Transport

Developing adaptive traffic signal control by actor-critic and direct exploration methods
Aslani, Mesgari, Seipel and Wiering

ice | proceedings

Proceedings of the Institution of Civil Engineers

https://doi.org/10.1680/jtran.17.00085
Paper 1700085
Received 30/06/2017 Accepted 16/01/2018
Keywords: communications & control systems/traffic engineering/transport management

ICE Publishing: All rights reserved

ice
Institution of Civil Engineers
publishing

# Developing adaptive traffic signal control by actor-critic and direct exploration methods

**Mohammad Aslani** MSc
Research Fellow, Department of Industrial Development, IT and Land Management, University of Gävle, Gävle, Sweden (corresponding author: maslani@mail.kntu.ac.ir)

**Mohammad Saadi Mesgari** PhD
Associate Professor, Department of Geospatial Information System (GIS), Faculty of Geodesy and Geomatics Engineering, K.N. Toosi University of Technology, Tehran, Iran

**Stefan Seipel** PhD
Professor, Department of Industrial Development, IT and Land Management, University of Gävle, Gävle, Sweden; Division of Visual Information and Interaction, Department of Information Technology, Uppsala University, Uppsala, Sweden

**Marco Wiering** PhD
Assistant Professor, Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen, Groningen, the Netherlands

**Designing efficient traffic signal controllers has always been an important concern in traffic engineering. This is owing to the complex and uncertain nature of traffic environments. Within such a context, reinforcement learning has been one of the most successful methods owing to its adaptability and its online learning ability. Reinforcement learning provides traffic signals with the ability to automatically determine the ideal behaviour for achieving their objective (alleviating traffic congestion). In fact, traffic signals based on reinforcement learning are able to learn and flexibly react to different traffic situations without the need of a predefined model of the environment. In this research, actor–critic is used for adaptive traffic signal control (ATSC-AC). Actor–critic has the advantages of both actor-only and critic-only methods. One of the most important issues in reinforcement learning is the trade-off between exploration of the traffic environment and exploitation of the knowledge already obtained. In order to tackle this challenge, two direct exploration methods are adapted to traffic signal control and compared with two indirect exploration methods. The results reveal that ATSC-ACs based on direct exploration methods have the best performance and they consistently outperform a fixed-time controller, saving average travel time by 21%.**

## Notation

| | |
|---|---|
| $A_s$ | finite action space |
| $a$ | action |
| $a_t$ | action at time $t$ |
| $b_n$ | maximum deceleration desired by vehicle $n$ |
| $C(s_t)$ | number of visits for state $s_t$ |
| $E$ | mathematical expectation |
| $k$ | number of actions |
| $L_n$ | distance from front bumper to front bumper at rest |
| $P^{a}_{s_t s'}$ | probability of going from state $s_t$ to $s'$ after taking action $a$ |
| $P(s, a)$ | exploitation term |
| $Q^{\pi}:(s, a) \rightarrow R$ | state-action value function |
| $R^{a}_{s_t s'}$ | average reward for the transition from state $s_t$ to $s'$ by taking action $a$ |
| $R_t$ | return |
| $r$ | reward signal |
| $S$ | state space |
| $s_t$ | state of the environment at time step $t$ |
| $T$ | reaction time of vehicles |
| $t$ | time step |
| $\mathrm{Veh}_i$ | number of vehicles on the $i$th approaching street of the associated intersection |
| $V_n(t)$ | speed of preceding vehicle ($n$) at time $t$ |
| $V_{n+1}(t)$ | speed of vehicle $n+1$ at time $t$ |
| $V^{\pi}:S \rightarrow R$ | state value function |
| $x_n(t)$ | position of vehicle $n$ at time $t$ |
| $x_{n+1}(t)$ | position of vehicle $n+1$ at time $t$ |
| $\alpha$ | learning rate of critic |
| $\beta$ | learning rate of actor |
| $\Gamma$ | exploration factor |
| $\gamma$ | discount factor |
| $\delta_{t+1}$ | TD-error |
| $\epsilon$ | probability of taking an exploration action |
| $\eta$ | constant |
| $\pi$ | policy |
| $\chi(s, a)$ | exploration term |
| $\omega$ | parameter controlling exploration rate in Boltzman exploration method |

## 1. Introduction

Population growth and thus the increase in social and economic activities in cities lead to an increase in the demand for transportation (Bhatta, 2010). The increase in demand for transportation in cities renders current infrastructures incapable of responding to transportation needs. Also, in today's world, which is the era of speed, advances and novel technologies, the number of vehicles is rapidly increasing. This increase in the demand of transportation and the number of vehicles has led to the emergence of traffic congestion in cities. This congestion imposes large expenses on societies at different levels and in various aspects.

One of the most effective solutions to this challenge is to employ intelligent transportation systems (ITS) (Bazzan and Klügl, 2013a; Chowdhury and Sadek, 2003). ITS, without the imposition of large expenses for constructing new streets, plays an effective role in improving the traffic congestion issue. Moreover, it provides a flexible approach to effectively manage and control traffic (Chowdhury and Sadek, 2003). Traffic control consists of different components, of which traffic signal control is the key to success of ITS. The main focus of this study is developing adaptive traffic signal control (Araghi *et al*., 2015; Bazzan and Klügl, 2013b).

Adaptive traffic control has been performed based on two approaches over previous decades: (*a*) a centralised approach; (*b*) a distributed approach. In the first approach, there is a central unit which directly monitors the performance of the whole system and tunes traffic signal parameters in response to traffic fluctuations. The centralised framework is used in most control algorithms and the use of a single central unit for calculating the optimal control parameters makes this approach so appealing. However, there are several drawbacks to the first approach. First, it primarily needs a reliable network connectivity to the central unit and this means that a communication network is always required, and any failure in the communication network leads to reverting to the fall-back plan, which is usually a stand-alone traffic control. The second handicap is its poor scalability for expanding the network size. In fact, the central computer is required to be largely updated in the case of adding extra traffic signal controllers. The computational complexity is another drawback of the centralised approach which prevents a system from updating its optimal control parameters on-line (NCHRP, 2010). The Sydney coordinated adaptive traffic system (Sims and Dobinson, 1980) and the split cycle offset optimisation technique (Hunt *et al*., 1981) are examples of a centralised approach.

In the second approach, each controller is responsible for its local decision making. The controllers are generally similar to the first approach with regard to adapting to traffic fluctuations; however, they have several advantages. They are not required to broadcast compulsory real-time control commands over the communication network. Thus, the system is able to work even during communication breakdown. They are computationally less demanding because of their locality characteristics. They are scalable and easy to expand in such a way that new controllers can be added without complicated algorithm design. Also, the failure of one controller would not result in the failure of the whole system (robustness) (Busoniu *et al*., 2008). The distributed approach can be broadly categorised into two types: (*a*) classical distributed approach and (*b*) modern distributed approach.

The main idea of the classical distributed approach is based on upstream vehicle detection and a reliable estimation/prediction of queue length and traffic flow. Examples of a system that uses a distributed approach are the optimised policies for adaptive control strategy (Gartner, 1983), the real-time hierarchical optimised distributed effective system (Head *et al*., 1992) and Prodyn (Henry *et al*., 1983). They usually assume that the routing choice of vehicles is constant, whereas in real traffic patterns there is a variable route choice because of the traffic fluctuations.

The modern distributed approach is based on self-learning in multi-agent systems (MAS) (Weiss, 1999). In this context, reinforcement learning as a type of machine learning method that does not need initial knowledge of the environment is beneficial (Sutton and Barto, 1998). In reinforcement learning, agents never see examples of the correct behaviour, but instead receive reward signals indicating the quality of the selected action in the given traffic condition (Aslani *et al*., 2017; Kaelbling *et al*., 1996; Sutton and Barto, 1998). The agents try to find the best sequence of actions that maximises the long-term rewards (return).

Wiering (2000) proposed a framework based on model-based reinforcement learning. Waiting times at junctions are learned/estimated, and used to select the traffic light settings. Waiting times are also propagated to individual vehicles to enable them to modify their routes. In fact, there is co-learning between traffic signals and vehicles; that is, value functions are learned by both traffic signals and vehicles. The results indicated that the proposed method outperforms different fixed-time controllers. Choy *et al*. (2003) proposed an MAS for traffic control in which, at the lowest level, each agent is responsible for controlling an isolated intersection and, at the middle level, there are some coordinator agents that coordinate multiple agents at the lowest level. Finally, at the highest layer, one agent controls all of the middle-level agents. The implementation of agents rests on neural network and fuzzy logic methods. In Medina *et al*. (2010), reinforcement learning was employed to control traffic signals. The state space includes the number of vehicles on approaching streets and the numbers of vehicles stopped on departing lanes approaching adjacent intersections. The proposed method was benchmarked against a fixed time controller. The results showed that the proposed method has better performances in terms of delay time and the number of stops. Jin and Ma (2015) employed Q-learning and state action, reward state action for adaptive signal control in the context of a group-based phasing technique. The proposed method was tested on a four-legged intersection. Simulation of urban mobility, an open-source traffic simulation tool, was used to simulate the traffic of the intersection. The results indicated that the learning-based adaptive signal controller outperforms a fixed time controller.

In reinforcement learning, agents should select actions in such a way that they efficiently explore the environment, and meanwhile exploit their obtained knowledge to avoid obtaining low signal rewards (Miyazaki *et al*., 1997; Sledge and Príncipe, 2017). Owing to these two conflicting objectives, the exploration–exploitation trade-off is an important issue in reinforcement learning. In order to tackle this issue, two direct exploration methods that consider the agent's history in the environment to guide exploration are adapted and compared with two indirect exploration methods (Thrun, 1992; van Otterlo and Wiering, 2012). Also, among all the different reinforcement learning methods, the actor–critic method (Konda and Tsitsiklis, 2003) is selected because of its suitable convergence features (Berenji and Vengerov, 2003). In all of the above-mentioned work, the simulated traffic system is quite simple and far from reality. In the current research an attempt is made to simulate traffic and drivers' behaviours as closely to reality as possible.

The rest of this paper is organised as follows: Section 2 describes the principles of reinforcement learning. The proposed adaptive traffic signal controllers based on the actor–critic algorithm and the direct exploration methods are presented in Section 3. The microscopic traffic simulation and the traffic network used are presented in Section 4. Experimental results are presented in Section 5 and finally the paper is concluded in Section 6.

## 2. Introduction of reinforcement learning

Reinforcement learning is a field of machine learning in which an agent aims to learn optimal behaviour by trial-and-error interactions with a dynamic environment. In other words, reinforcement learning is a promising approach to find optimal decisions in an unknown environment through interaction (Sutton and Barto, 1998).

At each time step ($t$), first the agent receives information about the state of the environment ($s_t$) from the state space ($S$) through its sensors, it then selects an action ($a_t$) from a finite action space ($A_s$). The selected action changes the state of the environment to a new state ($s_{t+1}$). After taking an action, the agent receives a scalar reward signal ($r_{t+1}$) depending on whether its action has led it closer to realising its objective. The goal of the agent is to collect as much reward as possible. In fact, the agent learns the optimal behaviour (policy) that maximises the sum of rewards in the future (return). The optimal policy is defined as the policy that receives the highest expected discounted cumulative reward (Equation 1).

1. $$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$

In this equation, $R_t$ is return, and $0 \leq \gamma < 1$ is the discount factor that represents the difference in importance between future rewards and instant rewards. $\gamma = 0$ makes the agent myopic by only considering the current reward, while $\gamma \rightarrow 1$ will make it far-sighted.

A policy, $\pi(s, a) = \Pr\{a_t = a | s_t = s\}$, maps state $s$ to a probability distribution over actions. Being in search of an optimal policy, an actor–critic method needs to rank states, in order to decide on a good action. A common way to rank states is by computing and using a so-called state value function $V^{\pi}: S \rightarrow R$. The state value function estimates the return that can be expected when starting in a specific state $s$ and taking actions determined by policy $\pi$. The state value function can be estimated by Equation 2, where $a$ is an action of the agent, $s_t$ is the current state of the environment, $P^a_{s_t s'}$ is the probability of going from state $s_t$ to $s'$ after taking action $a$, and $R^a_{s_t s'}$ is the average reward for the transition from states $s_t$ to $s'$ by taking action $a$.

2. $$V^{\pi}(s_t) = \sum_a \pi(s_t, a) \sum_{s'} P^a_{s_t s'} [R^a_{s_t s'} + \gamma V^{\pi}(s')]$$

$P^a_{s_t s'}$ and $R^a_{s_t s'}$ are defined according to Equation 3.

3. $$R^a_{s_t s'} = E\{r_{t+1} | s_t, a, s_{t+1} = s'\}, \; P^a_{s_t s'} = \Pr\{s_{t+1} = s' | s_t, a\}$$

In reinforcement learning, an attempt is made to find the optimal policy that maximises the state value function (Equation 4).

4. $$\pi^* = \text{argmax}_{\pi}(V^{\pi}(s)) \; \forall s$$

Since the optimal action should be selected in each state in control problems, it is also necessary to define a state–action value function $Q^{\pi}: (s, a) \rightarrow R$. The state–action value function estimates the expected sum of the discounted rewards for an agent starting at state $s$, taking action $a$ and then following policy $\pi$ thereafter (Equation 5).

5. $$Q^{\pi}(s_t, a) = \sum_{s'} P^a_{s_t s'} [R^a_{s_t s'} + \gamma V^{\pi}(s')]$$

Both the state value function and the state–action value function can be represented in tabular form. For a tabular form, storing and updating values are simple and fast. A well-known method for estimating value functions (state values or state–action values) is temporal-difference learning. Temporal-difference learning is a model-free method for policy evaluation that adjusts the estimated value of a state based on the immediate reward and estimated value of the next state. In fact, in temporal-difference learning, the value function is bootstrapped from the next time-step. The temporal-difference (TD)-error $\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ is measured between the value at state $s_t$, and the value at the subsequent state $s_{t+1}$, plus any reward $r_{t+1}$ accumulated along the way (Sutton, 1988). Different reinforcement learning methods have been proposed based on temporal-difference learning. As described before, the actor–critic method is employed in this research.

## 3. Adaptive traffic signal control based on actor–critic (ATSC-AC)

ATSC-AC is a learning traffic signal controller that is able to adaptively handle different traffic situations through the actor–critic method. The actor–critic algorithm is one of the promising approaches in traffic signal control because of its advantages over actor-only methods and critic-only methods (Grondman *et al.*, 2012). The actor–critic method has a separate memory structure for the policy and the value function. The critic monitors the ATSC-AC performance and estimates the value function, which is then used to update the actor's policy parameters in order to improve the performance of ATSC-AC. In other words, the actor shapes the policy $\pi$ and selects the actions and the critic predicts the expected return while following the policy $\pi$. The value function is estimated using temporal-difference learning. In actor–critic, the policy is not directly deduced from the value function. Instead, the policy is updated using only a small step size ($\beta$) – that is, a drastic change in the value function will not lead to oscillatory behaviour in the policy.

At the beginning of each phase, each ATSC-AC located at each intersection senses the traffic state at its own intersection ($s_t$), then selects a green time duration as an action ($a_t$) based on the knowledge acquired from the environment. It waits to the end of the phase whose duration is the sum of the selected green time duration and a yellow time duration (5 s). At the end of the phase, ATSC-AC receives a reward signal from the environment ($r_{t+1}$), that describes the quality of the selected action in the traffic environment. The new traffic state ($s_{t+1}$) is also sensed at the end of the phase (beginning of the next phase).

### 3.1 State definition

The state definition plays an important role in the learning ability of ATSC-AC. Different state representations with various state space sizes can be defined according to diverse views. ATSC-AC with a big state space needs more trials in order to converge to the optimal or near-optimal policy. Yet a state space that is too small prevents ATSC-AC from finding the best policy due to the lack of enough information from the environment. Another point is the matter of practicality of the state definition – that is, it should not require complex input information that cannot be provided by the traffic control infrastructure. In this research, the state space is represented by a vector in which each of its elements is the number of vehicles on each approaching street. Through this state definition, the traffic load is encoded to some extent.

### 3.2 Action definition

In order to define the action space of each ATSC-AC, first, the minimum and maximum green time durations are specified. Then, the interval between them is split into $k$ discrete values. In this research, the minimum and maximum green time durations are set to 10 and 90 s These values were obtained by a trial-and-error process. The interval is split into nine different values: [10, 20, 30, 40, 50, 60, 70, 80, 90] s. In fact, ATSC-AC selects a green time duration among the mentioned values.

### 3.3 Reward definition

The immediate reward signal is defined as the negative total number of vehicles waiting on all approaching streets to the associated junction (Equation 6).

6. $$r_{t+1} = -\sum_{i=1}^{N} (\mathrm{Veh}_i)$$

where $\mathrm{Veh}_i$ is the number of vehicles on the $i$th approaching street of the associated intersection. This reward function leads to the overall objective of having the smallest amount of vehicles waiting at each intersection.

### 3.4 Critic and actor update

After the reward signal is received, the state value function is updated based on temporal-difference learning (Equation 7).

7. $$\begin{aligned} V(s_t) &= V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \\ &= V(s_t) + \alpha\delta_{t+1}, \quad 0 < \alpha \leq 1 \end{aligned}$$

where $\alpha$ is the learning rate. In fact, the critic assesses the new state ($s_{t+1}$) based on the TD-error ($\delta_{t+1}$) to determine whether conditions have gone better or worse than expected. If $\delta_{t+1} > 0$, the tendency to select $a_t$ should be strengthened for the future, but if $\delta_{t+1} < 0$ the tendency should be weakened. The strengthening or weakening can be implemented by increasing or decreasing $P(s_t, a_t)$ in the actor (Equation 8).

8. $$P(s_t, a_t) = P(s_t, a_t) + \beta\delta_{t+1}$$

where $0 < \beta \leq 1$ is the learning rate of the actor.

### 3.5 Trade-off between exploration and exploitation

ATSC-AC needs to make a trade-off between exploration of different actions in the environment and exploitation of the obtained information. In order to increase the average return and state values during learning, ATSC-AC should select the best actions – that is, actions with the highest $P(s, a)$. However, ATSC-AC cannot find the best actions (policy) without exploration. Also, it cannot be certain that the best action in short-term optimisation is really the optimal action in long-term optimisation without exploring different actions. Therefore, ATSC-AC should be assured that all actions are tried often enough (exploration) while still following a good policy.

There are two approaches for exploration: (*a*) indirect exploration, which is mainly driven by randomness without considering the previous history of the learning process; (*b*) direct exploration, which uses further knowledge (the ATSC-AC history in the environment) in order to influence the parts of the environment that ATSC-AC will further explore. Clearly, the more efficiently ATSC-AC explores, the less time is required for learning. It has been indicated (Thrun, 1992) that direct exploration approaches are superior to indirect methods in terms of learning time and cost. Various direct exploration strategies have been proposed in the literature (van Otterlo and Wiering, 2012). In direct exploration, actions are evaluated based on the combination of their values (exploitation term, $P(s, a)$) and an exploration term $\chi(s, a)$ according to Equation 9, where $\Gamma \geq 0$ is the exploration factor that is decreased over time to converge to a greedy policy.

9. $$\mathrm{Eval}\,(s_t, a) = P\,(s_t, a) + \Gamma\,\chi\,(s_t, a)$$

In each time step, the action having the maximum evaluation (Eval) is chosen. In this research, two different direct exploration strategies are empirically studied and compared. The first direct exploration technique is the state counter-based method proposed by Thrun (1992). In this method, the exploration term considers the number of visits of the current state and that of the successor state (Equation 10).

$$\text{10.} \quad \text{Eval}\,(s_t,\, a) = P\,(s_t,\, a) + \Gamma \frac{C\,(s_t)}{E\,[C\,(s_{t+1})|s_t,\, a]}$$

In Equation 10, $C\,(s_t)$ is the number of visits for state $s_t$ and $E\,[C\,(s_{t+1})_t,\, a]$ is the expected counter value for the state that results from taking an action. It is estimated by Equation 11 during the learning process where $\eta$ is a constant factor (Wyatt, 1997).

$$\text{11.} \quad \hat{E}_{t+1}[C\,(s_{t+1})|s_t,\, a] = (1-\eta)\hat{E}_t[C\,(s_{t+1})|s_t,\, a] + \eta C\,(s_{t+1})$$

Algorithm 1 indicates how each ATSC-AC performs based on the state counter-based method.

Algorithm 1. ATSC-AC

```
Initialise $\alpha$, $\beta$, $\Gamma$ and $\eta$
Initialise $V(s)$, $P(s, a)$, $C(s)$
$A = [10, 20, 30, 40, 50, 60, 70, 80, 90]$ (s) is the action set
$t \leftarrow 0$
loop
  $s_t$, $a_t \leftarrow$ initial state and action of the episode
  $C(s_t) \leftarrow C(s_t) + 1$
  repeat
      Set $\Gamma$
      Set the current phase duration to $a_t$ + yellow time
      Wait to the end of the phase
      Observe the number of vehicles on each approaching street
      Calculate reward $r_{t+1} = -\sum_{i=1}^{N} (\text{Veh}_i)$
      Perceive the state $s_{t+1}$
      $C(s_{t+1}) \leftarrow C(s_{t+1}) + 1$
      $\delta_{t+1} \leftarrow r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$
      $V(s_t) \leftarrow V(s_t) + \alpha\delta_{t+1}$
      $P(s_t, a_t) \leftarrow P(s_t, a_t) + \beta\delta_{t+1}$
      $\hat{E}_{t+1}[C\,(s_{t+1})_t,\, a_t] \leftarrow (1-\eta)\hat{E}_t[C\,(s_{t+1})_t,\, a_t] + \eta C\,(s_{t+1})$
      $\text{Eval}\,(s_t,\, a_t) = P(s_t,\, a_t) + \Gamma \frac{C\,(S_t)}{\hat{E}_{t+1}[C\,(s_{t+1})_t,\, a_t]}$
      $a_{t+1} \leftarrow \text{argmax}_a$ Eval $(s_{t+1},\, a)$
      $t \leftarrow t + 1$
  until $s_t$ in terminal
end loop
```

The second exploration method, which is called the state–action counter-based method, keeps track of the number of visits of state–action pairs ($C\,(s,\, a)$) instead of states ($C\,(s)$). In this method, which is simpler than the state counter-based method, the actions are evaluated by Equation 12. It is evident that this method does not need to estimate the expected counter-value of the next state.

$$\text{12.} \quad \text{Eval}\,(s_t, a) = P\,(s_t,\, a) - \Gamma\,(c\,(s_t,\, a))$$

In this research, the mentioned direct exploration techniques are compared with two indirect exploration methods, namely Softmax and $\epsilon$-greedy (Sutton and Barto, 1998), in order to investigate how the direct exploration methods can affect the performance relative to indirect ones. The idea behind $\epsilon$-greedy is that the best action (action with the highest value, $\text{argmax}_{a' \in A_s} P\,(s,\, a')$) is exploited with the probability of $1-\epsilon$ and other actions are randomly uniformly explored with the probability of $\epsilon$ in each trial according to Equation 13, where $|A_s|$ is the number of actions. The value of $\epsilon$ decreases by time as the controller gains more knowledge in order to turn its attention towards the obtained knowledge.

$$\text{13.} \quad \Pr\,(s_t,\, a) = \begin{cases} 1-\epsilon+\dfrac{\epsilon}{|A_s|}, & \text{if } a = \text{argmax}_{a' \in A_s} P\,(s,\, a') \\ \dfrac{\epsilon}{|A_s|}, & \text{else} \end{cases}$$

Softmax selects actions based on their probability proportion of the estimated values ($P\,(s,\, a)$). In fact, Softmax offers a more structured exploration by selecting actions in proportion to their estimated values (Equation 14).

$$\text{14.} \quad \Pr\,(s_t,\, a) = \frac{\exp\,(\omega Q\,(s_t, a))}{\sum_{j=1}^{k} \exp\,(\omega Q\,(s_t,\, a_j))}$$

where $k$ is the number of actions, and the parameter $\omega$ controls the exploration rate. The $\omega$ variable is small at first in order to ensure enough exploration, but over time it increases to give a higher probability of choosing the optimal action in order to exploit more. In this research ATSC-ACs based on different exploration techniques are developed through the application programming interface of Aimsun (Aimsun, 2008; transporting simulation systems).

## 4. Microscopic traffic simulation

Nowadays, microscopic traffic simulation plays an important role in designing and evaluating traffic control strategies such as traffic signal control. The reason is that a mathematical treatment of a problem has been found to be inadequate owing to the complex nature of traffic. In microscopic traffic simulation, the flow of traffic is analysed by modelling driver–driver and driver–street interactions. Driver–driver interactions show how drivers react to other drivers (vehicles) on the streets and driver–street interactions describe how drivers react to different features of a street. In fact, the individual base elements (e.g. driver) of the traffic system are first specified in more detail. These elements are then linked together to form a complete top-level traffic system. The traffic network, the drivers (vehicles) and traffic signals are the three main components of a microscopic traffic simulation.

Figure 1 presents the traffic network employed, which consists of nine intersections and 48 streets. Each intersection is controlled by one ATSC-AC. The phase sequence is fixed in each ATSC-AC (Figure 2) and ATSC-AC determines the length of phases at the beginning of each phase (algorithm 1). The length of each street is 250 m with two lanes at each side and a 50 km/h speed limit. In order to make the traffic environment non-stationary, the traffic demand from each source sequentially switches from 250 (Veh/h) to 450 (Veh/h), and then to 650 (Veh/h) every 20 min. The values have been selected such that they prevent the network from being oversaturated during the simulation. In each junction, 33·3% of all vehicles go straight on, 33·3% turn left and 33·3% turn right. The traffic simulation is carried out for 400 h.

Driver behaviour models describe the actions – for example, acceleration, deceleration and lane change – of each driver in response to its surrounding traffic environment. In fact, the idea behind driver behaviour modelling is that drivers would like to travel at their desired speed on each street section, but their surrounding traffic environment (i.e. preceding vehicle, adjacent vehicles and traffic signals) limits their behaviour. In the driver behaviour modelling, at each simulation step, the position and speed of each vehicle is updated according to lane changing, acceleration and deceleration models (Barcelo, 2010; Panou *et al.*, 2007).

Regarding lane changing, there are basically two motivations for a driver changing driving lanes: (*a*) to reach the desired speed when a vehicle ahead is driving slowly; (*b*) to get into the turning lane. The vehicle that would like to turn at the next turning point needs to get closer to the turning lane so that it can turn. For more information about the employed lane-changing model, readers can refer to Gipps (1986a, 1986b). In order to model deceleration of vehicles, an approach that is based on collision avoidance is employed (Gipps, 1981). The idea behind this approach is that a driver tries to keep at a distance from the lead vehicle in such a way that, in the case of any emergency stop by the vehicle ahead, the follower can come to a complete stop without any collision. Based on this idea, the maximum allowed speed of the follower is calculated by Equation 15 (Gipps, 1986b), where $T$ is the reaction time of the vehicles, $L_n$ is the distance from the front bumper to the front

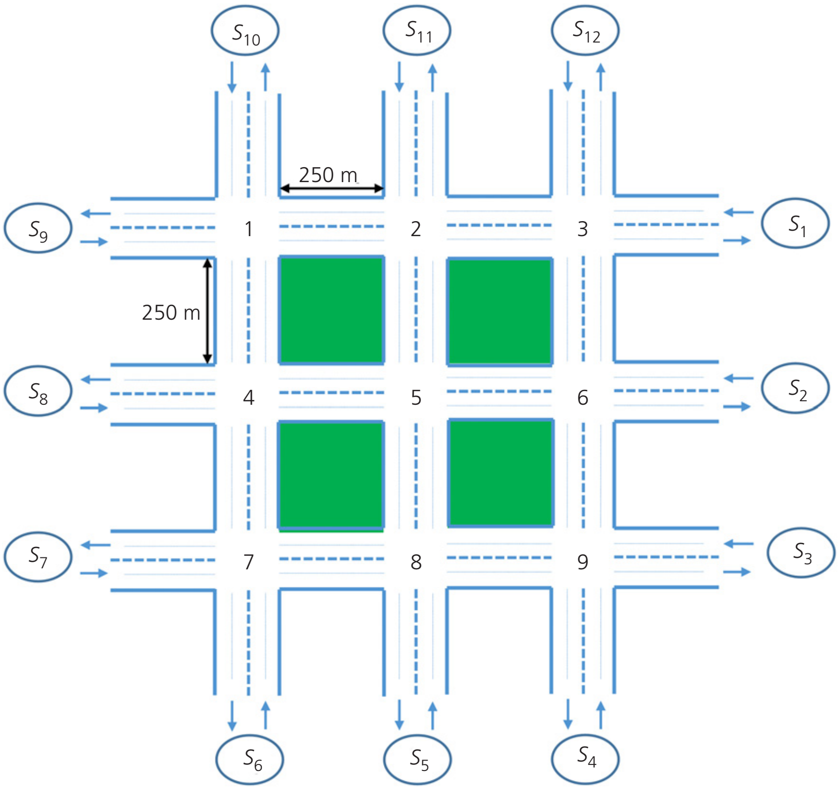


**Figure 1.** Traffic network

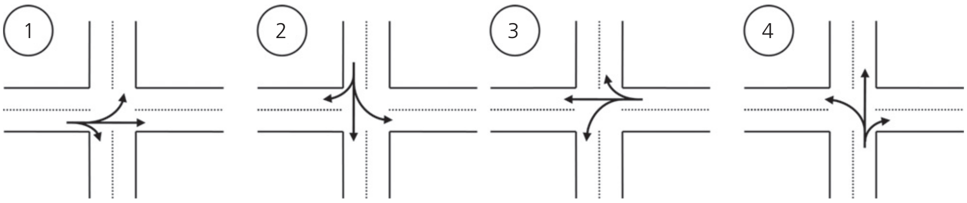


Figure 2. Order of phases

bumper at rest, $x_{n+1}(t)$ is the position of vehicle $n+1$ at time $t$, $x_n(t)$ is the position of the preceding vehicle ($n$) at time $t$, $V_{n+1}(t)$ is the speed of vehicle $n+1$ at time $t$, $V_n(t)$ is the speed of the preceding vehicle ($n$) at time $t$, and $b_n$ ($<0$) is the maximum deceleration desired by vehicle $n$. Acceleration represents the intention of a vehicle to achieve a certain desired speed.

15. $$V_{n+1}^{\text{Dec}}(t+T) = T\,b_{n+1} + \sqrt{T^2\,b_{n+1}^2(t) - b_{n+1}[2(x_n(t) - x_{n+1}(t) - L_n)] - T\,V_{n+1}(t) - \frac{V_n^2(t)}{b_n}}$$

The maximum speed to which a vehicle ($n$) can accelerate during the time period ($t$, $t+T$) is calculated by Equation 16 (Gipps, 1981), where $V_{n+1}^{*}(t)$ is the desired speed of the vehicle $n+1$ (the speed that a driver would like to drive), and $a_{n+1}$ is the maximum acceleration of vehicle $n+1$.

16. $$V_{n+1}^{\text{Acc}}(t+T) = V_{n+1}(t) + 2{\cdot}5\,a_{n+1}T\left[1 - \frac{V_{n+1}(t)}{V_{n+1}^{*}(t)}\right]\sqrt{0{\cdot}025 + \frac{V_{n+1}(t)}{V_{n+1}^{*}(t)}}$$

The final speed of the vehicle will be the minimum of speed in acceleration and deceleration mode (Equation 17).

17. $$V_{n+1}^{\text{Final}}(t+T) = \text{Min}\{V_{n+1}^{\text{Acc}}(t+T), V_{n+1}^{\text{Dec}}(t+T)\}$$

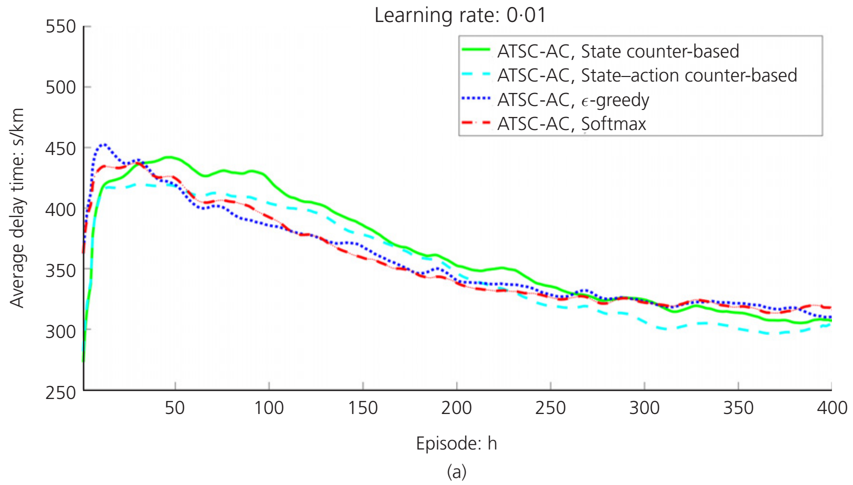
In the microscopic traffic simulation of this research, the reaction time ($T$) is set to 1 (s), the maximum acceleration ($a_n$) is set to 3 (m/s$^2$), the maximum deceleration is drawn from a Gaussian distribution with a mean of 6 (m/s$^2$) and a standard deviation of 0·5 (m/s$^2$) and the minimum headway is randomly selected from [1·5, 2·5] (s).

Traffic signals, the third component of the microscopic traffic simulation, are controlled by actor–critic (ATSC-AC). As is clear from Section 3.5, each ATSC-AC has a couple of parameters that should be tuned in order to achieve the best performance. The best value found for the learning rate of the actor and critic is 0·01. The discount factor is set to 0·99. $\eta$, the constant factor in Equation 11, is set at 0·5. It should be noted that these values were obtained by trial and error. Moreover, the value of $\Gamma$ (exploration factor) gradually decreases from 1·0 to 0·0 during the first 300 h (training period) and then it is kept constant at 0·0 over the last 100 h (hours between 300 and 400, the test period) in order to evaluate the learning performance of the system. Regarding Softmax exploration, the value of $\omega$ increases from 0·0 to 10·0 over the first 300 simulation hours and then it is kept constant at 10.0 over the last 100 h. In the $\epsilon$-greedy exploration method, the value of $\epsilon$ decreases from 0·8 to 0·0 over the first 300 h and then it is set to 0·0 during the last 100 h.

## 5. Results

Two performance indices, namely average travel time (s/km) and average delay time (s/km), are employed in order to evaluate the performance of the proposed ATSC-ACs based on the different exploration techniques. The average travel time is the average time that a vehicle needs to travel 1 km. The average delay time is the difference between the expected travel time (the time it would take to traverse the traffic network under ideal conditions) and the travel time. Figure 3 compares the performance of different exploration techniques in terms of the above-mentioned indices during 400 h of simulation.

It is evident that the direct exploration methods (counter-based methods) outperform indirect ones ($\epsilon$-greedy and Softmax) over the last 100 episodes. Also, ATSC-AC with the state–action counter-based method has the best performance. For a better representation of the results, the average performances of ATSC-AC in terms of average travel time and delay time over the last 100 episodes (test period) in which ATSC-ACs act in a greedy manner are presented in Table 1.



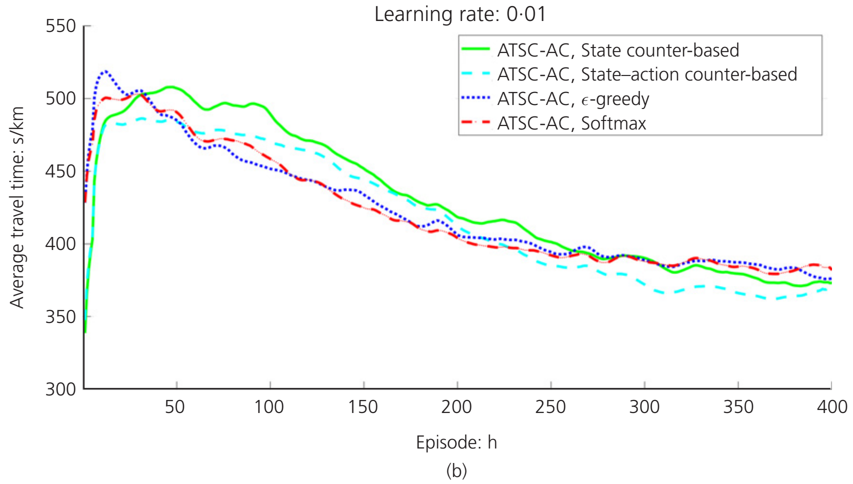


**Figure 3.** The learning performance of different ATSC-ACs (average of five times simulation): (a) average delay time; (b) average travel time

**Table 1.** Average performance and standard deviations over the last 100 episodes (results are averaged over five simulations)

| Controller | Average travel time: s/km | Average delay time: s/km |
|---|---|---|
| ATSC-AC, State counter-based | 379 ± 9 | 313 ± 9 |
| ATSC-AC, State–action counter-based | 367 ± 6 | 301 ± 6 |
| ATSC-AC, ϵ-greedy | 385 ± 7 | 319 ± 7 |
| ATSC-AC, Softmax | 385 ± 6 | 319 ± 6 |

As is clear, ATSC-AC with the state–action counter-based exploration method outperforms others because of suitable exploitation–exploration of different state–action pairs. In order to verify the performance of the proposed method, the best ATSC-AC is benchmarked against a fixed-time controller. The fixed-time controller employs a predetermined signal timing plan without considering the changes of traffic conditions (traffic fluctuations). In the fixed-time controller, the time intervals are the same every time the signal cycles, regardless of variations in traffic loads. It gives the longest green time to the heaviest traffic movement based on the historical information (Roess *et al.*, 2010). Table 2 compares the performance of the best ATSC-AC and the fixed time method in terms of average travel time and delay time. It is clear that the best ATSC-AC leads to saving average travel time and average delay time by 21% and 23%, respectively.

## 6. Conclusion

In this paper, an adaptive traffic signal controller based on actor–critic (ATSC-AC) was presented. Actor–critic has the advantages of both actor-only and critic-only. Also, it has more suitable convergence properties in comparison to actor-only and critic-only. Each ATSC-AC tries to alleviate the traffic congestion of its intersection. At the beginning of each phase, the algorithm senses the current traffic condition and selects a green time duration based on its knowledge obtained through interaction with the traffic environment.

Table 2. Comparison of ATSC-AC with fixed-time controller

| Controller | Average travel time: s/km | Average delay time: s/km |
|---|---|---|
| ATSC-AC, State–action counter-based | 367 | 301 |
| Fixed-time | 465 | 393 |
| Percentage improvement ATSC-AC compared with fixed-time | 21% | 23% |

Since ATSC-ACs do not possess enough knowledge of the traffic environment at the beginning of the simulation, they explore different green time durations regardless of their values. As time goes by and ATSC-ACs gain enough knowledge, they tend to exploit more by selecting those green times that have a fairly high value. In fact, they make a trade-off between exploration and exploitation. In order to do so, two direct exploration methods were adapted and their performances were compared with two indirect exploration techniques. In order to evaluate the proposed ATSC-AC, a 3 × 3 traffic network was employed, although the proposed method can be easily applied to larger traffic networks. The results indicate that ATSC-AC with direct exploration is the best controller and outperforms the fixed-time controller.

## REFERENCES

Aimsun (2008) http://www.aimsun.com (accessed 10/02/2018).

Araghi S, Khosravi A and Creighton D (2015) A review on computational intelligence methods for controlling traffic signal timing. *Expert Systems with Applications* **42(3)**: 1538–1550.

Aslani M, Mesgari MS and Wiering M (2017) Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies* **85**: 732–752.

Barcelo J (2010) Models, traffic models, simulation, and traffic simulation. In *Fundamentals of Traffic Simulation* (Barcelo J (ed.)). Springer, New York, NY, USA, pp. 1–62.

Bazzan ALC and Klügl F (2013a) *Introduction to Intelligent Systems in Traffic and Transportation*. Morgan and Claypool, Williston, VT, USA.

Bazzan ALC and Klügl F (2013b) A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review* **29(3)**: 375–403.

Berenji HR and Vengerov D (2003) A convergent actor-critic-based FRL algorithm with application to power management of wireless transmitters. *IEEE Transactions on Fuzzy Systems* **11(4)**: 478–485.

Bhatta B (2010) *Analysis of Urban Growth and Sprawl from Remote Sensing Data*. Springer-Verlag, Berlin/Heidelberg, Germany.

Busoniu L, Babuska R and Schutter BD (2008) A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **38(2)**: 156–172.

Chowdhury MA and Sadek AW (2003) *Fundamentals of Intelligent Transportation Systems Planning*. Artech House, Norwood, MA, USA.

Choy MC, Srinivasan D and Cheu RL (2003) Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* **33(5)**: 597–607.

Gartner NH (1983) OPAC: a demand-responsive strategy for traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board* **906**: 75–81.

Gipps PG (1981) A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological* **15(2)**: 105–111.

Gipps PG (1986a) A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological* **20(5)**: 403–414.

Gipps PG (1986b) Multsim: a model for simulating vehicular traffic on multi-lane arterial roads. *Mathematics and Computers in Simulation* **28(4)**: 291–295.

Grondman I, Busoniu L, Lopes GAD and Babuska R (2012) A survey of actor-critic reinforcement learning: standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42(6)**: 1291–1307.

Head KL, Mirchandani PB and Sheppard D (1992) Hierarchical framework for real-time traffic control. *Transportation Research Record* **1360**: 82–88.

Henry JJ, Farges JL and Tufal J (1983) The PRODYN real-time traffic algorithm. *IFAC Proceedings Volumes* **16(4)**: 305–310.

Hunt PB, Robertson DI, Bretherton RD and Winton RI (1981) *SCOOT – A Traffic Responsive Method of Coordinating Signals*. Laboratory TaRR, Crowthorne, UK.

Jin J and Ma X (2015) Adaptive group-based signal control by reinforcement learning. *Transportation Research Procedia* **10**: 207–216.

Kaelbling LP, Littman ML and Moore AW (1996) Reinforcement learning: a survey. *Journal of Artificial Intelligence Research* **4**: 237–285.

Konda VR and Tsitsiklis JN (2003) On actor-critic algorithms. *SIAM Journal on Control and Optimization* **42(4)**: 1143–1166.

Medina JC, Hajbabaie A and Benekohal RF (2010) Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure. In *Proceedings of 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), Funchal, Portugal*. IEEE, New York, NY, USA, pp. 525–530.

Miyazaki K, Yamamura M and Kobayashi S (1997) K-certainty exploration method: an action selector to identify the environment in reinforcement learning. *Artificial Intelligence* **91(1)**: 155–171.

NCHRP (2010) *NCHRP Synthesis 403: Adaptive Traffic Control Systems: Domestic and Foreign – State of Practice*. Transportation Research Board, Washington, DC, USA.

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

Roess RP, Prassas ES and Mcshane WR (2010) *Traffic Engineering*. Pearson Higher Education, Upper Saddle River, NJ, USA.

Sims AG and Dobinson KW (1980) The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Transactions on Vehicular Technology* **29(2)**: 130–137.

Sledge IJ and Príncipe JC (2017) Balancing exploration and exploitation in reinforcement learning using a value of information criterion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA*. IEEE London, pp. 2816–2820.

Sutton RS (1988) Learning to predict by the methods of temporal differences. *Machine Learning* **3(1)**: 9–44.

Sutton RS and Barto AG (1998) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.

Thrun SB (1992) *Efficient Exploration in Reinforcement Learning*. School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, USA, technical report CMU-CS-92-102.

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

Weiss G (1999) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, USA.

Wiering M (2000) Multi-agent reinforcement learning for traffic light control. In *Proceedings of 17th International Conference on Machine Learning, Stanford, CA, USA*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1151–1158.

Wyatt J (1997) *Exploration and Inference in Learning from Reinforcement*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, UK.