A Simulation Based Online Planning Algorithm for Multi-Agent Cooperative Environments

Extended Abstract

Rafid Ameer Mahmud
University of Dhaka
rafidameer-2016814404@cs.du.ac.bd

Fahim Faisal
University of Dhaka
fahim-2016714432@cs.du.ac.bd

Saaduddin Mahmud
University of Massachusetts Amherst
smahmud@umass.edu

Md. Mosaddek Khan
University of Dhaka
mosaddek@du.ac.bd

ABSTRACT

Multi-agent Markov Decision Process (MMDP) has been an effective way of modelling sequential decision making algorithms for multi-agent cooperative environments. However, challenges such as exponential size of action space and dynamic changes limit the efficacy of proposed solutions. This paper propose a scalable and robust algorithm that can effectively solve MMDPs in real time. Simulation, pruning, and prediction are the three key components of the algorithm. The simulation component enables real time solutions by using a novel iterative pruning technique which in turn makes use of the prediction component trained with self play data. The algorithm is self-sustained as it generates new training data from simulation and gradually becomes better. Furthermore, we show empirical results demonstrating the capabilities of the algorithm and compare them with existing MMDP solvers.

KEYWORDS

Cooperation Learning; Multi-agent Markov Decision Process; Transfer Learning; Graph Convolutional Network

ACM Reference Format:
Rafid Ameer Mahmud, Fahim Faisal, Saaduddin Mahmud, and Md. Mosaddek Khan. 2022. A Simulation Based Online Planning Algorithm for Multi-Agent Cooperative Environments: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Sequential decision making models for multi-agent environments hold the key to many real life problems such as autonomous vehicles [10], controlling robots [6], resource allocation [12], games with multiple types of agents such as Starcraft II [8], multi-agent path finding [9] and so on. Cooperation among agents and the curse of dimensionality are the two fundamental challenges of this domain. As the number of agents increases, the joint action space and number of states rise exponentially, rendering single agent planning algorithms impractical in these cases. In multi-agent systems, the sequential decision making problem can be modeled as a variant of Markov Decision Process (MDP) [11], called Multi-agent Markov Decision Process (MMDP) [2].

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

Over the years, a number of centralized and decentralized planning algorithms have been proposed to solve MMDPs. Best et al. [1] proposed a decentralized online planning algorithm to solve the dimensionality problem by using parallel MCTS trees and periodic communication. On the other hand, Aleksander et. al. [4] introduced an algorithm ABC, where agents do not use communication, rather they train their individual agent models one at a time and try to induce cooperation by predicting agent policies using the learned agent models. Guestrin et al. [5] showed that communication among agents can be represented as a coordination graph (CG) and joint value functions can be estimated from the higher order value factorization. Using this, Choudhury et al. [3] proposed simulation based *anytime online* planning algorithm FV-MCTS-MP, where factored value function is used with CG to incorporate communication among agents. The proposed solutions solve both the cooperation and dimensionality problem but fail to adapt to dynamically changing environments.

we introduce a simulation based anytime planning algorithm, that we call SiCLOP, for multi-agent cooperative environments. Specifically, SiCLOP tailors Monte Carlo Tree Search (MCTS) and uses Coordination Graph (CG) and Graph Neural Network (GCN) to learn cooperation and provides real time solution of a MMDP problem. It also improves scalability through an effective pruning of action space using Gibbs sampling. Additionally, unlike existing approaches, SiCLOP supports transfer learning, which enables learned agents to operate in different environments. We also provide theoretical discussion about the convergence property of our algorithm within the context of multi-agent settings. Finally, our extensive empirical results show that SiCLOP significantly outperforms the state-of-the-art online planning algorithms. [1]

2 SICLOP: MULTI-AGENT ONLINE PLANNING

Our algorithm ***Si**mulation based **C**ooperation **L**earner and **O**nline **P**lanner* (SiCLOP) is an MMDP solver with three components:

Simulation, SiCLOP-S. We tailor Monte Carlo Tree Search (MCTS) to present a simulation process that takes an MMDP problem as input. There are four steps in the simulation process. They are repeated multiple times within a fixed amount of time to solve the problem. In the first step, a leaf is reached from the *root* by sequentially calculating the scores of the children of a node with Eq. 1. The child with the highest score is then selected.

[1] See [7] for detailed discussion about the algorithm, proofs and experiments.

$$score_a = Q(s,a) + c * \sqrt{(\log N_i)/n_i} \quad (1)$$

Here, $Q(s,a)$ is the action value, c is the exploration constant, N_i and n_i are visit counts of current node and action. After choosing a leaf node, its child nodes need to be created. The number of possible child nodes is exponential in magnitude with respect to the number of agents. Therefore, in the second step, a small subset of joint actions are sampled from the joint action space with SiCLOP-P. The child nodes are then created and their *state values* are predicted with SiCLOP-NN. The *state values* are then backpropagated to their ancestors to complete the next three steps. At the end of generating the simulation tree, the most visited joint action in the root node is selected and returned.

Pruning, SiCLOP-P. We propose a novel joint action sampling method inspired by *Gibbs sampling* to effectively search over the joint policy space. It iterates over the agents to predict their best response policy BR_i using SiCLOP-NN by assuming other agent policies π_{-i} to be static. This process repeats for multiple cycles and at the end of each cycle, a joint action is sampled. At cycle k, the policy of agent i is updated with the conditional best response policy,

$$\pi_i^k = \Psi(\pi_i | \pi_1^k, ..., \pi_{i-1}^k, \pi_{i+1}^{k-1}, ..., \pi_n^{k-1}) \quad (2)$$

Here, the updated policy π_i^k is generated from the current static policies π_{-i}. After a fixed number of cycles, the set of sampled joint actions is returned.

Lemma 2.1. *In any iteration of sampling for agent i in group $\mathcal{N}$, the new sampled policy $\pi_i^* \in BR_i(\pi_{-i})$ will lead to non-decreasing reward and iterating through all the agents of $\mathcal{N}$ will lead to an equilibrium for that group.*

Proof. If the current joint policy of all the agents is π and the updated policy for agent i is π_i^*, then for all $s \in S$, the state value,

$$\forall i \in \mathcal{N} \quad V_i^{(\pi_i^*, \pi_{-i})}(s) \geq V_i^{(\pi_i, \pi_{-i})}(s) \quad (3)$$

If all the agents in the group acquire non-decreasing rewards, the combined reward will also be non-decreasing. As both the joint action space and reward are finite, iterating through the policy space will converge to a joint policy equilibrium for the group $\mathcal{N}$. In non-stationary situations, an equilibrium may not exist, and the method will converge to a set of equilibria in that case. □

Prediction, SiCLOP-NN. The neural network SiCLOP-NN is a GCN based policy predictor and *state value* estimator with two parts, feature extraction and prediction. The first part is used for graph embedding with multiple GCN layers. The second part takes the extracted feature vectors for each agent and generates the best response policies as boltzmann probability distribution and *state value* estimations. The *state values* are aggregated to construct a combined estimated *state value*. SiCLOP-NN takes the local information of the agents and a Coordination Graph (CG) as input. The definition of the local information can be defined according to the environment, for example, the cells within a limited range of an agent. The CG can also be constructed according to predefined rules. The neural network is trained on the simulation data collected from recent MMDP problem solutions as it better represents the currently learned policy generation. SiCLOP-NN trains and predicts on states consisting of a variable number of agents, with their internal interactions passed as CG into the GCN block. Thus, SiCLOP-NN's architecture allows making predictions in environments of any size and shape, allowing it to be transferred to other environments of varying setups.

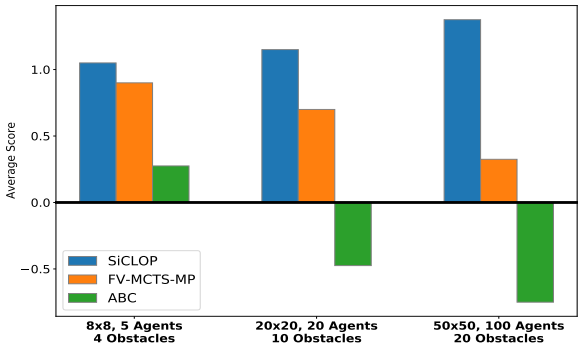

Figure 1: Performance comparison

3 EMPIRICAL RESULTS AND CONCLUSION

We have evaluated SiCLOP's performance in a grid world where the agents need to reach specific cells by cooperating and avoiding penalties like colliding with each other and obstacles. We tested the transfer learning capability by training the algorithm in a 10*10 grid with 4 agents and 15 obstacles, then deploying it in different environments, as shown in Table 1. The average score around 1.0 means the agents were able to complete their tasks successfully.

Environment List			
Grid Size	Agents	Obstacles	Avg. Score
20*20	20	24	1.06
50*50	100	200	1.15
100*100	150	250	1.32

Table 1: Transferring and testing in new environments

To compare with other planning algorithms, we tested ABC and FV-MCTS-MP on the same environment and MMDP problems, as shown in Figure 1. The negative average score denotes succumbing to the penalties. As we can see, SiCLOP outperforms other MMDP solvers as the environment becomes larger.

4 CONCLUSIONS

This paper introduces a scalable and adaptive algorithm to solve MMDP. Our algorithm SiCLOP uses local information of the agents to construct state dependant dynamic CGs and uses it to find optimum policies through effective pruning. It has been shown theoretically that our novel sampling process can find optimal policies for a group of agents. Combined with simulation and learning, SiCLOP is then able to recursively find better policies. We have shown that our algorithm can adapt to larger, dynamic, more realistic environments and outperform existing online MMDP solvers.

ACKNOWLEDGMENTS

This research is supported by the ICT Innovation Fund (2020–2021) of Bangladesh Government.

REFERENCES

[1] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. 2019. Dec-MCTS: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research* 38, 2-3 (2019), 316–337.

[2] Craig Boutilier. 1996. Planning, learning and coordination in multiagent decision processes. In *TARK*, Vol. 96. Citeseer, 195–210.

[3] Shushman Choudhury, Jayesh K Gupta, Peter Morales, and Mykel J Kochenderfer. 2021. Scalable Anytime Planning for Multi-Agent MDPs. *arXiv preprint arXiv:2101.04788* (2021).

[4] Aleksander Czechowski and Frans Oliehoek. 2020. Decentralized MCTS via Learned Teammate Models. *arXiv preprint arXiv:2003.08727* (2020).

[5] Carlos Guestrin, Daphne Koller, and Ronald Parr. 2001. Multiagent Planning with Factored MDPs.. In *NIPS*, Vol. 1. 1523–1530.

[6] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

[7] Rafid Ameer Mahmud, Fahim Faisal, Saaduddin Mahmud, Md Khan, et al. 2021. Learning Cooperation and Online Planning Through Simulation and Graph Convolutional Network. *arXiv preprint arXiv:2110.08480* (2021).

[8] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4295–4304.

[9] Oren Salzman and Ron Zvi Stern. 2020. Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems. In *19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2020*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 1711–1715.

[10] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2016. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv preprint arXiv:1610.03295* (2016).

[11] Richard S Sutton and Andrew G Barto. 2011. Reinforcement learning: An introduction.

[12] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 5571–5580.