University of Groningen

Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran

Aslani, Mohammad; Seipel, Stefan; Mesgari, Mohammad Saadi; Wiering, Marco

Published in:
Advanced Engineering Informatics

DOI:
[10.1016/j.aei.2018.08.002](https://doi.org/10.1016/j.aei.2018.08.002)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Aslani, M., Seipel, S., Mesgari, M. S., & Wiering, M. (2018). Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran. *Advanced Engineering Informatics*, 38, 639-655. <https://doi.org/10.1016/j.aei.2018.08.002>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Full length article

Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran

Mohammad Aslani^{a,*}, Stefan Seipel^{a,b}, Mohammad Saadi Mesgari^c, Marco Wiering^d

^a Department of Industrial Development, IT and Land Management, University of Gavle, Gavle, Sweden

^b Division of Visual Information and Interaction, Department of Information Technology, Uppsala University, Uppsala, Sweden

^c Faculty of Geodesy and Geomatics Engineering, K.N. Toosi University of Technology, Tehran, Iran

^d Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen, Groningen, The Netherlands

ARTICLE INFO

Keywords:

Reinforcement learning

System disturbances

Traffic signal control

Microscopic traffic simulation

ABSTRACT

Traffic signal control plays a pivotal role in reducing traffic congestion. Traffic signals cannot be adequately controlled with conventional methods due to the high variations and complexity in traffic environments. In recent years, reinforcement learning (RL) has shown great potential for traffic signal control because of its high adaptability, flexibility, and scalability. However, designing RL-embedded traffic signal controllers (RLTSCs) for traffic systems with a high degree of realism is faced with several challenges, among others system disturbances and large state-action spaces are considered in this research.

The contribution of the present work is founded on three features: (a) evaluating the robustness of different RLTSCs against system disturbances including incidents, jaywalking, and sensor noise, (b) handling a high-dimensional state-action space by both employing different continuous state RL algorithms and reducing the state-action space in order to improve the performance and learning speed of the system, and (c) presenting a detailed empirical study of traffic signals control of downtown Tehran through seven RL algorithms: discrete state Q-learning(λ), SARSA(λ), actor-critic(λ), continuous state Q-learning(λ), SARSA(λ), actor-critic(λ), and residual actor-critic(λ).

In this research, first a real-world microscopic traffic simulation of downtown Tehran is carried out, then four experiments are performed in order to find the best RLTSC with convincing robustness and strong performance. The results reveal that the RLTSC based on continuous state actor-critic(λ) has the best performance. In addition, it is found that the best RLTSC leads to saving average travel time by 22% (at the presence of high system disturbances) when it is compared with an optimized fixed-time controller.

1. Introduction

Traffic control is of a vital importance in densely populated cities. The ineffective control of traffic can cause significant costs for drivers owing to the increased wasted time, negative effects on the environment due to vehicle emissions and detrimental impacts on the economy due to increased fuel consumption [1]. The main components of a traffic control system are traffic signal control, ramp-metering, variable speed limit enforcement, and dynamic route guidance [2]. Within such a context, scheduling of traffic signals is a critical traffic control challenge. Inefficiency in traffic signal timing attributed to the inability of adapting to prevailing traffic conditions leads to different small congested areas that can in turn cause larger traffic jams [3]. In recent years, computational intelligence techniques such as fuzzy logic [4–6],

neural networks [7,8], and reinforcement learning (RL) [9–11] have shown their potential for designing adaptive traffic signal controllers. In this paper, RL [12] is applied because of its online learning ability to gradually improve its performance, its adaptability to different traffic states as well as its ability to work without knowing an explicit model of the stochastic traffic environment. An RL-embedded traffic signal controller (RLTSC) has the capability to learn through experience by dynamically interacting with the traffic environment in order to reach its objective. Each RLTSC examines different green time durations (actions) in different traffic situations (states) and determines the best sequence of them based on the received scalar reward signals (feedback) which indicate the quality of the selected action. Each RLTSC which controls one intersection learns over time to obtain a signal timing plan that optimizes the sum of rewards in the future (return).

* Corresponding author at: Department of Industrial Development, IT and Land Management, University of Gavle, Gavle, Sweden.

E-mail addresses: mohammad.aslani@hig.se, moh.aslani@gmail.com (M. Aslani), Stefan.Seipel@hig.se (S. Seipel), mesgari@kntu.ac.ir (M.S. Mesgari), m.a.wiering@rug.nl (M. Wiering).

<https://doi.org/10.1016/j.aei.2018.08.002>

Received 21 June 2017; Received in revised form 15 July 2018; Accepted 14 August 2018

Available online 08 October 2018

1474-0346/ © 2018 Elsevier Ltd. All rights reserved.

RL algorithms can be either model-based [13] or model-free [9,14,15]. Model-based approaches need to initially employ and/or learn the environmental model (i.e. traffic system) in order to compute the optimal policy (signal timing plan). On the other hand, a model-free approach (e.g. SARSA, Q-learning, and actor-critic) does not rely on the estimation of the environmental model; instead, it progressively acquires the optimal policy by interacting with the environment and getting experience [16]. Both approaches have their strengths and weaknesses regarding their convergence guarantees, convergence speed, and ability to plan [12]. However, in the model-based approaches, obtaining an accurate model of the environment can be challenging and a slight bias in the model may lead to a strong bias in the policy. Thus, we employ model-free approaches, including Q-learning, SARSA, and actor-critic, which also makes the research more appealing for field deployment.

Conventional model-free RL algorithms require storing distinct estimations of each state-action value (for SARSA and Q-learning algorithms) or each state value (for the critic part of actor-critic algorithms) in lookup tables. Although they are computationally less demanding, their learning process can be slow when the state-action space is large. The reason is that the agent needs to experience all possible states. In this context, there are a couple of solutions for coping with a large state-action space. A first step to the solution is to employ continuous-state RL. In this version of RL, the knowledge gained from observations in each state (traffic condition) is applied to similar states by means of function approximators [17]. The application of previously acquired knowledge to unseen states generally leads to faster convergence. To handle the complexity of the state space, the state space is first mapped onto a feature space by using a feature function. Then the values of states are approximated in the feature space, instead of the original state space. Finding good function approximators of appropriate complexity is the key to the success of continuous state RL. The second solution is to reduce the state-action space by efficiently removing unnecessary and less effective state variables and actions in order to decrease the number of interactions needed to learn the optimal timing plan. An ineffective reduction of the state-action space may lead to undesired results because it cannot provide the agent with the required information that might be useful in decision making. Thus, it should be investigated which state variables and actions based on the conditions of the study area should be included in the state-action space so that optimal signal timing plans can be learned successfully. However, even if some trivial state variables are eliminated, the state-action space can still be too large for successfully being handled in discrete state RL. The third solution which is the combination of solutions 1 and 2 uses both state-action space reduction and continuous state RL. The first contribution of the present paper is to adopt all these three solutions and compare their results.

Regarding function approximation in continuous state RL, there are several different types of function approximators that are categorized into linear and non-linear function approximators. The values of states are represented by a weighted linear sum of a set of non-linear extracted features in linear function approximators or are sometimes represented by non-linear approximators such as neural networks. While non-linear function approximators may approximate an unknown function with better accuracy, linear function approximators are better to understand, simpler to implement, and faster to compute [18]. Also, linear function approximators are able to estimate non-linear value functions due to the employed basis function [12]. Because of the advantages of linear function approximators, linear function approximators are employed in this paper. A popular method to extract features in a linear function approximator is tile coding that splits the state space into separate tiles and assigns one feature to each tile [19]. Striking an empirical balance between representational power and computational cost is one of the most important advantages of tile coding that makes it suitable to be employed in this research.

In this article, a detailed empirical study of traffic signals control in

downtown Tehran through seven discrete and continuous state RL algorithms, namely discrete state Q-learning(λ), discrete state SARSA(λ), discrete state actor-critic(λ), continuous state Q-learning(λ), continuous state SARSA(λ), continuous state actor-critic(λ) [12,18], and continuous state residual actor-critic(λ) [20] is conducted. The traffic congestion in downtown Tehran is very heavy. Its traffic control systems are old-fashioned and most traffic signals are still manually controlled by police officers that can lead to inefficient traffic control.

One of the most significant challenges in designing the RLTS, is the robustness of them against different system disturbances that almost always happen in real-world applications. System disturbances, based on their intensity level, can disturb either the normal performance or even the convergence of the system. We consider three different kinds of system disturbances: jaywalking, incidents, and sensor noise. In the study area, observation surveys illustrated that many pedestrians prefer making the illegal crossing to waiting for the green pedestrian signal. Jaywalking which is performed by impatient pedestrians during the red pedestrians' signal increase the stochasticity of the traffic environment and disturb the performance of the RLTSs. Incidents as a nonrecurring traffic congestion cause [21] that disturb the traffic flow, make unexpected delays in the movements of the vehicles and consequently make the traffic environment more nonstationary for the RLTSs. Moreover, due to the obsolescence of the traffic control infrastructures, traffic signal sensors can be noisy and imperfect. In fact, the RLTSs' observations of the state as well as the reward signal are noisy. That is, the observed number of vehicles waiting on the approaching streets are different than their true values. Sensor noise, unlike the first two factors that have an indirect effect, directly disrupts the RLTSs' performance. This represents a difficult class of learning problem owing to the stochastic nature of the traffic environment together with high-order dynamics (i.e. variable traffic flows) and sensor noise. The RLTSs should be able to autonomously respond to a changing environment with stochasticity and random shocks. In the literature, different RLTSs have been investigated, but without thoroughly examining the robustness of them against system disturbances. Thus, another contribution of this paper is the comprehensive investigation of the RLTSs' robustness to the system disturbances. In this context, a real-world microscopic traffic simulation of the upper downtown core of Tehran city is carried out. In this simulation, a three-dimensional traffic network (i.e. slope of the streets are considered) with real changing traffic flows over 6 h in the morning (6:00 am to 12:00 pm) is considered.

Outline of this paper. The remaining part of this paper is organized as follows: Section 2 reviews the related work and summarizes the gaps in the existing literature. Section 3 describes the operation of discrete and continuous state RL. Section 4 demonstrates the traffic network and microscopic traffic simulation of the central area of Tehran. Section 5 technically explains the design of the RLTSs and the way they work. System disturbances, namely incidents, jaywalking, and sensor noise are explained in Section 6. Section 7 describes a series of experiments and their results and Section 8 provides a discussion of our findings. Finally, Section 9 concludes the paper and proposes some directions for future work.

2. Related work

Traffic signal control has been one of the major challenges for controlling traffic congestion in urban areas. Different systems and methods based on complex mathematical models have been proposed in order to optimize traffic signal parameters in response to diverse traffic situations. The Split Cycle Offset Optimization Technique (SCOOT) [22] and the Sydney Coordinated Adaptive Traffic System (SCATS) [23] are two successful commercial systems that have been installed in more than one hundred cities worldwide. Although these systems have alleviated traffic congestion somewhat, they may not be efficient in handling the traffic networks without well-defined traffic flow patterns (e.g. morning and afternoon peak hours). Also, they

assume a default value for the saturation flow that rarely takes place in real-world cases due to incidents, special events, and inclement weather conditions. Similar frameworks that are, however, decentralized are Optimization Policies for Adaptive Control (OPAC) [24], PRODDYN [25], and the Real-Time, Hierarchical, Optimized, Distributed, and Effective System (RHODES) [26]. These systems are based on upstream vehicle detection and a reliable estimation/prediction of queue length and traffic flow. They need a reliable measurement of the temporal and spatial variation in traffic flow. Also, in the case of any disturbance (e.g. incidents) their performance decreases significantly. Generally, they work properly when the traffic is rather homogeneous and may face challenges in heterogeneous traffic patterns [27].

Murat and Gedizlioglu [28] developed an adaptive traffic signal controller based on a fuzzy inference system [29–31]. A fuzzy inference system includes knowledge-base that stores the available knowledge about the problem in the form of fuzzy if-then rules and fuzzy membership functions. In their research, the knowledge-base was manually determined by experts' knowledge and field studies. Their proposed controller consists of two parts; the first part determines the green time duration and the second part arranges the phase order. It was benchmarked against an actuated controller in an isolated intersection and the results show that it has the best performance. However, converting experts' knowledge into fuzzy rules and fuzzy membership function is a difficult task and may lead to incomplete, unnecessary and conflicting knowledge due to the complexity arising in the traffic environment.

Eriksen et al. [32] used UPPAAL STRATEGO tool to develop a controller for an isolated signalized intersection in Køge city of Denmark. Traffic signal controller was modeled as a network of multiple stochastic timed automata and UPPAAL STRATEGO was employed to find the best strategy [33]. The timed automata network represents the engineering knowledge of traffic signal control. SUMO was used for microscopic traffic modeling of the intersection. The proposed method was benchmarked against fixed-time and actuated controllers in three different traffic loads: low, medium, and high. The results revealed that the presented controller outperforms other controllers in high and medium traffic loads in terms of average delay time and queue length. However, the proposed method may face difficulties in handling heterogeneous traffic loads. Due to the above limitations and increasing volume of traffic every year; more flexible, robust, and computationally efficient approaches for controlling traffic networks of multiple controllers are necessary. In this context, RL [34,18,35] that has served different roles in automating and supporting knowledge-intensive tasks in different engineering fields such as civil [36,37], transportation [9,14,38,39], architecture [40], and control [41–43] can be beneficial. RL is a promising approach that provides traffic signal controllers with the ability of self-learning and can address the mentioned shortcomings. It enables traffic controllers to learn online to continuously improve their performance over time as well as adapt to the changing traffic environment [44]. In each RL-TSC, the required knowledge for optimal traffic signal control is automatically learned by interacting with the environment and stored in state values and/or action values.

The advantages of using RL, particularly discrete state Q-learning, to control an isolated two-phase signalized intersection were demonstrated by Abdulhai and Kattan [9]. An RL-embedded controller was benchmarked against a fixed-time controller and the results showed that the proposed controller either performs on a par with the fixed-time control scheme on constant and uniform flows or slightly outperforms the fixed-time controller for the variable traffic flow case. Wen et al. [45] proposed an adaptive traffic signal controller based on discrete state SARSA(λ) to control a four-legged signalized intersection. The performance of SARSA(λ) was compared with fixed-time and full-actuated controllers and the empirical results indicated that the proposed controller significantly outperforms them especially in over-saturated traffic demand. El-Tantawy and Abdulhai [46] employed discrete state Q-learning to control a simulated multi-phase intersection of downtown Toronto. The authors studied the effect of different state

definitions on the performance of the adaptive traffic signal controllers. They showed that the adaptive traffic signal controller based on Q-learning consistently outperforms a fixed-time controller regardless of the state definition. El-Tantawy et al. [10] extended their previous work from an isolated signalized intersection to multiple signalized intersections. The performances of two RL algorithms, discrete state Q-learning(λ) and SARSA(λ), were evaluated to control multiple intersections in a simulated traffic network of downtown Toronto with real-world settings. Also, the design of three different state definitions and four different reward definitions were outlined. Experimental results revealed that RL-embedded traffic signal controllers using the number of arriving vehicles to the green phase and queue length at the red phase for the state definition as well as a cumulative delay for the reward definition led to the best results.

Dusparic et al. [47] employed distributed W-learning proposed by Dusparic and Cahill [48] in order to control six signalized junctions of Cork city center simulated in VISSIM [49]. Distributed W-learning is an RL-based method for collaborative optimization that uses Q-learning and W-learning [50]. In the proposed method, each traffic signal considers not only the locally best action but also the actions suggested by the neighboring traffic signals and performs an action with the highest importance in terms of the defined objectives. The order of the queue length on each approaching street is used for the state space definition. The increase or decrease in the total number of vehicles waiting at the intersection in comparison to the previous time step is also included in the state space definition. Moreover, the action of each traffic signal is the phase that should be in effect next. The authors utilized Boltzman exploration [12] in order to balance between exploration and exploitation. The proposed method was evaluated against SCOOT in three different traffic flows: low (2–4 am), medium (10–12 pm), and high (5–7 pm). The results showed that the proposed adaptive traffic signal controller outperforms SCOOT in all three different traffic flows regarding the average number of stops.

Wiering [13] proposed a framework based on discrete state model-based RL to control multiple traffic signals. Their framework is car-centric where the traffic signals make a decision based on the information (waiting time) received from each vehicle. The value function that estimates the expected waiting times of vehicles are learned by both traffic signals and vehicles. Vehicles are also able to learn the policies for choosing the optimal paths. The results showed that their method clearly outperforms the designed fixed-time controller. Bakker et al. [51] extended Wiering's framework [13,52] by employing coordination graphs to describe the dependencies between the neighboring traffic signals. In order to find the optimal joint actions between neighboring traffic lights, the max-plus algorithm was used. The proposed method was tested on three hypothetical 2D traffic networks of different sizes. The authors reported that the max-plus algorithm outperforms Wiering's method. Wiering's framework was also extended by proposing a multi-objective traffic signal control system that optimizes the weighted sum of several reward functions [2]. The authors employed the Bayesian approach to estimate the parameters of the underlying multi-agent Markov Decision Process in order to increase the stability and robustness of the system against the changing environment dynamics. A continuous driver's behavior in the space and time dimension is another positive feature of their work. The authors indicated that the proposed approach outperforms TC-1 proposed by Wiering [13].

In traffic control problems with continuous state-action spaces or large state-action spaces, employing discrete state RL may result in a poor performance of the system. One of the fundamental ways to address this challenge is to integrate RL with function approximators [12] that generalize the knowledge gained from observations in each state to similar states. In this context, Richter et al. [53] employed the natural actor-critic algorithm to control signalized intersections of a 10×10 hypothetical and simplified traffic network. Each traffic signal received a local reward along with additional information from its neighboring

intersections. The results revealed that the natural actor-critic algorithm managed to outperform SAT (adaptive controller inspired by SCATS) and a vanilla policy-gradient algorithm [54]. Prashanth and Bhatnagar [55] proposed two methods to handle the large state-space challenge, Q-learning with function approximation and Q-learning with feature-based representation. The state space consists of queue lengths on different lanes and elapsed time for the red signal. The objective of the system is to minimize the queue lengths, while at the same time ensuring fairness so that no lane has a long red time. They benchmarked their proposed method against fixed-time, longest queue, and SOTL from [56] as well as discrete state Q-learning [9] on four hypothetical traffic networks. The authors reported that the proposed Q-learning algorithm with function approximation consistently outperforms other methods. In Pham et al. [57], a comparison between two methods, continuous state SARSA and distributed coordination of exploration and exploitation [58], in controlling traffic signals was made. Tile coding was employed as a method of approximating the value function for the states. The authors tested both methods on a simplified traffic network with 4 signalized intersections. The results revealed that continuous state SARSA has the best performance.

In most related work, the performance of adaptive traffic signal controllers based on RL were evaluated in hypothetical traffic networks with assumptive settings [13,52,2,55,57,53] and the effects of disturbance factors that exist in the real-world traffic environment were overlooked [46]. Also, the performance of a few RL algorithms, mostly algorithms based on Q-learning or SARSA, were assessed and compared. The applicability of RLTSs needs to be further evaluated before field trials can be considered, to ensure they are flexible enough. In this research, we design different practical traffic signal controllers based on seven discrete and continuous state RL methods and evaluate their performances in the realistically simulated traffic network of downtown Tehran. The robustness of the RLTSs against system disturbances is assessed in order to make sure that they are able to cope with all kinds of scenarios.

3. Discrete and continuous state Q-learning(λ), SARSA(λ), and actor-critic(λ)

RL, as a major branch of machine learning, is an approach for learning an optimal policy of an agent by interacting with its surrounding environment. The optimal policy is defined as the policy that receives the highest expected discounted cumulative reward (return). Learning the optimal policy in value-function-based RL [12] needs to estimate value functions. Value functions are either state-values (i.e. value of each state) or action-values that are also called Q-values (i.e. value of taking an action in a given state). These values represent the agent's knowledge from its environment. State-values estimate "how good" it is for the agent to be in a given state s and action-values estimate "how good" it is to choose a given action a in a given state s . The value of the state s under a policy π can be defined by Eq. (1).

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\} \quad (1)$$

In Eq. (1), $V^\pi(s)$ is the state-value that corresponds to the expected discounted cumulative reward (i.e. return) starting in state s and then following policy π . The factor r_{t+k+1} is the reward obtained when arriving to states, etc. $\gamma \in [0, 1)$ is a discounting factor that determines to what degree future rewards affect the value of the state s . The value of an action a in the state s is defined by Eq. (2), where $Q^\pi(s, a)$ is the state-action value, that corresponds to the expected return when starting in state s , taking action a and following policy π thereafter.

$$Q^\pi(s, a) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\} \quad (2)$$

In classical RL (i.e. discrete state RL), value functions are represented in a tabular form. In the tabular form, the state-values or action-values are stored in a look-up table and there is one output value for each input. Storing and updating the values are simple, intuitive, and fast in the tabular form. A well-known method for estimating value functions is temporal difference learning that adjusts the estimated value of a state based on the immediate reward and estimated value of the next state [59].

Currently, there are three well-known classical RL algorithms for learning the optimal policy based on temporal differences: Q-learning [60], SARSA [61,12], and actor-critic [62]. Q-learning is an off-policy RL algorithm that updates action-values on the basis of the best action. In Q-learning, as long as the agent visits all the state-action pairs infinitely often no matter how it explores, it converges to the optimal action-values. Q-learning is able to (at least conceptually) separate exploration from control [63]. SARSA is an on-policy RL algorithm that updates action-values on the basis of the experience obtained from following some policy. In SARSA, the agent should explore all the state-action pairs enough, and then exploit the obtained knowledge. Another on-policy method is actor-critic that, unlike Q-learning and SARSA, has a separate data structure to estimate state-values (the critic) and a mapping from states to a preference value for each action (the actor). An advantage of actor-critic is that in the case of having many actions there is no need to consider all actions' Q-values to select one of them [16].

Conventional Q-learning, SARSA, and actor-critic update the value functions with a one-step backup, whereas the received reward is often the result of a series of steps. Within such a context, eligibility traces [59] that propagate rewards back to the earlier visited states, diminished by a decay parameter for each step back, can be beneficial. Applying the eligibility traces concept to the Q-learning, SARSA, and actor-critic methods results in the Q-learning(λ), SARSA(λ), and actor-critic(λ) algorithms in which λ is the decay parameter [12].

Discrete state RL algorithms are capable of handling a limited number of states and actions and they suffer from long learning times when the state-action space is large or continuous, due to the need to explore all possible states separately. In fact, the tabular-based RL methods scale-up poorly to control problems with large and continuous state-action spaces. One of the fundamental ways to handle this challenge is the integration of RL algorithms with linear function approximators for learning the value functions. The linear function approximation approach can learn non-linear value functions due to the used basis function (e.g., tile coding). Linear function approximation takes samples from the desired value-function and tries to construct an approximation of the entire function by generalization. In a practical point of view, there is a natural metric on states such that close states have similar values so that the agent is able to deal with states never exactly experienced before and it can learn efficiently by generalizing from previously experienced states. In linear function approximation, the state-values are commonly approximated as a weighted linear sum of a set of features computed from the state variables (Eq. (3)).

$$V(s_t) = \theta^T \cdot \phi(s_t) = \sum_{i=1}^m \theta_i \phi_i(s_t) \quad (3)$$

where θ is a weight vector for state-values and $\phi(s_t)$ is a feature function that maps the state s_t into a vector $\langle \phi_1(s_t), \phi_2(s_t), \dots, \phi_m(s_t) \rangle$, where m is the total number of features and $\phi_i(s_t)$ is i -th feature of the feature function. Learning entails finding the weight vector θ corresponding to approximate optimal state-values. Linear function approximation possesses a quadratic error surface with a single minimum that can be found by gradient descent [64,12]. Moreover, linear function approximators are able to estimate very complex value functions because the mapping function itself can be arbitrary complex. Similarly, action-values are calculated by Eq. (4). This equation represents the mapping from state-action pairs to action-values through a parameterized

function.

$$Q(s_t, a_t) = (\theta')^T \cdot \Phi(s_t, a_t) \quad (4)$$

In this equation, θ' is a set of scalar weights and $\Phi(s_t, a_t)$ is a mapping function that is defined according to Eq. (5). Both θ' and Φ are $(m \times g)$ -dimensional vectors where g is the number of actions.

$$\begin{aligned} \Phi^T(s_t, a_t) &= [\varphi_1(s_t) \cdot b_1, \dots, \varphi_m(s_t) \cdot b_1, \varphi_1(s_t) \cdot b_2, \dots, \varphi_m(s_t) \cdot b_2, \dots, \varphi_1(s_t) \cdot b_g, \dots, \varphi_m(s_t) \cdot b_g] \\ b_i &= \begin{cases} 1, & \text{if } a_t = a_i \\ 0, & \text{if } a_t \neq a_i \end{cases} \end{aligned} \quad (5)$$

Choosing the right feature function type is critical for successful learning. Among different feature functions employed in linear function approximators, tile coding is one of the most exploited techniques. This approach approximates the value function by using a piecewise-constant function [19]. It divides the state space into a set of overlapping partitions called tilings. Each tiling consists of a set of non-overlapping tiles. Although the tilings could be located at random; in this research, they are located to cover the whole state space regularly. Each state variable is partitioned into the same number of tiles and then the tiling (grid) is created by combining the tiles for each state variable separately. In addition, the tilings are slightly offset in each dimension so that a particular state is mapped to different tiles in different tilings. Let DIM be the dimension of the state space, p_j is the number of tiles on the j -th dimension, and TL is the number of tiling layers. The membership value of the triggering state ($\varphi_i(s_t)$) to different tiles is either 0 or 1 and calculated by Eq. (6), where $\varphi_i(s_t)$ is the feature value of the i -th tile and m is the total number of tiles (features).

$$\begin{aligned} \varphi_i(s_t) &= \begin{cases} 0 & \text{if } s_t \notin \text{tile}_i \\ 1 & \text{if } s_t \in \text{tile}_i \end{cases} \quad 1 < i < m, \\ m &= (p_1 \times p_2 \times p_3 \times \dots \times p_{DIM}) \times TL \end{aligned} \quad (6)$$

The general form of updating θ in the employed continuous state RL methods is based on Eq. (7), where α is the learning rate, δ is the temporal difference error [59], e_{t+1} is the eligibility trace at time $t + 1$, and $\nabla_{\theta} V(s_t)$ is the gradient of $V(s_t)$ with respect to θ .

$$\begin{aligned} e_0 &= 0 \\ e_{t+1} &= \gamma \lambda e_t + \nabla_{\theta} V(s_t) = \gamma \lambda e_t + \phi(s_t) \\ \delta_{t+1} &= r_{t+1} + \gamma \phi_{t+1}^T \cdot \phi(s_{t+1}) - \phi_t^T \cdot \phi(s_t) \\ \theta_{t+1} &= \theta_t + \alpha \delta_{t+1} e_{t+1} \end{aligned} \quad (7)$$

The integration of RL algorithms with function approximation may be unstable and lead to poor results [65]. The residual algorithm [20] was proposed to address this concern. The residual algorithm is the direct application of gradient descent to the problem of minimizing the mean squared Bellman residual. This method updates both states s_t and s_{t+1} to achieve the convergence on the mean squared Bellman residual. Eq. (8) shows the general form of updating θ in the residual algorithm.

$$\begin{aligned} e_{t+1} &= \gamma \lambda e_t + (\nabla_{\theta} V(s_t) - \beta \gamma \nabla_{\theta} V(s_{t+1})) = \gamma \lambda e_t + (\phi(s_t) - \beta \gamma \phi(s_{t+1})) \\ \theta_{t+1} &= \theta_t + \alpha \delta_{t+1} e_{t+1} \end{aligned} \quad (8)$$

In Eq. (8), $\beta \in [0, 1]$ attenuates the effect of the successor state (s_{t+1}). In this research, β is adapted during the learning process [66]. The residual algorithm can be incorporated into Q-learning(λ), SARSA(λ), and actor-critic(λ). However, due to space limitations, we just employ the combination of the residual algorithm with actor-critic(λ) which we call continuous state residual actor-critic(λ).

4. Microscopic traffic simulation of the study area, downtown Tehran

Tehran city, the capital of Iran, which is located in the northern part of the country, includes an irregular traffic network with the total

length of 2700 km, of which 14% are highways and 27% are arterial roads [67]. Every day more than 15 million trips on the network with an average speed of 21 km/hour per vehicle are made [68]. Tehran has been experiencing a rapid population growth in recent decades. A nearly fivefold increase in the population of the city in 55 years has made the city the most densely populated area in Iran [69]. It is also considered as a crowded city according to the international standard because of its average density of 146 people per hectare [70]. This rapid population growth of the city has led to the increase in the number of vehicles from 5 vehicles per 1000 inhabitants in 1956 to 90 cars per 1000 inhabitants in 2011 [71]. Since the growing trend of the number of vehicles has not been proportional to the development of the traffic network, traffic congestion has become an endemic and destructive phenomenon in Tehran.

A traffic network located in the heart of the financial district of downtown Tehran is selected as the study area (Fig. 1). The traffic congestion in this area is so heavy due to the high concentration of tall buildings and businesses. As shown in Fig. 1, the traffic network consists of four major arterials, Motahari, Beheshti, Shariati, and Modares, and many intersections. Motahari, Beheshti, and Shariati are one-way arterials that run eastbound, westbound, and northbound respectively. Modares is a two-way arterial that runs north/south and has interchange connections to Beheshti and Motahari arterials. In this research, AIMSUN¹ is employed to build the testbed network. AIMSUN is a microscopic traffic simulation software program that is able to simulate the individual driver decisions based on various theories of driver behaviors such as car following, gap acceptance, and lane changing [72] to capture interactions between complex travel demand patterns and network supply phenomena. It describes vehicle movements as the result of a vehicle's interaction with other vehicles and with the road infrastructure based on the traffic rules. In order to carry out a microscopic traffic simulation the following information should be set: (a) traffic network and its characteristics, (b) traffic demand data, and (c) driver behavior parameters.

The traffic network and road characteristics such as speed limits on the sections and turnings as well as the capacity of the streets were entered into the software with a reasonable level of detail. These data were calculated and calibrated by the Tehran Comprehensive Transportation and Traffic Company². Moreover, the elevations of nodes in the traffic network were collected from the digital surface model of the study area and entered in AIMSUN. Considering the nodes elevations of the traffic network is for improving the quality of the microscopic traffic simulation as it affects the velocity of the simulated vehicles. It has to be underlined that it does not have bearing on the effectiveness of the RLTSs. Also, four vehicle types, because of their high proportions in the national fleet, are selected to be modeled: Pride, Peugeot, Peugeot 206, and Samand. They comprise more than 78% of all the vehicle types used in the study area. Moreover, in order to simulate pedestrian crossings, pedestrians walking facilities such as central refuge, crosswalk, and curb waiting areas are added to the traffic network. Jaywalkers count data for each intersection were also collected by fieldwork in this area. Regarding traffic demand data, the traffic demands over 6 h in the morning (between 6:00 am and 12:00 pm) on 26 April 2014 (workday) are defined in the form of 6 origin-destination matrices calculated by the municipality of Tehran. These 6 origin-destination matrices reflect the variations of traffic demands in the morning.

Driver behavior parameters that reflect the drivers' personalities in decision making in different traffic situations are also important in microscopic traffic simulation. These parameters affect the desired vehicles' movements such as lane changing, overtaking, traveling speed,

¹ Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks.

² <http://trafficstudy.tehran.ir/>.

Fig. 1. Traffic network.

and holding a safety gap to other vehicles. The most effective parameters on the traffic simulation, such as reaction time (sec), reaction time at stop (sec), minimum headway (sec), and speed acceptance (degree of acceptance of speed limits), were calibrated to bring the simulation closer to the observed behavior of drivers [73]. Table 1 shows the calibrated values of these parameters. It should be highlighted that we included as much detail (e.g. traffic demand, slope of the streets, and number of lanes) for which data is available in the simulation to ensure the results, to some extent, are reliable.

5. RLTSs: RL-embedded traffic signal controllers

In traffic signal timing design, cycle and phase play a critical role. A traffic signal cycle is one complete sequence of signal directions and a phase is the part of a cycle allocated to any combination of non-conflicting traffic movements. Each movement must be served by at least one phase of the cycle [74]. Each signalized intersection utilizes an RLTS to adapt the green time allocated to each phase in response to the current traffic situations. At the beginning of each phase of the cycle, the RLTS senses the local prevailing traffic condition (i.e. environment state) represented by a $[Ph, T, A_1, A_2, \dots, A_w, D_1, D_2, \dots, D_v]$ feature vector. The first component (Ph) is the index of the current green phase of the junction, the second component (T) is the current time of the day, w and v are the number of approaching and departing streets of the junction, $A_1, A_2, \dots, A_w$ show the number of vehicles

waiting on each approaching street, and $D_1, D_2, \dots, D_v$ indicate the number of vehicles waiting on each departing street connected to the neighboring intersections. The state space of each RLTS is $2 + v + w$ dimensional. For example, the RLTS which controls a junction with 4 approaching streets and 4 departing streets has a state space with the dimension of 10 in which the first two elements are the index of the current green phase and time of the day and the rest of the elements are the number of waiting vehicles on the approaching and departing streets. Since the traffic demand varies over time, considering the time variable (T) in the state space would seem to be helpful in dealing with the flow variations. Moreover, departing links variables (D_i) can provide the traffic signal controller with the ability of handling the spill-back phenomenon and indirect cooperation with the neighboring traffic signals.

After sensing the local traffic condition, the RLTS selects a green time duration, i.e., a value from [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70] s. Once a green time is selected, the RLTS waits to the end of the current phase duration which is the summation of the green and yellow interval. Then, it receives a scalar reward signal from the stochastic traffic environment based on how close the selected green time satisfies the RLTS's objective. The scalar reward signal provided to the RLTS is defined as the negative total number of vehicles waiting on all approaching streets to the associated junction. The received reward is then used to update the value (worthiness) of the selected green time. The value of each green time informs the RLTS how beneficial it

is to select the green time in the current traffic situation in order to satisfy the defined objective. The value of each green time is initially set to zero to represent that the RLTSs do not possess any prior knowledge of the stochastic traffic environment. Lacking any knowledge of the stochastic traffic environment means that the RLTSs will initially explore all green times in all different traffic situations by making random green time selections. The RLTSs iteratively interact with the stochastic traffic environment by selecting different green times, receiving rewards and updating the values of the green times to better reflect the true value of the actions, and to improve the ability of the RLTSs to make informed decisions. In this paper, the values are updated through the mentioned discrete state or continuous state RL methods. In all the employed continuous state RL, tile coding as an effective feature function is used. The fast and effective learning process through tile coding depends on choosing the right number of tilings and tiles. The nearly optimal number of tilings and tiles in each dimension, by using trial and error, are found to be 2 and 5 respectively. With these values, there would be a trade-off between speed and precision of the learning process.

Algorithm 1. Continuous state RLTS based on tile coding for updating state-values

```

1: Initialize  $\alpha, \gamma, \vec{\theta} = \vec{0}$ , and  $\vec{e} = \vec{0}$ 
2:  $t \leftarrow 0$  // time step
3: loop
4:  $s_t \leftarrow$  Initial state of the episode
5:  $F \leftarrow$  Set of active-tiles present in  $s_t$ 
6: repeat
7:  $a_t \leftarrow$  action given by  $\pi$  for  $s_t$ 
8:  $\vec{e} \leftarrow \lambda \gamma \vec{e}$ 
9: for all  $i \in F$  do
10:  $e(i) \leftarrow e(i) + 1$ 
11: end for
12: Set the current phase duration to  $a_t + \text{yellowInterval}$ 
13: Wait to the end of the phase
14:  $Ph \leftarrow$  Index of the current green phase
15:  $T \leftarrow$  Current time of the day
16: for  $i = 1$  to  $w$  do
17:  $A(i) \leftarrow$  Number of vehicles waiting on  $i$ -th approaching street
18: end for
19: for  $i = 1$  to  $v$  do
20:  $D(i) \leftarrow$  Number of vehicles waiting on  $i$ -th departing street
21: end for
22: Calculate reward  $r_{t+1} = -\sum_{i \in w} A(i)$ 
23:  $\delta_{t+1} \leftarrow r_{t+1} - \sum_{i \in F} \theta(i)$ 
24: Estimate  $s_{t+1}$  based on  $Ph, T, A$ , and  $D$ 
25:  $F \leftarrow$  Set of active-tiles present in  $s_{t+1}$ 
26:  $\delta_{t+1} \leftarrow \delta_{t+1} + \gamma \sum_{i \in F} \theta(i)$ 
27:  $\vec{\theta} \leftarrow \vec{\theta} + \alpha \delta_{t+1} \vec{e}$ 
28:  $t \leftarrow t + 1$ 
29: until  $s_t$  in terminal
30: end loop

```

Fig. 2 illustrates the arrangement of tiles and tilings on the j -th dimension in this research. In order to use tile coding, first, each state variable is normalized between 0 and 1, then the whole state space is covered by two sets of tiles (two tilings). Each tiling is offset 0.1 from another one on each dimension and the width of each tile is 0.2. Each

Table 1

The most effective parameters of the traffic simulation.

Parameters	Calibrated value
Reaction time (sec)	0.9
Reaction time at stop (sec)	1.2
Minimum headway (sec)	[1.5–2.53]
Speed acceptance	[1–1.3]

Fig. 2. The projected tiles and tilings on the j -th dimension of the state space that is $2 + v + w$ dimensional. s_t^j is the j -th dimension of the current state.

tile has a value, $\varphi_i(s_t)$, and a weight, θ_i (and/or a separate weight for each distinct action, θ_i^j). The value of a tile is 1 if the current state (s_t) falls in the region delineated by that tile, otherwise its value is zero (please refer to Eq. (6)). The weight of each tile is tuned during the learning process (please refer to Eqs. (7) or (8)). The approximated value of a given state (state-action) is the sum of the weights of all the activated tiles (please refer to Eqs. (3) or (4)). The way how state-values are updated in continuous state RLTSs is described in Algorithm 1.

The simulation is repeatedly carried out for 6 h in the morning and each 6 h is referred to as an episode. The whole simulation consists of 80 episodes. As the number of episodes increases, the RLTSs rely less on exploration through random green time selection and begin to exploit their obtained knowledge of the traffic environment by choosing those green times which possess a fairly high action value. The RLTS should strike a balance between exploration and exploitation in order to learn fast and efficiently. A straightforward and widely used approach to trade-off between exploration and exploitation is the ϵ -greedy method [16]. Through this method, the RLTS selects the green time that it believes has the best long-term effect with the probability of $1-\epsilon$, and it selects a green time uniformly at random, otherwise. ϵ is a tuning parameter to be changed during the simulation. The value of ϵ is initially high as the RLTSs do not possess sufficient knowledge of the traffic environment to make informed decisions. The value of ϵ declines to a value close to 0.0 over numerous episodes as the RLTSs' knowledge of the environment is raised. In this way, the RLTSs exploit what they have learned and choose only those green times that are most beneficial in terms of their long term objective. Specifically, the value of ϵ reduces from 0.8 to 0.0 during the first 70 episodes and then it is kept constant at 0.0 over the last 10 episodes (test period) in order to evaluate the learning performance of the system. Also, the discount factor is set to 0.99, which worked best in preliminary experiments.

Another effective fundamental parameter in learning performance is the learning rate (step size) that represents the rate of RLTS's learning throughout the course of the simulation as the RLTS discovers the characteristics of the traffic environment. Choosing the optimal value of the learning rate is prominent because a too small learning rate will lead to very slow learning and a learning rate which is too large will result in oscillations or divergence of the system. We have employed trial and error as the most reliable solution for finding the optimal learning rates of each RL method. Table 2 shows the selected learning rates. It should be noted that the explained RLTSs interact with AIMSUN through the application programming interface functions. Also, the RLTSs are implemented by C++ programming language.

Table 2
Learning rates obtained by trial and error for each RLTS.

Method of RLTSs	Learning rate
Discrete state Q-learning(λ)	0.25
Discrete state SARSA(λ)	0.3
Discrete state Actor-Critic(λ)	0.2
Continuous state Q-learning(λ)	0.0005
Continuous state SARSA(λ)	0.005
Continuous state Actor-Critic(λ)	0.25
Continuous state residual Actor-Critic(λ)	0.3

6. Incorporating system disturbances

Designing and implementing the RLTSs for real-world applications is not straightforward. A key issue that almost always happens for real-world traffic signal control and negatively affects the performance of the RLTSs is the occurrence of system disturbances. System disturbances can ruin the performance of the traffic signal controller and in turn, make the traffic congestion heavier. In this paper, system disturbances are considered to determine how the RLTSs handle the novel conditions. In fact, we evaluate the ability of the RLTSs to incorporate and adapt to the system disturbances and show how the adaptive decision-making ability of the RLTSs is affected by the disturbance events in the study area. Three different types of system disturbances are considered: (a) incidents, (b) jaywalking, and (c) sensor noise.

In the study area, due to the traffic congestion, incidents are likely to occur. Incidents (e.g. accidents, vehicle breakdowns, and spills) create nonrecurring congestions which relate to unexpected delays resulting in stop-start driving conditions and consequently increasing fuel consumption and air pollutants [75]. Sometimes incidents during the period of peak demand can make the traffic congestion more severe and several incidents within a short period of time at the bottlenecks of the traffic network can result in grid-lock. When an incident occurs, the main part of the street capacity is lost, the movement of traffic is disrupted, long queues are formed, a spillback phenomenon occurs, and particularly the traffic environment becomes more nonstationary for the RLTSs. Thus, the performance of the RLTSs while contending with the nonstationary environment should be assessed. It is worth noting that, the goal of incorporating different incident rates here is not

to recreate specific accidents or sudden events, but rather to determine if the RLTSs are able to logically react in the presence of the random shocks.

Pedestrians who jaywalk also play an important role in the traffic environment. According to a number of papers conducted on the pedestrians' crossing behavior [76–78], there are mainly two classes of pedestrians: (a) law-abiding pedestrians who only cross the streets during green man pedestrian signal, and (b) jaywalkers (impatient pedestrians) who may cross the streets when the pedestrian signal is red. Jaywalkers evaluate the gaps in the traffic streams to decide whether they are large enough to go through them safely. Since pedestrians who jaywalk trigger the vehicle reactions and the driver starts to slow down and/or changes the driving lane to give way to them, they have more interactions with vehicles than law-abiding pedestrians. Observation surveys showed that many pedestrians prefer jaywalking rather than waiting for green pedestrian signals [78,79]. In fact, the proportion of jaywalkers in comparison to law-abiding pedestrians is not small. Vehicle-jaywalker interaction happens where pedestrian volume is rather high [80]. Owing to the fact that the study area includes many commercial and administrative buildings that attract many people and, in turn, pedestrians, the vehicle-jaywalker interaction is observable daily there. Thus, illegal crossings of impatient pedestrians which disturb vehicle movements and consequently increase the stochasticity of the traffic environment are considered as the second system disturbance factor.

Another source of the system disturbance is the noise coming from the sensors. Data received from sensors in a real world traffic network can be affected by sensing errors due to, for example, the obsolescence of the existing traffic control infrastructure. Therefore, it is necessary to investigate whether the RLTSs perform well in the case of such inaccurate measurements. We add noise to the counted number of vehicles waiting on the streets to represent inaccuracy in counting that may occur in practice. In this manner, both state space and reward signals are affected by noise (see Fig. 3). The noise is drawn from a Gaussian noise distribution and added to the state variables representing a number of vehicles waiting on the approaching and departing streets. The noise can have positive or negative values for increasing or decreasing the counted values. However, the negative noise can decrease the measured number of vehicles to a minimum of zero. Although the true distribution is not perfectly Gaussian, this model is found to be adequate. As shown in Fig. 3, incidents, jaywalking, and

Fig. 3. System diagram and different parts of the RLTS. The RLTS receives through its sensors noisy data which is employed for the state and reward signal estimation. Based on the policy, the RLTS selects an action indicating the green time duration assigned for the current phase.

sensor noise are three elements which affect the performance of the system.

7. Experiments and results

Different experiments are conducted in order to achieve the following objectives: (a) evaluating the performance of different discrete and continuous state RL for controlling traffic signals in the traffic network of downtown Tehran, (b) assessing the robustness of the RLTSs against the above-mentioned system disturbances, and (c) identifying and removing trivial state variables in order to decrease the state space dimensionality and improve the learning performance of the RLTSs. In what follows, we discuss the individual experiments and corresponding results. In all experiments, three performance indexes, namely average travel time per vehicle (sec/km), average delay time per vehicle (sec/km), and average stop numbers (#/veh/km) are utilized in order to evaluate the performance of each traffic control system. Moreover, the average of 10 simulations per method is used for evaluating their learning performance.

7.1. Experiment 1: effect of eligibility traces

Using eligibility traces in updating the values of states and green times causes that the RLTSs look ahead all the way until the end of the episode rather than only one step in time. In this manner, all the rewards obtained later play a role in updating the values of states and green times by assigning more credit for recently visited states than those visited longer ago. As mentioned in Section 3, the decay parameter (λ) is between 0.0 and 1.0; with $\lambda = 0.0$ the current state is only eligible to be updated and with $\lambda = 1.0$ all the visited states are updated in each time step. Since there is no definitive proof to theoretically determine the best value of eligibility traces [81], the learning performance of seven different RLTSs each with the best learning rate and with different $\lambda = 0.0, 0.5$, and 0.9 are evaluated in order to find the best value of the decay parameter (Fig. 4). Interestingly, as shown in Fig. 4, the higher the value of λ , the better the performance of the actor-critic(λ) algorithms and the worse the performance of the Q-learning(λ) and SARSA(λ) algorithms. The performance improvement by reducing λ in discrete state Q-learning(λ) and SARSA(λ) algorithms was also concluded by El-Tantawy et al. [10]. It has to be highlighted that the performance of discrete state SARSA(λ) with $\lambda = 0.0$ and 0.5 are almost identical but $\lambda = 0.5$ slightly outperforms $\lambda = 0.0$. Regarding the different optimal values of λ , it can be inferred that the best value of λ not only depends on the problem at hand and RL design (e.g. state definition, reward definition, and action definition) but also depends on the type of employed RL method.

Table 3 compares the average performance, as measured by three performance indicators described earlier, of each RLTS over the last 10 episodes in which the policy is greedy. It is interesting to notice that Q-learning(λ) and SARSA(λ) algorithms are less sensitive than actor-critic(λ) to the value of λ , i.e. λ has a stronger influence on the performance of actor-critic(λ) in comparison to Q-learning(λ) and SARSA(λ). Also, the proper values of λ in all RLTSs result in the lower standard deviation of the performance indicators. This is due to the fact that the proper values of λ lead to a better convergence of the learning process. Among all different traffic signal controllers, continuous state residual actor-critic(λ) and continuous state actor-critic(λ) outperform others. The reason is that actor-critic keeps a separate policy independent of the value function. In this way, there is no need to consider all actions' Q-values in order to select one of them. Thus, it can lead to better performance in problems with large state-action spaces.

7.2. Experiment 2: effect of system disturbances

Since disturbances can cause unpredictable results, investigating the effect of them on the performance of the RLTSs is crucial. In this

experiment, it is supposed that the RLTSs have learned through a simulation environment without system disturbances and then they are deployed in the field in which there are incidents, jaywalkers, and sensor noise. In fact, we assume that the best policy is learned in the ideal traffic environment during the first 70 episodes and then the system disturbances are added over the last 10 episodes in order to see how the RLTSs perform while contending with the novel conditions. Two different levels of system disturbances, low and high, which consist of two levels of incident rates and sensor noise are defined in order to determine how they handle different disturbance levels (see Table 4). The impacts of the disturbances on the RLTSs' learning performance are provided in Fig. 5, from which some important points can be made.

The increasing or constant trends during the last 10 episodes of Q-learning(λ) and SARSA(λ), both discrete and continuous state, indicates the lack of robustness of these algorithms. By the way of contrast, the downward trends in the performance of actor-critic(λ) algorithms (discrete and continuous state) over the last 10 episodes proves the ability of them to adapt to the random shocks. Also, continuous state residual actor-critic(λ) has the best performance at the presence of the system disturbances. Among discrete state algorithms, only the discrete state actor-critic(λ) algorithm could improve its performance which shows the low robustness of discrete state Q-learning(λ) and SARSA(λ). Indeed, the RLTSs which can explore further after imposing random shocks can do well [82,83]. Table 5 compares the mean difference in performance between disturbance experiments (experiment 2) and the base case experiment (experiment 1) for the actor-critic(λ) algorithms over the last 10 episodes. It is evident that the continuous state actor-critic(λ) algorithm has the lowest mean differences and consequently the highest robustness property in comparison to others.

The noticeable point in this experiment is the reason why actor-critic(λ) algorithms over the last 10 episodes are able to explore and consequently adapt to system disturbances while following the greedy policy. The reason is that the values of most actions after episode 70 decrease because of the system disturbances and its effect on the traffic congestion. In fact, updated values of most actions prior to episode 70 (before adding disturbances) are higher than action values after this episode. As a result, once an action with the highest value is selected by the greedy policy, its value drops so that other actions with high values have the chance to be selected by the RLTS. In this manner, the RLTS even with the greedy policy can still explore some actions. Also, the critic's estimate provides the actor with the ability to update with gradients that have lower variance [84]. Actor-critic is able to react to smoothly changing states with smoothly changing actions. Moreover, as mentioned by Konda and Tsitsiklis [85], Grondman et al. [84], actor-critic methods usually have good convergence properties, in comparison to the critic-only methods (SARSA and Q-learning).

7.3. Experiment 3: effect of reduction in the size of the state-action space

The high dimensional state-action space can result in immature learning. In this experiment, it is determined which state variables and actions should be included in the RLTS's state-action space so that a better policy can be obtained and the size of the state-action space stays reasonable at the same time. It should be assured that the number of state variables and actions are not too large to enable the RLTSs to learn the policy with less training and is not also too small so that the state is detailed enough for mature learning. With this end in view, we start with our proposed initial state-action space and then present the effect of eliminating different state variables and actions. In the tests, all seven RLTSs with ϵ -greedy policy are utilized in order to identify the impact of reducing the size of the state-action space on the system performance. In the first test, the time variable (T) representing the current hour of the day is excluded from the state space. In this way, it can be evaluated whether considering T is beneficial in handling flow variations. The second test determines the impact of departing links variables ($D_1, D_2, \dots, D_r$) that are employed to provide the RLTSs with

Fig. 4. Average travel time (sec/km) for different RLTSs and for different values of the eligibility decay parameter. All the learning curves are the average of 10 simulations.

indirect cooperation abilities. Excluding departing links variables, on the one hand, results in a substantial reduction in the state space size, but on the other hand, leads to diminishing the indirect cooperation [27] between the traffic signals. In other words, the second test investigates the significance of indirect cooperation in comparison to the impact of low dimensionality of the state space. It determines whether considering the indirect cooperation between the traffic signals is essential regarding the topology of the simulated traffic network (i.e. fairly long distances between adjacent traffic signals). In the third test, the initial number of state variables is kept constant, but the number of actions declines from 14 to 7 (the new action set employed is [10, 20, 30, 40, 50, 60, 70] s). We investigate the impact of reducing the size of the action set while it also decreases the flexibility of the actions

(having actions with steps of 10 s is less flexible than 5 s). This allows us to determine whether the large number of different green times is crucial in traffic signal control. The forth test which is the combination of the former tests explores the impact of reducing the size of both state space and action set. In this test, we analyse the aggregate impact of removing time and departing links variables and decreasing the action set. The effect of removing other state variables such as Ph and $A_1, A_2, \dots, A_w$ are overlooked due to the obtained poor performance. Fig. 6 presents the learning performance of the RLTSs in different tests. In this figure the learning curves are compared to those with the initial state-action space. The following points can be concluded.

- None of the RLTSs with the initial defined state-action space leads

Table 3
Results of experiment 1 (results are averaged over 10 simulations).

RLTSCs		Average travel time (sec/km)	Average delay time (sec/km)	Average stop numbers (#/Veh/km)
Continuous state Actor-Critic (λ)	$\lambda = 0.0$	379 $\pm$ 14	325 $\pm$ 14	1.77 $\pm$ 0.01
	$\lambda = 0.5$	317 $\pm$ 14	264 $\pm$ 14	1.69 $\pm$ 0.02
	$\lambda = 0.9$	275 $\pm$ 08	222 $\pm$ 08	1.63 $\pm$ 0.01
Continuous state residual Actor-Critic (λ)	$\lambda = 0.0$	376 $\pm$ 21	322 $\pm$ 21	1.76 $\pm$ 0.01
	$\lambda = 0.5$	302 $\pm$ 17	249 $\pm$ 17	1.67 $\pm$ 0.02
	$\lambda = 0.9$	234 $\pm$ 13	181 $\pm$ 13	1.59 $\pm$ 0.01
Discrete state Actor-Critic (λ)	$\lambda = 0.0$	477 $\pm$ 11	423 $\pm$ 11	1.94 $\pm$ 0.01
	$\lambda = 0.5$	404 $\pm$ 10	350 $\pm$ 10	1.86 $\pm$ 0.01
	$\lambda = 0.9$	271 $\pm$ 09	218 $\pm$ 09	1.74 $\pm$ 0.01
Continuous state Q-learning (λ)	$\lambda = 0.0$	509 $\pm$ 07	456 $\pm$ 07	1.88 $\pm$ 0.01
	$\lambda = 0.5$	522 $\pm$ 08	468 $\pm$ 08	1.89 $\pm$ 0.01
	$\lambda = 0.9$	529 $\pm$ 08	475 $\pm$ 08	1.90 $\pm$ 0.01
Discrete state Q-learning (λ)	$\lambda = 0.0$	354 $\pm$ 14	301 $\pm$ 14	1.86 $\pm$ 0.01
	$\lambda = 0.5$	393 $\pm$ 14	340 $\pm$ 14	1.89 $\pm$ 0.01
	$\lambda = 0.9$	480 $\pm$ 26	427 $\pm$ 26	1.96 $\pm$ 0.03
Continuous state SARSA (λ)	$\lambda = 0.0$	508 $\pm$ 07	455 $\pm$ 07	1.88 $\pm$ 0.01
	$\lambda = 0.5$	520 $\pm$ 11	466 $\pm$ 11	1.89 $\pm$ 0.01
	$\lambda = 0.9$	533 $\pm$ 08	480 $\pm$ 08	1.91 $\pm$ 0.01
Discrete state SARSA (λ)	$\lambda = 0.0$	348 $\pm$ 22	294 $\pm$ 22	1.85 $\pm$ 0.02
	$\lambda = 0.5$	344 $\pm$ 16	291 $\pm$ 16	1.85 $\pm$ 0.01
	$\lambda = 0.9$	384 $\pm$ 14	330 $\pm$ 14	1.86 $\pm$ 0.01

Table 4
System disturbances.

Parameter		Low system disturbance	High system disturbance
Incident	Frequency	3 times per 6 h	4 time per 6 h
	Duration	10 (min)	20 (min)
	Length	10 (m)	20 (m)
	Number of blocked lanes	3	4
Sensor Noise	Starting time	Random between 6:00 am and 11:00 am	Random between 6:00 am and 11:00 am
	Mean of Gaussian distribution	0.0 vehicles	0.0 vehicles
	Standard deviation of Gaussian distribution	1.5 vehicles	3.0 vehicles
	Jaywalking	Real world data	Real world data

to the best performance (average performance over the last 10 episodes). It shows there are some unnecessary and redundant variables in the state-action space.

- Eliminating the time variable (T) from the state space in all RLTSCs not only results in a faster convergence speed but also brings about a better performance in comparison to the base case experiment (it has to be underlined that the performance of continuous state Q-learning(λ) does not change significantly by removing state variables). In fact, in spite of flow variations during the time, T could not efficiently affect the learning of the RLTSCs. This indicates that the optimal action, in the proposed regime, just depends on the traffic condition regardless of T. For example, the optimal action for the same traffic state at 10 am and 11 pm are probably identical.
- In test 2, the improvement in the performance of all RLTSCs is

observable. It can be inferred that the advantage of removing departing links variables which decreases the state space size outweighs the shortcomings of ignoring the indirect cooperation. The reason might be that the next lanes are so long and maybe have equal occupancies, that including them does not help, while it still adds input variables making the dimensionality higher and learning slower.

- In both continuous state Q-learning(λ) and SARSA(λ), reducing the state-action space size does not significantly affect the performance. This shows that the poor performance of them is not because of the high dimensionality of the state-action space. One possible explanation could be due to the type of function approximation (i.e. tile coding) that is not fully suitable for Q-learning(λ) and SARSA(λ) algorithms in the employed traffic environment. Another possibility is that the dynamics of the traffic environment (variable traffic flow and coevolving traffic signals) are so high to be fully handled by continuous state Q-learning(λ) and SARSA(λ).
- Test 4 in Q-learning(λ) and SARSA(λ) algorithms (both discrete and continuous state), test 2 in continuous state actor-critic(λ) algorithms and test 3 in discrete state actor-critic(λ) algorithm lead to the best performance.

Table 6 compares the performance of the best state-action spaces. It is obvious that continuous state actor-critic(λ) has the best performance in terms of average travel time, delay time, and stop numbers. In the next experiment, the best state-action spaces obtained in this experiment are employed in order to assess the robustness of the new state-action space against system disturbances.

7.4. Experiment 4: effect of system disturbances on the performance of new RLTSCs

In this experiment, the system disturbances from experiment 2 are taken into account in order to evaluate the robustness of the obtained RLTSCs from experiment 3. In fact, the analysis is performed with the reduced state-action space to evaluate whether this reduction can increase the robustness. Fig. 7 illustrates the effect of system disturbances on the RLTSCs' learning performance in terms of average travel time. The following points can be concluded from the results.

- Like experiment 2, continuous state SARSA(λ) and Q-learning(λ) have the constant or growing trend after encountering system disturbances. This shows the lack of robustness in handling the unexpected conditions even after reducing the size of the state-action space.
- Unlike experiment 2, discrete state Q-learning(λ) and SARSA(λ) algorithms show the downward trends over the last 10 episodes. It proves their ability in adapting to disturbances in the reduced state-action space. By considering results of experiments 2 and 4 for discrete state Q-learning(λ) and SARSA(λ), it is inferred that the performance of the RLTSCs under disturbances depends on not only the employed RL method but also the used state space.
- Comparing the learning curves of continuous and discrete state actor-critic methods over the course of last 10 episodes in experiments 2 and 4 shows that reducing the state-action space has increased the speed of adaptability and convergence of the RLTSCs when encountering the disturbances.
- The slopes of the learning curves of RLTSCs over the last 10 episodes show how fast they are in handling system disturbances. It is clear that continuous state actor-critic(λ) and discrete state SARSA(λ) are faster in handling disturbances.
- Table 7 shows the average performance of different RLTSCs over the last 10 episodes. It is obvious that continuous state actor-critic(λ) outperforms others.

Fig. 5. Average travel time (sec/km) for different RLTSs with system disturbances. All the learning curves are the average of 10 simulations. The learning curves over the last 10 episodes are zoomed-in.

Table 5

The mean difference in performance between disturbance experiment (experiment 2) and the base case experiment (experiment 1) over the last 10 episodes (results are averaged over 10 simulations).

Traffic signal controller	Level of system disturbances	Mean difference in average travel time (sec/km)	Mean difference in average delay time (sec/km)	Mean difference in average stop numbers (#/Veh/km)
Continuous state Actor-Critic (λ)	Low	68.8 ± 21.6	68.6 ± 21.7	0.21 ± 0.04
	High	100.4 ± 26.7	100.1 ± 26.6	0.23 ± 0.04
Continuous state residual Actor-Critic(λ)	Low	99.1 ± 26.5	98.9 ± 26.5	0.24 ± 0.03
	High	102.7 ± 26.3	102.4 ± 26.3	0.26 ± 0.03
Discrete state Actor-Critic(λ)	Low	99.7 ± 21.3	99.5 ± 21.3	0.21 ± 0.03
	High	153.6 ± 23.0	153.3 ± 23.0	0.28 ± 0.03

Fig. 6. Average travel time (sec/km) for different RLTSs with different inputs and sets of actions. All the learning curves are the average of 10 simulations.

Table 6

Results of experiment 3 (results are averaged over 10 simulations).

RLTSC	Average travel time (sec/km)	Average delay time (sec/km)	Average stop numbers (#/Veh/km)
Continuous state Actor-Critic(λ) (Test 1)	191 $\pm$ 10	137 $\pm$ 10	1.52 $\pm$ 0.01
Continuous state residual Actor-Critic(λ) (Test 1)	214 $\pm$ 12	161 $\pm$ 12	1.56 $\pm$ 0.02
Discrete state Actor-Critic(λ) (Test 3)	259 $\pm$ 12	205 $\pm$ 12	1.68 $\pm$ 0.01
Continuous state Q-learning(λ) (Test 4)	509 $\pm$ 5	455 $\pm$ 5	1.89 $\pm$ 0.01
Discrete state Q-learning(λ) (Test 4)	320 $\pm$ 11	266 $\pm$ 11	1.79 $\pm$ 0.02
Continuous state SARSA(λ) (Test 4)	498 $\pm$ 8	444 $\pm$ 8	1.9 $\pm$ 0.01
Discrete state SARSA(λ) (Test 4)	306 $\pm$ 8	252 $\pm$ 8	1.75 $\pm$ 0.02

Fig. 7. Average travel time (sec/km) for different RLTSs. All the learning curves are the average of 10 simulations. The learning curves over the last 10 episodes are zoomed-in.

8. Discussion

In this research, RL is embedded in a traffic signal control task that can be considered as a knowledge-intensive task as it requires large quantities of knowledge for optimal control. RL provides traffic signals with the ability to acquire the necessary knowledge for making reliable decisions by interacting with the traffic environment and without the need for human intervention. Efficiently embedding RL in traffic signals in real-world traffic environment encounters different challenges. Among others, system disturbances and high dimensional state-action space are regarded in this research. Considering these challenges is the key to success of moving towards putting the RLTSs into practice.

Regarding high dimensional state-action spaces, results indicated

that the controllers founded on continuous state RL generally lead to better performance and faster convergence. This is because continuous state RLTSs do not require to directly visit all states to estimate their values. The values of states are estimated by the values of similar states. However, the performance of continuous state RL is highly dependent on the suitability of the feature function and using inappropriate feature functions might significantly ruin the performance (e.g. continuous state Q-learning(λ) and SARSA(λ)).

Concerning system disturbances, results showed that all RLTSs are not very robust (although some methods are more robust than others) against system disturbances. The RLTSs based on continuous state SARSA(λ) and Q-learning(λ) fall short of robustness. Also, continuous state actor-critic(λ) is the most robust RLTS in comparison to others.

Table 7
Results of experiment 4 (results are averaged over 10 simulations).

Traffic signal controller	Level of system disturbances	Average travel time (sec/km)	Average delay time (sec/km)	Average stop numbers (#/Veh/km)
Continuous state Actor-Critic(λ)	Low	258 $\pm$ 28	205 $\pm$ 28	1.71 $\pm$ 0.05
	High	296 $\pm$ 23	242 $\pm$ 22	1.75 $\pm$ 0.05
Continuous state residual Actor-Critic(λ)	Low	271 $\pm$ 30	218 $\pm$ 30	1.71 $\pm$ 0.05
	High	310 $\pm$ 36	256 $\pm$ 36	1.77 $\pm$ 0.05
Discrete state Actor-Critic(λ)	Low	335 $\pm$ 15	282 $\pm$ 15	1.92 $\pm$ 0.02
	High	374 $\pm$ 24	320 $\pm$ 24	1.96 $\pm$ 0.05
Continuous state Q-learning(λ)	Low	598 $\pm$ 6	544 $\pm$ 6	2.04 $\pm$ 0.01
	High	600 $\pm$ 9	547 $\pm$ 9	2.04 $\pm$ 0.01
Discrete state Q-learning(λ)	Low	405 $\pm$ 20	351 $\pm$ 20	1.95 $\pm$ 0.04
	High	407 $\pm$ 17	353 $\pm$ 17	1.97 $\pm$ 0.03
Continuous state SARSA(λ)	Low	601 $\pm$ 6	548 $\pm$ 6	2.07 $\pm$ 0.01
	High	607 $\pm$ 8	553 $\pm$ 8	2.07 $\pm$ 0.01
Discrete state SARSA(λ)	Low	381 $\pm$ 26	328 $\pm$ 26	1.91 $\pm$ 0.03
	High	407 $\pm$ 34	353 $\pm$ 34	1.94 $\pm$ 0.05

This is because the critic's estimates provide the actor with the ability to update with gradients that have lower variance.

The traffic signals of the study area are mostly controlled by police officers in the periods of peak demands. Since modeling their operations is not the objective of this research, we do not benchmark their performance against the RLTSs. We, however, benchmark the best RLTS (i.e. continuous state actor-critic(λ)) against an optimized fixed-time method in three different disturbance levels in order to ensure that the best RLTS is able to outperform the fixed-time controller even in the presence of disturbances. The fixed time signal plan for each intersection is separately optimized by employing TRANSYT [86]. It is worth noting that among all the system disturbance factors, the sensor noise is not applicable to the fixed-time controller due to the fact that it does not work based on the information obtained from sensors. As shown in Table 8, the best RLTS results in lower average travel time, delay time, and stop numbers for all three system disturbance levels. The most notable improvement is average delay time. Also, the best RLTS outperforms the fixed-time controller at the presence of high system disturbances.

Another point is the matter of differences between real-world and simulated traffic that can affect the performance of the RLTSs. Certainly the microscopic traffic simulation of the study area cannot thoroughly replicate the real-world traffic, because there are some elements and factors that are unknown or very difficult to be modeled with acceptable accuracy. Therefore, the performance of different RLTSs in the simulated traffic environment might not be similar to that of in the study area. Here, we briefly discuss some possible pitfalls that could reduce the expected improvements in practice.

The first issue is the variation of traffic demand over time. In the used microscopic traffic simulation the time-interval of flow variations is one hour, i.e., the traffic flow changes in one-hour intervals. In real-world traffic, however, the variations in demand occur in shorter time-intervals. This issue that increases the nonstationarity and stochasticity of real-world traffic may disturb the performance of the RLTSs. Also, the variations of the traffic flow among the days of the week are neglected in this research and this is another pitfall that may impair the performance. Furthermore, drivers adhere to lane discipline in the

simulated traffic environment that makes the traffic pattern more homogeneous, whereas the real traffic pattern in the study area is, to some extent, heterogeneous due to the lack of lane discipline. This issue can afflict the improvements. Moreover, the reaction time of all drivers is not the same in real-world traffic that may lead to more stochasticity and in turn affect the performance. Additionally, it is assumed that jaywalkers cross the streets in a constant velocity. In reality, they, however, may increase their speed, depending on their age and gender, when a vehicle is approaching them.

9. Conclusion and future work

In this research, first, a real-world microscopic traffic simulation of downtown Tehran was carried out. Then, an empirical study of seven different RLTSs from both discrete and continuous state approaches was conducted. In the microscopic traffic simulation, traffic demand data for 6 h (between 6:00 am and 12:00 pm) in the form of origin-destination matrixes were assigned to the traffic network in order to simulate the morning traffic variations. To achieve high credibility in the traffic simulation, we utilized calibrated parameters describing the driver behaviors in the simulation.

The effects of eligibility traces on the performance of RL algorithms were investigated. It is found that the higher the value of λ is, the better the performance of actor-critic(λ) algorithms become and the worse the performance of Q-learning(λ) and SARSA(λ) algorithms become. Also, the results indicated that the state variables of departing links and time (i.e. $[T, D_1, D_2, \dots, D_v]$) are not beneficial. The continuous state actor-critic(λ) algorithm has the highest robustness property against traffic disturbances. Moreover, the results demonstrate the efficiency of decentralized and RL-embedded traffic signal control in urban traffic networks in comparison to the fixed-time method. Since the system is designed to operate in a decentralized manner, it is highly scalable to traffic networks of arbitrary sizes and additional intersections can be readily incorporated into the system.

The present research serves to the developments in the research line of intelligent traffic control and motivates the application of machine learning techniques to real-world domains. Future work can cover the

Table 8
Comparison between fixed time signal controller and the best RLTS.

System indexes	No system disturbances			Low system disturbances			High system disturbances		
	Fixed-time	RLTS	% Improvement	Fixed-time	RLTS	% Improvement	Fixed-time	RLTS	% Improvement
Travel time (sec/km)	264	191	27.6	339	258	23.9	379	296	21.9
Delay time (sec/km)	210	137	34.8	285	205	28.2	325	242	25.6
Stop numbers (#/Veh/km)	1.82	1.52	16.6	1.98	1.72	13.1	2.06	1.75	15.1

following issues: (a) Using non-linear function approximators. Given the successes of the non-linear function approximators in complex RL tasks, an empirical comparison between non-linear and linear function approximators in traffic signal control would be of added value. In this context, challenges which arise are the matter of high computational cost and unguaranteed convergence of the non-linear function approximators in multi-agent reinforcement learning tasks. (b) Evaluating the proposed traffic signal controllers in a real-world environment. No matter how accurate the simulation is, it cannot completely imitate the real-world and there still may be some factors overlooked in the simulation. For this reason, it would be interesting to test the proposed traffic signal controllers in a real-world environment in order to see whether the reported gains will also be obtained in practice. (c) Conducting experiments using a different traffic simulator (e.g. Paramics and Vissim) to ensure that outcomes are not the results of overfitting to the employed simulator (AIMSUN).

References

- [1] A. Stevanovic, J. Stevanovic, K. Zhang, S. Batterman, Optimizing traffic control to reduce fuel consumption and vehicular emissions: integrated approach with VISSIM, CMEM, and VISGAOST, *Transp. Res. Rec.: J. Transp. Res. Board* 2128 (2128) (2009) 105–113.
- [2] M.A. Khamis, W. Gomaa, Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework, *Eng. Appl. Artif. Intell.* 29 (2014) 134–151.
- [3] S.F. Smith, G.J. Barlow, X.-F. Xie, Z.B. Rubinstein, Smart urban signal networks: initial application of the SURTRAC adaptive traffic signal control system, in: *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling*, AAAI Press, 2013, pp. 434–442.
- [4] S. Araghi, A. Khosravi, D. Creighton, S. Nahavandi, Influence of meta-heuristic optimization on the performance of adaptive interval type-2-fuzzy traffic signal controllers, *Expert Syst. Appl.* 71 (2017) 493–503.
- [5] A.N.H. Zaied, W.A. Othman, Development of a fuzzy logic traffic system for isolated signalized intersections in the state of Kuwait, *Expert Syst. Appl.* 38 (8) (2011) 9434–9441.
- [6] P.G. Balaji, D. Srinivasan, Type-2 fuzzy logic based urban traffic management, *Eng. Appl. Artif. Intell.* 24 (1) (2011) 12–22.
- [7] D. Srinivasan, M.C. Choy, R.L. Cheu, Neural networks for real-time traffic signal control, *IEEE Trans. Intell. Transp. Syst.* 7 (3) (2006) 261–272.
- [8] K.-H. Chao, R.-H. Lee, M.-H. Wang, An intelligent traffic light control based on extension neural network, in: I. Lovrek, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems: 12th International Conference, KES 2008, Zagreb, Croatia, September 3–5, 2008, Proceedings, Part I*, Springer, Berlin, Heidelberg, 2008, pp. 17–24.
- [9] B. Abdulhai, L. Kattan, Reinforcement learning: introduction to theory and potential for transport applications, *Can. J. Civ. Eng.* 30 (6) (2003) 981–991.
- [10] S. El-Tantawy, B. Abdulhai, H. Abdelgawad, Design of reinforcement learning parameters for seamless application of adaptive traffic signal control, *J. Intell. Transp. Syst.: Technol. Plan. Oper.* 18 (3) (2014) 227–245.
- [11] M. Aslani, M. Saadi, M.W. Mesgari, Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events, *Transport. Res. Part C: Emerg. Technol.* 85 (2016) 732–752.
- [12] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, 1998.
- [13] M. Wiering, Multi-agent reinforcement learning for traffic light control, in: *17th International Conference on Machine Learning*, 2000, pp. 1151–1158.
- [14] S. El-Tantawy, B. Abdulhai, 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.* 14 (3) (2013) 1140–1150.
- [15] M. Aslani, S. Seipel, M. Wiering, Continuous residual reinforcement learning for traffic signal control optimization, *Can. J. Civ. Eng.* 45 (8) (2018) 690–702.
- [16] M. van Otterlo, M. Wiering, Reinforcement learning and Markov decision processes, in: M. Wiering, M. Otterlo (Eds.), *Reinforcement Learning: State-of-the-Art*, Springer, Berlin, Heidelberg, 2012, pp. 3–42.
- [17] L. Busoniu, R. Babuska, B.D. Schutter, D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*, Automation and Control Engineering Series, CRC Press, Florida, 2010.
- [18] C. Szepesvari, *Algorithms for Reinforcement Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning series, Morgan & Claypool Publishers, 2010.
- [19] J.S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), *J. Dyn. Syst. Meas. Contr.* 97 (1975) 220–227.
- [20] L. Baird, *Reinforcement Learning Through Gradient Descent* (Ph.D. thesis), Carnegie Mellon University, 1999.
- [21] M. Jeihani, P. James, A.A. Saka, A. Ardeshtari, Traffic recovery time estimation under different flow regimes in traffic simulation, *J. Traffic Transp. Eng.* 2 (5) (2015) 291–300.
- [22] P.B. Hunt, D.I. Robertson, R.D. Bretherton, R.I. Winton, SCOOT – A Traffic Responsive Method of Coordinating Signals, Report, 1981.
- [23] A.G. Sims, K.W. Dobinson, The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits, *IEEE Trans. Veh. Technol.* 29 (2) (1980) 130–137.
- [24] N.H. Gartner, OPAC: a demand-responsive strategy for traffic signal control, *Transp. Res. Rec.: J. Transp. Res. Board* 906 (1983) 75–81.
- [25] J.-J. Henry, J.-L. Farges, J. Tufal, The PROLYN real-time traffic algorithm, in: *Proceedings of the 5th IFAC/IFIP/IFORS Symposium on Control in Transportation Systems*, 1983.
- [26] K.L. Head, P.B. Mirchandani, D. Sheppard, Hierarchical framework for real-time traffic control, *Transp. Res. Rec.* 1360 (1992) 82–88.
- [27] S. El-Tantawy, Multi-agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC) (PhD thesis), University of Toronto, 2012.
- [28] Y.S. Murat, E. Gedizlioglu, A fuzzy logic multi-phased signal control model for isolated junctions, *Transp. Res. Part C: Emerg. Technol.* 13 (1) (2005) 19–36.
- [29] L. Zadeh, Fuzzy sets, *Inf. Control* 8 (3) (1965) 338–353.
- [30] E.H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *Proc. Inst. Electr. Eng.* 121 (12) (1974) 1585–1588.
- [31] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man Cybern.* SMC-15 (1) (1985) 116–132.
- [32] A.B. Eriksen, C. Huang, J. Kildebogaard, J.H. Taankvist, UPPAAL STRATEGO for intelligent traffic lights, in: *12th ITS European Congress*, Strasbourg, France, 2017.
- [33] A. David, P.G. Jensen, K.G. Larsen, M. Mikučionis, J.H. Taankvist, UPPAAL STRATEGO, in: C. Baier, C. Tinelli (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems*, Springer, Berlin, Heidelberg, 2015, pp. 206–211.
- [34] R.S. Sutton, A.G. Barto, Toward a modern theory of adaptive networks: expectation and prediction, *Psychol. Rev.* 88 (2) (1981) 135–170.
- [35] H.M. Schwartz, *Multi-Agent Machine Learning: A Reinforcement Approach*, Wiley, New Jersey, 2014.
- [36] B. Adam, I.F. Smith, Reinforcement learning for structural control, *J. Comput. Civ. Eng.* 22 (2) (2008) 133–139.
- [37] B. Adam, I.F.C. Smith, Active tensegrity: a control framework for an adaptive civil-engineering structure, *Comput. Struct.* 86 (23) (2008) 2215–2223.
- [38] J. Jin, X. Ma, I. Kosonen, An intelligent control system for traffic lights with simulation-based evaluation, *Control Eng. Pract.* 58 (Supplement C) (2017) 24–33.
- [39] M. Aslani, M.S. Mesgari, S. Seipel, Marco Wiering, Developing adaptive traffic signal control by actor-critic and direct exploration methods, *Proc. Inst. Civ. Eng. Transp.* (2018) <https://doi.org/10.1680/jtran.17.00085>.
- [40] M. Ruiz-Montiel, J. Boned, J. Gavilanes, E. Jimenez, L. Mandow, J.-L. Perez-de-la Cruz, Design with shape grammars and reinforcement learning, *Adv. Eng. Inform.* 27 (2) (2013) 230–245.
- [41] L.A. Scardua, J.J. Da Cruz, A.H. Real Costa, Optimal control of ship unloaders using reinforcement learning, *Adv. Eng. Inform.* 16 (3) (2002) 217–227.
- [42] T. Yasuda, K. Ohkura, K. Ueda, A homogeneous mobile robot team that is fault-tolerant, *Adv. Eng. Inform.* 20 (3) (2006) 301–311.
- [43] V. Koropoulis, A. Gusriadi, S. Hirche, D. Lee, An extremum-seeking control approach for constrained robotic motion tasks, *Control Eng. Pract.* 52 (Supplement C) (2016) 1–14.
- [44] P. Mannion, J. Duggan, E. Howley, An experimental review of reinforcement learning algorithms for adaptive traffic signal control, in: L.T. McCluskey, A. Kotsialos, P.J. Muller, F. Klugl, O. Rana, R. Schumann (Eds.), *Autonomic Road Transport Support Systems*, Springer, Cham, 2016, pp. 47–66.
- [45] K. Wen, S. Qu, Y. Zhang, A stochastic adaptive control model for isolated intersections, in: *IEEE International Conference on Robotics and Biomimetics*, 2007, ROBIO 2007, 2007, pp. 2256–2260.
- [46] S. El-Tantawy, B. Abdulhai, An agent-based learning towards decentralized and coordinated traffic signal control, in: *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 665–670.
- [47] I. Dusparic, J. Monteil, V. Cahill, Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning, *19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 2065–2070.
- [48] I. Dusparic, V. Cahill, Autonomic multi-policy optimization in pervasive systems: overview and evaluation, *ACM Trans. Auton. Adapt. Syst. (TAAS)* 7 (1) (2012) 11:1–11:25.
- [49] M. Fellendorf, P. Vortisch, Microscopic traffic flow simulator VISSIM, in: J. Barcelo (Ed.), *Fundamentals of Traffic Simulation*, Springer, New York, NY, 2010, pp. 63–93.
- [50] M. Humphrys, *Action Selection Methods Using Reinforcement Learning* (PhD thesis), University of Cambridge, 1997.
- [51] B. Bakker, S. Whiteson, L. Kester, F.C.A. Groen, Traffic light control by multiagent reinforcement learning systems, in: R. Babuska, F.C.A. Groen (Eds.), *Interactive Collaborative Information Systems*, Springer, Berlin, Heidelberg, 2010, pp. 475–510.
- [52] M. Wiering, J. Vreeken, J.V. Veenen, A. Koopman, Simulation and optimization of traffic in a city, in: *IEEE Intelligent Vehicles symposium*, 2004, pp. 453–458.
- [53] S. Richter, D. Aberdeen, J. Yu, Natural actor-critic for road traffic optimisation, *NIPS'06 Proceedings of the 19th International Conference on Neural Information Processing Systems*, MIT Press, 2007, pp. 1169–1176.
- [54] J. Baxter, P.L. Bartlett, L. Weaver, Experiments with infinite-horizon, policy-gradient estimation, *J. Artif. Intell. Res.* 15 (2001) 351–381.
- [55] L. Prashanth, S. Bhatnagar, Reinforcement learning with function approximation for traffic signal control, *IEEE Trans. Intell. Transp. Syst.* 12 (2) (2011) 412–421.
- [56] S.-B. Cools, C. Gershenson, B. D'Hooghe, Self-organizing traffic lights: a realistic simulation, in: M. Prokopenko (Ed.), *Advances in Applied Self-Organizing Systems*, Springer, London, 2013, pp. 45–55.
- [57] T.T. Pham, T. Brys, M.E. Taylor, Learning coordinated traffic light control, in:

- Proceedings of the Adaptive and Learning Agents Workshop (at AAMAS-13), 2013.
- [58] M.E. Taylor, M. Jain, P. Tandon, M. Yokoo, M. Tambe, Distributed on-line multi-agent optimization under uncertainty: balancing exploration and exploitation, *Adv. Complex Syst.* 14 (3) (2011) 471–528.
- [59] R.S. Sutton, Learning to predict by the methods of temporal differences, *Mach. Learn.* 3 (1) (1988) 9–44.
- [60] C. Watkins, P. Dayan, Q-learning, *Mach. Learn.* 8 (3) (1992) 279–292.
- [61] G.A. Rummery, M. Niranjan, On-line Q-learning Using Connectionist Systems, CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.
- [62] V.R. Konda, V. Borkar, Actor-critic like learning algorithms for Markov decision processes, *SIAM J. Control Optimiz.* 38 (1) (1999) 94–123.
- [63] S. Singh, T. Jaakkola, M.L. Littman, C. Szepesvari, Convergence results for single-step on-policy reinforcement-learning algorithms, *Mach. Learn.* 38 (3) (2000) 287–308.
- [64] R.S. Sutton, Generalization in reinforcement learning: successful examples using sparse coarse coding, in: D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, vol. 8, MIT Press, 1996, pp. 1038–1044.
- [65] X.-R. Cao, H.-F. Chen, Perturbation realization, potentials, and sensitivity analysis of Markov processes, *IEEE Trans. Automatic Control* 42 (10) (1997) 1382–1393.
- [66] L. Baird, Residual algorithms: reinforcement learning with function approximation, *Proceedings of the 12th International Conference on Machine Learning (ICML 95)*, Morgan Kaufman Publishers, 1995, pp. 30–37.
- [67] H. Behruz, A. Safaie, A.P. Chavoshy, Tehran traffic congestion charging management: a success story, in: 18th World Congress on Intelligent Transport Systems and ITS America Annual Meeting 2011, 2011, pp. 83–95.
- [68] TCTTC, Transportation at One Glance, Report, The Company of Comprehensive Studies of Transportation and Traffic of Tehran, 2007.
- [69] SCI, Year Book of Iran, Report, Statistical Center of Iran, 2014.
- [70] S. Lotfi, M.J. Koohsari, Measuring objective accessibility to neighborhood facilities in the city (a case study: zone 6 in Tehran, Iran), *Cities* 26 (3) (2009) 133–140.
- [71] S.E. Asgharpour, H. Zanjani, G. Taleghani, Impact of urbanization on population changes in metropolitan area of Tehran, Iran, in: 3rd International Geography Symposium GEOMED 2013 Symposium Proceedings, 2013, pp. 6–22.
- [72] J. Casas, J.L. Ferrer, D. Garcia, J. Perarnau, A. Torday, Traffic simulation with AIMSUN, in: J. Barcelo (Ed.), *Fundamentals of Traffic Simulation*, Springer, New York, NY, 2010, pp. 173–232.
- [73] DTT, Calibration of Traffic Engineering Software Based on the Traffic Conditions in Tehran. Report, Department of traffic and transportation, Municipality of Tehran, 2010.
- [74] P. Koonce, L. Rodegerdts, K. Lee, S. Quayle, S. Beaird, C. Braud, J. Bonneson, P. Tarnoff, T. Urbanik, Traffic Signal Timing Manual, Report FHWA-HOP-08-024, Federal Highway Administration, 2008.
- [75] J. Kwon, M. Mauch, P. Varaiya, Components of congestion: delay from incidents, special events, lane closures, weather, potential ramp metering gain, and excess demand, *Transp. Res. Rec.: J. Transp. Res. Board* 30 (1959) (2006) 84–91.
- [76] X. Wang, Z. Tian, Pedestrian delay at signalized intersections with a two-stage crossing design, *Transp. Res. Rec.: J. Transp. Res. Board* 2173 (2010) 133–138.
- [77] B. Li, A model of pedestrians' intended waiting times for street crossings at signalized intersections, *Transp. Res. Part B: Methodol.* 51 (2013) 17–28.
- [78] P. Onelcin, Y. Alver, Illegal crossing behavior of pedestrians at signalized intersections: factors affecting the gap acceptance, *Transp. Res. Part F: Traffic Psychol. Behav.* 31 (2015) 124–132.
- [79] O. Keegan, M. O'Mahony, Modifying pedestrian behaviour, *Transp. Res. Part A: Policy Pract.* 37 (10) (2003) 889–901.
- [80] Y. Zheng, T. Chase, L. Elefteriadou, B. Schroeder, V.P. Sisiopiku, Modeling vehicle-pedestrian interactions outside of crosswalks, *Simul. Model. Pract. Theory* 59 (Supplement C) (2015) 89–101.
- [81] R.S. Sutton, Open theoretical questions in reinforcement learning, in: P. Fischer, H.U. Simon (Eds.), *Computational Learning Theory: 4th European Conference, EuroCOLT'99 Nordkirchen, Germany, March 29–31, 1999 Proceedings*, Springer, Berlin, Heidelberg, 1999, pp. 11–17.
- [82] A. Marinescu, I. Dusparic, A. Taylor, V. Cahill, S. Clarke, P-MARL: prediction-based multi-agent reinforcement learning for non-stationary environments, in: G. Weiss, P. Yolum, R.H. Bordini, E. Elkind (Eds.), *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems*, 2015, pp. 1897–1898.
- [83] B.C. da Silva, E.W. Basso, A.L.C. Bazzan, P.M. Engel, Dealing with non-stationary environments using context detection, *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, ACM, 2006, pp. 217–224.
- [84] I. Grondman, L. Busoniu, G.A.D. Lopes, R. Babuska, A survey of actor-critic reinforcement learning: standard and natural policy gradients, *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* 42 (6) (2012) 1291–1307.
- [85] V.R. Konda, J.N. Tsitsiklis, On actor-critic algorithms, *SIAM J. Control Optimiz.* 42 (4) (2003) 1143–1166.
- [86] D. Robertson, TRANSYT-A Traffic Network Study Tool, Report, Transport and Road Research Laboratory, 1969.