Distributed Policy Iteration for Scalable Approximation of Cooperative Multi-Agent Policies

Extended Abstract

Thomy Phan
LMU Munich
thomy.phan@ifi.lmu.de

Kyrill Schmid
LMU Munich
kyrill.schmid@ifi.lmu.de

Lenz Belzner
MaibornWolff
lenz.belzner@maibornwolff.de

Thomas Gabor
LMU Munich
thomas.gabor@ifi.lmu.de

Sebastian Feld
LMU Munich
sebastian.feld@ifi.lmu.de

Claudia Linnhoff-Popien
LMU Munich
linnhoff@ifi.lmu.de

ABSTRACT

We propose *Strong Emergent Policy (STEP) approximation*, a scalable approach to learn strong decentralized policies for cooperative MAS with a distributed variant of policy iteration. For that, we use function approximation to learn from action recommendations of a decentralized multi-agent planning algorithm. STEP combines decentralized multi-agent planning with centralized learning, only requiring a generative model for distributed black box optimization. We experimentally evaluate STEP in two challenging and stochastic domains with large state and joint action spaces and show that STEP is able to learn stronger policies than standard multi-agent reinforcement learning algorithms, when combining multi-agent open-loop planning with centralized function approximation. The learned policies can be reintegrated into the multi-agent planning process to further improve performance.

KEYWORDS

multi-agent planning; multi-agent learning; policy iteration

ACM Reference Format:
Thomy Phan, Kyrill Schmid, Lenz Belzner, Thomas Gabor, Sebastian Feld, and Claudia Linnhoff-Popien. 2019. Distributed Policy Iteration for Scalable Approximation of Cooperative Multi-Agent Policies. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

1 INTRODUCTION

Cooperative multi-agent systems (MAS) are popular in artificial intelligence research and have many potential real-world applications like autonomous vehicles, sensor networks, and robot teams [4–6]. However, decision making in MAS is extremely challenging due to intractable state and joint action spaces as well as stochastic dynamics and uncertainty w.r.t. other agents' behavior.

Centralized control does not scale well in large MAS due to the *curse of dimensionality*, where state and joint action spaces grow exponentially with the number of agents [1, 3–7]. Therefore, decentralized control is recommended, where each agent decides its individual actions under consideration of other agents, providing better scalability and robustness [4–7]. Decentralized approaches to decision making in MAS typically require a *coordination mechanism* to solve joint tasks and to avoid conflicts [3].

Recent approaches to learn strong policies are based on *policy iteration* and combine planning with deep reinforcement learning, where a neural network is used to imitate the action recommendations of a tree search algorithm. In return, the neural network provides an action selection prior for the tree search [2, 13]. This iterative procedure, called *Expert Iteration (ExIt)*, gradually improves both the performance of the tree search and the neural network [2]. ExIt has been successfully applied to zero-sum games, where a single agent improves itself by self-play. However, ExIt cannot be directly applied to large cooperative MAS, since using a centralized tree search is practically infeasible for such problems [4, 5].

In this work, we propose *Strong Emergent Policy (STEP) approximation*, a scalable approach to learn strong decentralized policies for cooperative MAS with a distributed variant of policy iteration. For that, we use function approximation to learn from action recommendations of a decentralized multi-agent planner. STEP combines decentralized multi-agent planning with centralized learning, where each agent is able to explicitly reason about emergent dependencies to make coordinated decisions. Our approach only requires a generative model for distributed black box optimization.

2 METHOD

Given a *Multi-agent Markov Decision Process (MMDP)* $M = \langle \mathcal{D}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ [3] with a (finite) set of agents $\mathcal{D} = \{1, ..., N\}$, a (finite) set of states $\mathcal{S}$, a (finite) set of joint actions $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_N$, a transition probability function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \leftarrow [0, 1]$, and a global reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \leftarrow \mathbb{R}$, we extend the ExIt framework of [2, 13] to cooperative MAS to approximate a strong *joint policy* $\hat{\pi}(s_t) = \langle \hat{\pi}_1(s_t), ..., \hat{\pi}_N(s_t) \rangle \in \mathcal{A}$ for each state $s_t \in \mathcal{S}$. For that, we use function approximation to learn from action recommendations of a decentralized multi-agent planner to approximate strong *decentralized policies* $\hat{\pi}_i$ for each agent $i \in \mathcal{D}$, which are combined into a strong joint policy $\hat{\pi}$ for the MAS. The training procedure of STEP consists of a *planning* and a *learning* step.

In the planning step, a decentralized planning algorithm is executed for a state $s_t \in \mathcal{S}$ to recommend an action $a_{t,i}$ for each agent $i \in \mathcal{D}$ according to the relative action frequencies $p(a_{t,i}|s_t) \in [0, 1]$ calculated during planning. The individual actions are combined into a joint action $a_t = \langle a_{t,1}, ..., a_{t,N} \rangle$ and executed to observe a new state $s_{t+1} \in \mathcal{S}$ and a global reward $r_t = \mathcal{R}(s_t, a_t)$.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

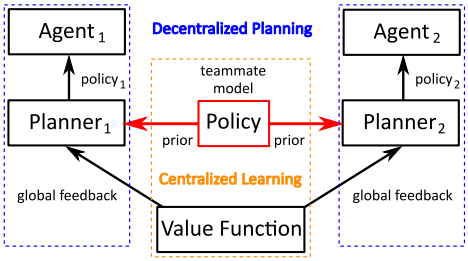
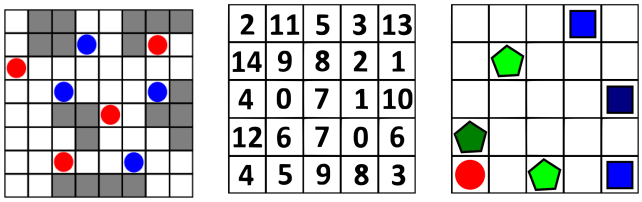


Figure 1: Architecture of STEP. The policy (red box) can be used as a prior for action selection and to predict other agents' behavior for coordination. The value function is used to evaluate leaf states for multi-agent planning [11].



(a) Pursuit & Evasion **(b) Smart Factory (SF)** **(c) An agent in SF [11]**

Figure 2: (a) Pursuit & Evasion ($N = 4$) with pursuers (red circles) and evaders (blue circles). (b) Machine grid of the Smart Factory (SF) with the numbers denoting the machine type. (c) An agent i (red circle) with $tasks_i = [\{9, 12\}, \{3, 10\}]$ in the SF of Fig. 2b. It should get to the green pentagonal machines first before going to the blue rectangular machines [11].

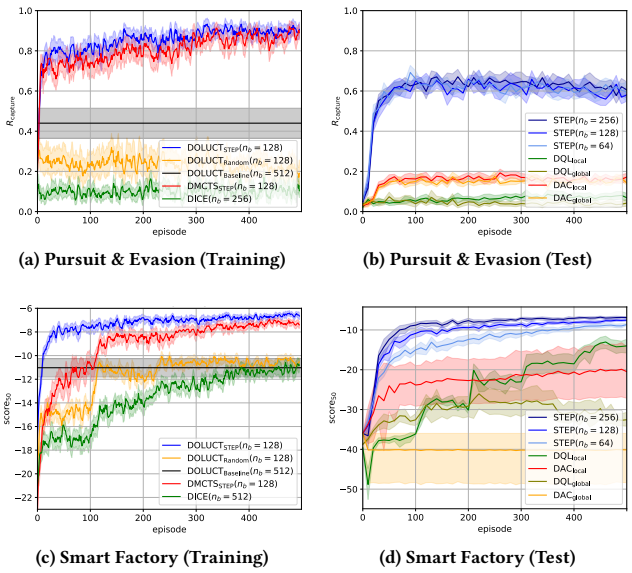
In the learning step, a parametrized function approximator $f_\theta = \langle \hat{\pi}, \hat{V} \rangle$ is used to approximate an optimal joint policy π^* by approximating optimal decentralized policies π_i^* for each agent $i \in \mathcal{D}$ and the optimal value function V^*. $\hat{\pi}_i$ is approximated by minimizing the cross-entropy loss between $p(a_{t,i}|s_t)$ and $\hat{\pi}_i(a_{t,i}|s_t)$, while $\hat{V}$ is approximated via temporal difference learning [14, 15].

f_θ can be reintegrated into the planning step to further improve performance by providing an action selection prior $\hat{\pi}_i$ similarly to [2, 13], a coordination mechanism to predict other agents' behavior via $\hat{\pi}$ [4], and a leaf state evaluator $\hat{V}$ to compensate for the limited search depth of the decentralized multi-agent planner [11]. The architecture of STEP is shown in Fig. 1.

3 RESULTS

We tested STEP in the *Pursuit & Evasion* domain (Fig. 2a and [16, 17]) with 2 agents and in the *Smart Factory* domain (Fig. 2b-c and [11]) with 4 agents. In the training phase, we applied STEP to both decentralized open-loop (DOLUCT) and closed-loop (DMCTS) planning, and compared the progress with different instances of DOLUCT using a random joint policy $\hat{\pi}$ or a baseline value function of $\hat{V}(s_t) = 0$ as well as a centralized open-loop version of DICE [9, 11]. In the test phase, we extracted the decentralized policies $\hat{\pi}_i$ approximated with STEP after every tenth training episode and compared them with *Distributed Q-Learning (DQL)* [16] and *Distributed Actor-Critic (DAC)* [6]. We implemented two variants of each DQL and DAC, where one variant was trained on the global reward $\mathcal{R}$ and the other one was trained on a decomposed local reward similarly to [7].

The results are shown in Fig. 3. Fig. 3a and 3c indicate that open-loop planning algorithms like DOLUCT are especially suited for STEP, when the domains are too complex to provide sufficient computation budget as already noted for single-agent problems [8, 10, 12, 18]. The approximated policies $\hat{\pi}_i$ of STEP with DOLUCT are able to clearly outperform standard multi-agent reinforcement learning algorithms like DQL and DAC in both domains. Providing a larger computation budget n_b seems to be beneficial when approximating strong decentralized policies with STEP as shown in Fig. 3b and 3d. The learned policies can be reintegrated into the planning process to further improve performance of the multi-agent planner as shown in Fig. 3a and 3c for DOLUCT and DMCTS.



(a) Pursuit & Evasion (Training) **(b) Pursuit & Evasion (Test)**

(c) Smart Factory (Training) **(d) Smart Factory (Test)**

Figure 3: Average training and test progress of $R_{capture}$ (Pursuit & Evasion) and $score_{50}$ (Smart Factory) different multi-agent planning and learning algorithms of 30 runs. Shaded areas show the 95 % confidence interval.

4 CONCLUSION

We proposed STEP, a scalable approach to learn strong decentralized policies for cooperative MAS with a distributed variant of policy iteration by combining decentralized multi-agent planning with centralized learning, where each agent is able to explicitly reason about emergent dependencies to make coordinated decisions, only requiring a generative model for distributed black box optimization. Our results show that STEP is able to produce stronger policies than standard multi-agent reinforcement algorithms, which can be reintegrated into the planning process to further improve performance. For the future, we plan to address partially observable domains by combining multi-agent planning with deep recurrent reinforcement learning for cooperative MAS.

REFERENCES

[1] Christopher Amato and Frans A Oliehoek. 2015. Scalable Planning and Learning for Multiagent POMDPs. In *29th AAAI Conference on Artificial Intelligence*.

[2] Thomas Anthony, Zheng Tian, and David Barber. 2017. Thinking Fast and Slow with Deep Learning and Tree Search. In *Advances in Neural Information Processing Systems*.

[3] Craig Boutilier. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*. Morgan Kaufmann Publishers Inc.

[4] Daniel Claes, Frans Oliehoek, Hendrik Baier, and Karl Tuyls. 2017. Decentralised Online Planning for Multi-Robot Warehouse Commissioning. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS.

[5] Daniel Claes, Philipp Robbel, Frans A Oliehoek, Karl Tuyls, Daniel Hennes, and Wiebe Van der Hoek. 2015. Effective Approximations for Multi-Robot Coordination in Spatially Distributed Tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS.

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. *32th AAAI Conference on Artificial Intelligence* (2018).

[7] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative Multi-Agent Control using Deep Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems*. Springer.

[8] Erwan Lecarpentier, Guillaume Infantes, Charles Lesire, and Emmanuel Rachelson. 2018. Open Loop Execution of Tree-Search Algorithms. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI Organization, 2362–2368. https://doi.org/10.24963/ijcai.2018/327

[9] Frans A Oliehoek, Julian FP Kooij, and Nikos Vlassis. 2008. The Cross-Entropy Method for Policy Search in Decentralized POMDPs. *Informatica* 32, 4 (2008), 341–357.

[10] Diego Perez Liebana, Jens Dieskau, Martin Hunermund, Sanaz Mostaghim, and Simon Lucas. 2015. Open Loop Search for General Video Game Playing. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM.

[11] Thomy Phan, Lenz Belzner, Thomas Gabor, and Kyrill Schmid. 2018. Leveraging Statistical Multi-Agent Online Planning with Emergent Value Function Approximation. In *Proceedings of the 17th Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS.

[12] Thomy Phan, Lenz Belzner, Marie Kiermeier, Markus Friedrich, Kyrill Schmid, and Claudia Linnhoff-Popien. 2018. Memory Bounded Open-Loop Planning in Large POMDPs Using Thompson Sampling. *33th AAAI Conference on Artificial Intelligence* (2018).

[13] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the Game of Go without Human Knowledge. *Nature* 550, 7676 (2017).

[14] Richard S Sutton. 1988. Learning to Predict by the Methods of Temporal Differences. *Machine learning* 3, 1 (1988).

[15] Richard S Sutton and Andrew G Barto. 1998. *Introduction to Reinforcement Learning*. Vol. 135. MIT Press Cambridge.

[16] Ming Tan. 1993. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *Proceedings of the 10th International Conference on International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc.

[17] Rene Vidal, Omid Shakernia, H Jin Kim, David Hyunchul Shim, and Shankar Sastry. 2002. Probabilistic Pursuit-Evasion Games: Theory, Implementation, and Experimental Evaluation. *IEEE transactions on robotics and automation* 18, 5 (2002).

[18] Ari Weinstein and Michael L Littman. 2013. Open-Loop Planning in Large-Scale Stochastic Domains. In *27th AAAI Conference on Artificial Intelligence*.