Learning to Communicate with Reinforcement Learning for an Adaptive Traffic Control System

Simon Vanneste, Gauthier de Borrekens, Stig Bosmans, Astrid Vanneste, Kevin Mets, Siegfried Mercelis, Steven Latré, Peter Hellinckx

Abstract Recent work in multi-agent reinforcement learning has investigated inter agent communication which is learned simultaneously with the action policy in order to improve the team reward. In this paper, we investigate independent Q-learning (IQL) without communication and differentiable inter-agent learning (DIAL) with learned communication on an adaptive traffic control system (ATCS). In real world ATCS, it is impossible to present the full state of the environment to every agent so in our simulation, the individual agents will only have a limited observation of the full state of the environment. The ATCS will be simulated using the Simulation of Urban MObility (SUMO) traffic simulator in which two connected intersections are simulated. Every intersection is controlled by an agent which has the ability to change the direction of the traffic flow. Our results show that a DIAL agent outperforms an independent Q-learner on both training time and on maximum achieved reward as it is able to share relevant information with the other agents.

1 Introduction

Traffic congestion is a worldwide problem which has a big environmental and economical impact. It has been estimated that in the EU, most traffic congestion's occur around urban areas and that these congestion's cost nearly 1% of the EU's GDP [1]. In this work, we will investigate traffic light control in order to improve congestion in urban area. Reinforcement learning (RL) is a machine learning method that searches for a policy which will maximize the future expected reward. RL has been

Simon Vanneste, Gauthier de Borrekens, Stig Bosmans, Astrid Vanneste, Kevin Mets, Siegfried Mercelis, Steven Latré, Peter Hellinckx

University of Antwerp - imec, IDLab, Faculty of Applied Engineering, Belgium, Sint-Pietersvliet 7, 2000 Antwerpen, e-mail: simon.vanneste@uantwerpen.be gauthier.deborrekens@uantwerpen.be, stig.bosmans@uantwerpen.be, astrid.vanneste@uantwerpen.be, kevin.mets@uantwerpen.be, siegfried.mercelis@uantwerpen.be, steven.latre@uantwerpen.be, peter.hellinckx@uantwerpen.be

demonstrated to successfully learn a policy in a wide area of environments like the Atari environments [8]. RL has also successfully been applied to intelligently managing traffic lights [15][19]. Single agent RL implementations have shown effective in simplified traffic environments, but suffer in more complex and realistic scenarios. To address this, Multi-Agent Reinforcement Learning (MARL) has been proposed, in which every traffic light is controlled by a single agent. These independent agents still have access to the full state of the environment. In this work, we will investigate a MARL system with agents that do not have access to the full state but only to their local observation for the Adaptive Traffic Control System (ATCS) environment. In order to overcome this partial observability, the agents have the ability to communicate with each other in order to share information about their observation with the other agents. The communication protocol is not predefined, but are fully enveloped in the reinforcement learning process, meaning the agents will learn how to communicate during the reinforcement learning process. If the agents do not discover the benefit of communication, no emergence will take place. Independent Q-learning and Differential Inter Agent Learning (DIAL) [3] are trained on a simplified traffic environment, simulated with the open-source simulator Simulation of Urban Mobility (SUMO) [6]. The FLOW framework is used as an interface between the RL algorithm and the simulated traffic environment [18]. The paper is structured as followed. In Section 2, we will discuss the related work. Section 3 describes the used reinforcement learning methods. The environment used to train these reinforcement learning agent is discussed in Section 4. Next, we describe the results in Section 5. Finally, we present our conclusion and future work in Section 6.

2 Related Work

This section will discuss the related work in the domain of MARL for the ATCS application and the related work in communication learning using MARL. El-Tantawy et al. [2] present a MARL ATCS called MARLIN-ATCS which is an independent Q-learning method that has a coordination system with neighbouring agents. This method is shown to outperform an independent Q-learner and real world controllers on the simulated network of Downtown Toronto. Van der Pol et al. [10] presented a multi-agent reinforcement learning system in which the global Q-function is learned as a linear combination of local subproblems. The subproblems are trained on the basic source problems containing two agents. Next, the global Q-function is used in the max-plus coordination [4] algorithm to optimize the actions of the agents. This methods allows the agent to learn an action policy while circumventing the non-stationarity problem and the high training costs of MARL. Tan et al. [14] presented a hierarchical multi-agent reinforcement learning system in which several regional controllers are combined with a centralized global agent. The regional controllers will all present a combination of actions which is further optimized by the centralized global agents. These methods focus on the challenges of large scale learning in

an ATSC system. In this work we focus on the challenge of learning in a decentralized Markov decision process (Dec-MDP) [9] in which agents learn to communicate in order to share information about their local observation.

The MARL domain is an active research domain in which several MARL communication learning methods have been developed. Sukhbaatar et al. [11] presented the CommNet RL method that will encode the communication information into the hidden state. This hidden state will be shared with the other agents and combined with their hidden state into the final hidden state which will be used by the agent to generate an action. This communication is learned end-to-end by allowing the gradients to flow through the agents based on the loss of the receiving agents. Foerster et al. [3] presented two methods to learn to communicate. Reinforced Inter-Agent Learning (RIAL) will use Q-learning to select both the action and the message to send to the other agents. Differentiable Inter-Agent Learning (DIAL) will train the communication end-to-end as discussed in section 3.2. Their experiments show that DIAL outperforms RIAL because of the direct feedback the receiving agent can give to the sending agent using the end-to-end training. DIAL has been extended by Vanneste et al. [16] to include a value-decomposition network [12]. The value-decomposition network can be used to improve the lazy agent problem in which a low performing agent is rewarded for the performance of the other agents. Another method to reduce the lazy agent problem while learning to communicate is by using a centralized critic as shown in the MADDPG method [7] and the MACC method [17].

3 Method

In this section, independent Q-learning and Differentiable Inter-Agent Learning are described. These two methods will be used to train our agent for the ATCS.

3.1 Independent Q-Learning

In the traditional single agent RL setting, the agent will receive the state $s_t \in S$ at timestep t and respond with an action $u_t \in U$. The environment will use this action to present the next state s_{t+1} and reward r_t to the agent which will be used to train the agent. Q-learning learns a Q-value for every state-action pair $Q(s, u)$ in which a higher Q-value represents a better action. The RL policy can be defined as $\pi(s) = argmax_u Q(s, u)$. Deep Q-learning will use a deep Q-network (DQN) with network parameters θ to represent this Q-function. This network will be trained by minimizing the loss function which is show in Equation 1. The discount factor γ is a value between 0 and 1 and represents the balance between long term and short term goals.

$$\mathscr{L}_i(\theta_i) = \mathbb{E}_{s_t, u_t, r_t, s_{t+1}} \left[(r_t + \gamma \max_{u_t} Q(s_{t+1}, u_{t+1}, \theta_i) - Q(s_t, u_t, \theta_i))^2 \right] \qquad (1)$$

An extension to DQN is independent Q-learning (IQL)[13] which allow us to train agents in a cooperative multi-agent setting. In IQL, every agent a will train an individual Q-network $Q^a(s_t, u_t^a; \theta^a)$ based on the global state s_t, the individual action of the agent u_t^a and the team reward r_t. The agent receive the same team reward in order to encourage cooperative behaviour.

3.2 Differentiable Inter-Agent Learning

IQL assumes that the full state s_t is available to every agent. In real-world applications, agents often have only a partial observation o_t of the environment. In this work, we will assume that our problem can be represented as a decentralized Markov decision process (Dec-MDP) [9] which means that the combination of all the observations of the different agents again represent the full state. In order to overcome this partial observability, the agents will have the ability to communicate with each other. The communication will allow the agent to send relevant observation information to other agent in order to recreate the state as required by IQL.

The Differentiable Inter-Agent Learning[3] (DIAL) method is an extension to the IQN in order to allow agent to learn an inter agent communication protocol. This communication protocol is not predefined but is learned simultaneously with the action policy. The action part of DIAL is identical to IQL apart from the fact that the action policy will also have access to the received messages. The DIAL method will learn the communication end-to-end by allowing the gradients to flow from the receiving agent back to the sending agents. These gradients are calculated in order to improve the IQN loss function of the independent Q-learner. DIAL can be used to learn continuous or discrete messages. In this work, we used continuous messages but this can be altered to discrete messages by including a discretise/regularise unit[3]. We have made a slight alteration to DIAL by splitting the action and communication policy without sharing any network parameters between these policies. This is possible because in our application, we are focusing on overcoming the partial observability of the Dec-MDP and no information is shared between the action and communication policy. This separation is however not possible when agents need to communicate strategic decisions with each other. The DIAL training process is illustrated in Figure 1.

4 Environment

Reinforcement learning requires an environment to provide observations, execute the actions and evaluate the behaviour of an agent. To simulate the traffic, the Simulation of Urban MObility (SUMO) simulator [6] is used. SUMO allows for large-
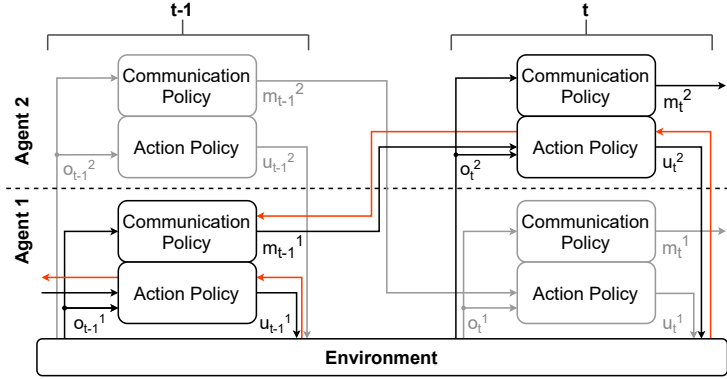
Fig. 1 DIAL with a separate action and communication policy.

scale, continuous simulation with precise control over many factors. The FLOW project [18] is a deep reinforcement learning framework that provides an interface for implementing and training RL algorithms on top of the SUMO simulator. It consists of both a network and an environment. The FLOW network contains all the information about the SUMO traffic simulation itself. It consists of a traffic layout defining the lanes and intersections. This layout is populated with vehicles and two traffic lights, each having a logic controller that defines their behaviour. For the vehicles, the built-in controller algorithm is used which will mimic human-like braking and acceleration behaviour. For the traffic lights, we substitute the controller with an IQL or DIAL agent. The FLOW environment acts as the interface between the RL agent and the simulation. It defines the observation space that is returned to the agents, performs the agent actions on the environment and defines the reward function. In Figure 2, our SUMO simulation is shown where the network consists of a grid of two intersections connected by a shared road. Both intersections contain traffic lights that are controlled by separate RL agents.

The 26 values of the observation are shown in Table 1. The first 12 values contain information on the speed, distance and position of the leading vehicle on each lane going toward the intersection. The edge number is used to identify which lanes the agents are observing, providing them information on their position in the environment. The following 12 values contain information on the traffic congestion, waiting queue and total vehicle waiting time of each of the four adjacent lanes. The final two bits indicate which direction the intersection is allowing through and if it is currently switching, indicated by a yellow light. Each agent also observes the communication messages which consist of 5 continuous values.

The action consists of a single value that is used to switch directions of the traffic on the intersection. The reward function aims to optimize global waiting times and is formulated as followed. Each vehicle contains an accumulated waiting time value that is incremented by one for each timestep its velocity is below 2 miles per hour. When the vehicle surpasses this velocity threshold, the total waiting time is decremented by 0.4 each timestep. The reward is the negative of the average of all
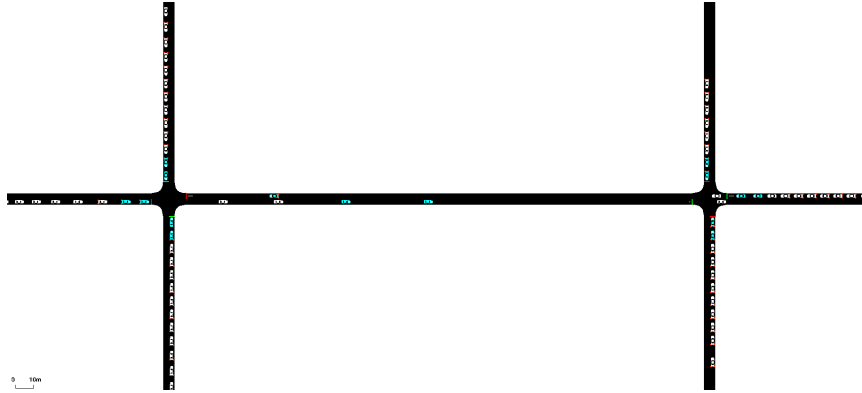
Fig. 2 A SUMO simulation with a network consisting of two intersections of four lanes, connected by a horizontal lane and currently being populated with vehicles.

0:3	4:7	8:11	12:15
Speed	distance to edge	edge number	total vehicles
16:19	20:23	24	25
waiting vehicles	waiting time	direction	currently yellow

Table 1 The 26 dimension observation space representing the current state of the environment.

accumulated waiting times at that timestep of both intersections. This reward was chosen in order to find a balance between waiting time and throughput on the intersections.

5 Results

This section will evaluate both IQL and DIAL in the traffic light environment. In the first subsection, we will evaluate the training behaviour of IQL and DIAL. In the second subsection, the trained policies are evaluated on different levels of inflow. These experiments are trained using a DIAL implementation within the RLlib [5] framework. In both experiments, the IQL and DIAL method have the same neural network architecture for the action policy. This action neural network consists of two layers of 256 neurons and a ReLu activation function. The final layer consists of a linear layer with and output size of 2. The hyperparameters for both methods are shown in Table 2. The DIAL method has an additional communication neural network which consists of no hidden layers as this network will make an encoding of the observation into a messages with 5 dimensions. This network could be expanded in order to reduce the size of the message or to move processing from the receiver towards the sender. In this work, we chose a linear mapping between observations and message in order to make the messages more understandable.

Parameter	Value
Gamma	0.99
Epsilon	1
Epsilon Decay	0.99995
Minimal Epsilon	0.05
Learning rate	0.0005
Optimizer	Adam

Table 2 The IQL and DIAL hyperparameters.

5.1 Training

The IQL and DIAL policies are trained using $25 * 10^3$ epochs and 35 million environment interactions. An epoch is defined as a training iteration in which we use 7 episodes with a fixed length of 200 timesteps. The agents are trained using a random inflow probability between 0.1% and 60%. Every configuration is trained five times using a different random seed. The average reward from these experiments are presented in Figure 3. At epoch $5 * 10^3$, DIAL starts to outperform IQL in average reward while having a smaller standard deviation. Between epoch $22 * 10^3$ and $25 * 10^3$, the standard deviation of IQL becomes smaller as the standard deviation of DIAL but the average reward of DIAL remains higher than the average reward of IQL. These results show that DIAL is outperforming IQL both in learning speed and in average reward.
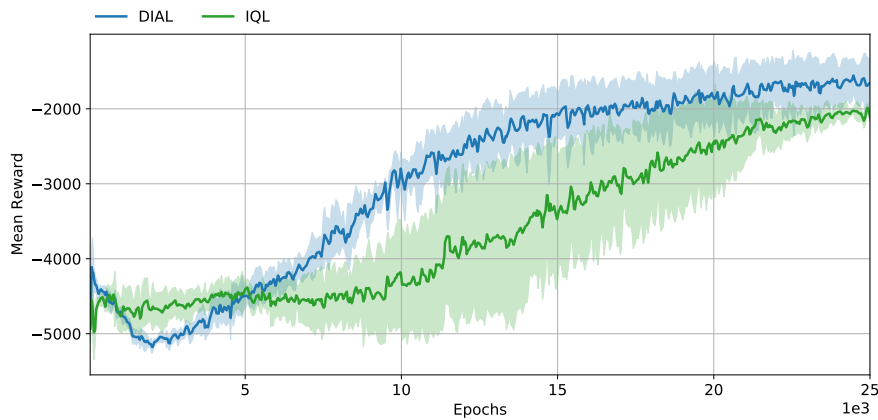

Fig. 3 The average training results of five runs for IQL and DIAL in the traffic light environment with the shaded area showing the standard deviation.

Table 3 shows the average reward of the best performing training iteration for both IQL and DIAL. These results show that DIAL is able to achieve a higher peak reward (8% improvement) and achieve this faster than IQL.

Method	Peak Performance Epoch	Average Peak Reward	Peak Performance Standard Deviation
IQL	$25.175 * 10^3$	-1986	54
DIAL	$24.75 * 10^3$	-1555	223

Table 3 The average reward of five runs from the best performing training iteration for the IQL and DIAL method.

5.2 Evaluation

Once the policies have been trained, a comparison on the performance is conducted by evaluating them on environments with fixed inflows as opposed to the random inflows used during training. Each policy iterates over ten fixed inflows and will be evaluated 200 times per inflow. The inflows are distributed evenly between the range of inflows used during training. The purpose of this experiment is to more accurately evaluate the behaviour of the trained policies without the random factor present during learning. An additional configuration is added to gain insight into the need for communication during evaluation. The additional configuration will be using DIAL with disabled communication which is achieved by sending the message to both agents that was generated by the communication policy when no vehicles were observed. This will indicate whether communication is used as a bias or if it is actually used during evaluation. If the communicating policy performance significantly drops when the agents are not allowed to communicate, communication is used for cooperation.
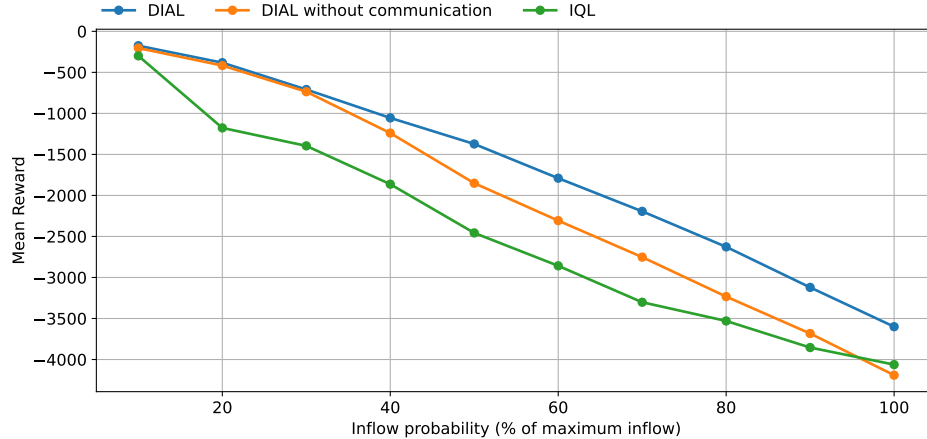

Fig. 4 Evaluation of the trained IQL and DIAL policies.

The result of this experiment can be found on figure 4. First, it is clear that DIAL outperforms IQL by a substantial amount. The difference is largest with average inflows ranging from 40% to 80%, slightly decreasing at the extremes. This indicates

that the emergence of communication leads to a more optimal traffic control behaviour. Finally, DIAL without communication is able to have similar performance for the low inflow probabilities but is outperformed by DIAL as the inflow probabilities increase. Since the default message used in the DIAL without communication was generated when no vehicles are observed, this message will be close to the message that DIAL will send when there is a low inflow probability. Therefore DIAL without communication and DIAL will have a similar average reward for these low inflow probabilities. DIAL without communication has a lower average reward than DIAL and IQL for an inflow probability of 100% as the agents are receiving message from a low inflow probability situation. This result shows that DIAL does not just have a better policy but has learned a valid communication protocol which increase the performance over a range of inflow probabilities.

6 Conclusion

In this work, a multi-agent reinforcement learning agent for an adaptive traffic control system is investigated. We compare independent Q-learning with the DIAL method which has the ability to learn communication between the agents. These methods are compared on an environment with two traffic light, using the SUMO simulator, in which the agents can only observe a part of the entire state. In this configuration, we demonstrate that DIAL is able to learn faster and achieve a higher maximum reward than an independent Q-learner. These methods are evaluated on a range of inflow probabilities and the results demonstrate that DIAL is able to achieve a better result on all of the evaluated settings. We believe that these results show the importance of learning a communication policy simultaneously with the action policy for real world use cases in which not every agent has access to the full state but the agents need to learn how to share information in order to overcome this limited view of the current state.

In future work, a more complex environment could be used that includes more agents that control more individual intersections in which the agent will learn to communicate with each other. This increase in number of agent will present certain challenges like the lazy agent problem (see Section 2). This problem could be minimized by using a centralized critic which has been demonstrated by the MADDPG method [7] and the MACC method [17]. Additionally, more communication topologies should be investigated. This will become especially important when the amount of agents increases and broadcasting every message becomes infeasible.

References

1. Commission, E.: Roadmap to a Single European Transport Area: Towards a Competitive and Resource Efficient Transport System: White Paper. Publications Office of the European Union (2011)
2. El-Tantawy, S., Abdulhai, B., Abdelgawad, H.: Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto. IEEE Transactions on Intelligent Transportation Systems **14**(3), 1140–1150 (2013)
3. Foerster, J.N., Assael, Y.M., De Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. arXiv preprint arXiv:1605.06676 (2016)
4. Kok, J.R., Vlassis, N.: Using the max-plus algorithm for multiagent decision making in coordination graphs. In: Robot Soccer World Cup, pp. 1–12. Springer (2005)
5. Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J.E., Jordan, M.I., Stoica, I.: RLlib: Abstractions for distributed reinforcement learning. In: International Conference on Machine Learning (ICML) (2018)
6. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE (2018). URL https://elib.dlr.de/124092/
7. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv:1706.02275 (2017)
8. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
9. Oliehoek, F.A., Amato, C.: A concise introduction to decentralized POMDPs. Springer (2016)
10. Van der Pol, E., Oliehoek, F.A.: Coordinated deep reinforcement learners for traffic light control. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016) (2016)
11. Sukhbaatar, S., Szlam, A., Fergus, R.: Learning multiagent communication with backpropagation. arXiv preprint arXiv:1605.07736 (2016)
12. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)
13. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the tenth international conference on machine learning, pp. 330–337 (1993)
14. Tan, T., Bao, F., Deng, Y., Jin, A., Dai, Q., Wang, J.: Cooperative deep reinforcement learning for large-scale traffic grid signal control. IEEE transactions on cybernetics **50**(6), 2687–2700 (2019)
15. Thorpe, T.L.: Vehicle traffic light control using sarsa (1997)
16. Vanneste, S., Vanneste, A., Bosmans, S., Mercelis, S., Hellinckx, P.: Learning to communicate with multi-agent reinforcement learning using value-decomposition networks. In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 736–745. Springer (2019)
17. Vanneste, S., Vanneste, A., Mercelis, S., Hellinckx, P.: Learning to communicate using counterfactual reasoning. arXiv preprint arXiv:2006.07200 (2020)
18. Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., Bayen, A.M.: Flow: A modular learning framework for autonomy in traffic (2017)
19. Zheng, G., Zang, X., Xu, N., Wei, H., Yu, Z., Gayah, V., Xu, K., Li, Z.: Diagnosing reinforcement learning for traffic signal control. arXiv preprint arXiv:1905.04716 (2019)