Growing Neural Gas in Multi-Agent Reinforcement Learning Adaptive Traffic Signal Control

Mladen Miletić[1,*], Ivana Dusparić[2] and Edouard Ivanjko[1]

[1] *University of Zagreb, Faculty of Transport and Traffic Sciences, Vukelićeva Street 4, 10000, Zagreb, Croatia*
[2] *Trinity College Dublin, School of Computer Science and Statistics, College Green, Dublin 2, Ireland*

Abstract

In recent years, there has been a significant increase in research interest in applying Reinforcement Learning (RL) to Adaptive Traffic Signal Control (ATSC). Urban traffic networks present a suitable environment for Multi-Agent (MA) ATSC systems, as each intersection can be managed by a single RL agent. However, the non-stationarity of the ATSC environment in Multi-Agent Reinforcement Learning (MARL) poses a challenge since the actions of one agent can directly affect the performance of its neighboring agents. To address this issue, this paper presents and compares several MARL ATSC approaches utilizing Growing Neural Gas (GNG) for state identification, implemented using a microscopic traffic simulator with a synthetic traffic model of nine intersections. This paper explores the effectiveness of various MARL ATSC approaches, including fully independent agents and those augmented with reward and state-sharing mechanisms. The results demonstrate that fully independent agents can enhance global traffic performance by optimizing local decisions. Furthermore, when agents share rewards and states, they achieve additional improvements in both local and global traffic conditions by fostering cooperative behavior and mitigating the impact of non-stationarity. In addition, this paper identifies the approach of centralized state identification with GNG, coupled with decentralized agent execution, as the most effective ATSC strategy. This configuration leverages the strengths of centralized data processing for accurate state representation while maintaining the flexibility and scalability of decentralized agent operation. Overall, the findings highlight the potential of GNG-based state identification in enhancing the performance of MARL ATSC systems.

Keywords

Growing Neural Gas, Reinforcement Learning, Adaptive Traffic Signal Control, Multi-Agent Systems

1. Introduction

The rising world population and the increasing urbanization trends have a direct impact on traffic in cities. It is estimated that further population growth will be mostly concentrated in the cities further increasing the mobility demands [1]. The recent COVID-19 pandemic caused a short-term shift in mobility demands and the modal split of daily commutes with more people being able to work from home and use online shopping services. However, initial analysis shows that this was only temporary as people are slowly returning to pre-pandemic behaviour [2]. Combined with the growing population the urban traffic system will be unable to handle the increase in mobility demand. Traditional solutions such as building additional infrastructure were proven to be ineffective in the long term as increased capacity attracted additional demand in an effect called Braess's paradox [3]. In addition, most older cities are not able to build any additional infrastructure due to a lack of available building space. A possible solution is seen in the introduction of Intelligent Transportation Systems (ITS) to improve the operational capacity of the existing transportation infrastructure [4].

In most urban areas, road traffic is primarily controlled by Traffic Signal Control (TSC) on intersections. While the primary task of TSC is to allow safe passage of vehicles entering the intersections it has a large effect on the traffic flow since opposing flows need to be temporarily stopped while waiting for their signal phase. For this reason, the intersections controlled by TSC are the primary bottlenecks in

ATT'24: Workshop Agents in Traffic and Transportation, October 19, 2024, Santiago de Compostela, Spain

[*] Corresponding author.

mladen.miletic@fpz.unizg.hr (M. Miletić); ivana.dusparic@tcd.ie (I. Dusparić); edouard.ivanjko@fpz.unizg.hr (E. Ivanjko)
https://fpz.unizg.hr/mmiletic (M. Miletić)
0000-0003-4437-4525 (M. Miletić); 0000-0003-0621-5400 (I. Dusparić); 0000-0002-2083-0672 (E. Ivanjko)

urban traffic networks. This negative effect is most noticeable with improperly adjusted controllers leading to even more congestion. TSC can operate with one of the three main control strategies [5]: (I) Fixed Time Signal Control (FTSC); (II) Traffic Actuated Signal Control (TASC); and (III) Adaptive Traffic Signal Control (ATSC).

The most commonly used TSC strategy is FTSC which uses pre-determined signal programs adjusted using historical traffic data. FTSC systems have low initial costs but are difficult to update and are not able to respond to changes in demand. TASC systems are an extension of FTSC that allows the signal program to change upon vehicle detection. TASC systems perform well in low-demand scenarios, but in high-demand scenarios, their operations are similar to FTSC. The most advanced type of TSC is ATSC which uses real-time traffic data to adapt signal programs to satisfy its operational objective. ATSC systems can operate on a network level with multiple intersections in the control loop. Many commercial ATSC systems are available such as SCOOT [6], SCATS [7], UTOPIA [8] and ImFlow [9]. Such ATSC systems can improve the traffic flow significantly but still require manual adjustment and fine-tuning to achieve good results. To overcome this problem, current state-of-the-art proposed ATSC approaches are based on Reinforcement Learning (RL) [5].

RL is a subset of algorithms and techniques of Machine Learning that focuses on learning by direct interaction of the controller usually called an agent with its environment [10]. The agent learns its control policy by interacting with the environment and receiving feedback on how successful its actions were. By applying the concept of RL to ATSC the traffic signal controller becomes an agent and its environment is the traffic network in which it operates. This approach removes the need for manual adjustment of ATSC parameters as now the agent can learn the control policy on its own. Another benefit of RL is that it can continuously learn even after deployment, meaning that if the traffic behavior changes the ATSC agent will be able to adapt its control policy. A common problem in RL is the identification of the environment state in high dimensional state space, especially for tabular RL algorithms such as Q-Learning. To overcome this problem a dimensionality reduction technique of Growing Neural Gas (GNG) can be used for state identification. This approach was successfully implemented in a single intersection traffic environment but no GNG-based controllers have been implemented on multiple intersections [11]. To allow scaling of ATSC to multiple intersections, the RL control is expanded to Multi-Agent Reinforcement Learning (MARL) systems by allowing each agent to control a single intersection. The use of MARL systems brings forth new challenges in ATSC since each agent operates independently but their selected actions influence neighboring agents [12]. For this reason, MARL agents are required to learn how to optimize local actions and how their actions influence neighboring agents.

This paper focuses on the evaluation of 3 different families of MARL ATSC approaches that use GNG for state identification to reduce the state-action complexity and help with the problem of non-stationarity. Following this introduction, the rest of the paper is organized as follows. Section 2 provides insight into previous and related research in MARL-based ATSC systems highlighting the contributions of this paper. Section 3 explains the relevant background and details on the use of RL and GNG in traffic environments. Section 4 expands the currently known MARL-based controllers by introducing GNG as a state identifier in RL. In section 5 the simulation environment and scenarios tested are explained in detail. Section 6 provides an overview and discussion of the obtained results, highlighting the most important observations. The paper ends with a conclusion section which includes a commentary on future work.

2. Related Work

The field of RL application in ATSC systems has attracted many researchers since traffic control problems provide a good environment for RL integration [13]. Older works focus on modeling ATSC controllers as a Markov Decision Process (MDP) by specifying the observable state space, actions an agent can take, and the reward for measuring the performance of the selected operational objective. In [14], authors test how different reward definitions affect the performance of RL-ATSC. They tested rewards defined by queue lengths, cumulative delay, and throughput. An edge was given to cumulative delay with a comment on the difficulties in its measurement in real-world scenarios. The authors also conclude that the performance of the reward function depends on the total traffic volume. Early forms of MARL-ATSC approaches are shown in papers [15, 12]. The former used independent agents, while the latter attempted to establish cooperation between agents to achieve even better performance. In [16], authors combine the idea of ATSC with the concept of Connected Vehicles (CVs) in a cooperative Multi-Agent (MA) environment.

It is evident from reviews [5, 13, 17] on RL-ATSC that a more modern approach is to solve the control problem by using Deep Reinforcement Learning (DRL). Papers [18, 19] both show significant traffic performance benefits from using DRL-ATSC. It is also shown, that traffic networks with a high number of intersections can be controlled with DRL-ATSC with noticeable performance benefits. The former also introduces two approaches to improve MA convergence stability by improving agent observability and introducing a discount factor that reduces the impact of states and rewards from other agents.

Considering the review of related works above, the most common problems identified in RL-ATSC applications are: (I) State-Action complexity; (II) Reward definition; and (III) Non-stationarity of the environment when applied in MARL configuration. The problem of high state-action complexity usually arises because the state space is continuous. To allow RL the state space representation needs to be created beforehand which can be problematic if there is no available data before learning. Ideally, the state representation would be built during the learning process as new previously unknown states might be encountered. For this reason, methods such as Growing Neural Gas (GNG) can be used to construct the space representation online. The growing characteristics of the GNG allow the network to adapt to newly encountered spaces without degrading previous knowledge [20, 21]. Previous works from the authors of this paper attempted to solve the problem of high State-Action complexity in traffic control by introducing Self Organizing Maps (SOM) [22] and GNG [11, 23] for state identification to offer an alternative to DRL approaches while maintaining a good convergence rate. Both approaches were successfully implemented on a single intersection, with both papers stating that future work should include scaling to multiple intersection networks. This paper answers the research gap identified in previous works by scaling the GNG-RL controller to multiple intersections in a MARL configuration. Hence, the main contributions of this paper are as follows:

- Scaling of the GNG-based RL-ATSC controller to GNG-based MARL-ATSC controller;
- Performance comparison of 15 GNG-based MARL-ATSC approaches.

3. Relevant background

In this section, the relevant background on RL and GNG for the purpose of traffic state identification is presented.

3.1. Reinforcement Learning

When applying RL algorithms for ATSC, in literature, the ATSC controller is usually modeled as an MDP tuple $\langle S, A, T, R, \gamma \rangle$. Here S is the set of environment states; A is the set of actions available to the controller; T is the transition probability of reaching state s' from state s after an action $a \in A$ is performed; R is the reward received from the environment after an action is performed; and $\gamma \in [0, 1)$ is the discount factor which determines the impact of future rewards [24]. Alternatively, ATSC can be modeled as a Partially Observable Markov Decision Process (POMDP) with a tuple $\langle S, A, T, R, \Omega, O, \gamma \rangle$, where the MDP is extended with Ω which is a set of observations, and O which is the conditional probability of those observations [25]. The POMDP model is more appropriate for ATSC since the actual traffic state is unknown to the controller and is instead represented by a vector of observations or measurements from the environment. However, the MDP model is usually used instead of POMDP to maintain simplicity in algorithm applications.

Once the ATSC controller is defined as an MDP the learning task is to find the optimal control policy to maximize the cumulative discounted reward. A common algorithm used to achieve this is Q-Learning based on Bellman's optimality equation [10]. The algorithm reaches the optimal control policy using the Q-value update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)), \tag{1}$$

where $Q(s_t, a_t)$ is the expected return of taking action $a \in A$ in state $s \in S$ in time step t, and α is the learning rate coefficient. After multiple iterations of each state-action combination, an optimal policy is formed by selecting actions with maximum Q-value for a given state. The main benefit of using the Q-Learning algorithm for ATSC is that it is a model-free approach that does not use state transition probabilities T which are difficult to model in traffic control.

3.2. Growing Neural Gas

The GNG [26] is a dimensionality reduction technique created as an extension to SOM which can be considered as a special type of neural network that uses competitive learning. The SOM or GNG network consists of a set of connected neurons sometimes called nodes. Each neuron is defined by a weight vector which is the representation of the neuron position in the input space. While training, the input signal is received, and the neuron closest to the input will become known as the Best Matching Unit (BMU). The BMU and neurons connected to it will then adjust their weight vectors to more closely match the received input using the equation:

$$W_i(k+1) = W_i(k) + \Theta(i, BMU, k)\alpha_{net}\big[X(k) - W_i(k)\big], \tag{2}$$

where $X(k)$ is the input signal vector in time step k; $W_i(k)$ is the weight vector of neuron i; $\Theta(i, BMU, k)$ is the neighborhood function which scales the weight update using the distance between neuron i and the BMU; and α_{net} is the learning rate of the network. The neighborhood function Θ is usually modeled with a Gaussian or a Ricker Wavelet function [27]. After training the network becomes a map of connected neurons with low dimensionality that preserves the topology of the original input data samples.

Unlike SOM, which has a fixed arrangement of neurons, the neurons and their connections in GNG can be added or removed during the learning process [28]. The primary benefit of the growing structure of GNG is that there is no need to specify the number of neurons in advance, but a limit on the total number can be imposed. A new neuron will be added to the GNG if the input signal has a large Euclidean distance from the current BMU. If the distance of the input signal is not as large but still significant a new neuron will be added with an edge forming to the current BMU. If the input signal is close to the current BMU, the existing BMU will be used instead of adding new neurons, and an edge will be formed between the BMU and the neuron which is the second closest to the input signal. By specifying neuron addition and connection distances as a hyperparameter of GNG, it is possible to set a desired level of detail a created topological map will be able to accommodate. As the network grows some neurons might become redundant or inert and will be then removed from the network to remain computationally efficient.

3.3. Traffic state identification

As discussed above, the observation space Ω for ATSC is defined by available traffic observation variables. As the number of variables increases the ATSC agent can have better insight into the actual environment state $s \in S$. However, the downside of adding more variables is the curse of dimensionality which can hinder the learning performance. In addition, most traffic observation variables are continuous, creating an n-dimensional continuous observation space, where n is the number of variables. To successfully apply tabular RL techniques, such as Q-Learning, in continuous space it is necessary to perform generalization from states that were previously experienced to the ones that have not

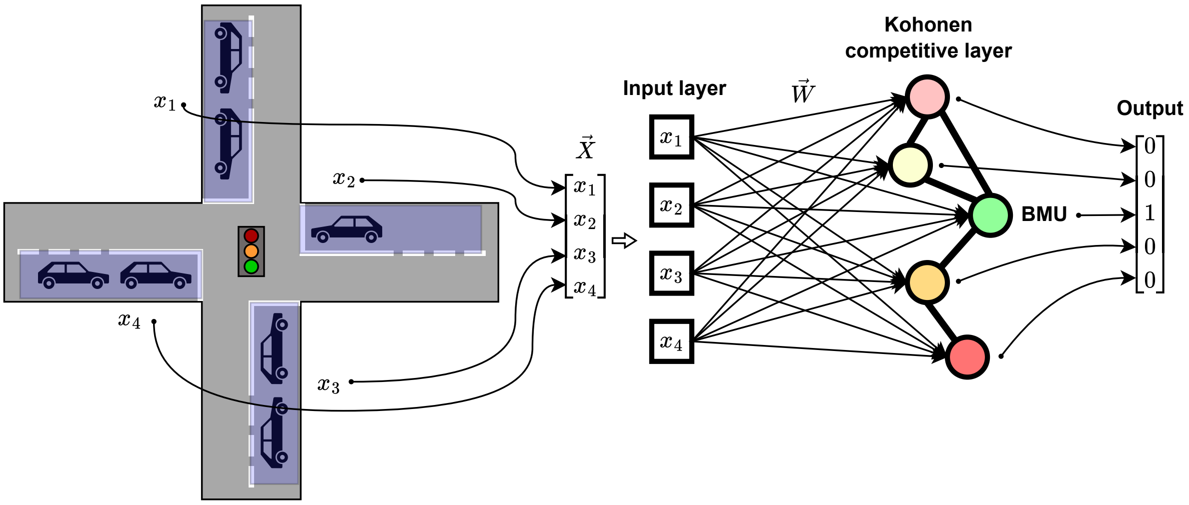


Figure 1: GNG-based traffic state identification

been [10]. A simple approach would be to split the state space into a number of finite and discrete segments, and then perform a generalization on those segments. It is difficult to efficiently define the state segments, especially if the distribution of states is unknown beforehand. To overcome this problem, the generalization can be done using GNG [11].

To use GNG for state space generalization, the length of the weights vector in each neuron must be the same as the number of traffic observation variables. Each visited state is sent to the GNG which will map the received state to the BMU of the trained GNG. This essentially means that the weights of neurons in GNG become origin points from which Voronoi polytopes are constructed. Each constructed polytope is a subspace of Ω that generalizes to one discrete state. The number of generalized states is equal to the number of neurons in the network. The entire process of using GNG for traffic state identification is shown in Figure 1. The growing properties of GNG allow for dynamic scaling of the number of identified states as the neuron map is constructed. A major benefit is that the GNG state map can be constructed simultaneously as the RL algorithm updates the control policy. Initially, the GNG will consist of a small number of neurons and a highly generalized state space, but this will be reduced as new neurons are added in later stages of learning.

4. Multi-Agent Reinforcement Learning Adaptive Traffic Signal Control

When extending the ATSC to multiple intersections, the control can be configured in a local or global manner. Localized control refers to each agent making a decision based solely on data gathered locally from the intersection it controls. Both the state space and reward from the environment are received locally, allowing the agent to adjust its control policy in a way that would maximize local traffic performance. This approach can also be considered a non-cooperative game in game theory, with each agent acting as a player. Localized control allows for good scalability and is simple to implement. It is also known to perform well in traffic networks with low demand, or in networks consisting of intersections located very far from one another. However, a significant problem of localized control is that it can lead to sub-optimal network-wide performance due to the actions of an agent on one intersection impacting the state and reward of its neighbors without any regard for the broader traffic situation. This problem can be partially overcome by introducing cooperation between agents. The simplest way to achieve cooperation is to modify the reward function to also include global performance metrics. However, the convergence in such an approach would be problematic as the agent will not be able to properly predict the received reward by only observing the local state. Further cooperative augmentation is the sharing of state variables between neighboring agents. This allows the agent a small level of insight into broader traffic control at the expense of slightly increasing state complexity, but remaining scalable to a high number of intersections.

The global control concept uses a single or coordinated group of agents to perform TSC using data gathered from the entire traffic network. This approach allows the agent to build their control policy for maximization of global traffic performance, even when local intersection performance would be lower. Global control can in theory achieve optimal performance, but it suffers from poor scalability as the state-action complexity increases exponentially as additional intersections are added to the control loop. In addition, it is computationally expensive and would require a robust communication system to be implemented.

While global control would be ideal, it is usually not feasible. For this reason, the proposal of this paper is to balance between global and local control by using individual agents that receive local rewards but have insight into the state of the entire network. This allows the agent to build a control policy that is finely tuned to the entire network's state but still uses local reward optimization. This approach can further be augmented by modifying the reward function of each agent to include a performance measure of neighboring agents giving them the incentive to select action that will not have a negative impact on the neighboring intersections' performance. In this type of control, the state space remains large and difficult to separate into discrete segments. To overcome this problem the GNG presented in section 3 can be used to reduce the dimensionality of the problem and identify the global state of the network keeping the state-action complexity low even when the number of intersections in the traffic network increases. The additional benefit of using GNG as a state identifier is that it will be able to operate even if some sensors go offline by using the value of zero for state variables with offline sensors.

5. Experimental Setup

In this section, all analyzed TSC approaches are presented with their hyperparameters and an explanation of the simulation environment used to implement each TSC approach.

5.1. Simulation environment

To evaluate the performance of various TSC approaches, the simulation environment consisting of a synthetic traffic model in microscopic traffic simulator PTV VISSIM is used. PTV VISSIM is a state-of-the-art microscopic traffic simulator capable of multi-modal traffic simulation [29]. A major benefit of PTV VISSIM is the developed COM interface which allows for external control of objects in VISSIM. This interface can be used to develop external traffic signal controllers, while VISSIM handles the simulation of vehicle behavior. The traffic model used for the analysis consists of 9 connected intersections arranged in a 3 by 3 grid as shown in Figure 2. There is a total of 12 vehicle inputs labeled a_1 to a_{12} with varying traffic demand lasting for a total of 16 simulation hours. Traffic demands are synthetically generated to be variable from The changes in traffic demand are shown in Figure 3. Each intersection is operated by a TSC in an FTSC regime with a cycle length of 60 seconds and an equal distribution of green times for both phases. This cycle length was selected to allow adequate traffic flow through the intersection according to the defined traffic demand. This FTSC regime will be used as a baseline TSC for comparison with GNG-based ATSC approaches.

5.2. Analyzed approaches and hyperparameters

A total of 3 ATSC approaches are presented for analysis and comparison to the FTSC Baseline. They are split into three main groups: Independent agents; State Sharing (SS); and Centralized State with Decentralized Agents (CSDA). In addition, each approach is augmented with four levels of varying Reward Sharing (RS).

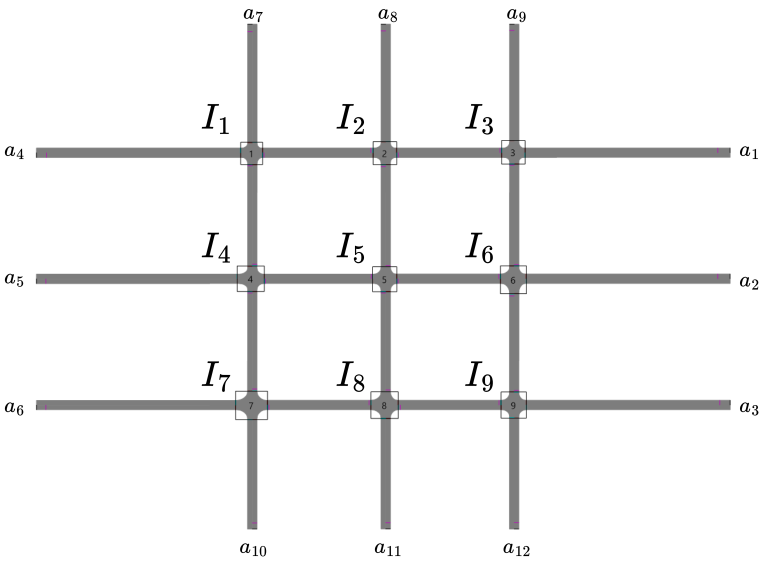


Figure 2: Traffic network of 9 connected intersections modelled in PTV VISSIM

5.2.1. Independent agents

The approach with Independent agents uses an RL agent at each intersection in the network. Each agent is only capable of observing its immediate surroundings or more precisely the queue lengths on each intersection approach. The average values of queue lengths are calculated for the previous time step by the VISSIM simulator. Those queue values are then given as input to the GNG of the agent. Each agent uses its own GNG with the following equations used to describe the parameters:

$$\alpha_{net}(k) = 0.95 * 0.9^{k-1} + 0.05, \tag{3}$$

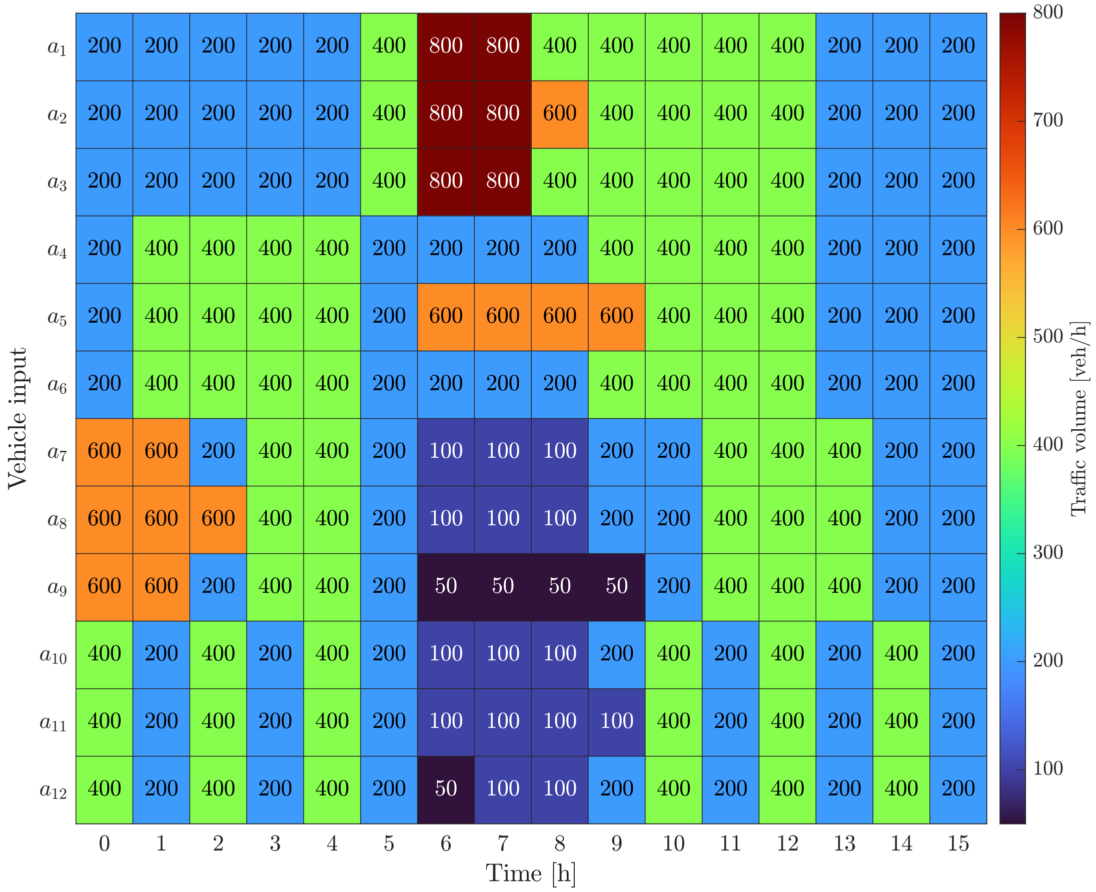
$$\Theta(k) = \exp\left(\frac{-d^2}{10 * 0.95^k}\right), \tag{4}$$

where k is the current training episode, and d is the Euclidean distance between the BMU and the input signal. The GNG will add a new isolated neuron to the network if the distance from the BMU to the input signal is higher than 20, if the distance is higher than 10 a neuron will be added and connected to the previously identified BMU. The maximum number of neurons in the network is limited to 150.

With GNG used as a state identifier, the RL component of the agent will use Q-Learning with hyperparameters $\alpha = 0.1$ and $\gamma = 0.8$. The action selection will use the ϵ-greedy policy with ϵ determined by Equation 5 to balance between exploration and exploitation of knowledge. The reward function will be defined as the reduction in lost time LT of vehicles passing through the intersection as shown in Equation 6. Lost time is a measure of how much time a vehicle lost because its speed was lower than desired because of its interaction with the environment. Each agent can choose from a set of five actions $A \in \{-10, -5, 0, 5, 10\}$ each representing a time change of the green split in the default signal program. The agent will take an action every $\Delta t = 300$ simulation seconds. This value was selected according to previous research as it allows for enough time for the new signal program to have an impact. An additional benefit is that the Δt value is a multiple of the cycle length, allowing for easier analysis.

$$\epsilon(k) = 0.95 * 0.9^{k-1} + 0.05 \tag{5}$$

$$r(k+1) = LT(k) - LT(k+1) \tag{6}$$



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a_1	200	200	200	200	200	400	800	800	400	400	400	400	400	200	200	200
a_2	200	200	200	200	200	400	800	800	600	400	400	400	400	200	200	200
a_3	200	200	200	200	200	400	800	800	400	400	400	400	400	200	200	200
a_4	200	400	400	400	400	200	200	200	200	400	400	400	400	200	200	200
a_5	200	400	400	400	400	200	600	600	600	600	400	400	400	200	200	200
a_6	200	400	400	400	400	200	200	200	200	400	400	400	400	200	200	200
a_7	600	600	200	400	400	200	100	100	100	200	200	400	400	400	200	200
a_8	600	600	600	400	400	200	100	100	100	200	200	400	400	400	200	200
a_9	600	600	200	400	400	200	50	50	50	50	200	400	400	400	200	200
a_{10}	400	200	400	200	400	200	100	100	100	200	400	200	400	200	400	200
a_{11}	400	200	400	200	400	200	100	100	100	100	400	200	400	200	400	200
a_{12}	400	200	400	200	400	200	50	100	100	200	400	200	400	200	400	200

Figure 3: Traffic demand volume for each network input

5.2.2. Reward sharing

Since the actions of one agent can affect the performance of neighboring agents the first level of cooperation is by introducing RS with a reward cooperation coefficient β. The main principle is that each agent reward is expanded by a partial reward from its neighbors as shown in the equation:

$$r(k+1) = LT(k) - LT(k+1) + \beta \left(\frac{1}{n} \sum_{m=1}^{n} LT_m(k) - LT(k+1)_m \right), \tag{7}$$

where n is the number of neighboring intersections, and LT_m is the lost time of vehicles on neighboring intersection m. Four different values of $\beta \in \{0.25, 0.5, 0.75, 1.00\}$ will be analyzed in combination with each TSC approach.

5.2.3. State sharing

The GNG of independent agents uses only queue lengths at their local intersections to determine the current state. In some traffic situations, an agent could be unaware of the increase in upcoming traffic demand from neighboring intersections. This makes it difficult for the agent to predict the future state and to select actions that would not help it to prepare for the upcoming traffic in advance. By incorporating simple SS each queue length value on the intersection has the added value of the queue length from the upstream intersection if it exists. The complexity of the GNG in this approach does not change, but the state identification includes some data from neighboring intersections. To implement this approach in realistic scenarios it would be imperative to know the desired direction of all vehicles in the queue to properly scale the number of vehicles that will move downstream.

5.2.4. Centralized state with decentralized agents

The primary reason for the use of MA systems in TSC for large networks is the problem of scalability since the state-action space required for a single agent would grow exponentially with the increasing number of state-defining variables. The GNG enables a high level of dimensionality reduction and is capable of identifying the state of the entire network. The state-action complexity still remains high but can be reduced by the use of decentralized agents at each intersection. In this CSDA approach, each agent receives the same state ID from the centralized GNG but builds its control policy by receiving local rewards. The GNG in this approach has 36 total inputs but all its hyperparameters and functions are kept the same as in the Individual agents approach.

6. Results and Discussion

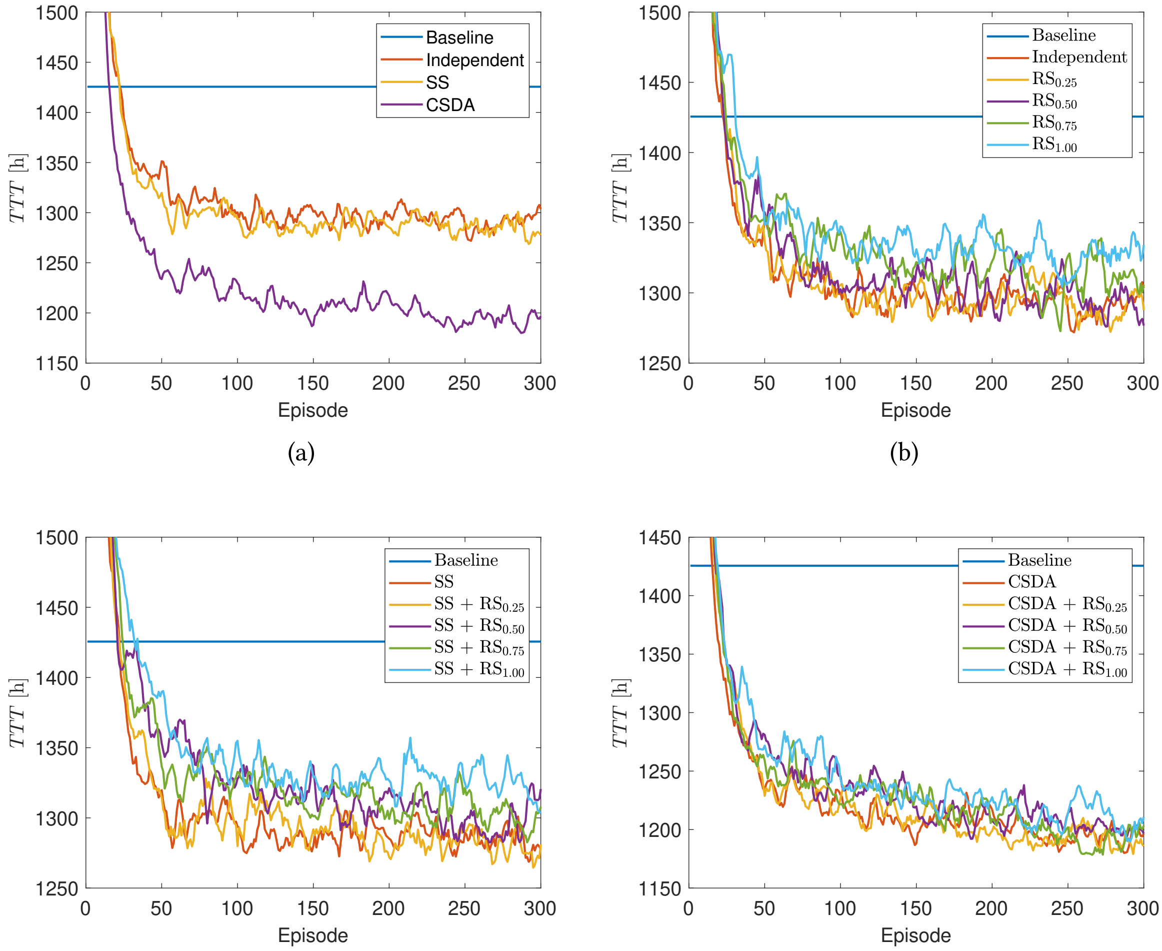
In this section, the obtained results for each TSC approach are presented and discussed.

6.1. Obtained results

The following network-level Measures of Effectiveness (MoEs) were used to evaluate the performance of each TSC approach: TTT as the Total Travel Time of all vehicles in the network; NS_{Tot} as the Total Number of Stops of all vehicles; and LT_{Avg} as the average lost time per vehicle. In addition, an analysis of total intersection-level lost time LT_{Tot} was performed to evaluate the local effects of each proposed approach since the reward definition in each analyzed TSC approach was modeled to reduce the lost time. For each MoE, the mean value for the last 50 episodes was calculated and used for the comparison. The obtained results for TTT are shown in Figure 4 with a moving average with a window of 5 to allow for easier inspection. The calculated mean results of LT_{Tot} for each intersection and approach are shown in Figure 5. Detailed numerical results for network-level MoEs are shown in Table 1 with the calculated improvement compared to the baseline scenario. In addition, the standard deviation σ for the last 50 episodes is calculated to evaluate the convergence stability. The best-performing approach is bolded in the table for easier identification.

Table 1
Mean network results for each TSC approach from episode 251 to 300

TSC approach	**MoE**								
	TTT [h]			NS_{Tot}			LT_{Avg} [s]		
	Value	[%]	σ	Value	[%]	σ	Value	[%]	σ
Baseline	1425.59	-	-	135694	-	-	58.58	-	-
Independent	1289.32	-9.56	15.68	119229	-12.13	2493	50.29	-14.15	0.95
RS$_{0.25}$	1286.33	-9.77	18.27	118596	-12.60	2917	50.11	-14.46	1.11
RS$_{0.50}$	1292.09	-9.36	17.32	119107	-12.22	2120	50.46	-13.86	1.05
RS$_{0.75}$	1312.15	-7.96	24.11	122087	-10.03	3525	51.68	-11.78	1.47
RS$_{1.00}$	1327.56	-6.88	21.06	124160	-8.50	3208	52.62	-10.18	1.28
SS	1285.00	-9.86	18.69	118392	-12.75	2921	50.03	-14.60	1.14
SS + RS$_{0.25}$	1280.26	-10.19	16.27	118323	-12.80	2788	49.74	-15.09	0.99
SS + RS$_{0.50}$	1297.43	-8.99	20.23	120075	-11.51	3115	50.79	-13.31	1.23
SS + RS$_{0.75}$	1301.23	-8.72	15.68	120901	-10.90	2338	51.02	-12.91	0.95
SS + RS$_{1.00}$	1325.28	-7.04	16.88	123514	-8.98	2878	52.48	-10.41	1.03
CSDA	1194.04	-16.24	14.08	105482	-22.26	2177	44.49	-24.05	0.86
CSDA + RS$_{0.25}$	1193.84	-16.26	16.28	105804	-22.03	2473	44.48	-24.07	0.99
CSDA + RS$_{0.50}$	1202.02	-15.68	12.74	107144	-21.04	1876	44.98	-23.22	0.78
CSDA + RS$_{0.75}$	**1190.25**	**-16.51**	**16.54**	**105332**	**-22.38**	**2611**	**44.26**	**-24.44**	**1.01**
CSDA + RS$_{1.00}$	1213.91	-14.85	18.86	108482	-20.05	2754	45.70	-21.99	1.15



(c)

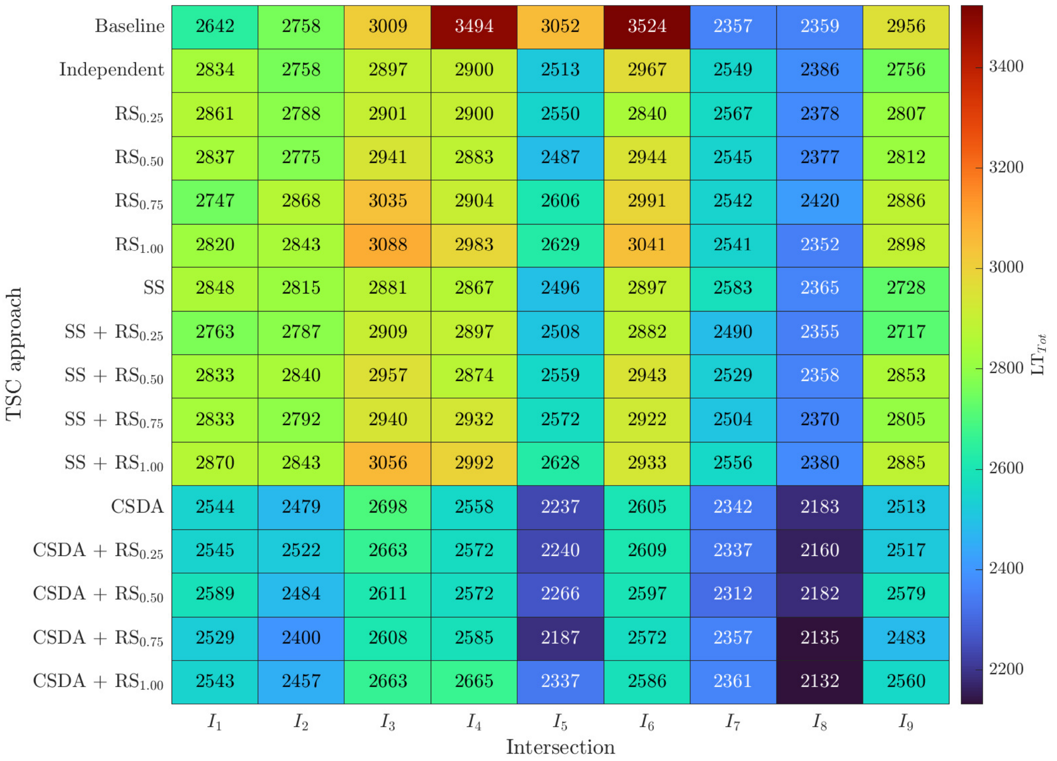
(d)

Figure 4: Moving average of the results of TTT for each analyzed TSC approach with window 5: **(a)** Comparison of TTT for each TSC approach without reward sharing; **(b)** Comparison of TTT for agents with varying levels of reward sharing; **(c)** Comparison of TTT for agents with state sharing and varying levels of reward sharing; **(d)** Comparison of TTT with centralized state identification and distributed agents and varying levels of reward sharing

6.2. Discussion

From the results presented in Table 1, it is evident that each analyzed TSC approach is capable of improving the traffic network performance when compared to the baseline controller which uses FTSC. This behavior is expected as any form of ATSC will be able to somewhat accommodate changing traffic demand. The Independent agents' approach reduced TTT by 9.56% on average by significantly improving the performance of the central east-west corridor with intersections I_4, I_5, I_6 as can be seen from LT_{Tot} in Figure 5. The performance of intersections I_1 and I_7 decreased somewhat because of the interaction with the agent on intersection I_4. This behavior is common in MA systems and is referred to as the problem of non-stationarity. The agents on I_1 and I_7 had no insight into the state of the central corridor and could not prepare actions to accommodate the sudden increase in traffic demand from right and left turn movements on I_4 resulting from increased throughput on I_4.

By expanding the Independent agents' control with varying levels of reward sharing from neighboring intersections a slight improvement in performance can be observed when the reward cooperation coefficient was set to 0.25. For higher values of the cooperation coefficient the performance slightly decreased. The use of reward sharing in TSC gives an incentive to the agents not to choose actions that will have a negative effect on neighboring intersections. Agents will instead tolerate lower local



TSC approach	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9
Baseline	2642	2758	3009	3494	3052	3524	2357	2359	2956
Independent	2834	2758	2897	2900	2513	2967	2549	2386	2756
$\mathrm{RS}_{0.25}$	2861	2788	2901	2900	2550	2840	2567	2378	2807
$\mathrm{RS}_{0.50}$	2837	2775	2941	2883	2487	2944	2545	2377	2812
$\mathrm{RS}_{0.75}$	2747	2868	3035	2904	2606	2991	2542	2420	2886
$\mathrm{RS}_{1.00}$	2820	2843	3088	2983	2629	3041	2541	2352	2898
SS	2848	2815	2881	2867	2496	2897	2583	2365	2728
SS + $\mathrm{RS}_{0.25}$	2763	2787	2909	2897	2508	2882	2490	2355	2717
SS + $\mathrm{RS}_{0.50}$	2833	2840	2957	2874	2559	2943	2529	2358	2853
SS + $\mathrm{RS}_{0.75}$	2833	2792	2940	2932	2572	2922	2504	2370	2805
SS + $\mathrm{RS}_{1.00}$	2870	2843	3056	2992	2628	2933	2556	2380	2885
CSDA	2544	2479	2698	2558	2237	2605	2342	2183	2513
CSDA + $\mathrm{RS}_{0.25}$	2545	2522	2663	2572	2240	2609	2337	2160	2517
CSDA + $\mathrm{RS}_{0.50}$	2589	2484	2611	2572	2266	2597	2312	2182	2579
CSDA + $\mathrm{RS}_{0.75}$	2529	2400	2608	2585	2187	2572	2357	2135	2483
CSDA + $\mathrm{RS}_{1.00}$	2543	2457	2663	2665	2337	2586	2361	2132	2560

Figure 5: Mean results for LT_{Tot} on each intersection and TSC approach from episode 251 to 300

rewards if the neighboring agents perform well. In the $\mathrm{RS}_{0.25}$ scenario, the performance of intersection I_6 was significantly increased, but the performance of neighboring intersections I_3, I_5 and I_9 decreased since they chose actions that would help the agent on I_6. With higher values of the reward cooperation coefficient, it becomes difficult for the agents to associate the obtained reward with respect to the current state-action pair since the state of neighboring agents is not known. The trend of decreasing performance with higher values of the reward cooperation coefficient is observable, but since the standard deviation of the results is high it is difficult to evaluate the impact of the reward cooperation coefficient properly.

By using the simple state-sharing augmentation the performance slightly increases. The agents can now prepare to meet the increasing traffic demand from neighboring intersections. When this approach is combined with reward sharing, a pattern of decreasing performance with a rising cooperation coefficient is observed, similar to scenarios without state sharing. Again, the best-performing approach was with a cooperation coefficient of 0.25. This increase in performance can be attributed to a better mapping between state actions and rewards since a part of neighboring agents' states are now shared.

The final group of tested approaches is with centralized state and distributed agents which outperformed all previous approaches by a great margin. With each agent having an insight into the entire network it was easier for each agent to select actions that would provide the best local benefit. The problem of non-stationarity is still present as the agents could not predict the actions of other agents and their effect on the network. By introducing reward sharing, the agents could now select actions that would benefit neighboring intersections if required. For CSDA approaches the best performing one was the approach with a cooperation coefficient of 0.75. This result shows that cooperation is easier when agents have more detailed information about the state of their neighbors. The GNG used for CSDA approaches handled the increased dimensionality of the input space without any loss of performance, but the scalability of GNG to larger networks remains an open question since it is expected that there are diminishing returns from including state variables from intersections that are further apart.

Inspection of results from Figure 4 shows how convergence is impacted by the chosen TSC approach. The Independent and state-sharing approaches seem to converge at around episodes 100 to 150, while the convergence of CSDA approaches seems slower and shows a tendency to continue converging even after the tested 300 episodes. Since all approaches use the same hyperparameters, this decrease in convergence speed can only be explained by slow adaptations of GNG neuron weights in later stages of learning. The total volume of the state space for CSDA approaches is much higher than the state space for other approaches. Thus, the limit of 150 neurons in the GNG might hinder performance since there would be more situations that require the addition of a neuron. This is confirmed by a large number of isolated neurons in the GNG of CSDA. However, by increasing the number of neurons, the total number of states would also increase which can also negatively impact the convergence.

The results of LT_{Avg} seem to correlate with the results of TTT, which is expected since vehicles spending less time in the traffic network are moving closer to their desired speed. The results for the NS_{Tot} also show a correlation with TTT. The environmental impacts were not directly analyzed as they are highly dependent on the emission model of vehicles, but considering the reduction in the total number of stops and travel time it can be expected that there would be a noticeable reduction in vehicle emissions.

7. Conclusion

This paper analyzed the performance of 3 different MARL ATSC approaches which are based on using the GNG for state identification. The performance was compared to the baseline FTSC approach to evaluate the full impact of each approach. Each approach improved the total performance of the traffic network, but the approach that uses centralized GNG for state identification of the entire network with decentralized agents performed the best. This result was further improved by expanding the reward function with reward sharing from neighboring agents. The approach that used only simple sharing of state variables had only a minor positive impact on the total performance. Future direction for the study should test additional state-sharing options such as including the state variables from downstream intersections which could prove beneficial when used with reward sharing.

While the CSDA approach performs the best, the problem of further scaling to more intersections remains an open question since adding more intersections would negatively impact the learning convergence of the agents. A possible research direction to overcome this convergence problem could be the introduction of knowledge sharing between neurons that share connections or have a small distance between each other if only a local subspace is considered.

Future work should also include the implementation in realistic traffic environments, with heterogeneous intersections. In such realistic scenarios, it would also be possible to perform the environmental analysis by including detailed vehicle information for the agents and modifying the reward function to a minimization of vehicle emissions. The addition of an offset-controlling agent could also help to improve traffic performance by enabling green wave control on corridors in the network that have a dominant traffic flow in one direction.

Acknowledgments

The author of this paper Mladen Miletić received a mobility grant from the Croatian Science Foundation under the project MOBDOK-2023-4074 that enabled the research presented in this paper. This work has been partly supported by the Croatian Science Foundation under the project IP-2020-02-5042 (DLASIUT), and by the Science Foundation Ireland under SFI Frontiers for the Future Grant No. 21/FFP-A/8957 (Clearway project). The authors would like to thank the company PTV for supporting the research in this paper. This research has also been carried out within the activities of the Centre of Research Excellence for Data Science and Cooperative Systems supported by the Ministry of Science and Education of the Republic of Croatia.

References

[1] A. A. Ceder, Urban mobility and public transport: future perspectives and review, International Journal of Urban Sciences 25 (2021) 455–479. doi:10.1080/12265934.2020.1799846.

[2] H. Brůhová Foltýnová, J. Brůha, Expected long-term impacts of the COVID-19 pandemic on travel behaviour and online activities: Evidence from a czech panel survey, Travel Behaviour and Society 34 (2024) 100685. doi:10.1016/j.tbs.2023.100685.

[3] D. Braess, Über ein Paradoxon aus der Verkehrsplanung, Unternehmensforschung 12 (1968) 258–268. doi:10.1007/BF01918335, [In German].

[4] Z. A. Cheng, M.-S. Pang, P. A. Pavlou, Mitigating traffic congestion: The role of intelligent transportation systems, Information Systems Research 31 (2020) 653–674. doi:10.1287/isre.2019.0894.

[5] M. Miletić, E. Ivanjko, M. Gregurić, K. Kušić, A review of reinforcement learning applications in adaptive traffic signal control, IET Intelligent Transport Systems 16 (2022) 1269–1285. doi:10.1049/itr2.12208.

[6] R. Bretherton, SCOOT Urban Traffic Control System—Philosophy and Evaluation, IFAC Proceedings Volumes 23 (1990) 237 – 239. doi:10.1016/S1474-6670(17)52676-2, IFAC/IFIP/IFORS Symposium on Control, Computers, Communications in Transportation, Paris, France, 19-21 September.

[7] P. Lowrie, The Sydney co-ordinated adaptive traffic system: Principles, methodology, algorithms, IEE CONF. PUBL.; ISSN 0537-9989; GBR; DA. 1982; NO 207; PP. 67-70 (1982).

[8] D. Pavleski, D. Koltovska-Nechoska, E. Ivanjko, Evaluation of adaptive traffic control system UTOPIA using microscopic simulation, in: 2017 International Symposium ELMAR, 2017, pp. 17–20. doi:10.23919/ELMAR.2017.8124425.

[9] J. Wahlstedt, Evaluation of the two self-optimising traffic signal systems Utopia/Spot and ImFlow, and comparison with existing signal control in Stockholm, Sweden, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 1541–1546.

[10] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[11] M. Miletić, E. Ivanjko, S. Mandžuka, D. K. Nečoska, Combining neural gas and reinforcement learning for adaptive traffic signal control, in: 2021 International Symposium ELMAR, 2021, pp. 179–182. doi:10.1109/ELMAR52657.2021.9550948.

[12] S. El-Tantawy, B. Abdulhai, Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC), in: 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 319–326. doi:10.1109/ITSC.2012.6338707.

[13] M. Noaeen, A. Naik, L. Goodman, J. Crebo, T. Abrar, Z. S. H. Abad, A. L. Bazzan, B. Far, Reinforcement learning in urban network traffic signal control: A systematic literature review, Expert Systems with Applications 199 (2022) 116830. doi:10.1016/j.eswa.2022.116830.

[14] S. Touhbi, M. A. Babram, T. Nguyen-Huu, N. Marilleau, M. L. Hbid, C. Cambier, S. Stinckwich, Adaptive Traffic Signal Control: Exploring Reward Definition For Reinforcement Learning, Procedia Computer Science 109 (2017) 513–520. doi:10.1016/j.procs.2017.05.327.

[15] I. Arel, C. Liu, T. Urbanik, A. Kohls, Reinforcement learning-based multi-agent system for network traffic signal control, IET Intelligent Transport Systems 4 (2010) 128–135(7). doi:10.1049/iet-its.2009.0070.

[16] W. Liu, G. Qin, Y. He, F. Jiang, Distributed Cooperative Reinforcement Learning-Based Traffic Signal Control That Integrates V2X Networks' Dynamic Clustering, IEEE Transactions on Vehicular Technology 66 (2017) 8667–8681. doi:10.1109/TVT.2017.2702388.

[17] M. Gregurić, M. Vujić, C. Alexopoulos, M. Miletić, Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data, Applied Sciences 10 (2020). doi:10.3390/app10114011.

[18] T. Chu, J. Wang, L. Codecà, Z. Li, Multi-agent deep reinforcement learning for large-scale traffic signal control, IEEE Transactions on Intelligent Transportation Systems 21 (2020) 1086–1095. doi:10.1109/TITS.2019.2901791.

[19] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, Z. Li, Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 3414–3421. doi:10.1609/aaai.v34i04.5744.

[20] M. Guériau, F. Armetta, S. Hassas, R. Billot, N.-E. El Faouzi, A constructivist approach for a self-adaptive decision-making system: Application to road traffic control, in: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), 2016, pp. 670–677. doi:10.1109/ICTAI.2016.0107.

[21] M. Guériau, N. Cardozo, I. Dusparic, Constructivist approach to state space adaptation in reinforcement learning, in: 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), 2019, pp. 52–61. doi:10.1109/SASO.2019.00016.

[22] M. Miletić, K. Kušić, M. Gregurić, E. Ivanjko, State complexity reduction in reinforcement learning based adaptive traffic signal control, in: 2020 International Symposium ELMAR, 2020, pp. 61–66. doi:10.1109/ELMAR49956.2020.9219024.

[23] M. Miletić, D. Čakija, F. Vrbanić, E. Ivanjko, Impact of connected vehicles on learning based adaptive traffic control systems, in: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022, pp. 3311–3316. doi:10.1109/SMC53654.2022.9945071.

[24] M. van Otterlo, M. Wiering, Reinforcement Learning and Markov Decision Processes, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 3–42. doi:10.1007/978-3-642-27645-3_1.

[25] M. Igl, L. Zintgraf, T. A. Le, F. Wood, S. Whiteson, Deep variational reinforcement learning for POMDPs, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 2117–2126.

[26] B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, volume 7, MIT Press, 1994. URL: https://proceedings.neurips.cc/paper_files/paper/1994/file/d56b9fc4b0f1be8871f5e1c40c0067e7-Paper.pdf.

[27] H. Hikawa, Y. Maeda, Improved learning performance of hardware self-organizing map using a novel neighborhood function, IEEE Transactions on Neural Networks and Learning Systems 26 (2015) 2861–2873. doi:10.1109/TNNLS.2015.2398932.

[28] Y. Prudent, A. Ennaji, An incremental growing neural gas learns topologies, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 2, 2005, pp. 1211–1216 vol. 2. doi:10.1109/IJCNN.2005.1556026.

[29] E. Joelianto, H. Sutarto, Simulation of Traffic Control Using VissimCOM Interface, Internetworking Indonesia Journal 11 (2019) 55.