Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 201 (2022) 24–31

Procedia Computer Science

www.elsevier.com/locate/procedia

The 13th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 22 - 25, 2022, Porto, Portugal

Information upwards, recommendation downwards: reinforcement learning with hierarchy for traffic signal control

Taylor de O. Antes, Ana L. C. Bazzan*, Anderson Rocha Tavares

Instituto de Informática / PPGC, Universidade Federal do Rio Grande do Sul, Caixa Postal 15064, 91.501-970 Porto Alegre, Brazil

Abstract

Traffic signal control (TSC) is a practical solution to the major problem of congestion in metropolitan areas. Reinforcement Learning (RL) techniques present powerful frameworks for optimizing traffic signal controllers that learn to respond to real-time traffic changes. Multiagent RL (MARL) techniques have been showing better results over centralized techniques (RL-based or not), where local intersection agents have partial observation of and control over the environment. Since in TSC the best decision does not depend only on local information, in the present paper we aim at increasing agents' views by using a hierarchical approach, where information is passed upwards, is then aggregated forming recommendations that are sent downwards. We divide the transportation network into regions, each controlled by a region agent; this is done at different hierarchical levels. The traffic signal controllers, located at the intersections, are the local agents at the hierarchy's bottom. Region agents can supervise intersection agents or other region agents. Evaluation of this approach in a synthetic traffic grid shows that the proposed hierarchical organization outperforms a fixed-time approach and an RL-based approach without hierarchy.

Keywords: Intelligent Transportation Systems; Smart Cities; Reinforcement Learning; Multiagent Systems

1. Introduction

Traffic congestion is known to cause serious problems such as increased travel time, fuel consumption and emissions. Due to the growing need for mobility in our society and the fact that it is not always feasible to increase the capacity of the infrastructure (e.g., the road network), a more efficient use of the existing infrastructure is necessary. Traffic signal control (TSC), one of the most traditional and important tools for traffic management, comes as a solution for the problem. One way to control traffic signals is via techniques of reinforcement learning (RL), where signal controllers learn how to dynamically respond to real-time changes in traffic. A traffic signal controller using an RL approach can be trained to learn a control policy, based on interactions with the traffic system.

In a centralized RL approach for TSC, traffic measurements in the network are collected and used to determine states and actions. One drawback of this kind of approach is that their application is usually infeasible for large traffic networks, given that the state and action spaces grow exponentially with the network size. On the other hand, in a

* Corresponding author.
Email address: bazzan@inf.ufrgs.br

1877-0509
10.1016/j.procs.2022.03.006

decentralized or multiagent RL (MARL), each controller is an agent that learns independently, in a more scalable approach. However, in this approach, agents optimize their policies based on their own local observations only (local view), thus disregarding other agents' states and actions.

In order to address such a gap, in this paper we propose the Information Upwards, Recommendation Downwards (IURD) approach, a hierarchically-organized RL framework. Specifically for TSC, we split the traffic network into regions, considering different hierarchical levels. Each region is controlled by a region agent. At the bottom level of the hierarchy, intersection agents (henceforth also called local agents) control the traffic signals. At other hierarchical levels, region agents supervise a set of intersection agents or a set of other region agents. Subordinate agents pass information to their supervisors about their own observations. The supervisors use their subordinates' information to learn and provide a region-wise recommendation. This recommendation is passed to the subordinates, which try to improve their policies, while also taking their supervisor's recommendation into account.

The type of organization of the collective of agents that underlies IURD helps to guide the learning of the intersection agents towards a better collective performance, while maintaining the scalability. We apply the proposed IURD framework in TSC using vector-based calculus as a simple but efficient method to aggregate information between agents of different hierarchical levels.

In summary, the main contributions of this paper are: (i) IURD, a hierarchically-organized RL framework that guides low-level controllers towards a better collective performance; and (ii) empirical evidence that IURD is promising as a scalable framework for RL in TSC.

2. Background and Related Work

An RL task can be formulated as a Markov decision process (MDP), i.e., by a tuple (S,A,T,R), where S is the set of states, A is the set of actions, $T : S \times A \to \Psi(S)$ is the state transition function where $\Psi(S)$ is a probability distribution over S, and $R : S \times A \to \mathbb{R}$ is the expected reward function. An agent interacts with the environment and tries to learn the optimal policy π^*, which maps each state $s \in S$ to an action $a \in A$, so that utility is maximized. The utility of each state-action pair is based on immediate and subsequent rewards that the agent receives when interacting with the environment.

Sarsa (State-Action-Reward-State-Action) is an RL algorithm in which the agent learns the utility of each state-action pair, here called the *Q-value*. The *Q-value* (*Q(s,a)*) reflects an agent's expected utility for performing the action a in a given state s and following its policy thereafter. *Q-values* can be learned directly from an experience tuple (s,a,r,s',a'), that denotes the fact that the agent, in the a state s, performed the action a, received reward r, moving to the next state s' and choosing the next action a'. Let $\alpha \in [0,1]$ be the learning rate and $\gamma \in [0,1]$ be the discount factor; the *Q-value* is then updated via Eq. (1). The learning rate determines to which extent the most recent interaction overrides the old experience and the discount factor determines the importance of future rewards.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s',a') - Q(s,a) \right] \tag{1}$$

In the present work, the agents use the Sarsa algorithm with linear function approximation, in which a state s is represented by a feature vector $(k_1(s),\ldots,k_n(s))$, and the action-value $Q(s,a)$ is approximated by $\tilde{Q}(s,a,w) = \sum_{i=1}^{n} k_i(s) \cdot w_i^a$, where $(w_1^a,\ldots,w_n^a)$ is a weight vector for action a. The learning problem is then to adjust the weights for each action to better approximate its value. Given an experience tuple (s,a,r,s',a'), each weight w_i^a is updated according to Eq. (2):

$$w_i^a \leftarrow w_i^a + \alpha \left[r + \gamma \tilde{Q}(s',a',w) - \tilde{Q}(s,a,w) \right] \times k_i(s) \ . \tag{2}$$

As for related work, traffic signal control has been extensively studied using various approaches, such as classical control theory, fuzzy systems, neural networks, and RL among others. Next, we briefly discuss some of the approaches that are closer to ours, focusing on RL-based ones.

In the RL front, challenges are the representation of the dynamics and how they affect the formulation of state, action, and reward. Existing MARL works model the traffic and intersection dynamics using different data sources, such as queue length, vehicle waiting times or flow. Surveys (e.g., [14, 15, 16]) on the various approaches offer a more detailed discussion of the subject.

To the best of our knowledge, works that take advantage of a hierarchical structure are rarely seen in the literature. In Abdoos et al. [2], a holonic multiagent system is used to model a traffic network. The hierarchy is divided into super-holons and sub-holons, where the former receive abstractions of their sub-holons and restrict the actions their subordinates can take while learning. In Abdoos et al. [3], a two-level Q-learning hierarchy is presented, with the superior one using tile coding to learn how to restrict the low-level agent's actions in order to improve the collective performance. In Bazzan et al. [5], a hierarchy of supervisors and supervised agents is presented, in which RL is not used at supervisors level. In [6], traffic networks are modeled with three hierarchical levels: intersections, zones and regions, each using a neural network. In [13], the authors used a feudal hierarchy with the MA2C [7] algorithm, dividing the network into regions and the agents into workers (low level intersection controller agents) and managers (high level region controller agents), obtaining significant improvement in traffic flow. Managers, however, considered only the information at the border of the controlled region to learn to set a goal for the workers inside it. Recently, [1] presented a two-level hierarchy that uses long-short term memory (LSTM) for traffic prediction in the higher level. The network is divided in regions and, using the traffic prediction, the region agents try to find the best joint action for the local controllers agents inside the controlled region. The local controllers uses a threshold mechanism to follow the recommended joint action or follow their own policy. The experiments show an improvement in average delay time using the proposed hierarchy.

Our approach is inspired by ideas of hierarchical [9], feudal [8] and holonic [10] learning. Differently from other works, we present a generic framework for organizing the task in a hierarchical manner, where higher level agents do not restrict their subordinates' actions.

3. Proposed Approach

The generic form of the IURD hierarchical organization is presented in Figure 1. A region agent a at hierarchical level l (R_a^l) has n subordinate agents at the level $l-1$ and is responsible for controlling a region. Subordinate agents send information (P) about their states to their supervisor agent. This information can be any combination of functions applied over the subordinates' partial observation. Therefore, the state of a region agent is an aggregation of its subordinates information. This works in all levels of the hierarchy. In short, if an agent has a supervisor, such a supervisor is informed about the agent's state.

Based on their current state, region agents choose what to recommend to their subordinates. Region agents do not restrict their subordinate actions; rather, they give an abstract recommendation.

In IURD, region agents' rewards can be any aggregation function over their subordinates' rewards. Also, agents that follow their supervisor recommendation receive extra reward, given by a function that maps the supervisor recommendation and the subordinate's action to a numerical value. Thus, as the subordinate agents are encouraged to follow recommendations, they seek collective and not just individual performance.

Henceforth, level zero agents, i.e., those that control intersections, are called intersection agents and an intersection agent i is represented by I_i.

The IURD framework for TSC employs a vector-based hierarchical organization (henceforth, VHO). The processed information, the supervisor recommendation, and the incentive function are calculated using a vector-based representation, as discussed next.

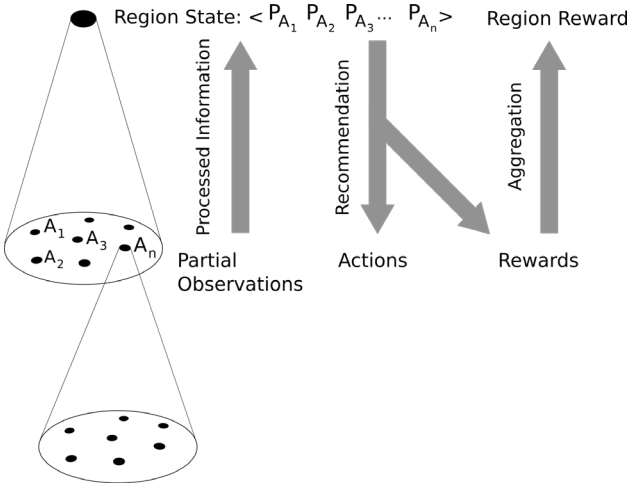
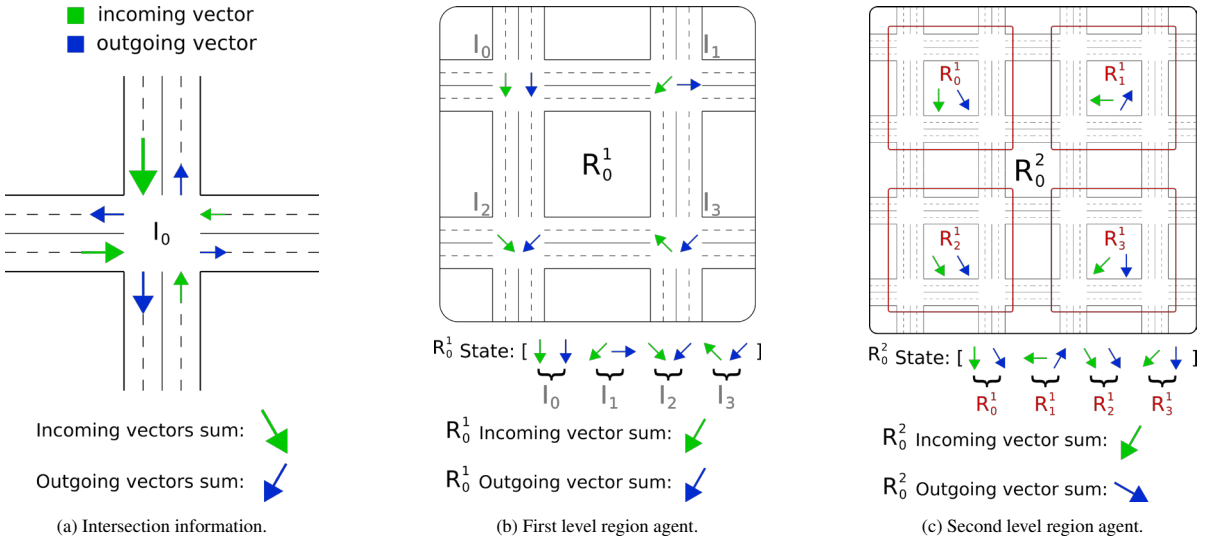


Fig. 1: IURD generic hierarchy

The information in the VHO is represented as vectors in their polar form. The angle represents the geographical orientation of the traffic flow and the magnitude represents the quantity of vehicles going in that direction. At each time step t, subordinate agents gather information for



(a) Intersection information. (b) First level region agent. (c) Second level region agent.

Fig. 2: Scheme of the VHO for processing flow of vehicles in IURD. For illustration, in the leftmost figure, the size of the arrows indicate magnitude of flows.

each traffic flow in its area, sum the vectors for the incoming and outgoing flows and pass these resulting vectors to their supervisor. In case this supervisor is also subordinate to other agent, it sums the incoming and outgoing vectors in its state and passes these vectors to the next higher level. This happens in each level of the hierarchical organization.

The supervisor recommendation ($\mathscr{A}^+$) is a vector that indicates a direction of traffic flow of the controlled region. In this structure, and based on its subordinate information, a region agent learns which of the 8 directions (E, NE, N, NW, W, SW, S, SE) needs to be prioritized. Figure 2 illustrates this procedure.

The reward function of the subordinate agents follows the equation $r' = r + r * \sigma(a, \mathscr{A}^+)$, where r' is the total reward, r is the reward obtained from interacting with the environment and σ is the incentive function that maps the subordinate's action (a) and the supervisor recommendation ($\mathscr{A}^+$) to a numerical value. In our case, the sigma function is defined as the cosine of the angle formed by the vectors that represent the supervisor recommendation and the subordinate's action.

4. Experiments

This section describes the experiments that were executed using SUMO, a microscopic traffic simulator [12]. We start by defining how the intersection and region agents function when the generic IURD framework is used in the TSC domain.

4.1. Intersection Agents

The intersection agent were modeled inspired by the one proposed in Alegre [4], as follows.

State. Let L be the intersection's lane set, o be the occupancy percentage of a lane, defined by the number of vehicles in the lane divided by the lane capacity, q be the queue, defined by the number of halting vehicles of a lane (vehicles are considered halting when their speed are lower than 0,1 m/s), and $\mathscr{A}^+$ be the supervisor recommendation. An intersection agent state is then defined as in Eq. (3).

$$s = [o_1, q_1, \cdots, o_L, q_L, \mathscr{A}^+] \ . \tag{3}$$

Action. At each time step t, the intersection agent chooses an action $a_t \in A$. Possible actions refer to the intersection signal phases (i.e., those traffic flows that receive green signal). If the current signal phase is chosen, it is extended for five extra seconds. This process must respect two constraints: a phase needs to last for a minimum time, and it cannot last for longer than a maximum green time.

Reward. The reward r of intersection agents is given by the difference between the vehicles' waiting time $w_{v,i}$ considering two successive actions. This aims at encouraging agents to reduce the waiting time at the intersections, improving traffic flow.

4.2. Region Agents

Region agents are modeled using the VHO approach, which was presented in Section 3. The state of a region agents is built using its subordinates' information; its action is one the possible recommendations. Recall that, as discussed, region agents add a value given by the incentive function $\sigma = \cos(.)$ to encourage their subordinate to follow their lead. The reward of region agents is defined as the mean of their subordinates' rewards.

4.3. Scenario

IURD using VHO was tested in a 4x4 grid network (16 intersections, Figure 3), as commonly found in the RL for TSC literature. The lanes have a length of 200m and a maximum speed of 50 km/h. Each intersection has two phases: one that allows green time to the flow of vehicles in the North-South direction (plus left turns); and similarly for the East-West direction. Agents choose an action every 5 seconds. The network is divided into four regions, each with four intersections. One group is B2, C2, B3, and C3. Others are similar.

There are 8 origin-destination (OD) pairs (A2F5, A3F4, A4F3, A5F2, B6E1, C6D1, D6C1, E6B1) that represent the trips, as illustrated in Figure 3. The demand is such that one vehicle is inserted in each OD pair every 7.2 seconds for the initial 10k steps. Each simulation is run for 25k steps, (approximately 7 hours).

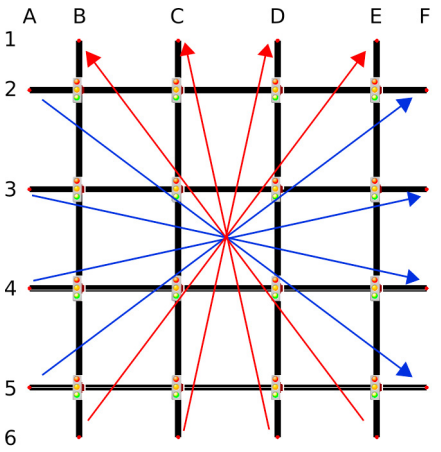


Fig. 3: Grid 4x4: topology and demand (OD pairs)

4.4. Metrics and Results

Using the scenario previously described, we compare three methods: the IURD method proposed, another RL-based method where agents learn individually (i.e., the standard Sarsa is employed at each intersection, without hierarchy, recommendation, or communication), and a fixed-time method (where phases have fix green time optimized by Webster's method that is implemented in SUMO[1]). Henceforth (and in the plots), we refer to them as IURD, IL (individual learning), and Webster respectively.

The values of the parameters in Eq. (2), as well as the exploration rate ε are given in Table 1 for both RL-based methods.

Table 1: Values of learning parameters.

par.	IL	IURD
α_i	1×10^{-3}	1×10^{-5}
α_r	—	1×10^{-6}
γ	0.95	0.95
ε	0.05	0.05

The results presented in this paper are derived from the average of 10 repetitions for each method. We note that the fixed-time (Webster) method that is implemented in SUMO yields the same output because the fixed times are pre-computed considering a demand that is given by the OD matrices, thus there is no need to repeat its execution.

In Table 2, the three methods are compared using the following metrics:

1. reward: average reward that intersection agents receive
2. trip completion: number of vehicles per second completing their trips

[1] Readers are referred to the software documentation at https://sumo.dlr.de/docs/Tools/tls.html

Table 2: Performance regarding different evaluation metrics.

Metrics	Method		
	Webster	**IL**	**IURD**
Reward ⇑	-3644	-1901 ± 141	-1384 ± 212
Trip completion flow [veh/s] ⇑	0.51	0.69 ± 0.03	0.74 ± 0.06
Time to process demand [s] ⇓	21792	16053 ± 714	14926 ± 1321
Trip avg. waiting time [s] ⇓	530	435 ± 41	393 ± 79
Trip avg. time loss [s] ⇓	660	543 ± 48	503 ± 87
Avg. depart delay [s] ⇓	2818	1351 ± 262	879 ± 304

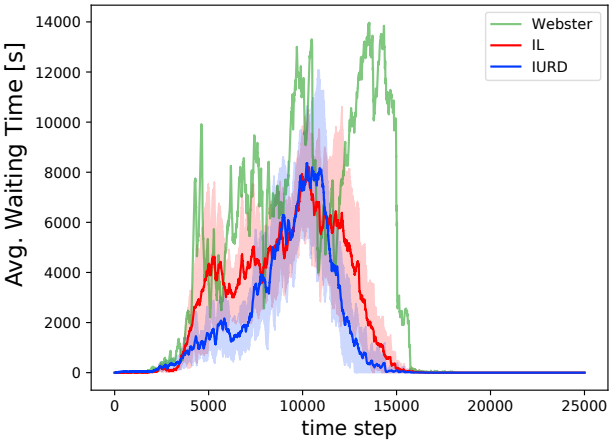


Fig. 4: Average waiting time at intersections.

3. time to process the whole demand, i.e., time when the last vehicle completed its trip
4. trip average waiting time (output by SUMO): time in which a vehicle has speed below 0.1m/s; this measure is an average over all vehicles
5. trip average time loss: time lost due to driving below the ideal speed (which is estimated by SUMO's car model)
6. average depart delay: time vehicles had to wait before they could start their journeys due to congestion

As seen in Table 2, the IURD method outperforms the others. Agents receive more reward; it takes less time for vehicles to complete their trips; there is much lower delay in the departures when IURD is used; it has a lower trip waiting time and time loss. The ANOVA and Tukey pairwise statistical tests confirm that these values are statistically different (95% confidence).

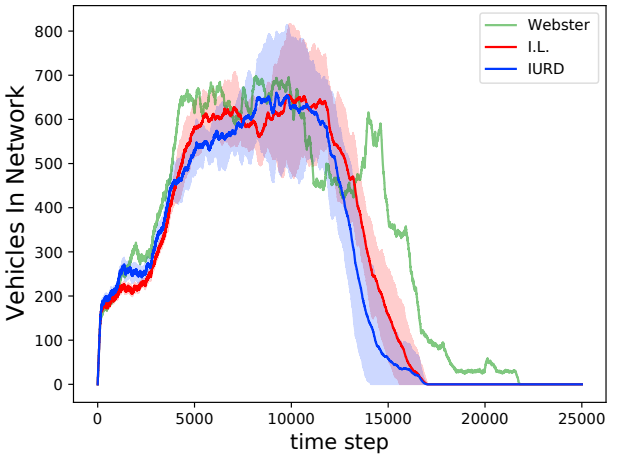
The performance related to the measures in Table 2 are also shown in aggregated ways in Figures 4 to 6. Figure 4 shows the average waiting time, which are influenced by items 1, 4, and 5 in that table. Figure 5 and Figure 6 depict the number of vehicle in the network and trips completed per hour respectively. Both relate to itens 2 and 3 in the table. The lines are the averages of the 10 simulations for each method and the shadows represent their standard deviations.

Recall that we insert vehicles in each origin for the first 10k steps (roughly 3 hours).

Regarding the average waiting time plot (Figure 4), we observed that the waiting times using IURD are lower than the other methods (roughly during all the time steps).

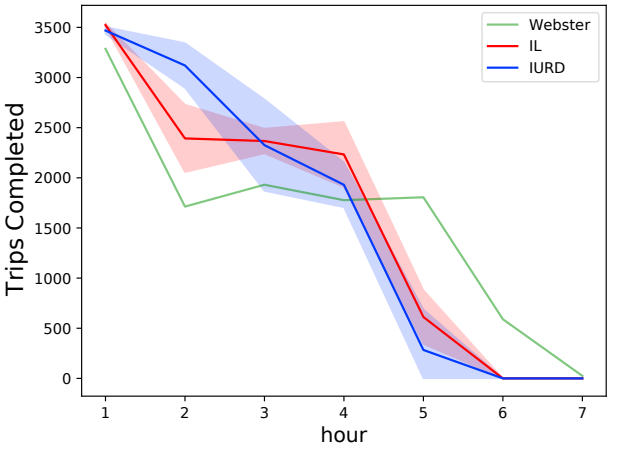
The number of vehicles in the simulation is associated with the depart delay because SUMO retains vehicles in their origins if there is congestion in the respective lane. Figure 5 shows that IURD and also IL deal with the vehicle demand faster than the Webster-based method.

Figure 6 refers to the number of trips completed per hour. We can see that the Webster-based method is more efficient in the beginning as the agents do not need to explore given that they use a fixed strategy. However, after some



R

Fig. 5: Number of vehicles in the simulation.



R

Fig. 6: Trips completed per hour.

time IURD shows the best results: as it avoids retention of vehicles, the trips take less time. Using IURD, by the fifth hour most of the trips are already completed.

5. Conclusion

This paper proposes the Information Upwards, Recommendation Downwards (IURD) framework for organizing RL in a hierarchical manner. It is applied for controlling traffic signal agents. In this domain, we discuss an instance of IURD, namely the Vector-based Hierarchical Organization (VHO). VHO uses vectors to aggregate the information on traffic flow. Such aggregated information is then used by supervisor agents to learn how to recommend actions to agents they supervise.

The experiments in a synthetic 4x4 grid showed that the proposed method outperformed two other traffic signal control methods, namely, one based on fixed time and an RL method without hierarchical organization.

Future investigations include experimentation with different incentive functions, such as punishing agents for not following supervisors' recommendations, or more complex forms of incentives for following recommendations. Also, the way that the regions are set play a role in the performance. This needs to be further investigated, as for instance by forming groups on the fly as proposed in [11].

Acknowledgments

Ana Bazzan is partially supported by CNPq (grant 307215/2017-2). This work was partially funded by CNPq and by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil) - Finance Code 001, and also partially sponsored by the German Federal Ministry of Education and Research (BMBF), Käte Hamburger Kolleg Cultures des Forschens/ Cultures of Research.

References

[1] Abdoos, M., Bazzan, A.L., 2021. Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. Expert Systems with Applications , 114580doi:https://doi.org/10.1016/j.eswa.2021.114580.

[2] Abdoos, M., Mozayani, N., Bazzan, A.L., 2013. Holonic multi-agent system for traffic signals control. Engineering Applications of Artificial Intelligence 26, 1575–1587. doi:10.1016/j.engappai.2013.01.007.

[3] Abdoos, M., Mozayani, N., Bazzan, A.L., 2014. Hierarchical control of traffic signals using Q-learning with tile coding. Appl. Intell. 40, 201–213. doi:10.1007/s10489-013-0455-3.

[4] Alegre, L.N., 2019. SUMO-RL. https://github.com/LucasAlegre/sumo-rl.

[5] Bazzan, A.L.C., de Oliveira, D., da Silva, B.C., 2010. Learning in groups of traffic signals. Eng. Applications of Art. Intelligence 23, 560–568. URL: http://www.sciencedirect.com/science/article/pii/S0952197609001699.

[6] Choy, M., Srinivasan, D., Cheu, R., 2003. Cooperative, hybrid agent architecture for real-time traffic signal control. IEEE Transaction on Systems, Man and Cybernetics- Part 1: Systems and Humans 33, 597–607.

[7] Chu, T., Wang, J., Codecà, L., Li, Z., 2019. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems 21, 1086–1095. doi:10.1109/TITS.2019.2901791.

[8] Dayan, P., Hinton, G.E., 1993. Feudal reinforcement learning, in: Hanson, S.J., Cowan, J.D., Giles, C.L. (Eds.), Advances in Neural Information Processing Systems 5. Morgan-Kaufmann, pp. 271–278.

[9] Dietterich, T.G., 1999. Hierarchical reinforcement learning with the MAXQ value function decomposition. arXiv:cs/9905014.

[10] Gerber, C., Siekmann, J., Vierke, G., 1999. Holonic Multi-Agent Systems. Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI.

[11] Labres, J.V.B., Bazzan, A.L.C., Abdoos, M., 2021. Improving traffic signal control with joint-action reinforcement learning, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01–08. doi:10.1109/SSCI50451.2021.9659871.

[12] Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo, in: The 21st IEEE International Conference on Intelligent Transportation Systems.

[13] Ma, J., Wu, F., 2020. Feudal multi-agent deep reinforcement learning for traffic signal control, in: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Auckland, New Zealand. pp. 816–824.

[14] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Far, B., Abad, Z.S.H., Bazzan, A.L.C., 2021. Reinforcement learning in urban network traffic signal control: A systematic literature review. URL: engrxiv.org/ewxrj, doi:10.31224/osf.io/ewxrj.

[15] Wei, H., Zheng, G., Gayah, V.V., Li, Z., 2019. A survey on traffic signal control methods. arXiv:1904.08117.

[16] Yau, K.L.A., Qadir, J., Khoo, H.L., Ling, M.H., Komisarczuk, P., 2017. A survey on reinforcement learning models and algorithms for traffic signal control. ACM Comput. Surv. 50. doi:10.1145/3068287.