Cooperative Multi-agent Reinforcement Learning Applied to Multi-intersection Traffic Signal Control

JAMES AULT, Texas A&M University, USA
GUNI SHARON, Texas A&M University, USA

A recent stream of publications proposed to model traffic signal control as a Markov decision process and optimize it with standard or adjusted reinforcement learning (RL) algorithms. While presenting compelling results for optimizing such controllers in stand-alone intersections, limited success was shown for cooperative control over multiple intersections. Still, current control schemes such as green-wave propagation show that coordinating and optimizing signal controllers over adjacent intersections has the potential to greatly reduce congestion and, as a result, emissions and travel times. In this paper we investigate the applicability of state-of-the-art multiagent RL (MARL) algorithms to the multi-intersection signal control domain. We show that, such algorithms suffer from inherent limitations in this domain which prevent them from converging to an optimized coordinated policy.

CCS Concepts: • **Computing methodologies** → **Multi-agent reinforcement learning**; • **Applied computing** → **Transportation**.

1 INTRODUCTION

Signalized intersections are a known bottleneck responsible for 12–55% of commute time in urban areas [12]. Recently proposed solutions suggest modeling traffic signal control (TSC) as a Markov decision process (MDP) and training the signal controller with reinforcement learning (RL). While presenting compelling results showing a reduction of up to 52% in average vehicle travel time when compared to fixed-time actuation, these RL solutions were shown to fail at coordinating the signal control over several intersections [2]. Domain-independent multiagent reinforcement learning (MARL) algorithms are designed to address similar coordinated control problems. In this paper, we investigate the performance of MARL benchmark algorithms [9] when applied to benchmark multiagent TSC (MATSC) tasks [2]. Our study found that despite strong results shown in other domains, cooperative MARL algorithms may not be particularly effective in MATSC tasks when compared to independent single-agent RL algorithms.

2 BACKGROUND

In reinforcement learning (RL) an agent is assumed to learn through interactions with the environment. The environment is commonly modeled as a Markov decision process (MDP) which is defined by: $\mathcal{S}$ – the state space, $\mathcal{A}$ – the action space, $\mathcal{P}(s_t, a, s_{t+1})$ – the transition function of the form $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, $R(s, a)$ – the reward function of the form $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, and γ – the discount factor. The agent is assumed to follow an internal policy π which maps states to actions, i.e., $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The agent's chosen action (a_t) at the current state (s_t) affects the environment such that a new state emerges (s_{t+1}) as well as some reward (r_t) representing the immediate utility gained from performing action a_t at state s_t, given by $R(s, a)$. The observed reward is used to tune the policy such that the expected sum of discounted reward, $J_\pi = \sum_t \gamma^t r_t$, is maximized.

2.1 Traffic Signal Control (TSC) as an MDP

A signalized intersection is composed of incoming and outgoing roads where each road is assembled from one or more lanes. At each time step, a signal controller is responsible to enable some combination of non-conflicting phases such that some utility measurement is optimized. A phase is a specific traffic movement through the intersection activated

Authors' addresses: James Ault, Texas A&M University, USA, jault@tamu.edu; Guni Sharon, Texas A&M University, USA, guni@tamu.edu.

by one or multiple physical signals. We adopt the TSC MDP formalization presented by Ault et al. [1]. The **state space ($\mathcal{S}$)** is defined by the state of incoming traffic and the currently enabled phases. The **action space ($\mathcal{A}$)** is all sets of non-conflicting phases to be assigned the right-of-passage (green light). The **transition function ($\mathcal{P}$)** is defined by the traffic progression following the signal assignment as determined by a simulated environment (we use the simulation of urban mobility, SUMO [3]). The **reward function ($\mathcal{R}$)** is the negative waiting time summed over all vehicles within 200m of an intersection. The **discount factor (γ)** is set to 0.99.

2.2 Multiagent RL (MARL)

In a multiagent MDP setting the action space is a Cartesian product of several sub action spaces $A \leftarrow A_1 \times ... \times A_n$ where each sub action space A_i is affiliated with a single agent i. MARL algorithms commonly belong to one of two classes, (1) *independent* – agents are trained without knowledge of other agents' actions and local observations and (2) *centralized training-decentralized execution* (CTDE) – agents' policy is conditioned only on local observations but can be trained with access to all agents' observations and actions.

The Extended PyMARL (EPyMARL) [9] codebase provides benchmark MARL implementations. In total it includes 9 multi-agent algorithms:

Independent algorithms: (1) Independent Q-Learning (**IQL**) [14]; (2) Independent Asynchronous Advantage Actor-Critic (**IA2C**) [8]; (3) Independent Proximal Policy Optimization (**IPPO**) [11]

CTDE algorithms: (4) Value Decomposition Networks (**VDN**) [13]; (5) **QMIX** [10]; (6) Multi-Agent DDPG (**MADDPG**) [7]; (7) Counterfactual Multiagent Policy Gradient (**COMA**) [6]; (8) Multiagent A2C (**MAA2C**); (9) Multiagent PPO (**MAPPO**) [15]

2.3 The Reinforced Signal Control (RESCO) toolkit

The Reinforced Signal Control (RESCO) toolkit gives a benchmark for MATSC. The tasks have between 3 and 21 heterogeneous signalized intersections spread across urban traffic networks. RESCO provides benchmark implementations of MARL algorithms specifically designed for coordinated MATSC. These include, MPLight [4] which uses upstream and downstream pressures in the state and reward on top of a shared-parameter specialized DQN model and Feudal Multiagent A2C (FMA2C) [5] which uses a hierarchy of *managing agents* to coordinate independent *worker agents*.

RESCO also includes independent MARL methods (not coordinated) which use only local rewards, instead of a single global reward that is common in EPyMARL. In each of the benchmark tasks the authors of [1] found that the independent methods outperform MPLight and FMA2C. The authors noted that while their independent PPO implementation could improve performance slighty (about 5% on average) over that of DQN, the later was considerably faster to converge (an order of magnitude fewer training episodes).

3 EXPERIMENTS

We compared all of the MARL algorithms implemented in EPyMARL against the benchmark TSC algorithms provided by RESCO on the 4 MATSC benchmark tasks ("Cologne Corridor", "Cologne Region", "Ingolstadt Corridor", "Ingolstadt Region"). For the tested algorithms, all hyper-parameters were set to those given by [1] and [9]. The general implementations in EPyMARL make 2 unique assumptions regarding the underlying environments that do not hold for TSC and, thus, require special modifications.

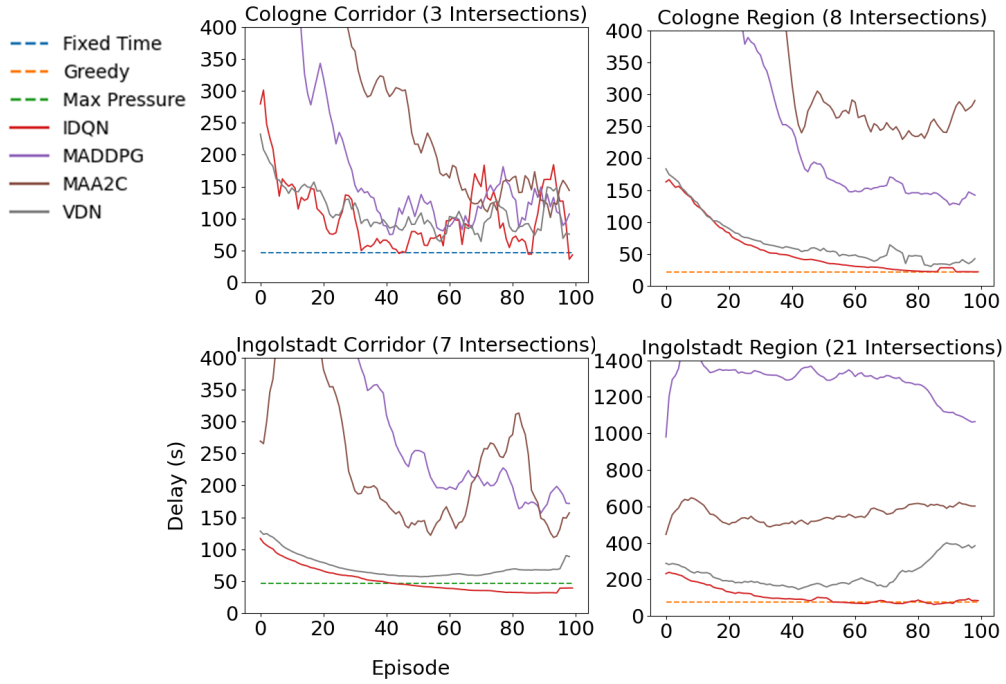

Fig. 1. Learning curves over 10 random seeds, with a sliding window average of 5 episodes. Dashed lines give the best static baseline method from RESCO for each task. Only the 4 best performing algorithms are shown out of 10, for ease of presentation.

Assumption 1 (parallel worlds): The environment can be instantiated arbitrarily. **TSC case:** A real-world TSC environment can not be arbitrarily instantiated multiple times (outside of a simulator). **Modification 1:** Algorithms in EPyMARL which expect parallel policy rollouts were adjusted to receive a single rollout.

Assumption 2: Optimization is performed episodically over a total of millions of environmental steps. **TSC case:** Evaluation in MARL research on TSC is typically only over hundreds of episodes [1, 2, 4], e.g., in RESCO each episode is set to 360 steps. **Modification 2:** EPyMARL was extended to support n-step execution of the implemented algorithms, such that policy gradient methods are rolled out for 10 environment steps instead of 10 episodes and Q-learning based methods perform a policy update after each environment step instead of each episode.

The codebase for our experiments is provided open-source at: *https://github.com/Pi-Star-Lab/epymarl_resco*

RESCO's independent DQN implementation denoted **IDQN** in Figure 1 performed significantly better than the three independent methods from EPyMARL. This is principally due to the difference in state representation and local rewards. IDQN aggregates each agent's observations in convolutional layers over lanes composing the same incoming road. Both QMIX and COMA failed to converge on all TSC tasks, likely due to sensitivity of the default hyper-parameters. Tuning these parameters on the validation tasks in RESCO may alleviate this issue. In the Ingolstadt Region task, only the independent methods IDQN, IA2C, and IPPO converge. Ingolstadt Region can have significant distance between the intersections (independent agents) reducing and varying the impact each agent has on another. For the Cologne Region task the independent methods all performed similarly. MAA2C and MADDPG converge to local optimums, however in this case VDN performs reasonably well (comparable to the best RESCO benchmark, IDQN). The discrepancy in the

VDN performance between Ingolstadt and Cologne (regional) is explained by the fact that the Cologne task has less than half the number of intersections in a much smaller area. The Cologne Corridor task presented the most challenging coordination problem, with low traffic arterial roads and short distances between intersections. In this task, none of the algorithms improved over a static fixed time controller and performance between them was not significantly different.

4 CONCLUSION

Our experimental study suggests that domain-independent MARL algorithms are not more effective in optimizing multi-intersection signal control when compared to RL algorithms that are specifically designed to solve such problems. Moreover, our study suggest that independent MARL methods perform better when compared to those explicitly targeting coordinated behavior. This trend seems to follow similar conclusions drawn from the original EPyMARL work where, in 19 of 25 tasks, independent MARL approaches performed best. We conclude that training coordinated behaviors in MARL is challenging in many domains including signal control, even when considering as few as three intersections.

REFERENCES

[1] James Ault, Josiah Hanna, and Guni Sharon. 2020. Learning an Interpretable Traffic Signal Control Policy. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

[2] James Ault and Guni Sharon. 2021. Reinforcement Learning Benchmarks for Traffic Signal Control. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track*.

[3] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. SUMO–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

[4] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward A thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3414–3421.

[5] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (2019), 1086–1095.

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[7] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[8] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[9] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*.

[10] 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.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[12] Guni Sharon. 2021. Alleviating Road Traffic Congestion with Artificial Intelligence. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4965–4969. https://doi.org/10.24963/ijcai.2021/704 Early Career.

[13] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

[14] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*. 330–337.

[15] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. arXiv:2103.01955 [cs.LG]