Decentralized Multi-agent Path Finding based on Deep Reinforcement Learning

Heng Lu

Metropolitan College, Boston University, Boston, MA 02215, America
hengl@bu.edu

Abstract. Multi-agent path finding (MAPF) problem has long been a focus of reinforcement learning researchers due to is potential applications to real world robot deployment. Nowadays, many efforts have been made to develop decentralized MAPF algorithms since decentralized ways tend to scale better in larger robot team compared with centralized algorithms. However, reviews on this topic are still lacking. This paper reviews some state-of-the-art decentralized MAPF algorithms. These algorithms are classified into three categories, i.e., imitation learning (IL) algorithms, graph neuro networks (GNN) and task decomposition algorithms. IL-based algorithm learns from expert data, GNN-based algorithms learn by incorporating GNN, and task decomposition methods decompose MAPF into easier subtasks. For each algorithm, first its formulation of MAPF problem, i.e., the structure of observations, actions and rewards, is introduced. Then its essential part is analyzed in detail. Finally, its advantages and limitations are investigated and comparisons with other algorithms are made. In the end, the paper is summarized and provides outlook to the field.

Keywords: multi-agent path finding, deep reinforcement learning, robotics

1 Introduction

Multi-agent path finding problem is crucial for the real-world deployment of large-scale autonomous robots, whose aim is to find collision-free paths for agents to reach their goals as soon as possible. Researchers have developed numerous methods for solving the problem, and all these methods can be classified into two classes, centralized or decentralized. Centralized approaches rely on a powerful central unit to compute all positions of robots based on entire information of the environment. However, centralized approaches are extremely computationally expensive and are considered as NP-hard problem, which makes them inaccessible in large-scale scenarios. Thus, much effort has been devoted to develop decentralized MAPF algorithms, which are less computationally expensive and thus may perform well in large-scale real-world cases. Deep reinforcement learning is known for its ability to improve agent's performance through interacting with the environment. It seems reasonable for allowing agents to find colli-

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_19

sion-free paths by trial and error in MAPF problems, and thus deep reinforcement learning methods are widely adopted in decentralized MAPF algorithms.

There already exist many literatures for MAPF review. However, most of them are either outdated or only focus on specific questions. This paper tries to give a review on some most relevant decentralized MAPF algorithms and analyze their advantages and disadvantages. In main body (Section 2), several algorithms are classified into three classes and analyzed in detail. Section 2.1 focuses on imitation learning-based algorithms. Section 2.2 concentrates on GNN-based methods. In section 2.3, task decomposition methods are analyzed. Section 3 concludes this study.

2 Main Body

In this part, several important algorithms in MAPF are briefly explained and their advantages and disadvantages are also analyzed.

2.1 Imitation Learning

PRIMAL, i.e., Pathfinding via Reinforcement and Imitation Multi-Agent Learning [1] is a framework for decentralized MAPF. In this work, agents are trained to imitate demonstrations given by an expert centralized path planner Operator Decomposition-recursive-M* (ODrM*), while only given information within their field of view (FOV). The FOV is separated into four channels, with each channel representing obstacles, other agents' positions, neighboring agents' goals and own goal positions. These goal positions are contained in observations only if they are in the agents' current FOV. The actions are sampled only from valid actions, which are learned by using a loss function. For learning a selfless policy, this framework relies on three methods: blocking penalty, training-time demonstration, and switching training environments. Blocking penalty penalizes agents with a large negative reward if agents, which stop at their destinations, prevent or significantly delay other agents from reaching their goals. At the start of each episode, the agents randomly decide on whether to involve reinforcement learning or imitation learning, where the imitation learning is done by minimizing behavior cloning loss and the reinforcement learning is done by asynchronous advantage actor critic(A3C) [2]. The hybrid structure is shown in Fig. 1. This framework performs well in environments with low obstacle densities, but its performance drops when the obstacle density increases. Moreover, it doesn't work well when the agent density is high, since the agents are trained with a varying world size but a fixed team size.

One main problem of PRIMER is that agents stop at goal destinations after reaching them, which may not hold in real-world warehouse situations. The authors then propose PRIMER2, which is a decentralized framework for lifelong MAPF [3]. In this setting, new goals are assigned to those agents which have reached their goals. Compared with PRIMER, this work especially considers corridor obstacles, and thus two delta maps and one blocking map are introduced to handle corridor case properly. The authors find decentralized coordination learning is crucial for a selfless policy.

Instead of simply adding a blocking penalty, the agents perform convention learning, where agents learn a cooperative policy adhere to a set of conventions, such as agents should never navigate into a narrow corridor if another agent in the corridor is moving along the opposite direction. These conventions are learned through a valid loss function. For one-shot MAPF task, PRIMER2 lags behind centralized pathfinding algorithms in medium team size, but it scales well to large team sizes while centralized planners are unable to generate results due to computational complexity. However, the quality of solutions given by PRIMER2 is lower than that given by centralized planner like ODrM*, but it still outperforms the quality given by its predecessor, PRIMER.

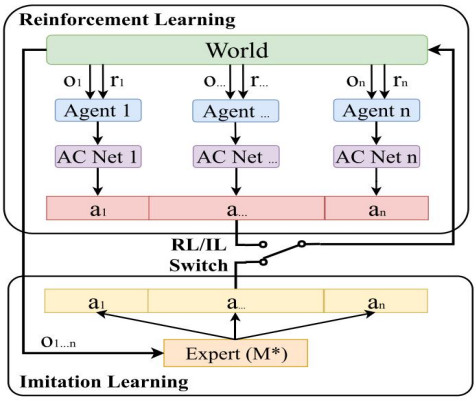

Fig. 1. Hybrid structure of PRIMAL [1]

Their later work extends PRIMER/PRIMER2 by incorporating a transformer based highly scalable communication mechanism, which is called scalable communication for reinforcement and IL-based MAPF(SCRIMP) [4]. In this work, the authors assume that communications won't be blocked by obstacles and agents have a one timestep communication delay. SCRIMP has three main parts: observation encoder which encodes observations into latent representation, communication block which is a transformer-encoder only network and takes messages from all agents sent at previous timestep as input to compute outputs, and output heads. This work also proposes value-based tie breaking to resolve inter-agent collisions and deadlocks. Moreover, each agent keeps an individual episodic buffer and generate intrinsic reward based on the maximum distance between its current position and stored positions to improve exploration. Compared with distributed heuristic learning with communication method (DHC) and prioritized communication learning (PICO), two state-of-art MAPF algorithms, SCRIMP has higher success rate and reaches more goals, and it even has similar performance with centralized ODrM*, although it doesn't have access to global information. The authors also find that when the communication range is restricted, SCRIMP performs even better in large team size, since some irrelevant information is ignored in this case.

2.2 Graph Neuro Network for MAPF

Graph Neuro Network approach for multi-robot path finding problem was introduced by [5]. The framework is shown in Fig. 2. In this paper, GNN is used as the communication network. The robots are represented as graph nodes, and existing edges indicates the connectivity between robots, where the edge weights are the strength of communication. The robots only have a local field of view of the environment, which are given to GNN for inter-robots communication after feature extraction. The robots are only allowed to communicate with neighboring robots within a predefined range. Robots select actions based on the outputs of GNN, and the action network is shared among all robots. In training stage, the agents are trained using imitation learning, where agents learn from expert experiences given by MAPF algorithms. In inference stage, since the trained network is not free from collision, the authors proposed collision shielding to further prevent collision, which means if the actions taken by two robots will result in a collision, then both robots will stop for that moment. This may cause the robots stuck in that state and finally fail to reach the goal. The authors also use online expert algorithm in training to overcome this issue. The most interesting result of this method is its generalization capability. While trained on small groups of robots, the method showed no noticeable performance drop when tested on larger groups of robots.

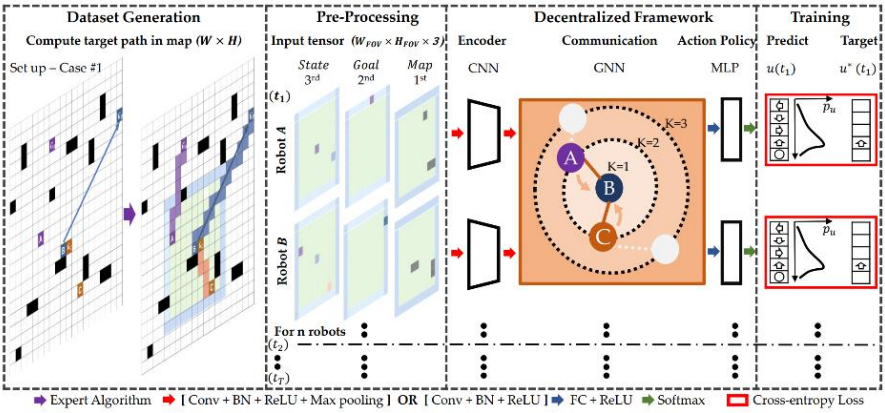

Fig. 2. Overview of GNN based MAPF framework [5]

Their later work, message-aware graph attention networks (MAGAT) has even greater capability to generalize to large scale robots path planning [6]. The main difference of MAGAT and simple GNN is the introduction of graph convolution and attention mechanisms, where attention is used to determine weights of edges based on relative importance of neighboring robots' features. This allows robots focus on the most important messages chosen by attention. By using graph convolution, MAGAT ensures permutation equivalence, which means that the trained MAGAT is resistant to the robot indices. Moreover, MAGAT is also time invariant, which suggests that the outputs given by MAGAT are consistent when the agents are encountered with the same situation in latter time steps. They also use ResNet blocks as the perception

module instead of simple CNN blocks in the previous work, and this proves to improve model performance when the shared messages are limited in size. The common limitation of the two works is the assumption that the communication of robots is done instantly without delay and thus these methods may not perform well in time-delayed case.

Jan Blumenkamp et al. tried a different way [7]. This work has two main parts, software part Robot Operating System 2 (ROS2) and communication network part, which is done by GNN. The ROS2 caches raw input data, including messages from neighbors and observations, for the policy. The ROS2 also performs a sim-to-real abstraction and the agents can't distinguish between real world and simulator environments, which makes it easier to deploy to the real world. The author demonstrates their work by a case study, where 5 robots are trained to navigate through a narrow passage without collision. The reward for the agents is designed to guide the robots to find efficient collision-free paths to their goals. Compared with [6], this work doesn't use expert trajectories when training, and this might be the reason for its relatively low success rate when running in a full decentralized mode. Moreover, in the case study, the author only shows the result for 5 robots, and thus the generalization ability remains a question. Nevertheless, its ability to deploy to the real world is promising.

2.3 Task Decomposition

Alexey Skrynnik proposed a hybrid policy learning (HPL) framework for MAPF [8]. The structure of HPL is shown in Fig. 3. In this paper, MAPF problem is decomposed into two subtasks: goal-reaching and collision-avoidance. The goal-reaching task is accomplished by curriculum learning which uses expert experiences to provide training curricula for agents. The collision-avoiding task can be done by either model-based Monte Carlo tree search method or model-free QMIX algorithm. The two tasks are learned in parallel in training phase. In inference phase, the paper combines policies learned in the two tasks by adding action probabilities given by these policies, which forms a hybrid policy which can resolve MAPF problem. Compared with standalone reinforcement learning methods, the hybrid policy learning method has significantly better performance. Moreover, this method also shows faster convergence compared with PRIMAL due to its better designed reward function, which gives positive reward even if the agent fails to reach the goal and penalizes the agent for distraction from an optimal path.

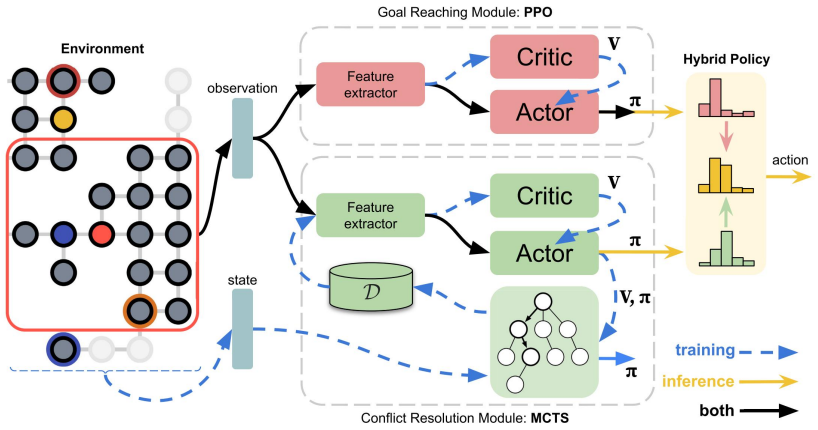

Fig. 3. Structure of HPL approach [8]

However, the HPL method shows poor scalability when the agent density is high, where dynamic obstacles become more significant. Globally guided reinforcement learning(G2RL) is a framework for addressing moving obstacles [9]. Different from other methods, G2RL assumes the agents know global information of static obstacles and a globally optimal path is calculated in the beginning of each run, which is also called the global guidance. In this stage, agents are guided by a novel reward function which gives dense rewards and encourages agents to follow the global guidance, but not as strictly as traditional imitation learning requires. In local RL stage, the agents choose actions based on current and history local observations using double deep Q network, where 3D CNN layers are used to extract spatial features in the local observations and LSTM layers are used to process temporal information. The success rate of G2RL agents outperforms that of PRIMAL agents in all testing scenarios, and this may arise from the introduction of global guidance which contains global information of all static obstacles and thus may overcome some situations where PRIMAL agents may get stuck. However, this requirement also restricts the availability of G2RL in some environments where the global information of static obstacles is inaccessible.

Many works in MAPF, e.g., G2RL and PRIMAL assume a bird-view of the environment and require the environment maps. However, the bird-view is not realistic in real world applications. In real world scenarios, robots hold a first-person view of the environment by using the information collected by their own sensory. Visuomotor Reinforcement Learning (VRL) is a MAPF framework which has good scalability and is free from expert demonstrations [10]. Different from G2RL, VRL uses GNN layers to allow communication between neighboring agents. VRL agents also receive a reward if they choose actions with the same direction indicated by optimal paths, which is calculated using A* in the discretized free space. The agents are first trained in simple environments to learn visual features of the destination, and then transferred into complex environments to improve agents' ability of avoiding collisions and obstacles. Compared with single agent visual pathfinding method, VRL

has much better performance in crowded environment. However, its scalability is much worse than MAGAT and PRIMAL, which are imitation learning-based method and adopt a bird-view of the environment. This may arise from the complexity of visual inputs compared with bird-view state inputs.

3 Conclusion

This paper analyzes three classes of decentralized MAPF algorithms. Imitation learning-based algorithms learn from experiences collected by some expert centralized planners. These algorithms tend to have great scalability in large teams although these agents are trained by using demonstrations in small teams. GNN based algorithms model individual robots as nodes and inter-robot communication as edges, and then use graph neuro networks to solve the problem, which are promising due to their agents' ability to exchange messages with neighboring agents and thus are able to better avoid collisions. However, these algorithms all show worse scalability in large teams despite of their high quality solutions in small teams. Task decomposition-based algorithms decompose the MAPF problems into subtasks, e.g., goal-reaching and task avoidance. However, the three classes are not mutually exclusive. The agents can learn from expert demonstrations while also use GNN to improve communications, which can further improve their collision avoidance capabilities and the algorithm's scalability.

However, these algorithms have some common limitations. Most algorithms investigated assumes a bird-view of the environment, which may not be generally available for real world environments. VRL agents hold a first-person view of the environment, but it has worse performance due to the complexity of visual objects. Future efforts can be made into developing high performance algorithms with a first-person view.

References

1. Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset. PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning. IEEE Robotics and Automation Letters, 4(3):2378–2385, Jul 2019.
2. Volodymyr Mnih, Adri`a Puigdom`enech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, Feb 2016
3. Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. PRIMAL2: Pathfinding via reinforcement and imitation multi-agent learning - lifelong. IEEE Robotics and Automation Letters, 6(2):2666–2673, April 2021
4. Yutong Wang, Bairan Xiang, Shinan Huang, and Guillaume Sartoretti. Scrimp: Scalable communication for reinforcement- and imitation-learning-based multi-agent pathfinding, Mar 2023
5. Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok. Graph neural networks for decentralized path planning. In Proceedings of the 19th International

Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20, page 1901–1903, Richland, SC, Dec 2020.
6. Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. Message-aware graph attention networks for large-scale multi-robot path planning, Nov 2021
7. Jan Blumenkamp, Steven Morad, Jennifer Gielis, Qingbiao Li, and Amanda Prorok. A framework for real-world multi-robot systems running decentralized gnn-based policies, Nov 2022
8. Alexey Skrynnik, Alexandra Yakovleva, Vasilii Davydov, Konstantin Yakovlev, and Aleksandr I. Panov. Hybrid policy learning for multi-agent pathfinding. IEEE Access, 9:126034–126047, 2021
9. Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. Mobile robot path planning in dynamic environments through globally guided reinforcement learning, May 2020
10. Zhe Liu, Qiming Liu, Ling Tang, Kefan Jin, Hongye Wang, Ming Liu, and Hesheng Wang. Visuomotor reinforcement learning for multirobot cooperative navigation. IEEE Transactions on Automation Science and Engineering, 19(4):3234–3245, 2022