Transportation Research Part C 85 (2017) 732–752

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events[☆]

Mohammad Aslani[a,*], Mohammad Saadi Mesgari[a], Marco Wiering[b]

[a] Faculty of Geodesy and Geomatics Engineering, K.N.Toosi University of Technology, Tehran, Iran
[b] Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen, Groningen, The Netherlands

ARTICLE INFO

Keywords:
Adaptive traffic signal controller
Reinforcement learning
Actor-critic
Pedestrians
Traffic disruptions

ABSTRACT

The transportation demand is rapidly growing in metropolises, resulting in chronic traffic congestions in dense downtown areas. Adaptive traffic signal control as the principle part of intelligent transportation systems has a primary role to effectively reduce traffic congestion by making a real-time adaptation in response to the changing traffic network dynamics. Reinforcement learning (RL) is an effective approach in machine learning that has been applied for designing adaptive traffic signal controllers. One of the most efficient and robust type of RL algorithms are continuous state actor-critic algorithms that have the advantage of fast learning and the ability to generalize to new and unseen traffic conditions. These algorithms are utilized in this paper to design adaptive traffic signal controllers called actor-critic adaptive traffic signal controllers (A-CATs controllers).

The contribution of the present work rests on the integration of three threads: (a) showing performance comparisons of both discrete and continuous A-CATs controllers in a traffic network with recurring congestion (24-h traffic demand) in the upper downtown core of Tehran city, (b) analyzing the effects of different traffic disruptions including opportunistic pedestrians crossing, parking lane, non-recurring congestion, and different levels of sensor noise on the performance of A-CATS controllers, and (c) comparing the performance of different function approximators (tile coding and radial basis function) on the learning of A-CATs controllers. To this end, first an agent-based traffic simulation of the study area is carried out. Then six different scenarios are conducted to find the best A-CATs controller that is robust enough against different traffic disruptions. We observe that the A-CATs controller based on radial basis function networks (RBF (5)) outperforms others. This controller is benchmarked against controllers of discrete state Q-learning, Bayesian Q-learning, fixed time and actuated controllers; and the results reveal that it consistently outperforms them.

1. Introduction

The continuous population increase and subsequently the growth in social and economic activities in cities lead to the rise in demand for transportation (Bhatta, 2010). The rising demand for transportation in the metropolises has made existing traffic infrastructures incapable of handling a lot of vehicles and brings about undesired everyday traffic congestions. Traffic congestions which are produced either by the routine traffic volumes (recurring congestion) or unexpected disruptions (non-recurring congestion

[☆] This article belongs to the Virtual Special Issue on "Agent technology".
[*] Corresponding author.
E-mail addresses: maslani@mail.kntu.ac.ir (M. Aslani), mesgari@kntu.ac.ir (M.S. Mesgari), m.a.wiering@rug.nl (M. Wiering).

http://dx.doi.org/10.1016/j.trc.2017.09.020
Received 21 December 2016; Received in revised form 22 September 2017; Accepted 22 September 2017
Available online 02 November 2017

events), mainly accidents, constructions, emergencies, break-downs, debris, or inclement weather conditions (Jeihani et al., 2015; Bifulco et al., 2016), have negative observable consequences such as long travel times, excess fuel consumption and increasing emission of local air pollutants. In order to reduce traffic congestion and its adverse negative effects, one of the most effective solutions is toward outfitting the existing infrastructure with intelligent transportation systems (ITS) which raise the capacity of existing transportation infrastructures without imposing a high cost of road construction (Chowdhury and Sadek, 2003; Bazzan and Kluegl, 2013a). ITS as systems utilizing synergistic technologies provide flexible approaches to effectively manage and control traffic. One of the significant components of ITS is adaptive traffic signal control (El-Tantawy et al., 2013; Islam and Hajbabaie, 2017).

Adaptive traffic signal control (Ma et al., 2016) is a strategy in which traffic signal timing parameters (e.g. cycle length, phase split and the duration of every controller phase) adapt based on actual traffic conditions and traffic fluctuations (e.g. number of waiting and approaching vehicles (WAVE) at the traffic signal) in order to achieve a set of specific objectives (e.g. minimizing the total number of WAVE). Adaptive traffic signal control can be modeled by self-learning in multi-agent systems (Weiss, 1999; Kluegl and Bazzan, 2012) because of the distributed and autonomous nature of traffic signal control (Adler et al., 2005; Katwijk et al., 2009). Moreover, the complexity arising in a traffic system due to the stochastic nature of the traffic patterns and complex effects of the actions performed by many other adaptive traffic signals makes it difficult to solve with preprogrammed traffic signal behaviors. Thus, a learning mechanism is necessary, such that traffic signals gradually find the global optimal solution on their own by direct interaction with the stochastic traffic environment. In this context, reinforcement learning (RL) (Sutton and Barto, 1998; van Otterlo and Wiering, 2012) as a promising method for generating, evaluating, and improving traffic signal decision making solutions is beneficial.

RL enables traffic signal controllers to learn and react flexibly to diverse traffic circumstances without the need of a predefined model of the stochastic traffic environment and also without the need of human intervention (Abdulhai and Kattan, 2003; Bazzan, 2009; El-Tantawy et al., 2014; Ozan et al., 2015; Mannion et al., 2016). Proper signal timing plans (policies) are learned through the experience of the traffic signals in their intersections (environment) rather than information retrieved from the relationship between correct input and output pairs of the traffic control system. After adaptive traffic signal controllers take actions, they receive single scalar reward signals depending on whether their actions have led them closer to realizing their objective. As a result, RL-embedded traffic signal controllers learn to obtain signal timing plans that optimize the sum of obtained rewards over time (return). By following given signal timing plans and processing the rewards, adaptive traffic signals can build estimates of the return. The function representing this estimated return is known as the value function (Sutton and Barto, 1998).

Numerous RL algorithms exist in the machine learning field. They mainly fall into one of the following three categories: 1-actor-only, 2-critic-only and 3-actor-critic, where the words actor and critic are synonyms for the policy and value function, respectively. Actor-only methods work with a parameterized family of policies. The benefit of a parameterized policy is that a spectrum of continuous actions can be generated, but the high variance in the estimation of the gradient makes learning slow, which is their weakness (Konda and Tsitsiklis, 2003). Critic-only methods such as Q-learning (Watkins and Dayan, 1992) and SARSA (Sutton and Barto, 1998) rely exclusively on value function approximation without an explicit function for the policy. Although they have a lower variance in the estimates of expected returns, to find the optimal actions in different states they need an optimization procedure in each state that makes them computationally more demanding especially if the action space is continuous. Actor-critic methods (Barto et al., 1983) are comprised of two parts, namely an actor and a critic. The critic is used to estimate the value function and the actor selects actions. The critic part evaluates the quality of the used policy and the actor uses the critic's information to update its policy parameters. Actor-critic methods have the advantages of both actor-only and critic-only methods. They are capable of producing continuous actions while the high variance in the estimation of gradients of actor-only methods is reduced by adding a critic (Grondman et al., 2012). Also, they are able to react to smoothly changing states with smoothly changing actions. In this research, actor-critic methods are employed because of their advantages over actor-only and critic-only methods.

Employing a discrete state actor-critic algorithm for traffic signal control which is naturally continuous leads to a combinatorial explosion of states and the well-known curse of dimensionality. Continuous state actor-critic algorithms use generalization that provides them with the ability of performing accurately in unseen situations. The success of continuous state actor-critic algorithms on traffic signal control hinges on effective function approximation which maps states to values via a parameterized function. Among the many function approximation schemes proposed, tile coding and radial basis functions (RBF) which strike an empirical balance between representational power and computational cost are applied in this research. Also, in order to increase the speed of learning, actor-critic methods based on eligibility traces are employed (Sutton and Barto, 1998).

Contributions of this paper. In this paper, continuous RL algorithms are applied to optimize traffic signal controllers in a traffic network modeling real variable traffic flows for 24-h on April 26, 2014, in downtown Tehran. Different actor-critic algorithms based on different types of function approximation are developed and compared on 6 different scenarios. In our traffic micro-simulations in addition to vehicles, impatient pedestrians and their interactions with vehicles are considered. Impatient pedestrians are the pedestrians who may cross junctions during red pedestrian signals if there are appropriate gaps. Impatient pedestrians may disturb vehicle movements and consequently the learning process of traffic signals by their crossing. Therefore, the effect of impatient pedestrians crossing on the traffic signals performance is examined in this research.

Since the study area is located in the administrative zone, the drivers for doing their administrative affairs usually park their vehicles beside the streets for a short time. This can in turn lead to reductions of the streets capacity. The causal effects of parked vehicles beside the streets on the learning of traffic signals are also assessed.

From a practical point of view, traffic signal's sensors can be noisy and imperfect, i.e., sensors may produce different observations from the same traffic condition. Also, reward signals as a feedback of the environment may be affected by noise. Thus, we investigate the effects of noisy states and reward signals on the performance of the learning traffic signals. Moreover, in order to make sure that in the case of any disruption in the traffic network the learning process is not disturbed, the impact of incidents as a non-recurring traffic congestion on the learning behavior of traffic signals is analyzed. Although all the above-mentioned points may not be novel separately, to the best of our knowledge, this is the first work considering the integration of all these points in RL-embedded traffic signal control.

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 technically describes discrete and continuous state actor-critic algorithms. An agent-based traffic simulation of downtown Tehran that comprises drivers, pedestrians and adaptive traffic signals is presented in Section 4. Section 5 demonstrates the six different scenarios and the performance of the actor-critic adaptive traffic signal controllers (A-CATs controllers) in each scenario. The discussion about the A-CATs' performances and choosing the best A-CATs controller according to all conducted scenarios is presented in Section 6. Finally, Section 7 concludes the paper and proposes some directions for future work.

2. Related work

Modern traffic signal control methods are categorized as reactive and adaptive traffic control systems (Gordon and Tighe, 2005). RL is an effective approach in machine learning that has been applied for designing adaptive traffic signal control (Bazzan, 2009). In what follows, some related studies which employed RL to find the optimal traffic signal policy are covered. For the comprehensive survey of existing methods in traffic control the readers are also referred to Bazzan (2009), Bazzan and Kluegl (2013b), Araghi et al. (2015) and Mannion et al. (2016).

One of the approaches in RL-embedded traffic signal control is to employ model-based RL. In this context, Wiering (2000) and Wiering et al. (2004) proposed a discrete model-based RL algorithm for minimizing the overall waiting time of vehicles by means of the position of each vehicle as the state of the traffic condition. The value functions that estimate the expected waiting times of vehicles are learned by both traffic signals and vehicles (co-learning), i.e. each vehicle estimates its own waiting time and sends it to the nearest adaptive traffic signal.

Khamis and Gomaa (2014) extended the framework of Wiering (2000) using Bayesian theory. RL-based traffic signals try to optimize the linear combination of multiple indexes including average travel waiting time, average travel time, average intersection waiting time, safety and speed control. The authors reported that their approach outperforms Wiering's TC-1. Their proposed system was tested on a simple grid-type network with the same lane numbers for all streets that makes the traffic pattern homogeneous and easy to be contended with.

Another approach in designing adaptive traffic signal control through RL is to use model-free RL. Within such a context, Abdulhai et al. (2003) employed discrete Q-learning (model-free RL) to design adaptive traffic lights to control an isolated two-phase intersection. The adaptive traffic signal controllers sense the length of a queue on four streets leading to the intersection as the state of the traffic and decide either to extend the current green phase or change it to the next one in such a way that the average number of waiting vehicles is minimized. They benchmarked the proposed controller against an optimized fixed time controller and the results showed that it outperforms the fixed time controller for variable traffic flows by 44%, and slightly outperformed the fixed time controller for constant and uniform flows.

Abdoos et al. (2013) employed an approach based on the integration of holonic multi-agent system and discrete Q-learning to control traffic signals in an unrealistic traffic network of 50 intersections. Intersections are arranged in thirteen holons by using a graph-based algorithm. The order of average queue lengths in the links leading to the junction is used for the state definition and the green time ratio among different links as a fixed order is employed in the action definition. The results showed that the holonic Q-learning outperforms an individual Q-learning algorithm.

El-Tantawy et al. (2014) compared discrete state Q-learning(λ) and SARSA(λ) based on three different state definitions and four different reward definitions in a real-world traffic network of downtown Toronto. Their results showed that the state definition rests on the number of arriving vehicles to the green phase and queue lengths at the red phases, which is to some extent similar to our research, is the best state definition. Also, cumulative delay was selected as the best reward function. It is worth noting that our reward definition was not examined in their research. They benchmarked Q-learning against a fixed time signal plan. The results showed that RL-based traffic signal control consistently outperformed the fixed time signal plan, regardless of the state representation or traffic conditions.

Darmoul et al. (2017) suggested a multi-agent architecture, which rests on concepts inspired by biological immunity, to control interrupted flow at signalized intersections. In the proposed architecture, each agent which controls a signalized junction adapts to disturbances through an artificial immune network that has similarities with Q-learning. Fluctuations in traffic volumes, variable traffic loads, and accidents were taken into account as the traffic disturbances. They benchmarked their method against a fixed-time controller and a distributed adaptation of the longest queue first on 1 × 3 and 2 × 3 hypothetical traffic networks. The results show that the proposed method outperforms fixed-time and longest queue first controllers at the presence of traffic disturbances.

The challenge for all discrete RL algorithms in complex traffic environments is to visit each state-action pair enough when the state-action space is huge. The naive solution is to decrease the number of states by coarse categorizing the state space that may result in poor performance of the system. The more reasonable solution is to employ continuous RL algorithms that have the ability to accurately perform on unseen data. In this context, Arel et al. (2010) proposed a novel method based on RL for adaptive traffic signal control. The goal of the system is minimizing the average delay, congestion and intersection cross-blocking. The control of the whole traffic network which consists of 5 intersections is the outcome of the collaboration between two types of autonomous agents: a central agent and outbound agents. The outbound agents determine traffic signal timing parameters by employing the longest queue first (LQF) algorithm as well as sharing their local traffic state with the central agent. The central agent, the only RL based traffic signal controller, has all traffic statistics of all neighboring intersections and learns a value function by utilizing a continuous Q-learning algorithm with a feed-forward neural network function approximation to be able to handle the large state space. Experimental results revealed that the continuous Q-learning algorithm outperforms the LQF approach at high demand levels. Although the authors claimed that their proposed system is extendable to a traffic network with many intersections, developing the neural network and its training are non-trivial and computationally expensive in a small traffic network, let alone in a real-world large-scale traffic network.

Prashanth and Bhatnagar (2011b) developed Q-learning with feature-based representations and Q-learning with function approximation in order to tackle the curse of dimensionality. They divided the congestion into three classes, low, medium and high, so that their method does not require precise information about traffic situations. They tested the proposed method on four simplified traffic networks. They benchmarked their methods against fixed time, longest queue and also the algorithms proposed in Abdulhai et al. (2003). Although experimental results demonstrated that Q-learning with function approximation outperforms others, their discretization of the congestion into three broad classes for the state representation is a coarse approximation. In another paper (Prashanth and Bhatnagar, 2011a), the authors compared the performance of continuous Q-learning with policy gradient actor-critic in two simplistic traffic networks. The results demonstrated that the policy gradient actor-critic method consistently outperforms the continuous Q-learning algorithm.

Abdoos et al. (2014) employed the integration of discrete RL and continuous RL to control 9 traffic signals on a 3×3 junction grid. There are two types of agents: bottom level agents and top level agents. The nine bottom level agents control local intersections independently by discrete Q-learning and three top level agents contribute to the learning of the lower members by using continuous Q-learning with a tile coding function approximation. The state of each bottom level agent is the order of the average queue length in the links leading to the intersection and their action is the green time ratio among different links in a fixed order. The state of the top level agents is estimated according to the feature vector of the members and their action is the action space restriction of bottom level agents. The authors mentioned that their method outperforms a standard Q-learning method in terms of delay time.

In most of the related works, the authors dealt with the problem in a naive way. I.e., they made either simplifying assumptions (e.g. a simplified traffic network with the same lane numbers for all streets, without noise, pedestrians, recurring congestion and non-recurring congestion) or almost impractical assumptions (e.g. computing the total waiting time of each vehicle during its travel by the traffic system) to develop new methods. In this paper, we design different A-CATs controllers, employ them on a real-world traffic network modeling the real variable traffic flows for 24-h and with different traffic disruptions (incidents, impatient pedestrians, parking lanes and noisy sensor data) and we also study their robustness against the traffic disruptions.

3. Background: discrete and continuous state actor-critic

3.1. Discrete state actor-critic

Actor-critic algorithms figure high among the most efficient and widely used algorithms in RL (Sutton and Barto, 1998). Their properties that comprise the advantages of actor-only and critic-only methods (Grondman et al., 2012) have made them one of the most preferred RL algorithms. The processing unit of the actor-critic method is composed of two parts: (1) a critic that estimates how the current state brings the agent closer to its long-term objective, and (2) an actor that selects an action. After each action selection, the critic assesses the new state to determine whether conditions have gone better or worse than expected. That evaluation is based on the temporal difference (TD) error:

$$TD = \delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t), \quad 0 \leqslant \gamma \leqslant 1 \tag{1}$$

In Eq. (1), s_t is the discretized state at time t,γ is the discount factor that represents the difference in importance between future rewards and instant rewards, r_{t+1} is the instant reward and $V(s_{t+1})$ as the state value indicates how well the state s at time $t+1$ is, based on the long-term objective.

Since past states are also responsible for the achieved TD error, eligibility traces are employed to make agents learn more rapidly. The new values of states based on the TD error and the eligibility traces are calculated by Eq. (2), where $0 < \alpha \leqslant 1$ is the learning rate for the critic, γ is the discount factor, s is the discretized state, λ is the decay parameter and $e_t(s)$ is the eligibility trace for the state s at time t.

$$\begin{aligned} &V(s) = V(s) + \alpha\delta_{t+1}e_{t+1}(s), \textit{for all } s \\ &e_{t+1}(s) = \begin{cases} \gamma\lambda e_t(s) & \textit{if} \quad s \neq s_t \\ \gamma\lambda e_t(s) + 1 & \textit{if} \quad s = s_t \end{cases} \quad 0 \leqslant \gamma, \lambda \leqslant 1 \end{aligned} \tag{2}$$

Since at the beginning of the simulation the agents do not have enough knowledge about which action in the current state is optimal in terms of long term reward, they need to explore different actions to find the optimal one. In order to implement exploration, the agents possess an ϵ–greedy exploration strategy in which actions that seem to be not optimal are selected with probability ϵ as:

$$Pr(s_t,a) = \begin{cases} 1-\epsilon + \frac{\epsilon}{|A_s|} & if \ a = argmax_{a'\in A_s} P(s_t,a') \\ \frac{\epsilon}{|A_s|} & else \end{cases} \tag{3}$$

In Eq. (3), $|A_s|$ is the number of actions, a is each action available in state s_t and the parameter ϵ is the probability that the agent explores by randomly selecting an action rather than exploiting its knowledge by selecting the action that it thinks is most beneficial for obtaining the highest long term reward. During the beginning of the simulation, the agents select a random action in order to gain knowledge from the environment. As the number of epochs (simulation steps) increases, they gradually change their method of action selection by turning their attention towards the learned knowledge. After selecting an action, the new values of all state-action pairs are updated by Eq. (4):

$$\begin{aligned} &P(s,a) = P(s,a) + \beta\delta_{t+1}e_{t+1}(s,a), for \ all \ s, \ a \\ &e_{t+1}(s,a) = \begin{cases} \gamma\lambda e_t(s,a) & Otherwise \\ \gamma\lambda e_t(s,a) + 1 & if \ s = s_t \ and \ a = a_t \end{cases} \quad 0 \leqslant \gamma, \ \lambda \leqslant 1 \end{aligned} \tag{4}$$

In this equation, $0 < \beta \leqslant 1$ is the learning rate of the actor and $P(s,a)$ indicates the tendency to select each action a in each state s. For employing the described discrete actor-critic method, it is necessary to discretize the state variables.

3.2. Continuous state actor-critic

Discretizing the continuous variables used in the definition of the state space and using a tabular representation naturally result to losing the generalization property, the ability of a system to perform accurately on unseen data, and subsequently increasing the number of required learning trials. In this context, a continuous state actor-critic algorithm which uses generalization performs more properly. In a practical point of view, there is a natural metric on the state space such that close states have similar values and thus, it enables the learning agents to handle states never exactly seen before.

More accurately, the value of each state to be approximated at time t ($V(s_t)$) is represented as a linear function $\theta^T\cdot\phi(s_t)$ where ϕ is a mapping function and θ is a vector of parameters (Sutton and Barto, 1998). In this article, we employ tile coding and radial basis functions (RBFs) as the two different mapping functions and perform a comparison between them.

The basic principle behind tile coding (Albus, 1975) is to approximate the value function by using a piecewise-constant function. This approach generalizes the state space into partitions called tilings. Each tiling consists of a set of non-overlapping grid cells. Each cell is called a tile. The tilings are slightly offset in each dimension so that a particular state is mapped to different tiles in different tilings. The membership value of the triggering state to different tiles is either 0 or 1. In this research, each state variable is partitioned into a finite set of tiles and then the tiling (grid) is created by combining the tiles in each state variable in a vector. Each tiling has the same number of partitions in each dimension.

Unlike tile coding, RBFs are continuous and the degree of membership to different RBFs are in the range of $[0,1]$ based on the distance between the triggering state and the center of RBFs. The value of an RBF is maximal at its center and drops off gradually away from the center. An RBF can be activated over a wide range of states. Eq. (5) is used to calculate the degree of membership to different RBFs in each state variable:

$$\varphi_i(s) = \exp\left(-\sum_{j=1}^{l} \frac{(s^j-\mu_{ij})^2}{2\sigma_{ij}^2}\right) \tag{5}$$

σ_{ij} and μ_{ij} are the standard deviation and center of RBF_i on the j-th dimension (j-th state variable). A larger standard deviation results in a flatter RBF. In this research, the centers are distributed evenly along each dimension, leading to m^l centers for l state variables and a given order m for the RBF.

Choosing the right number of tilings, tiles and RBF is critical for successful learning. Thus, to indicate the effect of the number of tilings, tiles and RBFs on the performance of the learning agents and find the optimal numbers, the continuous state actor-critic method is evaluated with different numbers of tilings (1, 3, 6, and 9), tiles (2, 5, and 8) and RBFs (2, 5, and 8). It is worth noting that σ values of RBFs are determined in a manner such that the distance between two consecutive RBF-centers is 2σ. The value function based on the TD error and eligibility traces is updated by Eq. (6), where $\phi(s_t) = [\varphi_1(s_t),\varphi_2(s_t),...,\varphi_n(s_t)]$ is a mapping function, n is the total number of features and θ is a vector of parameters for the critic.

$$\begin{aligned} &Z_0 = 0 \\ &Z_{t+1} = \gamma\lambda Z_t + \phi(s_t) \\ &\delta_{t+1} = r_{t+1} + \gamma\theta_t^T\cdot\phi(s_{t+1})-\theta_t^T\cdot\phi(s_t) \\ &\theta_{t+1} = \theta_t + \alpha\delta_{t+1}Z_{t+1} \\ &V(s_t) = \theta_{t+1}^T\cdot\phi(s_t) \end{aligned} \tag{6}$$

The state-action values are updated by Eq. (7).

$$
\begin{aligned}
&Z'_0 = 0\\
&Z'_{t+1} = \gamma\lambda Z'_t + \Phi(s_t,a_t)\\
&\theta'_{t+1} = \theta'_t + \beta(r_{t+1} + \gamma\theta^T_{t+1}\cdot\phi(s_{t+1}) - \theta^T_{t+1}\cdot\phi(s_t))Z'_{t+1}\\
&P(s_t,a) = (\theta'_{t+1})^T\cdot\Phi(s_t,a_t)
\end{aligned}
\tag{7}
$$

In Eq. (7), θ' is a vector of parameters for the actor and $\Phi(s_t,a_t)$ is defined according to Eq. (8). Both θ' and Φ are $(n\times k)$-dimensional vectors where n is the total number of features and k is the number of actions ($|A_s|$).

$$
\begin{aligned}
&\Phi^T(s_t,\,a_t) = [\varphi_1(s_t)\cdot b_1,\,\ldots,\,\varphi_n(s_t)\cdot b_1,\,\varphi_1(s_t)\cdot b_2,\,\ldots,\,\varphi_n(s_t)\cdot b_2,\,\ldots,\,\varphi_1(s_t)\cdot b_k,\,\ldots,\,\varphi_n(s_t)\cdot b_k]\\
&b_i = \begin{cases}1, & if\ \ a_t = a_i\\ 0, & if\ \ a_t \neq a_i\end{cases}
\end{aligned}
\tag{8}
$$

The space and time complexities are other prominent issues to be considered in using actor-critic methods. Basically, most actor-critic methods have a constant space complexity over time. The space complexity of the critic is the number of parameters of its linear function approximator (the total number of tiles). The space complexity of the actor is the number of parameters of its linear function approximator times the number of actions (total number of tiles times number of actions). For example, in tile coding with 1 tiling layer and 5 tiles on each state variable, we need to store 5^V parameters in the critic (5^V is the total number of tiles) and $5^V \times W$ parameters in the actor. V is the number of state variables and W is the number of actions. In the tabular setting, the space complexity will be the size of the state space times the size of the action space. The time complexity for the policy to converge to the optimal policy is infinite. The reason is that there is stochasticity involved, and according to the law of large numbers, only an infinite amount of experiences leads to an exact value estimate. In practice, however, RL algorithms have been shown to find optimal or almost optimal policies in a reasonable time, depending on the complexity of the learning task.

4. Agent-based traffic simulation in downtown Tehran as a traffic-congested area

Downtown Tehran, due to the city's largest concentration of tall buildings, businesses, retail centers and shops, has much more traffic congestion in comparison to other districts of Tehran. In 1980, the municipality of Tehran has put in place comprehensive demand management to restrict vehicle access to this area. The restricted traffic zone, sometimes called congestion charging zone, is defined in such a way that only a limited number of vehicles such as emergency vehicles, taxi and certain groups including doctors and journalists have permission to enter the area during specific hours of the day. Nevertheless, the traffic congestion in the area is still so heavy that it needs an efficient traffic control measure like traffic signal control in order to handle the traffic demand. Thus, a part of Tehran's downtown area is selected as the study area of this research (see Fig. 1). The study area consists of 18 traffic analysis zones (TAZ) defined by the municipality of Tehran[1]. Also, as depicted in Fig. 1 the elevation of the entire study area ranges from 1260 to 1321 m. The elevations are employed for generating the grade of each street as well as for improving fuel consumption and emission rates modeling.

An agent-based traffic simulation is a practical method to assist in understanding the impact of new policies in traffic. The agent-based traffic simulation can be seen as being composed of two interactive parts: agents (e.g. vehicles, pedestrians, and traffic signals), and their environment (e.g. roads, lanes, and intersections). Any traffic condition is the result of interactions between these two parts. In this research, the AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks) software[2] and its Application Programming Interface (API) are employed for the agent-based traffic simulation and traffic control. The AIMSUN software provides a large number of possible indicators (e.g. travel time, stop numbers and fuel consumption), which can represent global performance of a whole network or performance of partial networks. Detailed information about the software has been published in Casas et al. (2010). Since the main core in the development of a microsimulation model is the calibration of the simulation model parameters to bring the model closer to observed conditions, in the following subsections different features of the traffic simulation (e.g. traffic network, vehicles, pedestrians and traffic signals) along with their behavior and calibrated parameters are explained.

4.1. The traffic network as the environment of the agents

The environment in which agents interact is the traffic network that is made up of arterials, junctions and pedestrians walkway facilities. In the study area, there are 50 junctions that are connected through 16 arterials. Among all the arterials, four of them are major and vital called Motahari, Beheshti, Shariati and Modares. Fig. 2 shows the network topology. The Motahari, Beheshti and Shariati streets are one-way principle arterials that run eastbound, westbound and northbound respectively. The Modares highway is a two-way arterial that runs north/south and has interchange connections, with ramps, to the Beheshti and Motahari streets. Road characteristics such as speed limits on the sections and turnings and the capacity of streets are set in the software. The value of the streets' capacity, expressed in vehicles per hour, that indicates the maximum sustainable flow rate at which vehicles can be expected to traverse an arterial were calculated by the Tehran Comprehensive Transportation and Traffic Company[3]. Also, the elevation of

[1] http://www.tehran.ir/.

[2] http://www.AIMSUN.com, Transporting Simulation Systems (TSS).

[3] http://trafficstudy.tehran.ir/.

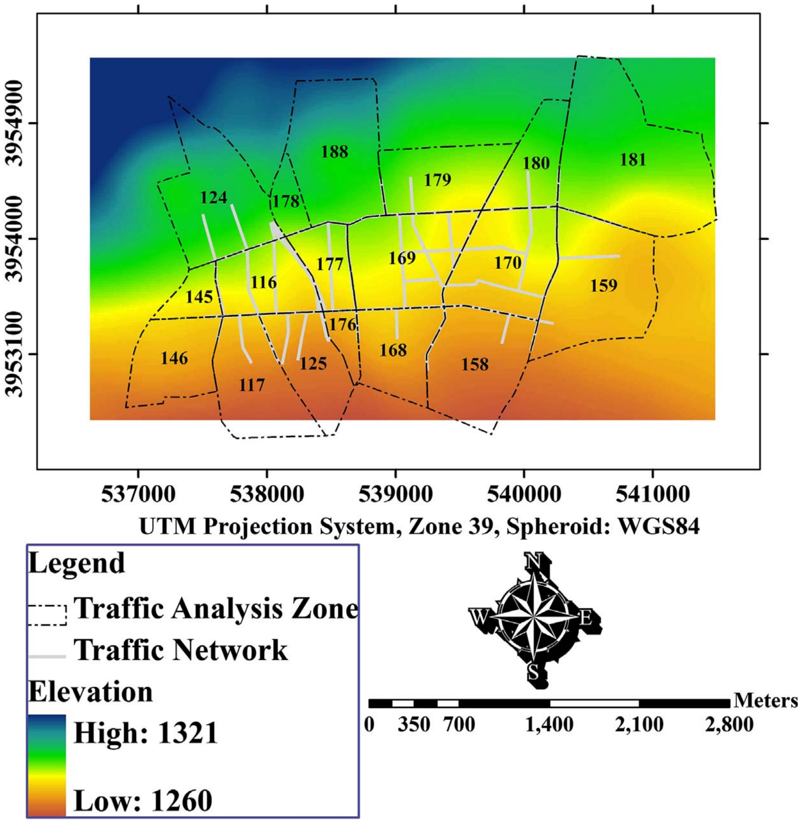


Fig. 1. Study area.

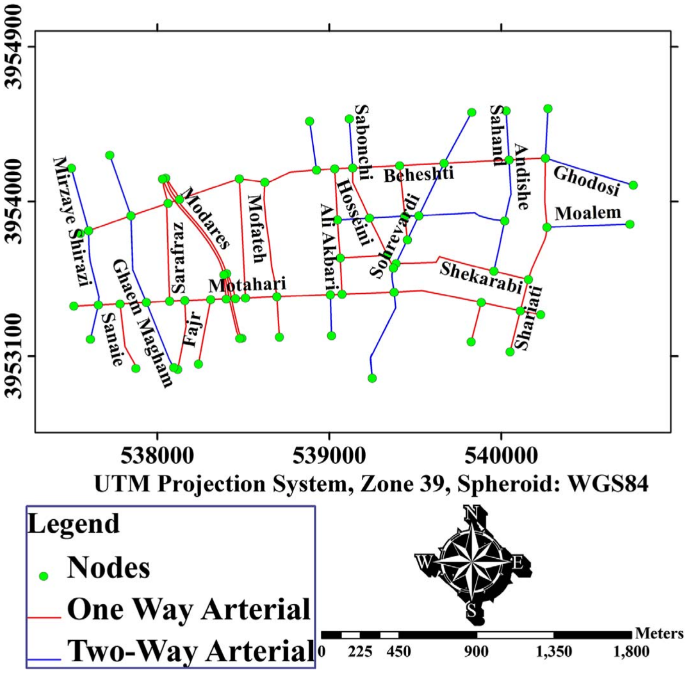


Fig. 2. Network topology.

Table 1
External properties of different vehicle types in the simulation.

Vehicle type	Proportion (%)	Length (mm)	Width (mm)	Maximum speed (km/h)	Maximum acceleration (m/s^2)	Deceleration (m/s^2)	Maximum deceleration (m/s^2)
Pride	27	3893	1605	140	1.26–1.46	3.10–3.80	4.65–5.75
Peugeot	22	4408	1694	190	2.32–2.52	3.30–3.90	5.00–5.90
Peugeot 206	19	3835	1652	170	1.77–1.97	3.30–3.90	5.00–5.90
Samand	10	4502	1720	185	2.13–2.33	3.30–3.90	5.00–5.90
Others	22	3900–4676	1600–1890	140–210	1.5–3.1	3.00–4.00	4.6–6.5

each street that makes the traffic network three-dimensional was entered in AIMSUN in order to improve the quality of the simulation. Since vehicles do not adhere to the lane discipline and drive between moving traffic, the number of vehicle queues (lanes) formed is different from the real lanes' number of streets. In fact, the practical number of lanes is more than the designed ones. In the present research, in order to make the traffic simulation more realistic, practical numbers of lanes are used instead of the nominal ones. The data of practical numbers of lanes were collected from field work and satellite images. In addition to the streets, pedestrians walking facilities such as central refuge, crosswalk and curb waiting area are added to the traffic network for simulating the pedestrians' crossing. It should be noted that the cross slope of all the crosswalks are overlooked in this research due to its trivial effects on the pedestrian movements.

4.2. Driver vehicles as the reactive agents

Driver-vehicle agents operate in a shared traffic environment and react to the neighboring driver-vehicle agents. The movement of a driver-vehicle agent depends on the external properties of the vehicle (vehicle type), such as length, width, maximum speed and acceleration, as well as internal characteristics of the human driver (driver behavior), such as reaction time, reaction time at stop, give way time, imprudent lane changing percentage, speed acceptance and overtaking maneuver percentage using a slower lane (undertaking maneuver).

From the external properties point of view, the national fleet in Iran consists of more than ten types of vehicles with different properties such as lengths, widths, maximum speed and maximum acceleration in the study area. According to the average proportion of each vehicle type, four vehicle types (Pride, Peugeot, Peugeot 206 and Samand) are chosen to be modeled. Given that the age of a vehicle can affect its properties including maximum speed, maximum acceleration, deceleration and maximum deceleration, these properties are mentioned in the bounded intervals to take into account the age of vehicles (see the first four rows in Table 1). Also, in order to consider the general effects of other types of vehicles, a new vehicle type under the name of 'Others' has been added to the traffic simulation. The external properties of this vehicle type are considered as interval variables to cover different vehicle types with different ages (see Table 1 last row).

Fuel consumption and emission rates are two other properties of the vehicles incorporated in the traffic simulation. The vehicle specific power (VSP) is employed in order to model the fuel consumption rate (Jimenez-Palacios, 1999). Regarding emission rates modeling, three different kinds of emissions including carbon monoxide (CO), hydrocarbons (HC) and oxides of nitrogen (NO_x) which are implicated in air pollution are modeled for the vehicles. Please see Appendix A for technical details of the fuel consumption and emission rates modeling.

Driver behavior modeling as one of the most crucial, if not the most crucial, aspect of the traffic simulation has a corresponding effect on the movement of driver-vehicle agents (Panou et al., 2007). The emergent driver behavior can be described as being composed of two levels: a strategic level and a maneuvering level. The strategic level specifies the general planning of the driver journey, e.g. the driver chooses the shortest path to reach the destination. At the maneuvering level which takes place while driving, some maneuvers and movements such as lane changing, overtaking, selecting the appropriate traveling speed and holding a safety gap to other vehicles are selected and executed to achieve short-term goals, e.g. avoiding traffic congestion on a specific segment of a street. Dijkstra's shortest path algorithm (Dijkstra, 1959) at the strategic level is used to find the routes with the minimum congestion. At each specific interval (15 min), the average travel times of all links are updated based on the traffic congestion, then drivers choose the route with the minimum travel time through Dijkstra's algorithm. Also, car-following, gap-acceptance and lane changing models at the maneuvering level (Casas et al., 2010) are employed for modeling the driver behavior. In order to bring the simulation closer to the observed behavior of drivers, the calibrated parameters of driver behavior models are used. These parameters were calibrated by DTT (2010). The most effective parameters on the traffic simulation and their calibrated values are:

- Reaction time (sec) as the parameter that plays a pivotal role in the car-following model is associated with the time that the following driver-vehicle agents respond to the speed changes of the preceding agent. The calibrated value is 0.9.
- Reaction time at stop (sec) is the time it takes a driver-vehicle agent to react to a change in the respective traffic signal or proceeding driver-vehicle agent. The calibrated value is 1.2.
- Minimum headway (sec) is the minimum possible temporal space between driver-vehicle agents. The calibrated range is [1.5–2.53].
- Speed acceptance is the degree of acceptance of speed limits. The calibrated range is [1–1.3].
- Undertaking maneuver (%) is the probability that a driver-vehicle agent overtakes the proceeding driver-vehicle agent by using a slower lane. The calibrated value is 15%.

4.3. Pedestrians as another group of reactive agents

Pedestrians are another type of reactive agents who have interactions with driver-vehicle agents while crossing the streets. The outcome of this interaction can be described as the pedestrian agent crossing behavior. According to a number of researches carried out on the pedestrian agent crossing behavior (Li, 2013; Wang and Tian, 2010; Onelcin and Alver, 2015), there are mainly two classes of pedestrian agents: (a) law-abiding pedestrian agents who only cross the streets when the pedestrian signal is green and (b) impatient ones who may cross the streets when the pedestrian signal is red. Impatient pedestrian agents, unlike law-abiding ones, have more interactions with driver-vehicle agents. Impatient pedestrian agents evaluate the gaps in traffic streams to decide whether it is large enough to go through it safely. Interestingly, observation surveys illustrated that many pedestrians prefer making the illegal crossing to waiting for the green pedestrian signal (Keegan and O'Mahony, 2003; Onelcin and Alver, 2015) i.e., the proportion of impatient pedestrian agents in comparison to law-abiding pedestrian agents is not small. Thus, impatient pedestrian agents due to their high interactions with driver-vehicle agents affect movements of driver-vehicle agents. In fact, the illegal crossing of pedestrian agents increases the stochasticity of the traffic environment for adaptive traffic signals.

In the pilot project that was carried out in the study area, it was observed that the green time served for driver-vehicle agents crossing one direction is sufficient to accommodate law-abiding pedestrian agents crossing another direction. Also, a lot of pedestrian agents are able to cross the street in a short time because of their seepage behavior. Furthermore, minimizing the driver-vehicle agents' waiting time can minimize the law-abiding pedestrian agents' waiting time to some extent. Therefore, in this research only the impact of pedestrian agents' illegal crossing behavior, as an element for producing chaos in the driver-vehicle agents' movements, on the learning behavior of traffic signals is considered. The speed and give way time of pedestrian agents are set in the range of 4–8 km/h and 10–30 sec respectively.

4.4. Modeling vehicle and pedestrian traffic demands as the recurring congestions and incidents as the non-recurring congestions

Traffic congestion is either recurring (daily expected traffic congestion) or non-recurring (unpredictable traffic congestion) (O'Flaherty, 1997). Both the recurring and non-recurring congestion result in stop-start driving conditions and consequently increasing fuel consumption and emission rates. The recurring congestion that exhibits a daily pattern usually occurs twice a day: during morning peak demand and evening peak demand. It is generally attributed to excess traffic demand, inadequate traffic capacity and poor traffic signal timing (Anbaroglu et al., 2014). In this research, the 24-h traffic demand with one-hour intervals for both the vehicles and impatient pedestrians are defined in terms of typical origin-destination (OD) matrices. In order to calculate the time-sliced OD matrices for all the vehicles of different types (Pride, Peugeot, Peugeot 206, Samand and Others), the primary OD matrix, collected from the municipality of Tehran, is adjusted by one-hour interval traffic count data obtained from traffic sensors (Cascetta, 2001). Regarding the fact that the traffic count data only indicate the total count of all types of vehicles, the adjusted OD matrices are general in terms of vehicle types. Assuming that the proportion of different vehicle types is constant for all the streets during 24-h, sub OD matrices for each kind of vehicle are generated by multiplying the adjusted OD matrices by the proportion of each vehicle type (Table 1). These 24 OD matrices that model morning and evening rush hours reflect the recurring traffic patterns. Fig. 3 shows the variations of traffic demand for the westbound street (Beheshti) approaching the Beheshti-Shariati intersection. It can be inferred that the maximum traffic flow volumes happen between 10:00–12:00 and 18:00–20:00. Regarding the pedestrian traffic demand, impatient pedestrians count data were collected by fieldwork. For instance, Fig. 4 shows impatient pedestrians count data for northbound crosswalks of the Beheshti-Shariati intersection between 16:00 and 20:00.

In addition to the recurring congestion, the non-recurring congestion that corresponds to unexpected delays and can be attributed to unexpected events like incidents (Kwon et al., 2006) is considered in the present research. Sometimes the incidents during periods of peak demand can make the recurring congestion more severe and even lead to grid-lock. Several incidents within a short period of time at bottlenecks in a traffic network result in grid-lock. In order to demonstrate the impact of the non-recurring congestion on the performance of A-CATs agents, various incidents with different lengths (e.g. 10, 15 and 20 meters) and durations (e.g. 15, 30 and 45 min) randomly occur in the traffic network (refer to Section 5.4).

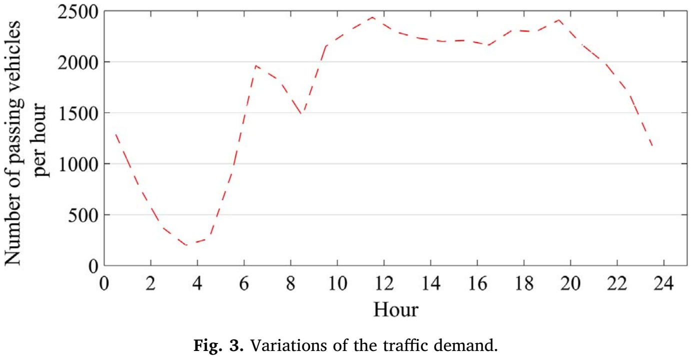
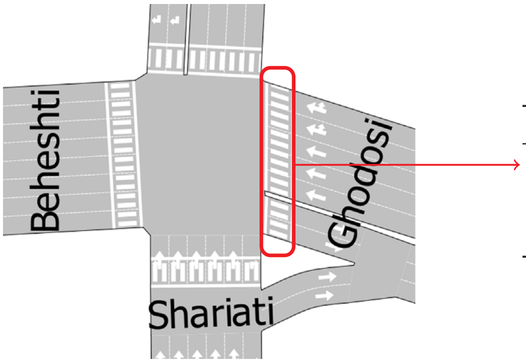


Fig. 3. Variations of the traffic demand.



Hour	Number of impatient pedestrians
16:00-17:00	22
17:00-18:00	31
18:00-19:00	34
19:00-20:00	10

Fig. 4. Impatient pedestrian count data.

4.5. A-CATs as the learning agents

An A-CATs agent is the learner and decision maker based on actor-critic methods (Section 3). It iteratively interacts with the stochastic traffic environment by receiving the prevailing traffic condition as a state of the environment and then chooses an action according to its past experiences (state-action value). In fact, traffic signal parameters are tuned in response to traffic fluctuations.

Each traffic signal has several principal components. Among others, a cycle and a phase are most important. A cycle is one complete sequence through all indications provided and a phase is a green interval plus a yellow interval in a cycle that is assigned to a fixed set of non-conflicting traffic movements (Koonce et al., 2008). In this research, each A-CATs agent determines the duration of each phase according to the current traffic condition.

The interaction frequency between A-CATs agents and the traffic environment is another challenge. The interaction frequency ranges from 1 sec (high interaction frequency) to 150 sec (low interaction frequency). The higher the interaction frequency, the higher the computational cost will be for the system. The lower the interaction frequency, the less adaptive A-CATs agents will be. In order to strike a balance between computational cost and adaptability, A-CATs agents interact with the traffic environment at the beginning of each phase. They sense the local traffic condition represented by a (1 + N)-dimensional feature vector. The first component is the index of the current green phase and each other component indicates the number of waiting and approaching vehicles (WAVE) on the streets leading to the junction. The advantage of this state definition is that the traffic load is encoded in the definition of the environment state, i.e. it has the benefits of being trivial to be measured by existing sensors. Another advantage is the management of vehicles with many passengers. For instance, a larger coefficient can be assigned to buses, taxis and other types of vehicles in public transportation with more passengers to provide them with more importance. The state of the environment can be presented to A-CATs agents in both continuous and discrete domains. In this research, both discrete and continuous state representations are employed (see Section 3 for more technical details).

The action of each A-CATs agent that follows a fixed phasing sequence scheme (El-Tantawy, 2012) is the duration of the current green phase, i.e., values of [0, 10, 20, 30, 40, 50, 60, 70, 80, 90] sec. The existence of zero in the action set makes the adaptive traffic signal agent more flexible in the sense that the phase can switch immediately if there is no traffic load on the streets associated with the current phase. A-CATs agents use the ϵ–greedy exploration strategy in order to trade-off between exploration and exploitation. It is worth noting that the traffic signal is switched to the next phase after serving the yellow time (5 sec). Also, the immediate reward signal provided to A-CATs agents after executing a given action is defined as the negative total number of WAVE on all the streets leading to the associated junction.

The agent-based traffic simulation is repeated for twenty days. Over the first six days, the exploration rate (ϵ) gradually decreases from 0.9 to 0.1. Between the seventh day and the thirteenth day, it is kept constant at 0.1 to enable A-CATs agents to explore various green time durations (training period). Over the last seven days, it is set to 0.0 so that A-CATs agents just employ the knowledge that they got from the training period and follow a greedy behavior (testing period). It should be noted that the traffic flow increases over the first four days in order to give A-CATs agents time to adapt to high traffic flows. Further in this paper, the performance during the first four days is omitted from the figures. The learning rates for the discrete state actor-critic and continuous state actor-critic based on RBF and tile coding algorithms are set to 0.15, 0.1 and 0.15 respectively. Also, the decay parameter (λ) is set to 0.9. These values were obtained based on a trial and error empirical parameter-tuning process. Moreover, each one hour is referred to as an episode. Algorithm 1 describes how each A-CATs agent works.

Algorithm 1. A-CATs

```
Time, t, tp, episodeTime ← 0
episodeLength ← the length of each episode
N ← number of streets leading to the junction
reactionTime ← reaction time of drivers
yellowTime ← the length of the yellow signal
set $\lambda$ and $\gamma$
$s_t$, $a_t$ ← initialize state and action
tp ← tp + $a_t$ + yellowTime
repeat
  episodeTime ← episodeTime + episodeLength
  set $\epsilon$ // $\epsilon$-greedy policy
  set OD-Matrix // traffic flow of traffic simulation
  repeat
    if abs(Time - tp) < reactionTime then
      observe number of WAVE on the streets leading to the junction
      $r_{t+1} \leftarrow -\sum_{i=1}^{N}$ (Number of WAVE)$_i$
      estimate the state, $s_{t+1}$, using WAVE and the current phase
      update critic (Eqs. (2) or (6))
      update actor (Eqs. (4) or (7))
      t ← t + 1
      select an action based on the $\epsilon$-greedy policy and $\epsilon$ (Eq. (3)), $a_t$
      tp ← tp + $a_t$ + yellowTime
    end if
    Time ← Time + reactionTime
  until abs(Time - episodeTime) < reactionTime
until Time > simulation time // end of simulation (20 days)
```

5. Experiments and results

In order to evaluate the impact of different traffic disruptions such as incidents, impatient pedestrians crossing, parking lanes and sensor noise on the learning of A-CATs, six kinds of scenarios are conducted. In the first scenario, the aim is to assess the learning of A-CATs without traffic disruption events, i.e. it is assumed that there are no incidents, impatient pedestrians, parking lanes and sensor noise in the traffic simulation. Each of the next four scenarios considers one of the disruption events in the whole traffic simulation. Finally, the last scenario aims to test the performance of A-CATs when the traffic disruption events are all activated in the traffic simulation. In all six scenarios, the performance of different types of A-CATs controllers (discrete state, continuous state based on tile

Table 2
Results of scenario 1 (results are averages over 10 simulations). The best 3 controllers for each index are shown in boldface.

Traffic signal controller	Average TT (sec/km)	Average SN (#/veh/km)	CO (kg)	HC (kg)	NO_x (kg)	Fuel (lit)
Fixed-Time	625 ± 21	2.07 ± 0.02	290 ± 9.5	13.2 ± 0.52	29.1 ± 0.6	5657 ± 150
Actuated	283 ± 24	1.72 ± 0.02	224 ± 10.7	9.33 ± 0.60	27.2 ± 0.7	5252 ± 273
Discrete State	262 ± 11	1.65 ± 0.02	205 ± 4.5	8.26 ± 0.25	25.4 ± 0.4	4992 ± 122
Tile coding (1,2)	316 ± 13	1.66 ± 0.02	202 ± 5.4	8.26 ± 0.29	24.7 ± 0.4	5005 ± 120
Tile coding (1,5)	**254 ± 18**	1.61 ± 0.03	200 ± 9.7	8.05 ± 0.51	25.0 ± 0.8	4800 ± 205
Tile coding (1,8)	297 ± 9	1.71 ± 0.04	220 ± 5.6	9.11 ± 0.29	26.6 ± 0.5	5352 ± 125
Tile coding (3,2)	277 ± 22	1.62 ± 0.04	**180 ± 3.6**	**7.14 ± 0.18**	**22.9 ± 0.6**	**4489 ± 122**
Tile coding (3,5)	**246 ± 13**	**1.60 ± 0.01**	197 ± 5.1	7.87 ± 0.29	24.8 ± 0.4	**4786 ± 114**
Tile coding (3,8)	295 ± 10	1.70 ± 0.01	217 ± 4.6	8.94 ± 0.24	26.3 ± 0.4	5245 ± 123
Tile coding (6,2)	288 ± 21	**1.60 ± 0.04**	**194 ± 8.4**	**7.82 ± 0.44**	**24.1 ± 0.7**	4868 ± 189
Tile coding (6,5)	259 ± 8	1.64 ± 0.01	203 ± 3.3	8.16 ± 0.18	25.3 ± 0.2	4914 ± 73
Tile coding (6,8)	315 ± 11	1.72 ± 0.02	225 ± 2.9	9.38 ± 0.15	26.8 ± 0.2	5409 ± 95
Tile coding (9,2)	328 ± 25	1.73 ± 0.03	210 ± 6.6	8.66 ± 0.37	25.2 ± 0.5	5057 ± 150
Tile coding (9,5)	279 ± 22	1.66 ± 0.03	213 ± 10.2	8.71 ± 0.55	25.9 ± 0.8	5111 ± 264
Tile coding (9,8)	329 ± 28	1.76 ± 0.04	233 ± 8.1	9.79 ± 0.44	27.5 ± 0.6	5613 ± 182
RBF (2)	541 ± 34	1.87 ± 0.05	229 ± 20.8	10.11 ± 0.99	24.7 ± 2.0	5529 ± 530
RBF (5)	**253 ± 10**	**1.61 ± 0.04**	**193 ± 8.5**	**7.73 ± 0.41**	**24.1 ± 0.9**	**4658 ± 205**
RBF (8)	277 ± 16	1.64 ± 0.02	203 ± 8.3	8.29 ± 0.43	24.9 ± 0.7	4919 ± 185

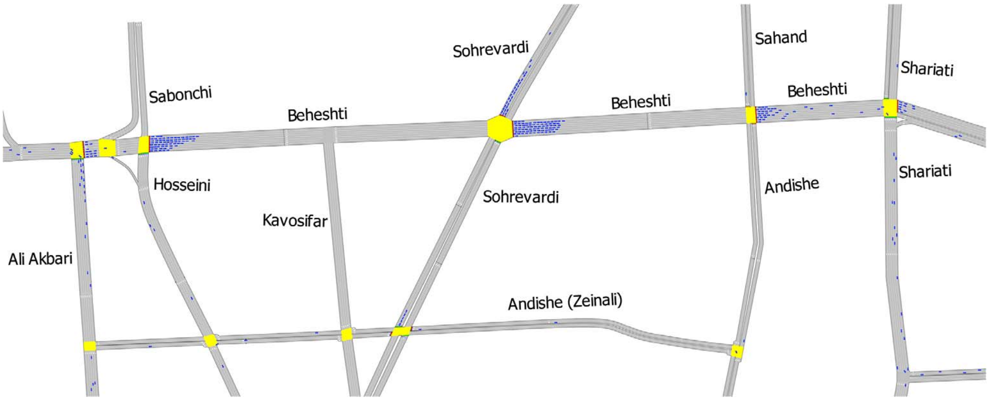


Fig. 5. Traffic simulation.

coding and continuous state based on RBF) is evaluated based on the following indexes: (a) average travel time (TT) per vehicle (sec/km), (b) average stop numbers (SN) (#/veh/km), (c) total CO, HC and NO_x emissions (kg) and (d) total fuel consumption (lit).

Morning and afternoon rush hours are two crucial periods of time in terms of traffic control because most people commute and desire to have a short travel time. Thus, assessing the performance of different types of A-CATs controllers in these two rush hours separately is of importance. However, in this research due to space limitations, only the results of the morning rush hour are presented.

Since the performance of continuous A-CATs controllers is highly dependent on the number of tilings, tiles and RBFs, the performance of different A-CATs controllers based on different numbers of tilings (1, 3, 6, and 9), tiles (2, 5, and 8) and RBFs (2, 5, and 8) are compared in order to find the optimal set-up of the function approximator in each scenario. In Tables 2 to 7, Tile coding (X, Y) refers to the tile coding method with “X” as the number of tilings and “Y” as the number of tiles in each dimension and RBF (Z) refers to “Z” as the number of RBFs in each dimension.

5.1. Scenario 1 (Base-case scenario: traffic network without traffic disruption)

Scenario 1 rests on the traffic network without incidents, sensor noise, impatient pedestrians and parking lanes. Fig. 5 shows the simulation of traffic in the northern part of the study area. Fig. 6 presents the learning performance of some of the best A-CATs controllers in terms of average TT. It is clear that Tile coding (3,5) outperforms others. Table 2 compares the average performance of different A-CATs over the last seven days by different indexes. At first glance, it is evident that based on the different performance indexes there are different best traffic signal controllers. Nonetheless, the A-CATs controllers based on Tile coding (3,5) and RBF (5) are the best ones based on the summation of ranks in different indexes. From the environmental standpoint, Tile coding (3,2) is the most valuable controller because it leads to the lowest CO, NO_x and HC pollution, as well as fuel consumption.

The A-CATs controllers based on RBF (2), Tile coding (1,2) and Tile coding (9,2) have the worst performance among all the learning controllers. In fact, they do not learn well to adapt their timing plan to the traffic fluctuations. Therefore, we decided to overlook them in the next scenarios in which we test more complex traffic environments. The significant performance difference between the best and the worst learning controllers proves the vital role of function approximation in designing A-CATs controllers. We also benchmarked the A-CATs controllers against fixed time and actuated controllers. In the fixed time controller, the timing of signals is fixed during the whole simulation. In the actuated traffic controller, there are some detectors on all the streets leading to the associated intersection in order to register whether or not there is a vehicle waiting or approaching. Detectors allow for variable

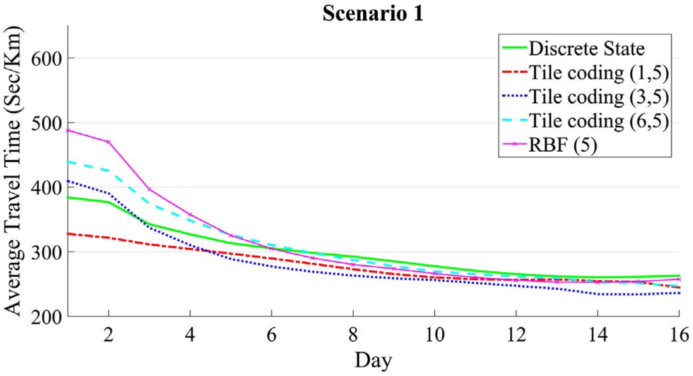


Fig. 6. Average TT (sec/km) for different A-CATs controllers.

Table 3
Results of scenario 2 (averages and standard deviations are computed using 10 simulations). The best 3 controllers for each index are shown in boldface.

Traffic signal controller	Average TT (sec/km)	Average SN (#/veh/km)	CO (kg)	HC (kg)	NO$_x$ (kg)	Fuel (lit)
Discrete State	296 ± 20	1.76 ± 0.02	209 ± 8.2	8.48 ± 0.45	25.7 ± 0.6	5134 ± 164
Tile coding (1,5)	**258 ± 22**	**1.72 ± 0.02**	**199 ± 8.0**	**7.88 ± 0.45**	**25.2 ± 0.5**	**4891 ± 175**
Tile coding (1,8)	321 ± 14	1.82 ± 0.02	222 ± 5.3	9.18 ± 0.29	26.8 ± 0.4	5431 ± 118
Tile coding (3,2)	354 ± 25	1.78 ± 0.03	214 ± 8.7	8.88 ± 0.48	25.6 ± 0.6	5219 ± 147
Tile coding (3,5)	**264 ± 12**	**1.75 ± 0.02**	**203 ± 6.3**	**8.14 ± 0.34**	**25.5 ± 0.5**	**4952 ± 127**
Tile coding (3,8)	347 ± 20	1.83 ± 0.02	230 ± 7.1	9.62 ± 0.39	27.3 ± 0.5	5594 ± 163
Tile coding (6,2)	407 ± 35	1.81 ± 0.03	222 ± 9.0	9.40 ± 0.48	25.8 ± 0.7	5452 ± 217
Tile coding (6,5)	297 ± 9	1.81 ± 0.03	217 ± 5.8	8.86 ± 0.31	26.5 ± 0.5	5249 ± 120
Tile coding (6,8)	369 ± 16	1.88 ± 0.03	239 ± 7.9	10.11 ± 0.42	27.9 ± 0.7	5784 ± 182
Tile coding (9,5)	319 ± 20	1.84 ± 0.04	224 ± 8.5	9.28 ± 0.45	27.0 ± 0.7	5377 ± 211
Tile coding (9,8)	387 ± 21	1.91 ± 0.02	246 ± 5.1	10.47 ± 0.28	28.5 ± 0.3	5924 ± 145
RBF (5)	**289 ± 16**	**1.74 ± 0.05**	**200 ± 10.4**	**8.09 ± 0.52**	**24.7 ± 0.9**	**4822 ± 241**
RBF (8)	345 ± 22	1.80 ± 0.04	219 ± 7.5	9.14 ± 0.39	25.9 ± 0.6	5259 ± 206

phase lengths by sensing momentary fluctuations of traffic flows (Roess et al., 2010). By comparison with the optimized actuated controller performance, the best A-CATs controller saves total fuel consumption by 11%.

5.2. Scenario 2 (Traffic network with impatient pedestrians)

Pedestrians with illegal crossing behavior are a prominent element in the learning of A-CATs controllers because their crossing during red pedestrian signals brings about chaos in the movement of driver-vehicle agents and consequently makes the environment more stochastic. We aim to consider how the learning process of A-CATs controllers as well as traffic congestion are affected by the crossing of impatient pedestrians. As shown in Table 3, the A-CATs controllers based on Tile coding (1,5), RBF (5) and Tile coding (3,5) have the best performance based on the summation of ranks in different indexes. Moreover, as it can be inferred from Table 3, five tiles in each dimension on average has a better performance in comparison to two and eight tiles. The reason is because the excess decrease and/or increase in the number of tiles in each dimension can bring about too much or too little generalization. Comparing the best A-CATs in scenario 2 with scenario 1 reveals that illegal crossing behavior leads to a 4% increase in the average TT in the study area.

5.3. Scenario 3 (Traffic network with parking lanes)

Parking beside the streets in the study area is common. This is owing to the fact that the study area is located in the administrative and commercial zone and drivers park their vehicles beside the streets to handle their businesses. This can decrease the number of lanes, and therefore the capacity of the streets and consequently make traffic congestion more severe for the same traffic demand. The data of parking places in the study area were collected from both field work and satellite images. This scenario aims to assess the negative effect of capacity reduction, that is due to the parking lanes, on the performance of A-CATs controllers. In Table 4, the performance of different A-CATs controllers after considering the parking lanes are shown. The A-CATs controllers based on RBF (5) and Tile coding (6,5) outperform others considering all the indexes. A comparison between the best A-CATs in scenario 1 and 3 indicates that parking lanes increase average TT by 4% in the study area.

Table 4
Results of scenario 3 (results are averaged over 10 simulations). The best 3 controllers for each index are shown in boldface.

Traffic signal controller	Average TT (sec/km)	Average SN (#/veh/km)	CO (kg)	HC (kg)	NO$_x$ (kg)	Fuel (lit)
Discrete State	325 ± 23	1.79 ± 0.05	220 ± 6.9	9.02 ± 0.37	26.7 ± 0.5	5137 ± 171
Tile coding (1,5)	273 ± 19	1.80 ± 0.08	215 ± 8.2	8.39 ± 0.37	27.9 ± 1.0	5218 ± 201
Tile coding (1,8)	309 ± 29	1.77 ± 0.05	217 ± 9.6	8.82 ± 0.53	26.9 ± 0.7	5115 ± 187
Tile coding (3,2)	499 ± 39	1.77 ± 0.05	214 ± 21.8	9.20 ± 1.07	**24.2 ± 2.0**	5118 ± 539
Tile coding (3,5)	**257 ± 31**	1.95 ± 0.22	223 ± 26.7	8.42 ± 1.01	30.1 ± 3.6	5532 ± 659
Tile coding (3,8)	293 ± 30	1.75 ± 0.10	210 ± 12.4	8.45 ± 0.63	**26.3 ± 1.3**	**4847 ± 226**
Tile coding (6,2)	505 ± 68	2.02 ± 0.04	248 ± 19.7	10.7 ± 1.13	28.0 ± 1.1	5805 ± 374
Tile coding (6,5)	**261 ± 17**	**1.73 ± 0.06**	**205 ± 9.1**	**8.04 ± 0.38**	26.5 ± 1.3	**4977 ± 273**
Tile coding (6,8)	362 ± 52	1.79 ± 0.04	226 ± 10.3	9.42 ± 0.63	26.8 ± 0.5	5104 ± 136
Tile coding (9,5)	283 ± 19	**1.74 ± 0.03**	**208 ± 6.5**	**8.27 ± 0.31**	26.4 ± 0.8	5016 ± 235
Tile coding (9,8)	406 ± 58	1.86 ± 0.05	243 ± 13.1	10.34 ± 0.79	28.0 ± 0.6	5424 ± 116
RBF (5)	**269 ± 28**	**1.68 ± 0.09**	**200 ± 14.7**	**7.83 ± 0.66**	**25.7 ± 1.6**	**4862 ± 358**
RBF (8)	346 ± 36	1.80 ± 0.06	225 ± 11.7	9.26 ± 0.59	27.3 ± 1.1	5312 ± 263

Table 5
Three defined incident rates.

Incident rate	Incident frequency	Incident duration	Incident length	Number of blocked lanes	Starting time
Low	4 times a day	15 (min)	10 (m)	2	Random between 7:00 and 19:00
Medium	6 times a day	30 (min)	15 (m)	3	Random between 7:00 and 19:00
High	8 times a day	45 (min)	25 (m)	4	Random between 7:00 and 19:00

5.4. Scenario 4 (Traffic network with incidents)

Robustness of adaptive traffic signals to non-recurring traffic congestion is a key factor in designing A-CATs controllers. When an incident (accident, vehicle breakdown and spill) 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 the environment becomes more non-stationary for A-CATs controllers. The higher incident rates (higher incident frequency, longer incident duration, a higher number of blocked lanes and longer incident length) lead to more traffic congestion and more non-stationarity of the environment. We should be assured that, when incidents occur in the traffic network, A-CATs controllers do not deliver poor performance and they can still converge to near-optimal solutions. To this end, three different incident rates: low, medium and high are defined to thoroughly evaluate the non-recurring congestion effect on the performance of A-CATs controllers (see Table 5). 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 A-CATs controllers are able to logically react in the presence of the random shocks. Table 6 indicates the average travel time per vehicle and the total fuel consumption for different traffic signal controllers in each incident rate. In the low incident rate, RBF (5) and Tile coding (1,5) perform best; in the medium incident rate, RBF (5) and Tile coding (6,2) perform best and finally in the high incident rate, RBF (5) and Tile coding (1,5) perform best in terms of rank summation of different indexes. Also, in all three incident rates, Tile coding (9,8) has the poorest performance. Therefore, based on the performance of different A-CATs controllers, the A-CATs controllers based on RBF (5) and Tile coding (1,5) can be selected as the best ones. The results show that the high incident rate ruins the performance of the best A-CATs up to 13% in terms of average TT when compared to scenario 1.

5.5. Scenario 5 (Noisy sensor data and reward signal)

In this scenario, we investigate the performance of different A-CATs controllers through a perspective of noise-sensitivity. The objective of this scenario is to evaluate the impact of noise on the performance of A-CATs controllers. To achieve this purpose, it is assumed that the sensors are imperfect and noisy and may provide A-CATs controllers with noisy data for identifying the current traffic condition based on the current observation. It is also assumed that environmental feedback on the green time interval quality is noisy. In fact, the number of WAVE which is employed for identifying the current state and reward signals may contain noise. More specifically, Gaussian noise is added to the number of WAVE to make the observation and reward signals noisy. It is worth noting that a negative noise can decrease the measured number of WAVE to a minimum of zero. We consider two different noise levels, low and high, with a zero mean and standard deviation of 1.5 and 3.0 vehicles respectively to study how the performance of different A-CATs controllers are affected by increasing the noise level. Table 7 presents the performance of traffic A-CATs controllers with different noise levels. Due to space limitations only the three performance indexes, average TT, total HC emission and total fuel consumption are presented. The A-CATs controller based on RBF (5) has the best performance based on the summation of ranks in different indexes for both low noise and high noise levels. Thus, according to the performance of different A-CATs controllers, the A-CATs controller based on RBF (5) is the best controller in this scenario. A comparison between the best A-CATs in scenario 1 and 5 indicates that low

Table 6
Results of scenario 4 (results are averaged over 10 simulations). The best 3 controllers for each index are shown in boldface.

	Low incident rate		Medium incident rate		High incident rate	
Traffic signal controller	Average TT (sec/km)	Fuel (lit)	Average TT (sec/km)	Fuel (lit)	Average TT (sec/km)	Fuel (lit)
Discrete State	279 ± 14	5030 ± 116	301 ± 18	5267 ± 170	316 ± 17	5351 ± 169
Tile coding (1,5)	**255 ± 15**	**4908 ± 197**	**280 ± 18**	5159 ± 158	291 ± 20	5171 ± 189
Tile coding (1,8)	297 ± 8	5353 ± 91	305 ± 12	5414 ± 126	313 ± 18	5554 ± 180
Tile coding (3,2)	330 ± 32	5023 ± 297	342 ± 24	**4778 ± 242**	384 ± 23	**5122 ± 197**
Tile coding (3,5)	**264 ± 14**	5027 ± 145	**274 ± 19**	5099 ± 98	**287 ± 18**	5198 ± 225
Tile coding (3,8)	305 ± 12	5385 ± 121	310 ± 12	5492 ± 133	317 ± 14	5509 ± 131
Tile coding (6,2)	311 ± 35	**4876 ± 224**	331 ± 35	**4619 ± 250**	362 ± 40	5219 ± 151
Tile coding (6,5)	**260 ± 22**	5038 ± 264	**276 ± 23**	5202 ± 216	**281 ± 12**	5235 ± 128
Tile coding (6,8)	318 ± 17	5498 ± 160	325 ± 15	5479 ± 123	330 ± 9	**5031 ± 120**
Tile coding (9,5)	283 ± 13	5094 ± 111	285 ± 9	5081 ± 163	**288 ± 15**	5251 ± 151
Tile coding (9,8)	329 ± 24	5629 ± 173	350 ± 7	5727 ± 82	362 ± 15	5860 ± 133
RBF (5)	279 ± 24	**4679 ± 202**	281 ± 26	**4678 ± 229**	290 ± 23	**4864 ± 165**
RBF (8)	292 ± 14	5095 ± 200	299 ± 19	5092 ± 144	310 ± 26	5261 ± 288

Table 7
Results of scenario 5 (results are averaged over 10 simulations). The best 3 controllers for each index are shown in boldface.

	Low noise			High noise		
Traffic signal controller	Average TT (sec/km)	HC (kg)	Fuel (lit)	Average TT (sec/km)	HC (kg)	Fuel (lit)
Discrete State	289 ± 6	8.64 ± 0.153	5121 ± 57	318 ± 22	9.26 ± 0.442	5387 ± 188
Tile coding (1,5)	**264 ± 10**	8.32 ± 0.216	4968 ± 96	**276 ± 7**	**8.55 ± 0.272**	**5102 ± 134**
Tile coding (1,8)	309 ± 12	9.56 ± 0.269	5541 ± 113	323 ± 7	9.61 ± 0.095	5588 ± 64
Tile coding (3,2)	277 ± 19	**7.83 ± 0.382**	**4800 ± 149**	315 ± 34	**8.85 ± 0.521**	**5197 ± 107**
Tile coding (3,5)	**258 ± 14**	8.33 ± 0.31	4949 ± 137	**287 ± 20**	9.04 ± 0.498	5239 ± 182
Tile coding (3,8)	303 ± 7	9.33 ± 0.156	5387 ± 65	328 ± 14	9.65 ± 0.371	5591 ± 139
Tile coding (6,2)	293 ± 29	**8.01 ± 0.753**	**4792 ± 297**	355 ± 11	9.22 ± 0.601	5317 ± 262
Tile coding (6,5)	282 ± 20	8.89 ± 0.35	5187 ± 125	294 ± 14	8.99 ± 0.368	5221 ± 159
Tile coding (6,8)	333 ± 16	9.84 ± 0.376	5641 ± 161	360 ± 27	10.37 ± 0.527	5860 ± 235
Tile coding (9,5)	271 ± 17	8.86 ± 0.395	5143 ± 168	294 ± 14	9.17 ± 0.282	5315 ± 107
Tile coding (9,8)	338 ± 15	9.83 ± 0.235	5649 ± 131	387 ± 15	10.74 ± 0.12	6082 ± 91
RBF (5)	**259 ± 19**	**7.99 ± 0.549**	**4741 ± 215**	**277 ± 20**	**8.13 ± 0.382**	**4877 ± 207**
RBF (8)	298 ± 11	8.54 ± 0.391	4976 ± 182	321 ± 11	9.07 ± 0.344	5224 ± 161

noise level increases average TT by 5% in the study area. The good performance of RBF (5) is expected because of its continuous manner in approximating the value function.

5.6. *Scenario 6 (Traffic network with incidents, noisy sensor data and reward signal, impatient pedestrians and parking lanes)*

Scenario 6 is a fusion of the previous scenarios and considers all the traffic disruptions including incidents, noisy sensor data and reward signals, impatient pedestrians and parking lanes simultaneously in order to assess the total impact of them. This scenario is implemented based on two experiments, optimistic and pessimistic. In the optimistic experiment the noise and incident rates are low while in the pessimistic experiment, it is assumed that the noise and incident rates are high. Table 8 indicates average TT, total HC emission and total fuel consumption for different A-CATs controllers. It can be seen that Tile coding (6,5) and RBF (5) perform best in the optimistic experiment and RBF (5) and Tile coding (1,5) perform best in the pessimistic experiment. Since RBF (5) has a very good performance in both experiments, it can be selected as the best controller in this scenario. Comparing the best A-CATs in scenario 6 with scenario 1 shows that the combined traffic disruptions in the optimistic and pessimistic experiments respectively increase average TT by 18% and 28%. Fig. 7 presents the convergence behavior of some A-CATs for the pessimistic experiment in terms of average TT. It is clear that Tile coding (1,5) performs best for this performance measure.

6. Discussion

In order to provide a better perception of the best A-CATs controllers in all scenarios, Table 9 summarizes the two best A-CATs controllers in all scenarios. It is evident that RBF (5) is always among the best two controllers in all six scenarios. This is because of the continuous manner of RBF in comparison to tile coding. To give an indication, Fig. 8 compares the smoothed learning curves of RBF (5) in scenarios 1 and 2. It can be inferred that the presence of impatient pedestrians leads to 40 sec/km increase in average TT. Also, we compare the performance of RBF (5) with the fixed time and actuated control systems in scenario 6. Tables 10 and 11 demonstrate the comparison results in both optimistic and pessimistic experiments. It is found that RBF (5) results in lower average

Table 8
Results of scenario 6: fusion of all traffic disruptions (results are averaged over 10 simulations). The best 3 controllers for each index are shown in boldface.

	Optimistic			Pessimistic		
Traffic signal controller	Average TT (sec/km)	HC (kg)	Fuel (lit)	Average TT (sec/km)	HC (kg)	Fuel (lit)
Discrete State	357 ± 34	9.13 ± 0.542	5162 ± 132	426 ± 20	10.20 ± 0.344	5401 ± 164
Tile coding (1,5)	**304 ± 17**	**8.37 ± 0.326**	5012 ± 105	**340 ± 17**	**8.77 ± 0.37**	**5203 ± 276**
Tile coding (1,8)	384 ± 19	9.91 ± 0.345	5458 ± 80	410 ± 32	10.44 ± 0.636	5651 ± 197
Tile coding (3,2)	608 ± 43	11.95 ± 0.445	6107 ± 174	612 ± 29	11.15 ± 0.532	5855 ± 255
Tile coding (3,5)	**298 ± 31**	8.40 ± 0.589	**4986 ± 312**	**340 ± 29**	**9.06 ± 0.50**	**5213 ± 153**
Tile coding (3,8)	366 ± 29	9.54 ± 0.411	5336 ± 175	400 ± 31	9.98 ± 0.563	5600 ± 154
Tile coding (6,2)	530 ± 32	10.5 ± 0.641	5283 ± 317	533 ± 57	11.08 ± 0.815	5753 ± 213
Tile coding (6,5)	**301 ± 20**	**8.23 ± 0.308**	**4996 ± 38**	**346 ± 28**	9.20 ± 0.444	5375 ± 117
Tile coding (6,8)	394 ± 57	9.94 ± 0.976	5477 ± 285	449 ± 24	10.87 ± 0.311	5801 ± 77
Tile coding (9,5)	327 ± 44	8.65 ± 0.879	5018 ± 333	396 ± 34	9.92 ± 0.572	5490 ± 85
Tile coding (9,8)	448 ± 30	10.86 ± 0.465	5728 ± 112	520 ± 24	11.95 ± 0.442	6113 ± 183
RBF (5)	312 ± 39	**7.9 ± 0.671**	**4579 ± 170**	380 ± 25	**8.58 ± 0.67**	**4756 ± 348**
RBF (8)	351 ± 29	8.9 ± 0.507	5072 ± 95	416 ± 41	9.71 ± 0.681	5289 ± 169

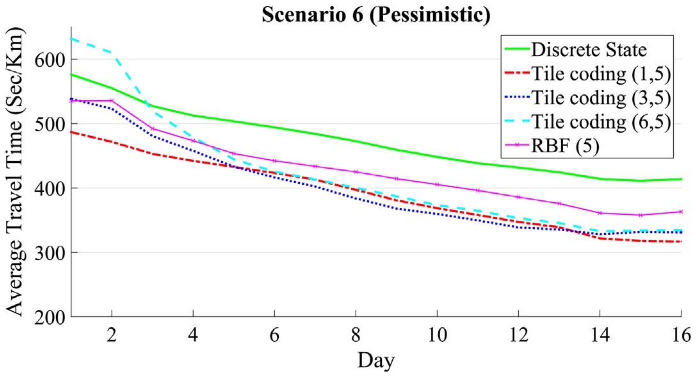


Fig. 7. Average TT (sec/km) for different A-CATs controllers.

Table 9
Best A-CATs controllers in different scenarios.

Scenario #	Sub-scenario	Two best A-CATs controllers
1 (Base scenario)		Tile coding (3,5), RBF (5)
2 (Impatient pedestrians)		Tile coding (1,5), RBF (5)
3 (Parking lane)		RBF (5), Tile coding (6,5)
4 (Incident)	Low incident rate	RBF (5), Tile coding (1,5)
	Medium incident rate	RBF (5), Tile coding (6,2)
	High incident rate	RBF (5), Tile coding (1,5)
5 (Noisy sensor data and reward signal)	Low noise rate	RBF (5), Tile coding (6,2)
	High noise rate	RBF (5), Tile coding (1,5)
6 (Fusion of all traffic disruptions)	Optimistic	Tile coding (6,5), RBF (5)
	Pessimistic	RBF (5), Tile coding (1,5)

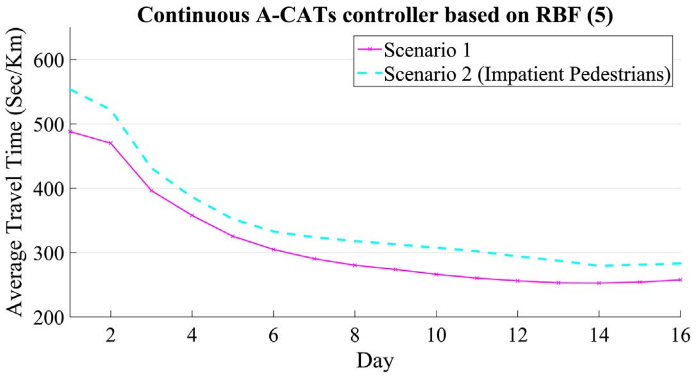


Fig. 8. Learning curves of the A-CATs controllers based on RBF (5) in scenarios 1 and 2.

Table 10
A comparison of the fixed time controller, the actuated controller and the A-CATs RBF (5) controller in the optimistic experiment.

	Fixed time	Actuated	A-CATs RBF (5)	% Improvements A-CATs RBF (5) vs. Fixed time	% Improvements A-CATs RBF (5) vs. Actuated
Avg. TT (sec/km)	763	377	311	59.2	17.5
CO (kg)	294	245	195	33.7	20.4
HC (kg)	13.5	10.4	7.9	41.5	24.0
NO_x (kg)	29.5	28.4	24	18.6	15.5
Fuel (Lit)	5700	5379	4579	19.7	14.9

TT, total emission and fuel consumption in comparison to the fixed time and actuated control systems. The most significant improvements are for average TT and the total HC emission.

In order to show the efficiency of A-CATs controllers, some of the best A-CATs are benchmarked against discrete state Q-learning (λ) (Watkins and Dayan, 1992; Sutton and Barto, 1998) and Bayesian Q-learning (Dearden et al., 1998) in the first scenario. Both

Table 11
A comparison of the fixed time controller, the actuated controller and the A-CATs RBF (5) controller in the pessimistic experiment.

	Fixed time	Actuated	A-CATs RBF (5)	% Improvements A-CATs RBF (5) vs. Fixed time	% Improvements A-CATs RBF (5) vs. Actuated
Avg. TT (sec/km)	791	449	380	52	15.3
CO (kg)	300	262	205	31.6	21.8
HC (kg)	13.8	11.4	8.6	37.7	24.6
NO_x (kg)	30.1	29.4	24.2	19.6	17.7
Fuel (Lit)	5882	5579	4756	19.1	14.8

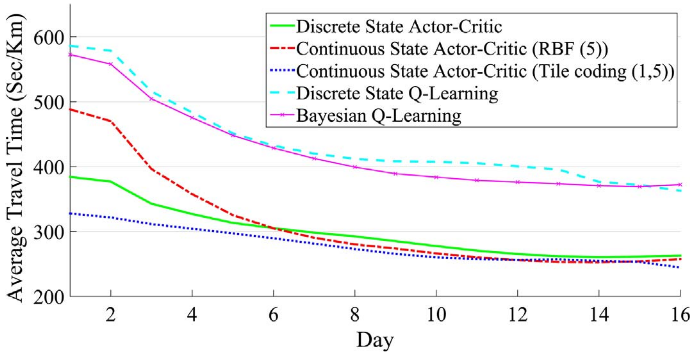


Fig. 9. Comparing the performance of continuous state actor-critic with discrete state Q-learning and Bayesian Q-learning for scenario 1.

discrete state Q-learning(λ) and Bayesian Q-learning are critic-only algorithms. Discrete state Q-learning(λ) employs an undirected exploration strategy. However, Bayesian Q-learning offers a principled approach to handle the exploration-exploitation dilemma by explicitly quantifying the value of exploration (Teacy et al., 2012; Ghavamzadeh et al., 2015). It maintains and propagates a probability distribution over the Q-values in order to compute the uncertainty the agent has about its estimation of the Q-values of each state (Duff, 2002). The myopic value of perfect information (VPI) is used to select actions and mixture updating for value updates (Dearden et al., 1998) is used in the employed Bayesian Q-learning (please refer to Appendix B for more details). In discrete state Q-learning(λ), ϵ-greedy is used to trade-off between exploration and exploitation and λ is set to 0.9. Fig. 9 compares the learning performance of A-CATs with discrete state Q-learning(λ) and Bayesian Q-learning. It is found that the A-CATs controllers outperform both discrete state Q-learning(λ) and Bayesian Q-learning. Also, it can be seen that Bayesian Q-learning has a slightly faster convergence speed in comparison to discrete state Q-learning(λ).

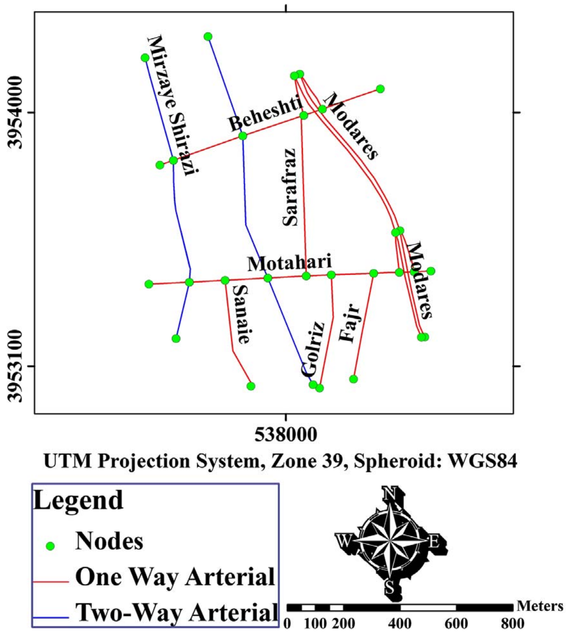


Fig. 10. Network topology of the small network.

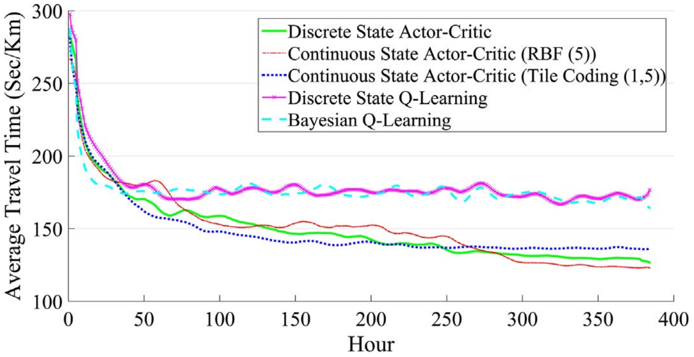


Fig. 11. Comparing the performance of continuous state actor-critic with discrete state Q-learning and Bayesian Q-learning in the simplified traffic environment.

In order to ensure seamless transferability of the system design and results, the performance of the A-CATs controllers is tested on a smaller network (Fig. 10) with a constant traffic flow. They are benchmarked against discrete state Q-learning(λ) and Bayesian Q-learning (Fig. 11). As shown in Fig. 11, A-CATs RBF (5) outperforms others in the simplified traffic environment.

7. Conclusion and future work

This paper has demonstrated the essence of discrete and continuous A-CATs controllers on a real-world traffic network. Within such a context, first, an agent-based traffic simulation of the study area was carried out. Then different A-CATs controllers based on discrete and continuous state actor-critic approaches were designed and the impact of different disruption events on their learning behavior was investigated.

Particularly, the paper considers the following features/dimensions: (1) traffic disruptions, (2) discrete and continuous state actor-critic approaches, and (3) function approximation definitions. It is found that the continuous A-CATs controller with the optimal function approximation outperforms the discrete one. In terms of function approximation, it is found that the excess decrease and/or increase in the number of tiles and RBFs in each dimension can bring about reducing good generalization in continuous RL. The RBF (5) function approximation results in better performances in comparison to tile coding. The reason is because of its continuous manner. Moreover, it is found that A-CATs RBF (5) leads to significant improvements against fixed time and actuated controllers.

This work intended to indicate that proposing and evaluating RL-embedded traffic signal controllers in an ideal and hypothetical traffic environment is not sufficient. This is one of the main reasons why RL-embedded traffic signal controllers are still only in papers and not on the streets. We hope this paper is a proper starting point for a closer scrutiny of different adaptive traffic signal controllers to achieve greater certainty in putting them into practice.

The work presented in this paper opens up various horizons for researchers. Future work can cover the following topics: (a) Designing a weather-sensitive traffic signal controller. The weather conditions affect driver behavior by changing their driving speeds, headways, reaction times, to name but a few. In fact, drivers become more cautious and accelerate more slowly in inclement weather. Therefore, optimal policies learned in typical clear conditions may not be optimal in other weather conditions. (b) Considering waiting time of both driver-vehicle agents and pedestrian agents to control traffic signals. The A-CATs controllers currently do not take into account pedestrians. This can lengthen their waiting time. When pedestrians wait too long, they get impatient and may cross the street during red pedestrian signals that bring about pedestrian fatal accidents. Thus, considering both vehicles and pedestrians in designing RL-embedded traffic signal controllers will be beneficial. (c) Modeling smaller vehicles (e.g. motorbikes and bikes) and their seepage behavior (passing between stationary vehicles) in the traffic environment and develop A-CATs controllers to take them into account. (d) Gaining further insights into function approximation and employing nonlinear function approximation in designing A-CATs controllers. (e) This research focused on discrete action RL that can, to some extent, decrease the flexibility of A-CATs controllers. Employing continuous action RL algorithms in order to provide A-CATs controllers with more flexibility in choosing green time durations could lead to better results.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Fuel consumption and emission rates modeling

In the traffic simulation, fuel consumption and emission rates are modeled. Concerning vehicle fuel consumption rate modeling, a modeling technique featuring engine power, the vehicle specific power (VSP), is employed (Jimenez-Palacios, 1999). VSP is defined as the instantaneous power demand of an engine per unit mass of a vehicle and has a high correlation with fuel consumption (Frey et al., 2008; Song and Yu, 2011). It depends on the speed, acceleration (deceleration) and roadway grade on the basis of second-by-

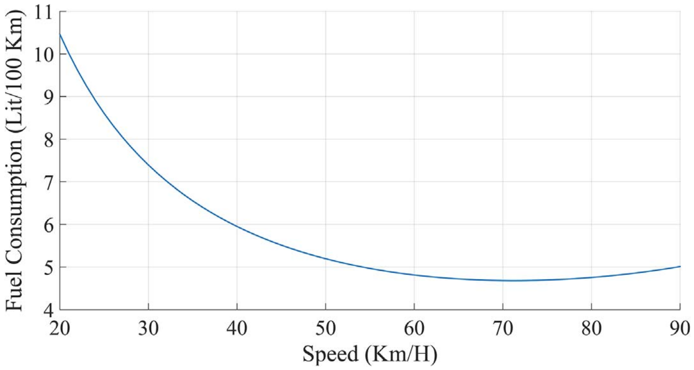


Fig. A.1. Fuel consumption rate at different speeds for a vehicle cruising at constant speed.

second cycles. The VSP values for urban vehicles are calculated by Eq. (A.1) (Jimenez-Palacios, 1999).

$$VSP = \frac{Power}{Mass} = v \times [1.1 \times a + 9.81 \times \sin(\mathrm{atan}(g)) + 0.132] + 0.000302 \times v^3 \tag{A.1}$$

VSP is vehicle specific power (m^2/s^3), v is the vehicle speed (m/s), $a$is the vehicle acceleration/deceleration rate (m/s^2) and g is the road grade.

According to the study conducted by the Air Quality Control Company of Tehran (AQCC, 2014), the exponential function fits the relationship well between the VSP and the fuel consumption rate for the vehicles manufactured in Iran (Eq. (A.2)).

$$Fuel\ Consumption\ Rate\left(\frac{\mathrm{lit}}{\mathrm{sec}}\right) = A_1 \times e^{B_1 \times VSP} + C_1 \times VSP + D_1 \tag{A.2}$$

In Eq. (A.2), A_1,B_1,C_1 and D_1 are the constant coefficients of the exponential function. The constant coefficients for the vehicles manufactured in Iran have been calibrated by the Air Quality Control Company of Tehran (AQCC, 2014). For example, the fuel consumption rate with the unit of liters per 100 km at different speeds for a vehicle cruising at a constant speed when the terrain is flat is presented in Fig. A.1. It is evident that the speed at which the fuel consumption rate is at its lowest value is around 71 km/h. Also, by replacing VSP with zero in Eq. (A.2) (i.e. idling state) the calculated fuel consumption rate is going to be 0.5286 (ml/sec). It should be noted that we assume that all the cars are gasoline vehicles and there are no electric vehicles. In order to model the energy consumption of electric vehicles, readers are referred to (Souza et al., 2016; Wang et al., 2013).

Regarding emission rates modeling, three different kinds of emissions including CO, HC and NO_x which are implicated in air pollution are modeled for the vehicles. Emission rates depending on the average speed of a vehicle can be expressed in the form of a 6th order polynomial function (Eq. (A.3)) (TRL, 1999; Boulter et al., 2009).

$$BER\left(\frac{\mathrm{g}}{\mathrm{km}}\right) = \frac{A_2 + B_2 v + C_2 v^2 + D_2 v^3 + E_2 v^4 + F_2 v^5 + G_2 v^6}{v} \tag{A.3}$$

BER is the basic emission rate in g/km for an urban vehicle on a road with a grade of zero, A_2-G_2 are coefficients and v is the average speed of the vehicle in km/h.

Apart from the average speed that affects the emission rate other parameters such as vehicle mileage, engine temperature, vehicle load and road grade affect the emission rate. However, only the latter one is considered in the present research. The road grade can increase or decrease resistance force against the tractive force of the vehicle. Increases or decreases in the resistance force have a corresponding impact on the emission rate. It is, however, naive to assume that extra emission produced on the uphill streets is entirely compensated by reduced emission on the downhill streets. To put it simply, the basic emission rate should not be used as such, but in conjunction with a scaling factor which is designed to address the effect of the road grade. The grade scaling factor can be calculated as the 6th order polynomial function by Eq. (A.4), where v is the average speed of the vehicle in km/h and a-g are coefficients for each pollutant type and grade class.

$$Grade\ scaling\ factor\ (\%) = a + bv + cv^2 + dv^3 + ev^4 + fv^5 + gv^6 \tag{A.4}$$

The coefficients in Eqs. (A.3) and (A.4) have been calibrated for different pollutants (CO, HC and NO_x) and different grade classes for the vehicles manufactured in Iran by AQCC (2014).

Appendix B. Bayesian Q-learning

Bayesian Q-learning is a Bayesian approach to Q-learning, in which the exploration-exploitation dilemma is handled by using a probability distribution over Q-values ($Q(s,a)$). Maintaining this distribution makes the problem of optimizing the exploration-exploitation trade-off straightforward. This distribution measures the uncertainty in the current estimation of state-action values. To be more specific, it assumes a normal distribution with an unknown mean $\mu_{s,a}$ and a precision $\tau_{s,a} = 1/\sigma_{s,a}$, where $\sigma_{s,a}$ is the unknown standard deviation of the distribution, over the total discounted reward (return). In order to model uncertainty through the standard Bayesian approach, Dearden et al. (1998) assumes that the joint distribution of $\mu_{s,a}$ and $\tau_{s,a}$ follows a normal-gamma distribution, i.e. $p(\mu_{s,a},\tau_{s,a}) \sim NG(m_{s,a},\lambda_{s,a},\alpha_{s,a},\beta_{s,a})$, where $\rho_{s,a} = \langle m_{s,a},\lambda_{s,a},\alpha_{s,a},\beta_{s,a} \rangle$ are hyperparameters that can be updated according to Theorem 1.

Theorem 1 (*Dearden et al., 1998; Teacy et al., 2012; DeGroot and Schervish, 2002*). *Let* $p(\mu,\tau) \sim NG(m,\lambda,\alpha,\beta)$ *be a prior density over the unknown parameters for normal distribution of return and let* $w = \{r_i\}_{i=1}^{n}$ *be a set of* n *independent samples of return with sample mean* $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} r_i$ *and sum of squares* $s^2 = \sum_{i=1}^{n}(r_i-\overline{x})^2$. *Then* $p(\mu,\tau|r_1,...,r_n) \sim NG(m',\lambda',\alpha',\beta')$ *where* $\lambda' = \lambda + n, m' = (\lambda m + n\overline{x})/\lambda', \alpha' = \alpha + n/2$ *and* $\beta' = \beta + s^2/2 + n\lambda(\overline{x}-m)^2/(2\lambda')$.

The hyperparameters provide not only a good estimation of $Q(s,a), m_{s,a}$, but also a representation of uncertainty that can be used to guide exploration in different ways. In this research, myopic *VPI* (Dearden et al., 1998; Teacy et al., 2012) is used for action selection. In this method, an exploration bonus is added to the expected Q-values that estimates the expected improvement in decision quality resulting from the new information (Eq. (B.1)).

$$a^* = \underset{a}{argmax}(E[Q(s,a)] + VPI(s,a)) \tag{B.1}$$

In this research, the *VPI* method developed by Teacy et al. (2012) is employed. The *VPI* for choosing action a in state $s, VPI(s,a)$, is calculated by Eq. (B.2).

$$VPI(s,a) = \begin{cases} (m_{s,a_2}-m).\ pr(\mu|\mu < m_{s,a_2}) + \vartheta_p(m_{s,a_2}), & for\ a = a_1 \\ (m-m_{s,a_1}).\ pr(\mu|\mu > m_{s,a_1}) + \vartheta_p(m_{s,a_1}), & otherwise \end{cases}$$

$$\vartheta(x) = \frac{\Gamma\left(\alpha-\frac{1}{2}\right)\sqrt{\beta}\left(1+\frac{\lambda(x-m)^2}{2\beta}\right)^{0.5-\alpha}}{\Gamma(\alpha)\Gamma\left(\frac{1}{2}\right)\sqrt{2\lambda}} \tag{B.2}$$

In Eq. (B.2), a_1 is the current best action in state s with the expected reward m_{s,a_1} and a_2 is the second best action with the expected reward m_{s,a_2} (Teacy et al., 2012). Another issue is to update the estimate of the distribution over Q-values after each transition. Since only immediate rewards can be directly observed, equations of Theorem 1 cannot be used directly. In this context, Mixture updating (Dearden et al., 1998) is used to update the hyperparameters $\rho_{s,a} = \langle m_{s,a},\lambda_{s,a},\alpha_{s,a},\beta_{s,a} \rangle$. In Mixture updating, the hyperparameters are calculated by Eq. (B.3), where $f(x)$ is the inverse of $g(y) = \log(y)-\psi(y)$ and $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function.

$$\begin{aligned} m &= \frac{E[\mu_{s,a}\tau_{s,a}]}{E[\tau_{s,a}]} \\ \lambda &= (E[\mu_{s,a}^2\tau_{s,a}]-E[\tau_{s,a}]\mu_0^2)^{-1} \\ \alpha &= \max(1 + \epsilon ,f(\log E[\tau_{s,a}]-E[\log\tau_{s,a}]) \\ \beta &= \frac{\alpha}{E[\tau_{s,a}]} \end{aligned} \tag{B.3}$$

$E[\tau_{s,a}],E[\mu_{s,a}\tau_{s,a}],E[\mu_{s,a}^2\tau_{s,a}]$ and $E[\log\tau_{s,a}]$ can be approximated by numerical integration (Dearden et al., 1998).

References

Abdoos, M., Mozayani, N., Bazzan, A.L.C., 2013. Holonic multi-agent system for traffic signals control. Eng. Appl. Artif. Intell. 26 (5–6), 1575–1587.

Abdoos, M., Mozayani, N., Bazzan, A.L.C., 2014. Hierarchical control of traffic signals using Q-learning with tile coding. Appl. Intell. 40 (2), 201–213.

Abdulhai, B., Kattan, L., 2003. Reinforcement learning: introduction to theory and potential for transport applications. Can. J. Civ. Eng. 30 (6), 981–991.

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. J. Transp. Eng. 129 (3), 278–285.

Adler, J.L., Satapathy, G., Manikonda, V., Bowles, B., Blue, V.J., 2005. A multi-agent approach to cooperative traffic management and route guidance. Transp. Res. Part B: Methodol. 39 (4), 297–318.

Albus, J.S., 1975. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). J. Dyn. Sys. Meas. Control 97, 220–227.

Anbaroglu, B., Heydecker, B., Cheng, T., 2014. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transp. Res. Part C: Emerg. Technol. 48, 47–65.

AQCC, 2014. The Coefficient of Emissions in the Warm State for Gasoline Light Duty Vehicles of Iran. Report, Air Quality Control Company of Tehran Municipality.

Araghi, S., Khosravi, A., Creighton, D., 2015. A review on computational intelligence methods for controlling traffic signal timing. Expert. Syst. Appl. 42 (3), 1538–1550.

Arel, I., Liu, C., Urbanik, T., Kohls, A.G., 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4 (2), 128–135.

Barto, A.G., Sutton, R.S., Anderson, C.W., 1983. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Trans. Syst. Man Cybern. SMC-13 (5), 834–846.

Bazzan, A.L.C., 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Auton. Agents Multi-Agent Syst. 18 (3), 342–375.

Bazzan, A.L., Kluegl, F., 2013a. Introduction to Intelligent Systems in Traffic and Transportation. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool.

Bazzan, A.L.C., Kluegl, F., 2013b. A review on agent-based technology for traffic and transportation. Knowl. Eng. Rev. 29, 375–403.

Bhatta, B., 2010. Analysis of urban growth and sprawl from remote sensing data. In: Advances in Geographic Information Science. Springer Verlag, Berlin Heidelberg.

Bifulco, G.N., Cantarella, G.E., Simonelli, F., Velona, P., 2016. Advanced traveller information systems under recurrent traffic conditions: network equilibrium and stability. Transp. Res. Part B: Methodol. 92, 73–87.

Boulter, P., Barlow, T., MacCrae, I., 2009. Emission Factors 2009: Report 3 – Exhaust Emission Factors for Road Vehicles in the United Kingdom. Report, TRL.

Casas, J., Ferrer, J.L., Garcia, D., Perarnau, J., Torday, A., 2010. Traffic simulation with AIMSUN. In: Barcelo, J. (Ed.), Fundamentals of Traffic Simulation. Springer, New York, NY, pp. 173–232.

Cascetta, E., 2001. Transportation systems engineering theory and methods. Applied Optimization, vol. 49 Springer.

Chowdhury, M.A., Sadek, A.W., 2003. Fundamentals of intelligent transportation systems planning. In: Artech House Its Library. Norwood, MA.

Darmoul, S., Elkosantini, S., Louati, A., Said, L.B., 2017. Multi-agent immune networks to control interrupted flow at signalized intersections. Transp. Res. Part C: Emerg. Technol. 82 (Suppl. C), 290–313.

Dearden, R., Friedman, N., Russell, S.J., 1998. Bayesian Q-learning. In: Proceedings of Fifteenth National Conference on Artificial Intelligence. AAAI Press, pp. 761–768.

DeGroot, M.H., Schervish, M.J., 2002. Probability and Statistics. Addison-Wesley, New York.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

DTT, 2010. Calibration of Traffic Engineering Software based on the Traffic Conditions in Tehran. Report, Department of Traffic and Transportation, Municipality of Tehran.

Duff, M.O., 2002. Optimal Learning: Computational Procedures for Bayes-adaptive Markov Decision Processes (Ph.D. Thesis). University of Massachusetts Amherst.

El-Tantawy, S., 2012. Multi-agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC) (Ph.D. Thesis). University of Toronto.

El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013. 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), 1140–1150.

El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2014. Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. J. Intell. Transp. Syst. Technol. Plan. Oper. 18 (3), 227–245.

Frey, H.C., Zhang, K., Rouphail, N.M., 2008. Fuel use and emissions comparisons for alternative routes, time of day, road grade, and vehicles based on in-use measurements. Environ. Sci. Technol. 42 (7), 2483–2489.

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., 2015. Bayesian reinforcement learning: a survey. Found. Trends. Mach. Learn. 8 (5–6), 359–492.

Gordon, R.L., Tighe, W., 2005. Traffic Control Systems Handbook. Report.

Grondman, I., Busoniu, L., Lopes, G.A.D., Babuska, R., 2012. A survey of actor-critic reinforcement learning: standard and natural policy gradients. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 42 (6), 1291–1307.

Islam, S.B.A., Hajbabaie, A., 2017. Distributed coordinated signal timing optimization in connected transportation networks. Transp. Res. Part C: Emerg. Technol. 80 (Suppl. C), 272–285.

Jeihani, M., James, P., Saka, A.A., Ardeshiri, A., 2015. Traffic recovery time estimation under different flow regimes in traffic simulation. J. Traffic Transp. Eng. 2 (5), 291–300.

Jimenez-Palacios, J.L., 1999. Understanding and Quantifying Motor Vehicle Emissions with Vehicle Specific Power and Tunable Infrared Laser Differential Absorption Spectrometer Remote Sensing (Ph.D. Thesis). Massachusetts Institute of Technology.

Katwijk, R.v., Schutter, B.D., Hellendoorn, J., 2009. Multi-agent control of traffic networks: algorithm and case study. In: Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC 2009). IEEE, pp. 316–321.

Keegan, O., O'Mahony, M., 2003. Modifying pedestrian behaviour. Transp. Res. Part A: Policy Pract. 37 (10), 889–901.

Khamis, M.A., Gomaa, W., 2014. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. Eng. Appl. Artif. Intell. 29, 134–151.

Kluegl, F., Bazzan, A.L.C., 2012. Agent-based modeling and simulation. AI Mag. 33 (3), 29–40.

Konda, V.R., Tsitsiklis, J.N., 2003. On actor-critic algorithms. SIAM J. Control Optim. 42 (4), 1143–1166.

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., Bonneson, J., Tarnoff, P., Urbanik, T., 2008. Traffic Signal Timing Manual. Report FHWA-HOP-08-024, Federal Highway Administration.

Kwon, J., Mauch, M., Varaiya, P., 2006. 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), 84–91.

Li, B., 2013. A model of pedestrians' intended waiting times for street crossings at signalized intersections. Transp. Res. Part B: Methodol. 51, 17–28.

Ma, W., An, K., Lo, H.K., 2016. Multi-stage stochastic program to optimize signal timings under coordinated adaptive control. Transp. Res. Part C: Emerg. Technol. 72 (Suppl. C), 342–359.

Mannion, P., Duggan, J., Howley, E., 2016. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In: McCluskey, L.T., Kotsialos, A., Muller, P.J., Kluegl, F., Rana, O., Schumann, R. (Eds.), Autonomic Road Transport Support Systems. Springer, Cham, pp. 47–66.

O'Flaherty, C.A., 1997. Traffic planning strategies. In: Transport Planning and Traffic Engineering. Butterworth-Heinemann, Oxford, pp. 132–153.

Onelcin, P., Alver, Y., 2015. Illegal crossing behavior of pedestrians at signalized intersections: factors affecting the gap acceptance. Transp. Res. Part F: Traffic Psychol. Behav. 31, 124–132.

Ozan, C., Baskan, O., Haldenbilen, S., Ceylan, H., 2015. A modified reinforcement learning algorithm for solving coordinated signalized networks. Transp. Res. Part C: Emerg. Technol. 54 (Suppl. C), 40–55.

Panou, M., Bekiaris, E., Papakostopoulos, V., 2007. Modelling driver behaviour in european union and international projects. In: Cacciabue, P.C. (Ed.), Modelling Driver Behaviour in Automotive Environments: Critical Issues in Driver Interactions with Intelligent Transport Systems. Springer, London, pp. 3–25.

Prashanth, L.A., Bhatnagar, S., 2011a. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In: 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1640–1645.

Prashanth, L.A., Bhatnagar, S., 2011b. Reinforcement learning with function approximation for traffic signal control. IEEE Trans. Intell. Transp. Syst. 12 (2), 412–421.

Roess, R.P., Prassas, E.S., McShane, W.R., 2010. Traffic Engineering. Pearson Higher Education, New Jersey.

Song, G., Yu, L., 2011. Characteristics of low-speed vehicle-specific power distributions on urban restricted-access roadways in beijing. Transp. Res. Rec. J. Transp. Res. Board. 2233, 90–98.

Souza, M.d., Ritt, M., Bazzan, A.L.C., 2016. A bi-objective method of traffic assignment for electric vehicles. In: IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 2319–2324.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. MIT Press, Cambridge.

Teacy, W.T.L., Chalkiadakis, G., Farinelli, A., Rogers, A., Jennings, N.R., McClean, S., Parr, G., 2012. Decentralized Bayesian reinforcement learning for online agent collaboration. In: In: Conitzer, V., Winikoff, M., van der Hoek, W., Padgham, L. (Eds.), Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 1. IFAAMAS, pp. 417–424.

TRL, 1999. Methodology for Calculating Transport Emissions and Energy Consumption. Report, Transport Research Laboratory.

van Otterlo, M., Wiering, M., 2012. Reinforcement learning and Markov decision processes. In: Wiering, M., Otterlo, M. (Eds.), Reinforcement Learning: State-of-the-Art. Springer, Berlin, Heidelberg, pp. 3–42.

Wang, X., Tian, Z., 2010. Pedestrian delay at signalized intersections with a two-stage crossing design. Transp. Res. Rec. J. Transp. Res. Board. 2173, 133–138.

Wang, Y., Jiang, J., Mu, T., 2013. Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing. IEEE Trans. Intell. Transp. Syst. 14 (3), 1331–1345.

Watkins, C., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3), 279–292.

Weiss, G., 1999. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge.

Wiering, M., 2000. Multi-agent reinforcement learning for traffic light control. In: 17th International Conference on Machine Learning, pp. 1151–1158.

Wiering, M., Vreeken, J., Veenen, J.V., Koopman, A., 2004. Simulation and optimization of traffic in a city. In: IEEE Intelligent Vehicles Symposium, pp. 453–458.