algorithms for solving MAPF problems. It bases on the assumption that the central planner has all the global information about the environment in command, such as the location of obstacles, the starting and target positions of each agent, etc. Two subclasses of centralized planning algorithm will be described in details in this section: optimal centralized planning algorithm and suboptimal centralized planning algorithm.

3.1 Optimal Centralized Planning Algorithm

As shown in Table 2, optimal centralized planning algorithm can be divided into four categories: Extension of A* Search, Increasing Cost Tree Search, Conflict-Based Search, and Reduction Method. This type of algorithm needs to satisfy both optimality and integrity, in other words when a MAPF problem has its solutions, the algorithm must output the optimal solution.

Table 2. Comparison of optimal centralized planning algorithm

Algorithm	Advantages	Disadvantages	Applicable problem size
Extension of A* Search	An extension of one-dimensional algorithm, which is easy to understand and simple to describe	With slow solving speed and small application range	Small size (2~30)
Increasing Cost Tree Search	With fast solving speed, be suitable for environment of large-scale agents	Be Difficult to operate	Medium size (2~60)
Conflict-Based Search	Be simple and fast to implement	Be prone to failure in environment of high density	Medium size (2~60)
Reduction Method	With fast solving speed	Be difficult to prove its reduction	Medium size (2~60)

3.1.1 Extension of A* Search

As already mentioned in single-agent path finding, A* Search is a classical search algorithm that is also applicable to MAPF problem. The open list and the closed list are the two vertex lists that it maintains with. Initially the start node is placed in the open list. In each iteration, the open list is searched for neighboring nodes and the start node is put into the closed list. A* Search will calculate the following values and select the one in open list with the minimum $g(n) + h(n)$.

1) $g(n)$ is the shortest path's length from the root node to n node.
2) $h(n)$ is the heuristic path cost from n node to the target node in estimation.

3.1.2 Increasing Cost Tree Search

The high-level search and the low-level search comprise the two tiers of Increasing Cost Tree Search [4]. High-level search is to find the cost c_i of each agent a_i in the optimal solution of the MAPF problem. While the low-level search is to verify that for a given $(c_1, c_2, \cdots, c_N)$, there are several optimal policies $(\pi_1, \pi_2, \cdots, \pi_N)$. This algorithm can also be accelerated by specific pruning strategies. A common strategy is to select all pairs of agents $(a_i, a_j), i, j \in \{1,2, \cdots, N\}, i \neq j$. When there not exists a conflict-free combination of paths between a pair (a_i, a_j), the low-level search will stop the follow-up search immediately.

3.1.3 Conflict-Based Search

Conflict-Based Search (CBS) is one of the most mainstream and effective MAPF methods, which is a two-level algorithm. CBS starts by planning a shortest path for one agent, ignoring the others. Then it generates two subtrees and adds an additional constraint depending on whether one of the robots executes the current command or not. Afterwards, the routes are re-routed in the respective subtrees, with one route for agent A on the left and one route for agent B on the right. The previous process will be repeated until a node is found with a path without any collisions (Figure 3).

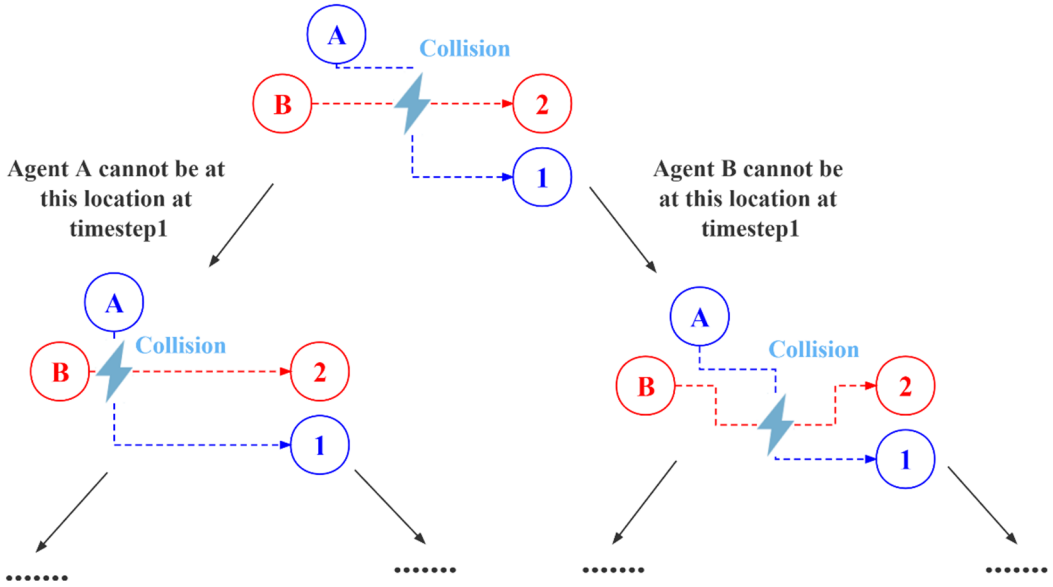

Fig 3. The Searching Tree of CBS

3.1.4 Reduction Method

The three classes of optimal centralized planning algorithm described above are all essentially search-based algorithms, while the Reduction Method is different from them. Reduction Method indicates that it is possible to reduce the MAPF problem into other standard problems such as Boolean Satisfiability Problem (SAT), Constraint Satisfaction Problem (CSP) and Travelling salesman problem (TSP). Once the correctness of the reduction has been proven, the existing efficient solvers for these problems can be used to deal with MAPF problem successfully.

The difficulty with the Reduction Method is the proof of the reduction, which generally requires strong mathematical skills. In general, these methods are faster to solve than the search-based algorithms described above. But they cannot guarantee the efficiency of the corresponding solvers after the proof of the reduction (Table 3).

Table 3. Main Categories of Reduction Method

Standard Problem	Introduction	Principle of reduction
SAT	Derived from the concept of classical propositional logic in mathematical logic regarding the satisfiability of formulas	Encode the structure of the map, the location of the agents and the constraints as bool variables [5]
CSP	A common optimization problem to solve the system of equations while satisfying the constraints	Abstract the map of the MAPF problem into known sub-maps [6]
TSP	A classical problem in graph theory that requires the path distance to be the smallest of all paths	Improve the IPL model and create a model indicator to evaluate the efficiency [7]

3.2 Suboptimal Centralized Planning Algorithm

Since the MAPF is a Non-deterministic Polynomial problem, algorithms for solving the optimal solution generally take a long execution time. In order to expedite the problem, it is necessary to sacrifice some optimality of the results. Suboptimal centralized planning algorithms are initiated in such demand.

These approaches have two drawbacks: only one agent is permitted to act at a time step, despite the integrity of the algorithms being ensured, and the quality of the outputs may be far below that of optimum algorithms [8]. The classification and comparison of suboptimal algorithms is shown in Table 4.

Table 4. Comparison of suboptimal centralized planning algorithm

Algorithm	Advantages	Disadvantages	Applicable problem size
Search Based Algorithms	Easy to implement, generally with good results	With slow solving speed and small application range	Small size (2~30)
Rule Based Algorithms	The result can be achieved quickly	The result may be much worse than the optimal solution. Completeness is only available on specific maps	Big size (2~120)
Reduction Based Algorithms	Take use of some existing solving algorithms	Be difficult to prove its reduction	Based on the standard problem

4. Distributed Execution Algorithms

Although classical centralized planning algorithms have been able to solve most of MAPF problems. However, they all have the crucial assumption that the central processor has all the information about the environment in command, but this assumption is difficult to reach in practical situations. With the development of technology, decentralized approaches are becoming more and more popular, where agents plan paths by communicating with other agents within a certain distance during their interaction with the environment, with better generalization, and can be extended to large-scale agents. This type of approach is known as distributed execution algorithms and its theory is based on and improved by RL.

4.1 Dynamic Programming

The dynamic programming algorithm was proposed by the American mathematician Bellman in his study of the optimization of multi-stage decision. The term "dynamic " means that the problem can be solved by a slew of smaller problems, and "programming" means an optimization strategy. The MAPF method based on dynamic planning is shown in Table 5 [9].

4.1.1 Bellman Equation

The theoretical basis for dynamic programming is based on the Bellman equation and the conclusions are given here directly.

$$\text{V-based:} \quad V^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} [R^a_{ss'} + \gamma V^{\pi}(s')] \tag{4}$$

$$\text{Q-based:} \quad Q^{\pi}(s,a) = \sum_a P^a_{ss'} [R^a_{ss'} + \gamma \sum_{a'} Q^{\pi}(s',a')] \tag{5}$$

Among them, $P^a_{ss'}$ is the transferring probability between pre and post states and $R^a_{ss'}$ is the reward for environmental feedback, while taking action a and transferring from s to s'.

4.1.2 Evaluation of the Methods

The use of dynamic programming on MAPF problems is divided into two main parts: Approximate Dynamic Programming (ADP) and Deep Dynamic Programming (DDP).

ADP, which use the generalization power of neural networks to value function approximation or strategy function approximation to get the reward function, thus not solving the Bellman equation directly, solving the "curse of dimensionality" problem.

DDP combines a learning neural heuristic with a deep strategic dynamic programming for solving the MAPF [10]. The model is based on a per-customer vertex feature vector gained from the Graph Neural Network (GNN) and finally constructs the optimal solution in the basis of this vector.

4.1.3 Limitations Analysis

Dynamic programming algorithms are widely used in MAPF. However, there are still numerous problems, such as curse of dimensionality, system agnosticism and low efficiency of real-time solution. Although approximate dynamic programming can effectively avoid these problems, they are also less robust due to the use of neural networks.

Table 5. Comparison of methods for MAPF based on dynamic programming

Algorithm	Results of optimization
ADP	Better optimization than the chosen centralized planning algorithm
ADP based on various rollout strategies [11]	A rollout strategy combined with dynamic decomposition can significantly reduce the solution time for large-scale MAPF problems
DDP [10]	On a MAPF with an agent scale of 100, the DDP approach outperforms all types of DRL-based algorithms
ADP based on look-up tables [12]	Better optimized than ADP based on rollout strategies

4.2 Research of Reinforcement Learning

A class of learning issues in the field of ML is known as RL, which is learned through interaction and feedback with the environment (Figure 4). Agents gradually develop their perception of the world through a sequence of activities, learning the characteristics of the environment so that their actions can accomplish their goals as rapidly as possible. A complete RL can be built as a Markov Decision Process (MDP) as (S, A, R, ρ, γ).

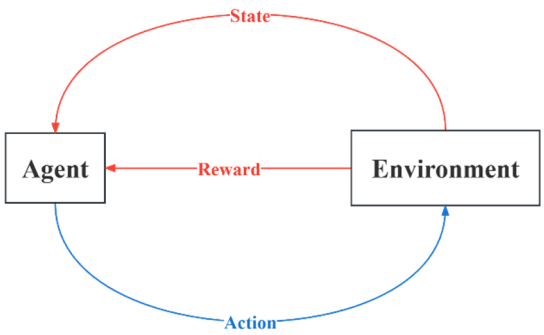

Fig 4. Fundamentals of Reinforcement Learning

4.2.1 Classification of RL

Depending on the type of output, RL can be divided into two categories and a special exception of fusion: value-based methods, policy-based methods, and Actor-Critic Method (AC). A policy-based

approach outputs the probability of the action to be taken next, and then the action is selected based on the probability (Figure 5). A value-based approach, on the other hand, outputs the value of taking various actions in the next step, and then selects the action based on the maximum value. For the AC method, the Actor will make an action based on probability and the Critic will give a value for the action made.

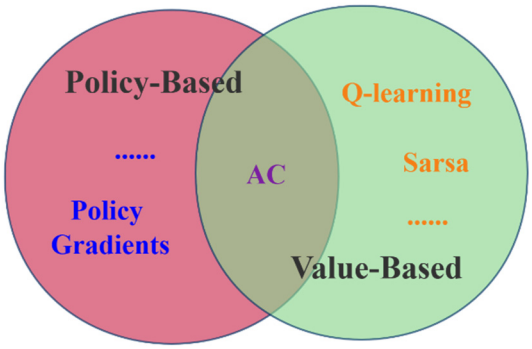

Fig 5. Classification of RL by the type of output

4.3 RL-based Algorithms for MAPF

There are many challenges in using RL methods to solve MAPF problems, such as sparse environmental rewards and complex environmental dynamics. Any reinforcement learning algorithm applied directly to the MAPF problem suffers from slow learning speed and poor learning quality. Fortunately, researchers have found out several solutions to the deficiencies upper, which can be broken down into three categories: Expert Demonstration Algorithms (EDA), Improved Communication Algorithms (ICA) and Task Decomposition Algorithms (TDA). Representative algorithms for each category are shown in Table 6.

4.3.1 Expert Demonstration Algorithms

The expert presentation approach uses a combination of reinforcement learning and Imitation Learning (IL). The most widely used is Pathfinding via Reinforcement and Imitation Multi-agent Learning (PRIMAL) [13], a new MAPF structure, whose algorithmic principle is shown in Figure 6.

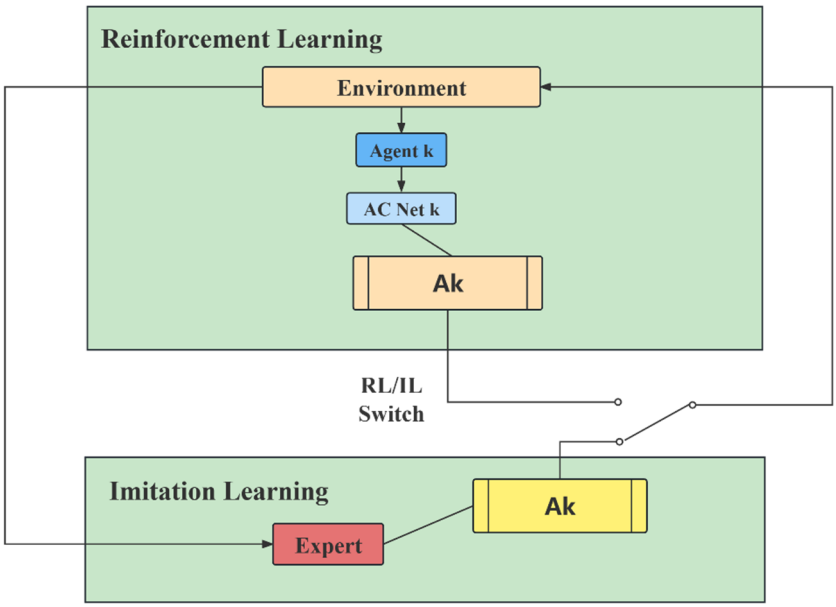

Fig 6. RL/IL framework structure diagram

4.3.2 Improved Communication Algorithms

In a MAPF of large-scale agents, in order to enhance coordination and cooperation, intelligences often communicate with each other. RL often uses broadcast communication, where information is disseminated to all other agents within a certain range. One of the best-known ones is the Learning Selective Communication (DCC) [14], it splits the training process into four modules: local observation of the input, observation of the encoding, selection of the judgement unit, and communication. This process is quite representative.

4.3.3 Task Decomposition Algorithms

MAPF is a highly difficult challenge in big dynamic settings because the intelligences must accomplish their objectives effectively while avoiding conflicts with other intelligences or dynamic items. An important approach to solving this challenge is to decompose the task and then use RL methods such as Deep Monte Carlo Tree Search, Q Mix Networks and Policy Gradient to calculate the actions of the agents [15].

Table 6. Comparison of methods for MAPF based on RL

Category	Represented methods	Features of each method
EDA	PRIMAL MAPPER [16] GLAS [17]	● Can be extend to 1,000-agents-scale environment ● Can achieve better results with less computational cost ● Achieve high success rate at all obstacle densities
ICA	PIOC DCC MAGAT [18]	● With higher success rate and faster learning ● With improved communication efficiency ● Close to the performance of centralized planning
TDA	HPL [19] G2RL [20] VRL [21]	● Significantly better than independent RL methods ● Provides excellent generalizability ● With superior scalability in large environments

5. Prospection

By contrasting the algorithms in different frameworks, this paper summarizes the future research directions in the following five areas, based on the challenges that MAPF has not yet solved:

1. In terms of distributed execution algorithms, although new algorithms are emerging all the time, there is still a lack of case studies to analyze which algorithms are more advantageous.

2. The conditions and areas of applicability of each algorithm can already be deduced in theoretical studies, so whether it is possible to fuse algorithms from different orientations and make full use of each other's strengths as the way to create hybrid algorithms with higher efficiency.

3. The MAPF problem has assumed that the actions of the agents are discrete, with only five actions: back and forth, left and right, and wait, and does not take into account the speed of the intelligences' movements, whereas most of the intelligences in the real-world MAPF problem have continuous actions. So how to upgrade the model to continuous actions still needs to be investigated.

6. Conclusion

This paper first introduces the elaborate concepts of MAPF, and then presents the MAPF centralized planning algorithm and the distributed execution algorithm. The centralized planning algorithms achieve high solution speed and quality in solving MAPF problems in static environments, while the RL-based distributed execution algorithms perform well in real-time replanning scenarios. The next research effort will focus on combining the advantages of both algorithms to create a more robust and efficient solution.

References

[1] Liu Qingzhou, WU Feng. Research progress of multi-agent path planning. Computer Engineering, 2020,46(4):1-10.

[2] Yan J J, Zhang Q S, Hu X P. Review of path planning techniques based on reinforcement learning. Computer Engineering, 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, Artificial Intelligence, 2013, 195: 470-495.

[5] Surynek P. Towards optimal cooperative path planning in hard setups through satisfiability solving, Proceedings of the Pacific Rim International Conferences on Artificial Intelligence. Washington D. C, USA: IEEE Press, 2012:564-576.

[6] Ryan M. Constraint-based multi-robot path planning, Proceedings of International Conference on Robotics and Automation. Washington D. C., 2010:38-54.

[7] Shu Xiangxiang. Research on optimal path planning of multi robots. China Computer and Communication, 2017 (15):62-64.

[8] Spiralris D K. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications, https://www.computer.org/csdl/proceedings-article/focs/1984/0715921/12OmNC3FGi5.

[9] Niu Pengfei, Wang Xiaofeng. Survey on Vehicle Reinforcement Learning in Routing Problem. College of Computer Science and Engineering, North Minzu University, 2022,58(01):41-55.

[10] Kool W, Van Hoof H, et al. Deep policy dynamic programming for vehicle routing problems. arXiv: 2102. 11756,2021.

[11] Goodson J C, Ohleman J W. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. Operations Research, 2013, 61 (1) :138-154.

[12] Zhang C, Delart N P, Zhao L, et al. Single vehicle routing with stochastic demands: approximate dynamic programming. Eindhoven: Technishe Universiteit Eindhoven, 2013.

[13] Sartoretti G, Kerr J, Shi Y, et al. Primal: path finding via reinforcement and imitation multi-agent learning. IEEE Robotics & Automation Letters, 2019, 4(3):2378-2385.

[14] Ma Z, Luo Y, Pan J. Learning selective communication for multi-agent path finding. arXiv: 2109. 05413, 2021.

[15] Skrynnik A, Yakovleva A, Davtdov V, et al. Hybrid policy learning for multi-agent pathfinding. IEEE Access, 2021, 9: 126034-126047.

[16] Liu Z, Chen B, Zhou H, et al. Mapper: multi-agent path planning with evolutionary reinforcement learning inmixed dynamic environments, 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020: 11748-11754.

[17] Rivre B, Hongig G W, Yue Y, et al. Glas: global-to-local safe autonomy synthesis for multi- robot motion planning with end-to-end learning. IEEE Robotics and Automation Letters, 2020,5(3):4249-4256.

[18] I Q, Lin W, Liu Z, et al. Message-aware graph attention networks for large-scale multi-robot path planning. IEEE Robotics and Automation Letters, 2021,6(3):5533-5540.

[19] Skrynnik A, Yakovlva et al. Hybrid policy learning for multi-agent pathfinding, IEEE Access, 2021, 9: 126034-126047.

[20] Ngb, Liu Z, Li Q, et al. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robotics and Automation Letters,2020,5(4):6932-6939.

[21] Iu Z, Liu Q, Tang L, et al. Visuomotor reinforcement learning for multirobot cooperative navigation. IEEE Transactions on Automation Science and Engineering, 2021:1-12.