Using Reinforcement Learning to Control Traffic Signals in a Real-World Scenario: An Approach Based on Linear Function Approximation

Lucas N. Alegre, Theresa Ziemke, and Ana L. C. Bazzan

***Abstract*— Reinforcement learning is an efficient, widely used machine learning technique that performs well in problems with a reasonable number of states and actions. This is rarely the case regarding control-related problems, as for instance controlling traffic signals, where the state space can be very large. One way to deal with the curse of dimensionality is to use generalization techniques such as function approximation. In this paper, a linear function approximation is used by traffic signal agents in a network of signalized intersections. Specifically, a true online SARSA(λ) algorithm with Fourier basis functions (TOS(λ)-FB) is employed. This method has the advantage of having convergence guarantees and error bounds, a drawback of non-linear function approximation. In order to evaluate TOS(λ)-FB, we perform experiments in variations of an isolated intersection scenario and a scenario of the city of Cottbus, Germany, with 22 signalized intersections, implemented in MATSim. We compare our results not only to fixed-time controllers, but also to a state-of-the-art rule-based adaptive method, showing that TOS(λ)-FB shows a performance that is highly superior to the fixed-time, while also being at least as efficient as the rule-based approach. For more than half of the intersections, our approach leads to less congestion and delay, without the need for the knowledge that underlies the rule-based approach.**

***Index Terms*— Traffic signal control, reinforcement learning, function approximation, multiagent systems.**

I. Introduction

TRAFFIC signal control is a challenging real-world problem. Current solutions to this problem, such as adaptive systems are often centralized or at least partially centralized. This is the case when there are several area controllers that are in charge of portions of the urban network. Less sophisticated alternatives are manual interventions from traffic operators or the use of fixed-time signal plans. However, in the era of big data and increasing computing power, other paradigms are becoming more and more prominent, as for instance, those derived from machine learning in general, and reinforcement learning (RL) in particular. In RL, traffic signal controllers located at intersections can be seen as autonomous agents that learn while interacting with the environment.

The use of RL is associated with challenging issues. One of them regards the environment being dynamic, thus making it necessary for agents to be highly adaptive. Moreover, agents must react to changes in the environment at individual level while also causing an unpredictable collective pattern, as they act in a coupled environment. Furthermore, the data needed in order to learn good policies may be staggeringly high-dimensional. Therefore, traffic signal control poses many challenges for standard techniques of multiagent RL.

To understand these challenges, let us first discuss the single agent case, where one agent performs an action once in a given state, and learns by getting a signal (reward) from the environment. RL techniques are based on estimates of values for state-action pairs (the so-called Q-values). These values may be represented as a table with one entry for each state-action pair. This works well in single agent problems and/or when the number of states and actions is small. However, in [1] Sutton and Barto discuss two drawbacks of this approach: first, the memory necessary to store these tables grows exponentially with the dimension of the state space, making it an impractical solution to real-world applications. Second, a long exploration time is required to fill such tables accurately. Those authors then suggest that generalization techniques may help in addressing this so-called curse of dimensionality.

An efficient representation of the states is a key factor that may limit the use of the standard RL algorithms in problems that involve several agents. Moreover, in scenarios in which the states are represented as continuous values, estimation of the state value by means of tabular Q-values may not be feasible. To deal with this problem, in this paper a true online SARSA(λ) algorithm [2] with Fourier Basis linear function approximation [3] is used. Henceforth, we refer to our approach by TOS(λ)-FB. As discussed ahead, this option is based on the fact that non-linear function approximation has several drawbacks, one being the lack of convergence guarantees and error bounds.

Manuscript received August 11, 2020; revised March 23, 2021; accepted May 20, 2021. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) through the project "Optimization and Network Wide Analysis of Traffic Signal Control" and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) under Grant 001. The work of Lucas N. Alegre was supported by the Brazilian Research Council, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), under Grant 140500/2021-9. The work of Ana L. C. Bazzan was supported by the Brazilian Research Council, CNPq, under Grant 307215/2017-2. The Associate Editor for this article was B. De Schutter. *(Lucas N. Alegre and Theresa Ziemke contributed equally to this work.) (Corresponding author: Ana L. C. Bazzan.)*

Lucas N. Alegre and Ana L. C. Bazzan are with the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre 91501-970, Brazil (e-mail: lnalegre@inf.ufrgs.br; bazzan@inf.ufrgs.br).

Theresa Ziemke is with the Transport Systems Planning and Transport Telematics Department, Technische Universität Berlin, 10623 Berlin, Germany (e-mail: tziemke@vsp.tu-berlin.de).

Digital Object Identifier 10.1109/TITS.2021.3091014

TOS(λ)-FB was implemented in the open-source agent-based transport simulation MATSim [4], which was then used to compare our approach to others, namely a fixed-time scheme and a rule-based adaptive signal control algorithm based on Lämmer and Helbing [5]. The results show that TOS(λ)-FB is able to show competitive performance in both an isolated intersection scenario, as well as in a network of intersections. This is especially notable, as the adaptive approach (and also the fixed time one) were designed specifically for dealing with the control of signals, whereas the RL-based approach needs no domain knowledge. To the authors' best knowledge, only a few works in the literature (especially those stemming from the RL area) include comparison to a state-of-the-art adaptive approach. More often than not, comparison of RL approaches is made only to a fixed-time scheme.

The remainder of this paper is organized as follows. The next section discusses background and related work. Section III describes TOS(λ)-FB. For evaluation of the proposed approach, experiments and their results are presented and discussed in Section IV, whereas Section V discusses the obtained results and future work.

II. Background and Related Work

In this section, we first introduce some concepts on traffic signal control (Section II-A) and give more details about one method in particular, which is used as comparison (Section II-B); then we discuss related work that is based on RL; the last subsection presents the simulation environment MATSim.

A. Traffic Signal Control

Traffic signals controllers can operate in several ways, defined over various dimensions. The first of these dimensions regards whether the signal is pretimed or based on a traffic-responsive strategy. Hence, to follow most of the RL literature, we interchangeably use the terms pretimed and fixed-time, as well actuated and adaptive, even if not fully correct.

In contrast to pretimed signals that cyclically repeat a given signal plan, traffic-responsive signals react to current traffic by adjusting signal states based on sensor data (e.g., from upstream inducting loops). They can, therefore, react to changes in demand and reduce emissions and waiting times more efficiently. One distinguishes different levels of adjustment: *actuated* signals (these use a fixed-time base plan and adjust parameters like green split, cycle time or offset); semi-actuated; and fully adaptive.

A variety of traffic-responsive traffic signal control algorithms have been developed. An overview is given, e.g., by Friedrich [6]. Here we briefly list some along with their references and note that these cover various generations and technological basis: PASSER [7]; Prodyn [8]; OPAC [9]; SOTL [10]; TUC (*Traffic-responsive Urban Traffic Control*) [11]; and TUC combined with predictive control [12]. Two popular approaches in this class are SCATS [13] and SCOOT [14]. Adaptive signal control is also very popular within research on RL; these are covered in Section II-D.

Next, we detail one state-of-the-art fully-adaptive approach that is categorized as rule-based, namely the one devised by Lämmer and Helbing [5]. The approach was shown to significantly improve waiting times while also granting stability in contrast to many other adaptive approaches [15], [16]. Hence, it is a suitable approach to compare TOS(λ)-FB with.

B. Lämmer and Helbing's Rule-Based Adaptive Traffic Signal Control Algorithm

The idea of the self-controlled signals proposed by Lämmer and Helbing [5] is to minimize waiting times and queue lengths at intersections, while also granting stability through minimal service intervals. The algorithm combines two strategies. The first is the optimizing strategy, which selects the signal phase i to be served next as the one with the highest priority index. This takes into account outflow rates and queue lengths of waiting and approaching vehicles that are registered by sensors. Given a prediction of the expected queue length $\hat{n}_i(t, \tau)$ at time $\tau > t$ and the maximum outflow rate q_i^{max} for phase i, one can derive the expected required green time for clearing the queue at time t. The second strategy is the stabilizing strategy, which ensures that each link is served at least once during a specified minimal service interval to prevent spillbacks. Links that have to be stabilized are added to a stabilization queue. If the queue is non-empty, the phase corresponding to the first element of the queue is switched to green for a guaranteed green time g_i^s depending on the average capacity utilization. If the stabilization queue is empty, the optimizing strategy takes over.

An assumption of Lämmer's algorithm is a queue-representation of traffic flow: if a link i is served, vehicles can leave the link with a constant outflow rate q_i^{max}, which is assumed to be known. Additionally, queues are assumed to be non-spatially, i.e., the algorithm does not account for vehicles spilling back to upstream lanes or links. Demand is supposed to be manageable on average with the desired cycle time T to ensure stability. The reader is referred to [5], [15] for more details, as well as to [17], [18] for a more recent extension/implementation (in MATSim).

C. Reinforcement Learning

In RL, an agent's goal is to learn an optimal control policy π^*, which maps a given state to the best appropriate action by means of a value function. We can model a RL problem as a Markov decision process (MDP) composed of a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a set of states; $\mathcal{A}$ is a set of actions; $\mathcal{T}$ is the transition function that models the probability of the system moving from a state $s \in \mathcal{S}$ to a state $s' \in \mathcal{S}$, upon performing action $a \in \mathcal{A}$; $\mathcal{R}$ is the reward function that yields a real number associated with performing an action $a \in \mathcal{A}$ when one is in state $s \in \mathcal{S}$; and $\gamma \in [0, 1)$ is the discount factor for future rewards. An experience tuple $\langle s, a, s', r \rangle$ denotes the fact that the agent was in state s, performed action a and ended up in s' with reward r. Let t denote the t^{th} step in the policy π. In an infinite horizon MDP, the cumulative reward in the future under policy π is defined by the

action-value function:

$$Q^{\pi}(s,a) = \mathbb{E}\Big[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau} | s_t = s, a_t = a, \pi\Big]. \quad (1)$$

Since the agent's objective is to maximize the cumulative reward, if it learned the optimal Q-values $Q^*(s,a)$ for all state-actions pairs, then the optimal control policy π^* is as follows[1]:

$$\pi^*(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^*(s,a), \quad \forall s \in \mathcal{S}. \quad (2)$$

RL methods can be divided into two categories: *model-based* methods assume that functions $\mathcal{T}$ and $\mathcal{R}$ are available. *Model-free* methods, on the other hand, do not require agents to have access to information about the environment.

D. Adaptive Traffic Signal Control Based on RL

There are many studies that use RL to improve traffic signal performance. Due to space restrictions, we refer the reader to some survey papers ([20]–[23]), which cover different aspects of the whole agenda. We note that [22] combines the survey with a good introduction for those unfamiliar with traffic signal control concepts.

Using model-free RL for traffic signal control is especially promising, as one does not need a lot of domain knowledge (as opposed to, e.g., rule-based approaches). Rather, the signal controller learns a policy by itself, i.e., RL is used in order to learn a policy that maps states (normally queues at junctions) to actions (normally keeping/changing the current split of green times among the lights of each phase).

However, issues may arise with the aforementioned curse of dimensionality. In fact, depending on the specific formulation (e.g., how states and action spaces are defined), the search space can be very high. For instance, consider an intersection with four incoming approaches with three lanes per approach. If we define the state as the queue length for each lane discretized in 10 levels, we end up with $10^{(4\times3)}$ distinct possible states. The reader is referred to [23] for several variants of such formulations.

Here we focus on the several dimensions that can be used to classify the research, stressing that there are hundreds of works that deal with signal control using RL.

One of such dimensions is decentralization vs. centralization. While the former is more popular, there are centralized approaches as well (e.g., [24]), where a single entity holds the MDP for all traffic signals: a central authority receives information about the queue lengths and elapsed time in all intersections and makes a decision about timings at each signal. On the other hand, the approaches in [21], [25] and many others are decentralized. Each junction learns independently (normally using QL).

A second dimension that is worth mentioning is whether tabular methods are used or, rather, function approximation. Since most of these works use QL, and thus approximate the Q-function as a table, they may fall prey to the curse of dimensionality. This is especially the case when one deals with realistic scenarios, as, e.g., those beyond 2-phase intersections. In order to address this, a few works used function approximation. For instance, [26] uses tile coding. However, the definition of states only considers queue length.

Recently, many studies have achieved impressive results using deep neural networks to approximate the Q-function (e.g., DQN [27], [28]). However, RL with linear function approximation has guaranteed convergence and error bounds [29], whereas non-linear function approximation is known to diverge in multiple cases [1], [30]. Beyond these theoretical results, linear learning methods are also of interest because they are very efficient in terms of both data and computation. Moreover, linear function approximation relies on a significantly fewer number of parameters, facilitating interpretation. Thus, if the Q-function can be linearly approximated with sufficient precision, linear function approximation methods are preferable.

A third classification dimension is whether the RL-based scheme is employed in an isolated intersection or in a network. Other dimensions are associated with the way each approach formulates states, actions, and compute rewards. More on this can be found in the surveys.

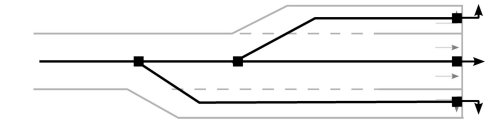
Finally, an important dimension is how evaluation is performed. The bulk of the literature performs just basic evaluation, either by comparing their proposed approaches to a fixed-time approach (and in the majority there is no information about how such timing was computed as, e.g., whether or not Webster's method [31] was employed, which should be the case to guarantee a minimum of fairness in the comparison); or by comparing to a random controller; or by comparing to tabular QL. Very few works present a comparison against a fully adaptive approach as we do here (by comparing against the scheme detailed in Section II-B). Moreover, most of these works do not give details about the adaptive approach used for comparison. In [32] the actual actuated controller used was not mentioned. Authors in [33] just mentioned that they have compared to GLIDE, a "modified version of SCATS used in Singapore". In [34], it is mentioned that "a mix of fixed-time control, semiactuated control, and SCOOT control" was used. Authors in [35] used SAT, an "algorithm that roughly emulates SCATS' behavior of saturation balancing", while [36] compared to SOTL [10]. In summary, a very small fraction of the works mentioned in the surveys did indeed compare to fully adaptive schemes.

E. Transport Simulation

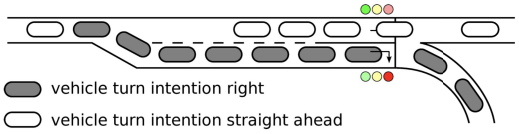
The agent-based transport simulation MATSim [4], which is used in this study, is able to run large-scale real-world simulations in reasonable time as, e.g., the open Berlin scenario [37]. Because of its agent-based structure, agent-specific waiting times and varying queue lengths over time at traffic lights can be directly analyzed and compared.

In MATSim, traffic flow is modeled mesoscopically by spatial first-in-first-out (FIFO) queues. Vehicles at the head of a queue can leave a link when the following criteria are fulfilled: (1) The link's free-flow travel time has passed, (2) the flow capacity of the link is not exceeded in the given

[1]For converge guarantees, in the case of QL, please see [19].



(a) Graph structure of a link with multiple lanes.



(b) Multiple queues; spillback is captured correctly.

Fig. 1. Links with multiple lanes in MATSim. Each lane is represented by its own FIFO queue. Traffic signal control for different turning moves is captured. Vehicles on different lanes can pass each other, unless the queue spills over. Source: [40].

time step, and (3) there is enough space on the next link. Despite this simple modeling approach, congestion as well as spillback can be modeled.

The traffic signal control module was developed by Grether as an extension to MATSim [38]. If a signal exists on a link, leaving the link is not possible while it shows red. First studies focused on fixed-time signals, but also approaches for traffic-responsive signal control have been implemented [17], [18], [39]. Separated waiting queues at intersections can be modeled in MATSim by lanes (see Fig. 1), which is especially useful to model protected left turns. Signals and lanes in MATSim are more extensively described by Grether and Thunig [40].

Events of vehicles entering or leaving links and lanes are thrown on a second-by-second time resolution in the simulation. Sensors on links or lanes that detect single vehicles can be easily modeled by listening to these events. As in reality, the maximum forecast period of such sensors is limited – vehicles can only be detected when they have entered the link. In the simulation, responsive signals use these sensor data to react dynamically to approaching vehicles. For every signalized intersection, the control unit is called every second to decide about current signal states. With that, any control scheme (in our case, RL-based) can be easily plugged in.

MATSim has many other functionalities. Readers interested in them – i.e., how agents adapt their plans and how long-term effects can be analyzed – are referred to [4]. Further, an example on how to start a MATSimsimulation using the RL signal control, which is detailled next, can be found at http://matsim.org/javadoc → signals → RunSarsaLambdaSignalsExample.

III. Methods

In this section, we first discuss the method used for function approximation, then give details about the formulation of state and action space, as well as rewards, for the specific domain of signal control. To close the section, we discuss the pseudo-code in more detail.

A. Function Approximation With the True Online SARSA(λ) and Fourier Basis Functions

As aforementioned, our approach (TOS(λ)-FB) implements the true online SARSA(λ) algorithm [41], a modification of the traditional SARSA(λ) that was demonstrated to have better theoretical properties and outperform the original method [42]. As detailed later, the state space can be very large for intersections with multiple approaches and lanes. In order to deal with high dimensional state spaces, the Q-function was linearly approximated using the Fourier basis scheme [3].

When linear approximation is used, the Q-function $Q(\mathbf{s}, a)$ for a given state vector $\mathbf{s}$ and discrete action a is approximated as a weighted sum of a set of m basis functions $\phi_1, \ldots, \phi_m$, as in Eq. 3, where $\boldsymbol{\theta}$ is the vector of learned weights.

$$Q(\mathbf{s}, a) = \boldsymbol{\theta} \cdot \boldsymbol{\phi}(\mathbf{s}, a) = \sum_{i=1}^{m} \theta_i \phi_i(\mathbf{s}, a) \tag{3}$$

The Fourier series is one of the most commonly used continuous function approximation methods, presenting solid theoretical foundations. In [3], it was empirically shown that Fourier basis outperforms other commonly used approximations methods such as polynomial and radial basis functions in continuous RL domains.

When applying Fourier series to the RL setting, it is possible to drop the sin terms of the series.[2] Then, for a nth order Fourier approximation, each basis function ϕ_i is defined as in Eq. 4, where $\mathbf{c}^i = [c_1^i, \ldots, c_k^i]$ is a vector that attaches an integer coefficient $c^i_{1 \leq j \leq k} \in \{0, \ldots, n\}$ to each feature in $\mathbf{s}$, and k is the state space dimension.

$$\phi_i(\mathbf{s}, a) = \begin{cases} \cos(\pi \, \mathbf{c}^i \cdot \mathbf{s}), & \text{if } a = a_t \\ 0, & \text{if } a \neq a_t \end{cases} \tag{4}$$

The set of basis functions $\phi_1, \ldots, \phi_m$ is obtained by systematically generating different coefficient vectors $\mathbf{c}^i$. Each coefficient $c^i_j \in \{0, \ldots, n\}$ determines (through the inner product in Eq. 4) the ith basis function's frequency along the jth state dimension. Hence, as we increase the order n of the approximation, basis functions with higher frequency coefficients are also considered. Importantly, when more than one coefficient is non-zero in a basis function, interactions between different state features are captured. For example, the effect of the queue lengths in the Q-value estimate probably depends on which is the currently active signal phase.

From now on, we denote $\boldsymbol{\phi}$ and Q as shorthand for the vector containing the values of all basis functions $\phi_i(\mathbf{s}_t, a_t)$ and the action-value estimate $Q(\mathbf{s_t}, a_t)$, respectively. The elements in $\boldsymbol{\phi}$ that correspond to the current action a_t take on the values of the Fourier basis; the elements corresponding to other actions have zero value (as in Eq. 4).

After the execution of action a_t, the weights $\boldsymbol{\theta}$ are updated via gradient descent, following the true online SARSA(λ) with linear function approximation update rule, as in Eq. 5, where $\delta = r_t + \gamma \, Q(\mathbf{s}_{t+1}, a_{t+1}) - Q(\mathbf{s}_t, a_t)$ is the temporal difference error and Q_{old} is a scalar temporary variable initialized with zero and set to $Q_{old} \leftarrow Q(\mathbf{s}_{t+1}, a_{t+1})$ after every step.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha(\delta + Q - Q_{old})\mathbf{e} - \alpha(Q - Q_{old})\boldsymbol{\phi} \tag{5}$$

The eligibility traces vector $\mathbf{e}$ – which is used to address the *credit assignment* problem – is updated as in Eq. 6. Each weight update also takes into account previously visited states,

[2]For detailed explanation, please see [3].

which are credited accordingly to the values accumulated on the vector $\mathbf{e}$. The parameter λ controls the decay of the eligibility traces at each time step.

$$\mathbf{e} \leftarrow \gamma\,\lambda\mathbf{e} + \boldsymbol{\phi} - \alpha\gamma\,\lambda(\mathbf{e}\cdot\boldsymbol{\phi})\boldsymbol{\phi} \tag{6}$$

Given the base learning rate α, each weight θ_i is updated with the scaled learning rate $\alpha_i = \alpha/||\mathbf{c}^i||_2$, as proposed in [3]. Both the weights and eligibility traces vectors are initialized with zeros.

In order to address the exploration-exploitation dilemma, the *ε-greedy* exploration strategy is used to choose actions: the action with the highest Q-value is selected with a probability of $1-\varepsilon$ and a random action is selected with probability ε.

Next we give the formulations that are specific to the domain of traffic signal control.

B. State Space

In RL problems, the definition of state space strongly influences the agents' behavior and performance. In traffic signal control, for instance, information related to the level of congestion in the approaching lanes is fundamental in order to appropriately choose the next active signal phase.

In the present setting, the agent observes a state vector $\mathbf{s}_t \in \mathbb{R}^k$ at each time step t. This vector partially represents the true state of the controlled intersection and is defined as in Eq. 8. In this definition, $\rho_a \in \{0,1\}$ is a binary feature active when $a \in \mathcal{A}$ is the current selected signal phase, and $\tau \in [0,1]$ is the elapsed time of the current signal phase divided by the maximum green time g_{max}. Let E be the set of all links of the intersection; L the set of all approaching lanes; C_e and C_l the storage capacity of the links and lanes, respectively; $V_{e,t}$ the set of vehicles on link $e \in E$ at the time t; and $V^q_{l,t}$ the set of queued vehicles on lane $l \in L$ at the time t. Then, $\omega_e \in [0,1]$ and $q_l \in [0,1]$ (as defined in Eq. 7) are the density of the link $e \in E$ and the queue occupation of the lane $l \in L$, respectively.

$$\omega_e = \frac{|V_{e,t}|}{C_e}, \quad q_l = \frac{|V^q_{l,t}|}{C_l}, \ \forall e \in E, \ \forall l \in L \tag{7}$$

$$\mathbf{s}_t = [\rho_1, \ldots, \rho_{|\mathcal{A}|}, \tau, \omega_1, \ldots, \omega_{|E|}, q_1, \ldots, q_{|L|}] \tag{8}$$

This state definition is inspired by [43], where authors achieved similar performance levels, even when using more complex state definitions (e.g., including positions of each vehicle in the approaching lanes).

Normally, the RL signal control is only allowed to change the active signal phase after a number of seconds Δ is elapsed. Here we use, as common in the literature, $\Delta = 3$ seconds. This means that, in general, one time step for the traffic signal agent corresponds to three seconds of simulation. This reduces the complexity and the size of the state space, without significantly reducing the performance.

C. Action Space

At each time step t, the traffic signal controller chooses a discrete action $a_t \in \mathcal{A}$. In our setting, the set of actions $\mathcal{A}$ is the set of signal phases the traffic signal controller can choose to activate. There are restrictions in the action selection: the agent can keep the active signal phase only if the elapsed time is less than the maximum green time g_{max}. Additionally, when the agent changes the active signal phase, it must wait yellow + all-red $+g_{min}$ seconds before acting again. These restrictions ensure the feasibility of the signal controller for real-world applications. The minimum-green time (g_{min}) restriction, in particular, ensures that pedestrians moving during a particular phase can safely pass through the intersection.

Algorithm 1 True Online SARSA(λ) With Fourier Basis

Input: α, λ, γ, ε, $\{\mathbf{c}^i\}_{1\le i\le m}$, Δ, g_{min}, g_{max}, yellow, all-red

$\theta \leftarrow \mathbf{0}$; $\mathbf{e} \leftarrow \mathbf{0}$; $Q_{old} \leftarrow 0$
Observe state $\mathbf{s}_0$ (Eq. 8)
Choose action a_0 based on $\mathbf{s}_0$ with ε-greedy scheme
Compute features ϕ_i with $\mathbf{s}_0, a_0$ and $\mathbf{c}^i$, $\forall i \in m$ (Eq. 4)
for t in $0 \ldots \infty$ **do**
  Observe state $\mathbf{s}_{t+1}$
  Compute reward r_t (Eq. 9)
  if elapsed-time $\ge g_{max}$ **then**
    Choose action $a_{t+1} \in \mathcal{A} \setminus a_t$ with highest Q-value
  else
    Choose action a_{t+1} with ε-greedy scheme
  end if
  Compute features ϕ'_i with $\mathbf{s}_{t+1}, a_{t+1}$ and $\mathbf{c}^i$, $\forall i \in m$
  $Q \leftarrow \boldsymbol{\theta}\cdot\boldsymbol{\phi}$ (Eq. 3)
  $Q' \leftarrow \boldsymbol{\theta}\cdot\boldsymbol{\phi}'$
  $\delta \leftarrow r_t + \gamma\,Q' - Q$
  $\mathbf{e} \leftarrow \gamma\,\lambda\mathbf{e} + \boldsymbol{\phi} - \alpha\gamma\,\lambda(\mathbf{e}\cdot\boldsymbol{\phi})\boldsymbol{\phi}$ (Eq. 6)
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha(\delta + Q - Q_{old})\mathbf{e} - \alpha(Q - Q_{old})\boldsymbol{\phi}$ (Eq. 5)
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi}'$; $Q_{old} \leftarrow Q'$
  if $a_t = a_{t+1}$ (Keep current signal phase) **then**
    Wait Δ seconds
  else
    Wait yellow + all-red $+g_{min}$ seconds
  end if
end for

D. Reward

After taking action a_t, the traffic signal controller receives a scalar reward $r_t \in \mathbb{R}$. As in [43] the reward is defined as the change in cumulative delay, as given in Eq. 9, where D_{a_t} and $D_{a_{t+1}}$ represent the cumulative delay at the intersection before and after executing the action a_t.

$$r_t = D_{a_t} - D_{a_{t+1}} \tag{9}$$

On its turn, the cumulative vehicle delay D, for any time t, is computed as in Eq. 10, where d^v_t is the delay of vehicle v at time t and $V_{l,t}$ is the set of vehicles on lane $l \in L$ at time t.

$$D_t = \sum_{l\in L}\sum_{v\in V_{l,t}} d^v_t \tag{10}$$

E. TOS(λ)-FB Traffic Signal Controller

To give an overview of TOS(λ)-FB, we now summarize it in Alg. 1. Because each agent follows this procedure in

a decentralized manner, each traffic intersection can have its own estimates for the Q-values and follow its own policy.

The algorithm receive as input the parameters related to RL (α, λ, γ and ε); the coefficient vectors $\{\mathbf{c}^i\}_{1\leq i\leq m}$, which control the linear approximation with Fourier basis (Eq. 4); the time between actions Δ; the values for minimum and maximum green time (g_{min} and g_{max}); and the duration for yellow and all-red periods.

In lines 8-11 we ensure that the constraint on the maximum green time g_{max} is satisfied: if the elapsed time for the current signal phase is greater than g_{max}, the agent must choose an action a_{t+1} different of a_t. Otherwise, the ε-greedy scheme (see Section III-A) is followed.

Lines 14-18 refer to how the weights θ, as well the eligibility traces vector $\mathbf{e}$ of the linear approximation are updated (Eq. 5 and Eq. 6, respectively), taking into account the current Q-values estimates (Eq. 3).

Finally, in lines 20-23 we address the constraints mentioned in Sec. III-B and Sec. III-C. If an agent chooses to keep the current signal phase, then the next action selection occurs after Δ seconds (as explained in Sec. III-B). Otherwise, it must respect the yellow, all-red and minimum green times before being allowed to change the signal phase again.

IV. Evaluation

The TOS(λ)-FB is evaluated in a network of traffic signals (discussed in Section IV-B). In that scenario, we compare TOS(λ)-FB with a fixed-time controller optimized by a mixed-integer program [44], and a version of our proposed controller with radial basis functions (RBF) instead of Fourier basis. As mentioned, contrarily to the majority of the works in the literature, we also compare to a state-of-the-art traffic-responsive signal approach, namely the one by Lämmer and Helbing (henceforth called Lämmer), presented in Section II-B. As a motivating example, we also include a discussion that relates to an isolated intersection scenario (see Section IV-A), where we compare our method with the traditional tabular SARSA(λ).

In our experiments, a period of 86400 seconds (one day) is simulated. The results presented in all following figures are averaged over 20 runs with different random seeds, and the shadowed area in the plots depicts the standard deviation regarding delay or queue length, accordingly. The lines were smoothed with a moving average window of 300 seconds (5 minutes) for better clarity.

According to [45] the order of the Fourier approximation was set to $n = 7$ for all the following experiments. As we are dealing with intersections with multiple lanes and signal phases, and the number of Fourier basis functions grows exponentially on the number of dimensions of the state space, it is necessary to restrict the number of Fourier basis. We can meet this condition by placing constraints on the coefficient vectors $\mathbf{c}^i$, thus reducing the number of basis functions (Eq. 4). In our experiments, we limited the number of coefficient vectors $\mathbf{c}^i$ by constraining them to have at most two non-zero elements, as adding more coefficients did not improve the results. Hence, we can still capture the relations between pairs of features in the state space. Furthermore, a learning rate of $\alpha = 10^{-6}$ was used. The discount factor was set to $\gamma = 0.95$, $\lambda = 0.1$ and the exploration rate was set to $\varepsilon = 0.01$ (this latter means that the agent is mostly taking the action with the highest Q-value, but still exploring with a fixed low chance). These values are common in the literature and produced the best results after extensive experimentation.

A. Isolated Intersection: Scenario and Results

To be able to comprehensively evaluate the efficiency of our RL control we first applied it to an isolated multi-lane intersection with various set-ups, especially different traffic saturation conditions. The results are discussed in detail in a previous preliminary study [46]. Also, we evaluated the use of other state or reward definitions and discussed the choice of the order of the Fourier basis approximation. This section only gives an overview of the most important findings of this previous study.

The isolated intersection featured here has four incoming approaches with multiple lanes. In the horizontal direction, there is a dedicated left turning lane in each traffic approach, as well as three lanes for straight traffic. In the vertical direction, there are two lanes for straight traffic. Traffic signals are grouped into three non-conflicting signal phases: straight traffic in horizontal direction; left-turning traffic in horizontal direction; vertical direction. While switching between two signal phases, there is an all-red period of one second. The minimum green time g_{min} for a signal phase is five seconds.

The fixed-time controller optimized with Webster's method [31] has a cycle time of 40 seconds and splits green times according to average flow rates (see ahead for details on the demand). Traffic-responsive approaches do not have a fixed cycle time. In particular, for Lämmer's algorithm, a desired and a maximal cycle time can be defined (for this scenario 40 and 60 seconds are used, respectively). For our RL-based control, a maximal green time g_{max} of 30 seconds per signal phase is used.

Regarding the demand in the base set-up, we have 1800 vehicles on average per hour in each horizontal approach. Additionally, there are 180 vehicles on average per hour, which turn left at the intersection, in each horizontal approach. In the vertical direction, there are 600 vehicles on average per hour from each approach—all going straight. Furthermore, arrival rates are stochastic: vehicles are inserted as platoons with a platoon size that is exponentially distributed around an expected value of five. Also, the time gap between vehicle platoons is exponentially distributed; its expected value is the platoon size divided by the average flow value.

1) Tabular Vs. Linear SARSA(λ): In order to transform the continuous state space defined in Section III-A to a discrete state space for the tabular SARSA(λ), the continuous attributes were discretized in equally distributed bins/intervals. In order to allow a fair comparison, the same discount factor, value of λ and exploration rate were used for both methods, except for the learning rate, which was set to $\alpha = 0.1$ for the tabular SARSA(λ).

In Fig. 2 the average delay per vehicle at each second of the simulation is depicted for true online SARSA(λ) with Fourier

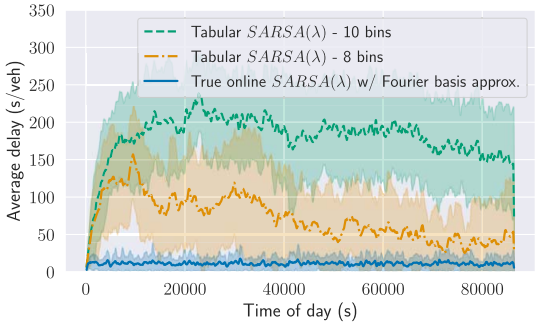


Fig. 2. Average delay for tabular and linear function approximation RL traffic signal controllers in the isolated intersection scenario.

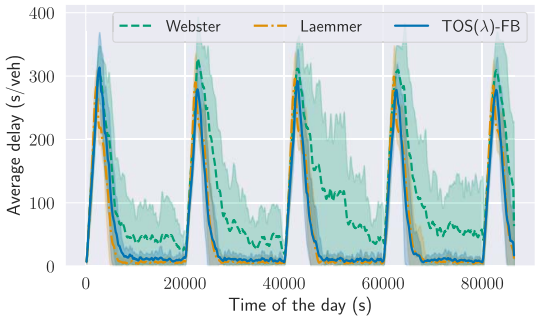


Fig. 3. Average delay per vehicle in the isolated intersection scenario with varying demand as described in Section IV-A 2).

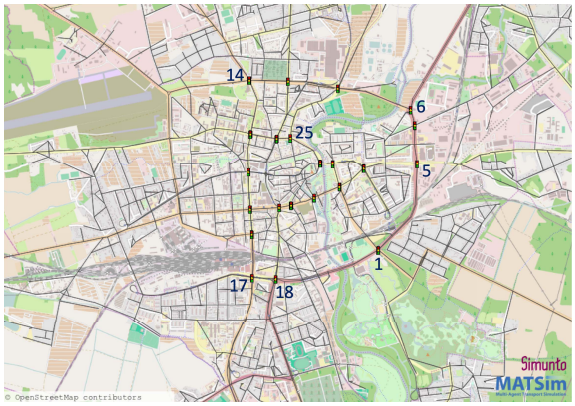


Fig. 4. Inner-city network of Cottbus with 22 signalized intersections.

basis linear function approximation and for tabular SARSA(λ) with 8 vs. 10 discretization bins of the q and ω features (see Eq. 8). Reducing the number of bins from 10 to 8 significantly speeds up the learning and reduces the delay, as the number of discretization bins exponentially increases the size of the state space. However, by reducing the number of bins, different states (in which different actions are optimal) are perceived as the same, thus leading to sub-optimal performance in the long run. In short, the aim of this experiment is to show that the usage of function approximation not only avoids the curse of dimensionality, but also introduces generalization. With that, the TOS(λ)-FB yields a much faster learning curve and overall lower delay values.

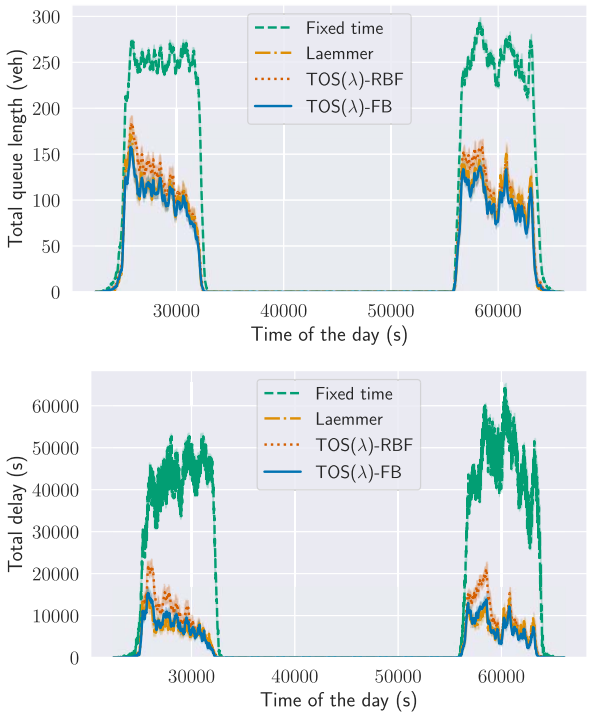


Fig. 5. Total queue length and delay over time in the Cottbus scenario.

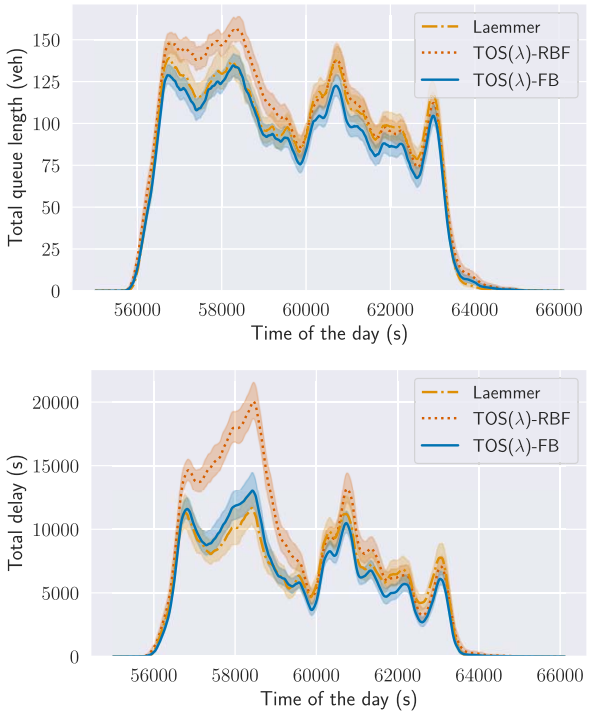


Fig. 6. Total queue length and delay over time in the afternoon peak.

2) Comparison With Fixed-Time and Rule-Based Signals: Fig. 3 depicts the results obtained when we interleave periods of 2000 seconds each, varying the demand. More specifically, we consider five such periods in which the demand in the horizontal approaches are twice those from the base set-up

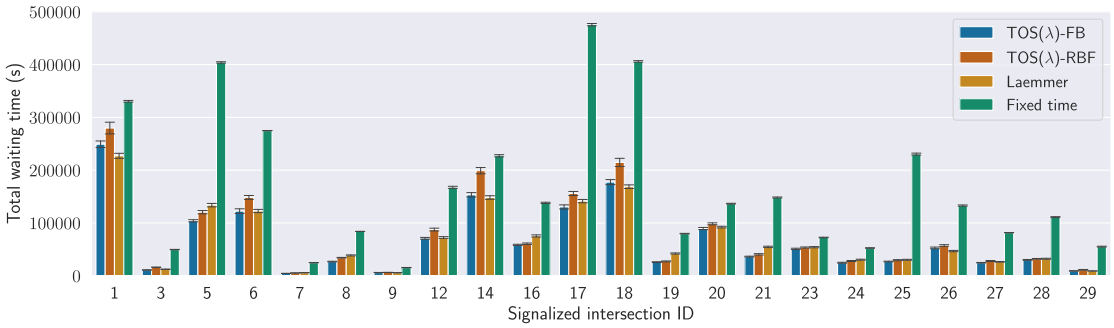


Fig. 7. Total waiting time at the 22 signalized intersections in the Cottbus scenario.

just mentioned. This aims at analyzing the effect of fluctuating demand on the performance of the RL-based controller.

One can see that TOS(λ)-FB is able to handle overload situations and quickly reduces the queues afterward. Recall that, contrarily to rule-based approaches, the RL control does not require domain knowledge.

B. Network of Intersections – City of Cottbus

To be able to test TOS(λ)-FB for signal control in a more realistic scenario with multiple signalized intersections, this study considers a scenario of the city of Cottbus, Germany. Input data for network, demand, optimized fixed-time signal plans and Lämmer's rule-based adaptive signals are taken from previous studies [18], [38], [44], which are here summarized. The network consists of approx. 10000 links and 4000 nodes and captures a region of about 1800 km^2 around the city. Daily home-work-home activity chains of around 33000 commuters living and working in the region are simulated using MATSim. In Fig. 4 we show a zoom of such area, depicting 22 signalized intersections in the inner city of Cottbus. Most of the signalized intersections have separate lanes and signal timings for left or right turns.

To be able to compare the signal control approaches with different traffic saturation conditions, we run the Cottbus scenario with different capacity factors, which scale the flow and storage capacity of links accordingly, to simulate higher or lower congestion levels. A smaller capacity factor leads to lower flow and storage capacity values on the links and therefore corresponds to a higher congestion level, and vice versa. The base case scenario taken from the previous studies uses a capacity factor of 0.7 (i.e., 70% of the original link capacities are used because only home-work-home trips are simulated). Here, we run simulations with a range of different values: 0.5, 0.6, 0.7, 0.8 and 0.9.

Unless noted, the values of the parameters are the same from the isolated intersection scenario. One such exception regards the maximum green time g_{max}, which was set to 60 seconds to be consistent with the Cottbus scenario and, therefore, comparable to the fixed-time and Lämmer's control. However, this maximum green time was hardly reached by TOS(λ)-FB.

C. Results for the Network Case

In Fig. 5 we compare the total queue length and the total delay resulted with TOS(λ)-FB, TOS(λ)-RBF (which corresponds to our method with radial basis function [1] approximation instead of Fourier basis), Lämmer, and fixed-time controllers. For fairness of comparison, we used for the RBF approximation of TOS(λ)-RBF the same number of basis used for TOS(λ)-FB. The RBF functions (which are modeled as Gaussian curves) were created using $n = 7$ different centers evenly distributed along each dimension of the state space. We can observe that the fixed-time controller produces much higher delay and queue length and oscillates more than the other controllers. TOS(λ)-FB resulted in slightly better results than Lämmer, especially in the afternoon peak. Although TOS(λ)-RBF shows good performance (similar to Lämmer in some periods of the simulation), it never results in lower queue lengths or total delay when compared to TOS(λ)-FB. Notice that these results confirm the findings of [3] regarding the use of the Fourier basis to approximate the Q-function.

For clarity, we zoom in Fig. 6 by removing the fixed time curve from the comparison and focusing on the afternoon peak. It can be seen that TOS(λ)-FB yields lower queue lengths during the whole period (top plot in Fig. 6). Regarding the total delay (bottom plot), TOS(λ)-FB starts worse, but afterward is able to beat Lämmer. This shows that although being related, both metrics (queue length and delay) are not perfectly correlated: a signal controller can reduce delay at cost of maintaining larger queues. Despite that, in the long term TOS(λ)-FB finds a policy that is able to outperform Lämmer in both metrics.

Fig. 7 highlights individual results for each one of the 22 signalized intersections. One can see that different intersections present very different patterns. TOS(λ)-FB shows advantage over Lämmer in situations with less congestion, resulting in lower waiting times for almost all intersections in which the waiting time is less than 200000 seconds for the fixed time control.

For the other, more congested intersections (1, 5, 6, 14, 17, 18, 25; highlighted in Fig. 4)—in which the waiting time is above 200,000 seconds for the fixed controller—TOS(λ)-FB performs better in at least half of them. The fixed-time controller results in significantly worse results for all intersections, which demonstrates the advantage and importance of adaptive controllers.

Fig. 8 shows the results for the comparison of different traffic saturation conditions. It depicts the total waiting time for different capacity factors, whereas higher capacity factors

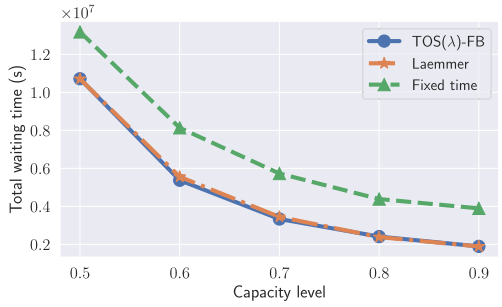


Fig. 8. Overall total waiting time in the Cottbus scenario for different traffic saturation conditions (the higher the capacity factor, the lower the demand level).

correspond to overall lower levels of saturation (as described in Section IV-B). One can see that changing the demand level seems to impact all methods equally. This especially ensures, that TOS(λ)-FB performs very well and comparable to Lämmer's approach also in situations with very high and, respectively, very low traffic saturation.

V. Conclusion and Future Work

As discussed in Section II, reinforcement learning is increasingly being proposed as a method for controlling traffic signals, as it is able to adapt to traffic patterns on the fly. In the present paper, it is shown that specific techniques from RL can help to improve the performance of traffic signal control, and even outperform state-of-the-art rule-based adaptive signal control algorithms. It was argued that tabular RL methods may not be feasible due to the curse of dimensionality. When it is possible to employ them, it is often the case that they need long learning times before convergence in the case of realistic intersections with more than two signal phases and when a more complex definition of state is used.

To address these issues, we use an RL algorithm with linear function approximation (the true online SARSA(λ) with Fourier basis functions) which, to the authors' best knowledge, was not used for traffic signal control before. Moreover, it can be argued that this kind of function approximation is more interpretable as compared to non-linear functions, e.g., those related on neural networks.

Our method TOS(λ)-FB was implemented in MATSim and compared to fixed-time and rule-based adaptive signal control both, in an isolated intersection scenario, as well as in a real-world scenario (city of Cottbus, Germany). In the former case, it can be seen that TOS(λ)-FB shows at least comparable results, without the need for domain knowledge that underlies rule-based and fixed time methods. To the authors' knowledge, this kind of comparison with other than fixed-time approaches is rarely in the RL literature and is, therefore, a key feature of this work.

In the case of the Cottbus network, which has 22 signalized intersections, when considering the overall measure (over all intersections), TOS(λ)-FB has a better performance in the afternoon peak. Considering individual intersections, in several of these TOS(λ)-FB shows advantage over the rule-based adaptive approach. Moreover, for more congested intersections, TOS(λ)-FB performs better in at least half of them. It is also worth mentioning that the fixed-time controller results in significantly worse results for all intersections, which demonstrates the advantage and importance of adaptive controllers.

Furthermore, we observed that different intersections present very different patterns, with some being critical for the network and some not. Hence, a possible future direction is to take a closer look at critical intersections and study how their learned policies impact adjacent intersections in the network. Similarly, an interesting future work is to combine our method with route choice algorithms in scenarios where both traffic signals and vehicles adapt simultaneously.

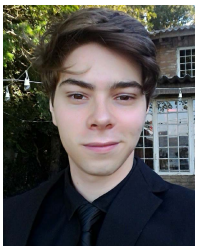
Acknowledgment

The authors would like to thank Prof. Kai Nagel for his support and supervision during the internship of Lucas N. Alegre at his group at TU Berlin.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[2] H. van Seijen, A. R. Mahmood, P. Pilarski, M. Machado, and R. Sutton, "True online temporal-difference learning," *J. Mach. Learn. Res.*, vol. 17, pp. 1–40, Sep. 2016.
[3] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the Fourier basis," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 380–385.
[4] A. Horni, K. Nagel, and K. W. Axhausen, Eds., *The Multi-Agent Transport Simulation MATSim*. London, U.K.: Ubiquity, 2016.
[5] S. Lämmer and D. Helbing, "Self-control of traffic lights and vehicle flows in urban road networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 4, pp. 4–19, 2008.
[6] B. Friedrich, "Adaptive signal control—An overview," in *Proc. 9th Meeting Euro Work. Group Transp.*, Bari, Italy, 2002, pp. 571–574.
[7] E. C. P. Chang, J. C. K. Lei, and C. J. Messer, "Arterial signal timing optimization using PASSER-II," Texas Transp. Inst., College Station, TX, USA, Tech. Rep. 467, 1988.
[8] J. Henry, J. L. Farges, and J. Tuffal, "The PRODYN real time traffic algorithm," in *Proc. Int. Fed. Autom. Control (IFAC) Conf.*, R. Isermann, Ed. Baden-Baden, Germany: IFAC, 1983, pp. 307–312.
[9] N. H. Gartner, "OPAC—A demand-responsive strategy for traffic signal control," *Transp. Res. Rec.*, vol. 906, pp. 75–81, 1983.
[10] C. Gershenson, "Self-organizing traffic lights," *Complex Syst.*, vol. 16, no. 1, pp. 29–53, 2005.
[11] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Eng. Pract.*, vol. 10, no. 2, pp. 183–195, Feb. 2002.
[12] L. B. de Oliveira and E. Camponogara, "Multi-agent model predictive control of signaling split in urban traffic networks," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 1, pp. 120–139, Feb. 2010.
[13] P. Lowrie, "The Sydney co-ordinated adaptive traffic system: Principles, methodology, algorithms," in *Proc. Int. Conf. Road Traffic Signalling*, Sydney, NSW, Australia, 1982, pp. 67–70.
[14] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "SCOOT—A traffic responsive method of coordinating signals," Transp. Road Res. Lab., Berkshire, New Bedfor, MA, USA, TRRL Lab., Tech. Rep. 1014, 1981.
[15] S. Lämmer and D. Helbing, "Self-stabilizing decentralized signal control of realistic, saturated network traffic," Santa Fe Inst., Santa Fe, NM, USA, Working Paper 10-09-019, 2010.
[16] S. Lämmer, "Die selbst-steuerung im praxistest," *Straßenverkehrstechnik*, vol. 3, pp. 143–151, 2016.
[17] N. Kühnel, T. Thunig, and K. Nagel, "Implementing an adaptive traffic signal control algorithm in an agent-based transport simulation," *Procedia Comput. Sci.*, vol. 130, pp. 894–899, 2018.
[18] T. Thunig, N. Kühnel, and K. Nagel, "Adaptive traffic signal control for real-world scenarios in agent-based transport simulations," *Transp. Res. Procedia*, vol. 37, pp. 481–488, Jan. 2019.

[19] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.
[20] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Auto. Agents Multi-Agent Syst.*, vol. 18, no. 3, pp. 342–375, Jun. 2009.
[21] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*, T. L. McCluskey, A. Kotsialos, P. J. Müller, F. Klügl, O. Rana, and R. Schumann, Eds. Cham, Switzerland: Springer, May 2016, pp. 47–66.
[22] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explor. Newslett.*, vol. 22, no. 2, pp. 12–18, Jan. 2021.
[23] K.-L.-A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 1–38, Oct. 2017.
[24] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1640–1645.
[25] K. J. Prabuchandran, A. N. Hemanth Kumar, and S. Bhatnagar, "Decentralized learning for traffic signal control," in *Proc. 7th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2015, pp. 1–6.
[26] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Hierarchical control of traffic signals using Q-learning with tile coding," *Int. J. Speech Technol.*, vol. 40, no. 2, pp. 201–213, Mar. 2014.
[27] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," presented at the NIPS Deep Learn. Workshop, 2013.
[28] E. Van Der Pol, "Deep reinforcement learning for coordination in traffic light control," Ph.D. dissertation, Faculteit der Natuurwetenschappen, Wiskunde en Informatica, Univ. Amsterdam, Amsterdam, The Netherlands, 2016.
[29] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690, May 1997.
[30] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*. San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 30–37.
[31] F. V. Webster, "Traffic signal setting," Road Res. Lab., London, U.K., Tech. Rep. 39, 1958.
[32] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 732–752, Dec. 2017.
[33] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intell. Transp. Syst.*, vol. 4, no. 3, pp. 177–188, Sep. 2010.
[34] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.
[35] A. Salkham, R. Cunningham, A. Garg, and V. Cahill, "A collaborative reinforcement learning approach to urban traffic control optimization," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Dec. 2008, pp. 560–566.
[36] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA: Association for Computing Machinery, 2018, pp. 2496–2505.
[37] D. Ziemke, I. Kaddoura, and K. Nagel, "The MATSim open berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data," *Procedia Comput. Sci.*, vol. 151, pp. 870–877, Jan. 2019.
[38] D. S. Grether, "Extension of a multi-agent transport simulation for traffic signal control and air transport systems," Ph.D. dissertation, Fakultät Verkehrs- und Maschinensysteme, TU Berlin, Berlin, Germany, 2014.
[39] D. Grether, J. Bischoff, and K. Nagel, "Traffic-actuated signal control: Simulation of the user benefits in a big event real-world scenario," in *Proc. 2nd Int. Conf. Models Technol. ITS*, Leuven, Belgium, 2011, pp. 11–12.
[40] D. Grether and T. Thunig, "Traffic signals and lanes," in *The Multi-Agent Transport Simulation MATSim*, A. Horni, K. Nagel, and K. W. Axhausen, Eds. London, U.K.: Ubiquity, 2016, ch. 12.
[41] H. Van Seijen and R. S. Sutton, "True online TD(λ)," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China: JMLR.org, 2014.
[42] H. van Seijen, A. R. Mahmood, P. Pilarski, M. Machado, and R. Sutton, "True online temporal-difference learning," *J. Mach. Learn. Res.*, vol. 17, pp. 1–40, Sep. 2016.
[43] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *Procedia Comput. Sci.*, vol. 130, pp. 26–33, Jan. 2018.
[44] T. Thunig, R. Scheffler, M. Strehler, and K. Nagel, "Optimization and simulation of fixed-time traffic signal control in real-world applications," *Procedia Comput. Sci.*, vol. 151, pp. 826–833, Jan. 2019.
[45] T. Ziemke, L. N. Alegre, and A. L. C. Bazzan, "A reinforcement learning approach with Fourier basis linear function approximation for traffic signal control," in *Proc. 11th Workshop Agents Traffic Transp. (ATT)*, vol. 2701, no. 9, 2020, pp. 55–62.
[46] T. Ziemke, L. N. Alegre, and A. L. C. Bazzan, "Reinforcement learning vs. rule-based adaptive traffic signal control: A Fourier basis linear function approximation for traffic signal control," *AI Commun.*, vol. 34, no. 1, pp. 89–103, Feb. 2021.



Lucas N. Alegre received the B.Sc. degree *(cum laude)* in computer science from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2021, where he is currently pursuing the Ph.D. degree with the Institute of Informatics. In the Winter of 2020, he worked as an Intern Researcher with the Technische Universität Berlin, Germany. His research interests include reinforcement learning, machine learning, artificial (and neuro-inspired) intelligence, and their applications to real-world problems.

Theresa Ziemke received the M.Sc. degree in mathematics from the Technische Universität Berlin (TU Berlin), Germany, in 2014. Since 2015, she has been a Research Associate with the Transport Systems Planning and Transport Telematics (VSP) Group and the Combinatorial Optimization and Graph Algorithms (COGA) Group, TU Berlin. Her research interests include modeling and simulation of transport (MATSim), traffic control and traffic optimization, algorithmic game theory, graph and network optimization, and efficiency of equilibrium.

Ana L. C. Bazzan received the Ph.D. degree from the University of Karlsruhe (now KIT), Germany, in 1997. She is currently a Full Professor with UFRGS, Porto Alegre, Brazil. Her main research interests include multi-agent systems, reinforcement learning, complex systems, and agent-based simulation. She is a member of the IFAAMAS Board. She is also a fellow of the Alexander von Humboldt Foundation. She served/serving as an Associate Editor for *Autonomous Agents and Multiagent Systems* journal and *Advances in Complex Systems* journal.