SCRIMP: Scalable Communication for Reinforcement- and Imitation-Learning-Based Multi-Agent Pathfinding

Extended Abstract

Yutong Wang
National University of Singapore
Singapore, Singapore
e0576114@u.nus.edu

Bairan Xiang
National University of Singapore
Singapore, Singapore
e1011060@u.nus.edu

Shinan Huang
National University of Singapore
Singapore, Singapore
e1010775@u.nus.edu

Guillaume Sartoretti
National University of Singapore
Singapore, Singapore
mpegas@nus.edu.sg

ABSTRACT

In this paper, we propose SCRIMP, a multi-agent reinforcement learning approach for multi-agent path finding. Our method learns individual policies from very small FOVs (3x3), by relying on a highly-scalable global/local communication mechanism based on a modified transformer. We further introduce a state-value-based tie-breaking strategy to improve performance in symmetric situations and intrinsic rewards to encourage exploration while mitigating the long-term credit assignment problem. Empirical evaluations indicate that SCRIMP can outperform other state-of-the-art learning-based planners with larger FOVs and even yield similar performance as a classical centralized planner.

KEYWORDS

Multi-Agent Pathfinding; Multi-Agent Reinforcement Learning; Communication Learning

ACM Reference Format:
Yutong Wang, Bairan Xiang, Shinan Huang, and Guillaume Sartoretti. 2023. SCRIMP: Scalable Communication for Reinforcement- and Imitation-Learning-Based Multi-Agent Pathfinding: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Multi-Agent Path Finding (MAPF) is an NP-hard problem, aimed at generating collision-free paths for a set of agents from their initial position to their assigned goal on a given graph. Recently, Multi-Agent Reinforcement Learning (MARL) has been used to produce fast and scalable but suboptimal solutions to the MAPF by relying on partial observability. However, this approach severely limits the information available to agents, making cooperation difficult. Introducing a communication mechanism in MAPF allows agents to share information within a team, mitigating the risk of partial observability and facilitating teamwork. However, designing scalable communication mechanisms is challenging since, as the team size increases, agents are more likely to be overwhelmed by messages that essentially introduce noise (i.e., the *chatter problem*).

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

Build upon our previous works [2, 7], this work propose a decentralized MAPF planner SCRIMP, where agents make individual decisions based on their own field of view (FOV) and on highly-scalable global/local communications based on Transformer. We further propose a learning-based stochastic tie-breaking strategy and an intrinsic reward structure. Throughout this work, to increase speed and simplify implementation on robots with limited-range onboard sensing, we train agents with a small 3×3 FOV. Evaluation results show that SCRIMP can scale to larger teams without retraining and outperform other state-of-the-art learning-based planners with larger FOVs. SCRIMP also perform similarly to the classical centralized planner ODrM* [11], and even achieve a higher success rate in a complex task requiring joint maneuvers.

2 LEARNING TECHNIQUES

2.1 Transformer-based Communication Learning

Our communication mechanism is based on the transformer model [10], which is able to integrate information over long sequences, scale to large amounts of data, and capture dynamic pairwise relationships, thus helping mitigate the *chatter problem*. To stabilize the training of transformer, we removed dropout layers and modified its submodule structure, as recommended in [6].

In our method, the observation of each agent is first encoded by seven convolutional layers, two max pooling layers, three fully connected layers, and one Long Short-Term Memory (LSTM) unit. In parallel, messages from the previous timestep are processed by the Transformer-Based Communication Block, **where only the transformer encoder is used**. These messages are first augmented with the unique agent ID using a sinusoidal positional embedding, and passed through multiple iterative computation blocks. The multi-head self-attention in each computation block encodes not only the high-level information of messages but also their dynamic interrelationships. For each agent i, at each of the h independent attention heads, its own message is linearly projected to a Query by learnable weights w_q, while all messages are linearly projected to the Keys and Values by W_k, W_v. The relationship between the self-message of agent i and all messages is then computed as:

$$\alpha_i = \text{softmax}\left(\frac{\boldsymbol{w}_q \dot{m}_i^{t-1} \cdot \left(\boldsymbol{W}_k \left[\dot{m}_1^{t-1}, \ldots \dot{m}_n^{t-1}\right]\right)^\top}{\sqrt{d_K}}\right), \tag{1}$$

where $\dot{m}_i^{t-1}$ is the messages processed by the previous layers; $\sqrt{d_K}$ is the dimension of the Keys.

The final joint output of the communication block is concatenated with the output and input of the LSTM cell to generate a message for the next time step, predicted state values, the agent's policy, and a "blocking" output [7]. Since the communication block is placed before the output heads, agents' decisions are conditioned on their own observations and the messages exchanged within the team. In addition, because the encoder is trained using the gradient of all agents' losses, it learns to capture useful information and ignore noise/mitigate chatter to reduce the losses of all agents, leading to enhanced, larger-scale cooperation.

2.2 Value-Based Tie Breaking

We expect large gains in performance if agents can reach a meaningful consensus on the priority of competing moves before a vertex or swap conflict occurs, thereby effectively breaking symmetries. In this work, we propose to let agents check for conflicts before executing actions, and then (weighted-)randomly sample which agent is allowed to perform its preferred action, based on a learning-based *priority probability*. Other agents then re-sample a new action according to their distribution of actions, where their former choice is masked. The priority probability is constructed using the predicted difference in team state-value (sum of individual state values [9], representing long-term collective benefit) and the normalized Euclidean distance between the agent and its goal (considering "target symmetry" [3]). Their weighted sums are fed into a softmax layer to generate the final priority probability, and the agent allowed to move is sampled based on this probability.

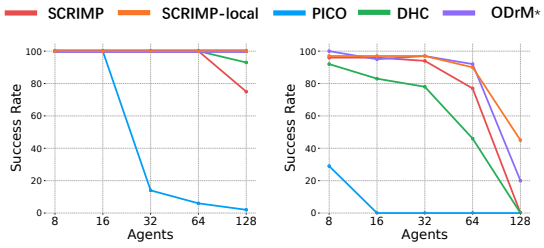
2.3 Episodic Buffer for Improved Exploration

Exploring the state-action space efficiently is a major challenge in RL [1], especially in MAPF tasks where agents need many low-reward actions before significant signals of finding and staying on goal (i.e., the *long-term credit assignment problem*). Encouraging exploration has been explored in various ways, including by introducing intrinsic rewards based on state familiarity. For MAPF in a gridworld, our proposed method generates intrinsic rewards based on visited grid cell coordinates (x, y) to encourage agents to explore more new area in each episode, thus increase probability of reaching goal and mitigate the long-term credit assignment problem.

In our method, each agent keeps an independent buffer with a limited capacity M, which starts each episode empty. At each step, each off-goal agent first calculates the Euclidean distance between its current location and all stored locations in the buffer. If the maximum distance exceeds threshold τ, the agent gives itself a bonus reward:

$$ri_i^t = \varphi(\beta - \delta), \quad \delta = \begin{cases} 1, \text{if max distance } < \tau \\ 0, \text{if max distance } \geq \tau, \end{cases} \tag{2}$$

where $\tau, \varphi \in \mathbb{R}^+$ and $\beta \in \{0, 1\}$ are pre-defined hyperparameters. The final reward is calculated as $r_i^t = ri_i^t + re_i^t$. After calculating intrinsic rewards, each agent determines whether its current maximum distance exceeds threshold ρ; if so, the current location is added to the episodic buffer [8]. When the buffer's capacity exceeds M, a random element in the buffer is replaced with the current element. The above process continues until the episode ends, at which point the buffer is emptied and a new episode begins.



(a) 0% static obstacle rate (b) 30% static obstacle rate

Figure 1: Success rate of different algorithms. The world sizes for 8, 16,32,64 **and** 128 **agents are** 10×10,20×20,30×30,40×40 **and** 40×40, **respectively. All results are calculated by running 100 episodes with randomly generated maps.**

3 COMPARISON WITH OTHER MAPF PLANNERS

We compare SCRIMP[1] with two other state-of-the-art RL methods, namely the GNN-based communication method DHC with 9×9 FOV [5], and the ad-hoc routing communication method PICO [4] with 11×11 FOV. We also show the results of the bounded-optimal centralized planner ODrM* as a reference. To further investigate the practicality of SCRIMP in scenarios where global communication may not be available, we extend SCRIMP with global communication into SCRIMP-local, a local communication model with a restricted communication range of 5. The experimental setup and the hyperparameters of our models are available in the Supplementary Material[2].

Testing results in Figure 1 show that the Success Rate(SR) of our models outperforms that of DHC and PICO in almost all cases. SCRIMP-local is able to yield similar performance as ODrM*. In particular, it still achieves 45% SR in the most difficult task requiring a high degree of agent cooperation (128 agents, 30% obstacle density), while the SR of ODrM* is only 20%, and DHC and PICO simply cannot solve the task. Moreover, local communication outperforms global communication as the number of agents increases due to the enhanced noise filtering mechanism.

4 CONCLUSION

This work introduces a new RL approach SCRIMP to MAPF that relies on a highly scalable local/global communication mechanism based on a modified transformer. We further propose to break ties between agents using a stochastic priority-based system and employ intrinsic rewards to encourage single-episode exploration while mitigating the long-term credit assignment problem. Experimental results show that SCRIMP with local/global communication outperforms other learning-based planners, while maintaining scalability to larger teams. In most cases, our method performs similarly to a classical centralized planner, and achieve higher success rate in the complex task.

[1]The full code is available at https://github.com/marmotlab/SCRIMP
[2]Available at https://drive.google.com/file/d/1DWJb4flLXljCmFefxG6ozVjTMyiqtDWy

ACKNOWLEDGMENTS

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1.

REFERENCES

[1] Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. 2021. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157* (2021).

[2] Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. 2021. PRIMAL _2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. *IEEE Robotics and Automation Letters* 6, 2 (2021), 2666–2673.

[3] Jiaoyang Li, Graeme Gange, Daniel Harabor, Peter J Stuckey, Hang Ma, and Sven Koenig. 2020. New techniques for pairwise symmetry breaking in multi-agent path finding. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 30. 193–201.

[4] Wenhao Li, Hongjun Chen, Bo Jin, Wenzhe Tan, Hongyuan Zha, and Xiangfeng Wang. 2022. Multi-Agent Path Finding with Prioritized Communication Learning. *arXiv preprint arXiv:2202.03634* (2022).

[5] Ziyuan Ma, Yudong Luo, and Hang Ma. 2021. Distributed heuristic multi-agent path finding with communication. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 8699–8705.

[6] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. 2020. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*. PMLR, 7487–7498.

[7] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, TK Satish Kumar, Sven Koenig, and Howie Choset. 2019. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters* 4, 3 (2019), 2378–2385.

[8] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. 2018. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274* (2018).

[9] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296* (2017).

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[11] Glenn Wagner and Howie Choset. 2015. Subdimensional expansion for multirobot path planning. *Artificial intelligence* 219 (2015), 1–24.