Expert Systems with Applications xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Review

A review on machine learning methods for controlling traffic signal timing

Sahar Araghi *, Abbas Khosravi, Douglas Creighton

Centre for Intelligent Systems Research (CISR), Deakin University, Victoria 3216, Australia

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Traffic signal timing
Machine learning
Q-learning
Neural network
Fuzzy logic system
Isolated intersection

ABSTRACT

Urban traffic as one of the most important challenges in modern city life needs practically effective and efficient solutions. Artificial intelligence methods have gained popularity for optimal traffic light control. In this paper, a review of most important works in the field of controlling traffic signal timing, in particular studies focusing on *Q*-learning, neural network, and fuzzy logic system are presented. As per existing literature, the intelligent methods show a higher performance compared to traditional controlling methods. However, a study that compares the performance of different learning methods is not published yet. In this paper, the aforementioned computational intelligence methods and a fixed-time method are implemented to set signals times and minimize total delays for an isolated intersection. These methods are developed and compared on a same platform. The intersection is treated as an intelligent agent that learns to propose an appropriate green time for each phase. The appropriate green time for all the intelligent controllers are estimated based on the received traffic information. A comprehensive comparison is made between the performance of *Q*-learning, neural network, and fuzzy logic system controller for two different scenarios. The three intelligent learning controllers present close performances with multiple replication orders in two scenarios. On average *Q*-learning has 66%, neural network 71%, and fuzzy logic has 74% higher performance compared to the fixed-time controller.

1. Introduction

The increasing volume of traffic in cities has a significant effect on the road traffic congestion and therefore the time it takes for road users to reach their destination. Widening roads and increasing their capacity is not sufficient by itself, as the intersection then become a bottleneck. Bottlenecks cannot be prevented, however, the way intersections are controlled has room for improvement.

Since the 1960's different methods have been presented to manage intersections and for controlling traffic signals' timing. As one of the first traffic signal controllers, fixed-time or pre-timed controllers applied historical data to determine appropriate time for traffic signals (Cai, 2010). Fixed-time method is not based on current traffic demands and cannot handle unexpected conditions in traffic. Based on the historical traffic volume, the cycle time is divided to several phases. A fixed amount of time is required for clearing the intersection and starting the next phase after each phase, called the safety time. The safety time increases per hour for the case of shorter cycle time. Therefore, there is a lower overall capacity for intersections with shorter cycle times. On the other hand, longer waiting times and longer queues are the consequences of longer cycle times. Webster proposed a formula based on the flow rate of each lane in a link to solve this issue. This formula is useful to find an optimal cycle and appropriate duration for green time in each phase (Webster, 1958). As fixed-time methods cannot predict traffic demand accurately, they are not appropriate for situations such as accidents, and other disturbances that may disrupt traffic conditions.

The next step for improving the control method was actuated or real time controllers. This type of controller emerged in the 1970's. Traffic-actuated control methods utilize inductive detectors to observe the actual traffic situation. The traffic-actuated controller must have the ability to determine whether the last vehicle of the queue formed at the stop line during the red phase has passed. This detection is useful for having efficient extension or termination of green time, and it is performed by measuring the gap between vehicles. The green time is terminated when the gap between vehicles is larger than the threshold maximum gap. The optimal placement of detectors at an intersection impacts the performance of actuated method. In addition, by increasing the number of detectors the accuracy of the system is improved. In actuated methods, a pre-specified block period time is considered

* Corresponding author.
E-mail addresses: saraghi@deakin.edu.au (S. Araghi), abbas.khosravi@deakin.edu.au (A. Khosravi), douglas.creighton@deakin.edu.au (D. Creighton).

http://dx.doi.org/10.1016/j.eswa.2014.09.003
0957-4174/© 2014 Published by Elsevier Ltd.

for extending the green time of a phase. Therefore, detection of sparse traffic can have a considerable influence on delay time (Koonce et al., 2008). MOVA (Vincent & Peirce, 1988), LHVORA (Kronborg & Davidsson, 1993), and SOS (Kronborg & Davidsson, 1996) are samples of actuated traffic control system.

Parameters like time, day, season, weather, and some unpredictable situations such as accidents and special events are highly influential on traffic load. Traffic-adaptive control systems was created to take these elements in account in order to more efficiently predict green times. Fixed-time and actuated method do not use a control policy or a parameterized value function. Furthermore, these systems do not utilize accumulative information for improving their performances. In adaptive traffic control systems, the traffic condition is sensed and monitored continuously and the timing of traffic signals is adjusted accordingly. It is useful to note that adaptive controllers and real-time ones are two different concepts, however, it is possible to have a system with both abilities. The controllers with real-time ability in response to sensory inputs are real-time systems, in which the parameters of the controller and internal logic remain unchanged. Alternatively, one of the special features of adaptive systems is their characteristic in adjusting their parameters and internal logic in response to the significant change of the environment (Abdulhai, Pringle, & Karakoulas, 2003).

Both SCATS (Sims & Dobinson, 1980) and SCOOTS (Hunt, Robertson, & Bretherton, 1982) are famous adaptive systems that gather data of the traffic flow in real-time at each intersection to control timing of traffic lights. To obtain traffic information SCATS counts vehicles at each stop line, and SCOOTS applies a set of advanced detectors placed upstream of the stop line. Using these detectors, SCOOTS gives a higher resolution of the traffic condition such as traffic flow and number of cars in the queue before they reach the stop line. SCATS and SCOOTS both use centralized control. OPAC (Gartner, Tarnoff, & Andrews, 1991) and RHODES (Mirchandani & Head, 2001) are two other adaptive traffic controlling systems that use distributed control. These systems are run locally and coordination between intersections are done by communication between neighbors. As an example, when an intersection releases number of vehicles informs the next intersection about the time and number of vehicles to expect.

The use of artificial intelligence methods to control traffic signals started in 1990's (Malej & Brodnik, 2007). Multiple optimization and estimation methods have been applied for adaptive control. Machine learning techniques are beneficial to create adaptive controllers with the ability to address unpredictable traffic condition issues. More explanation about applying machine learning techniques in controlling traffic signals will be presented in Section 2.

In this paper a review of intelligent methods have been applied for controlling traffic signals are presented. To have a comparison between three key intelligent methods have used in this area, *Q*-learning, neural network (NN), and fuzzy logic systems (FLS) controllers are implemented for a single intersection. Implementing these methods for an isolated intersection gives the opportunity to focus on the behavior of each method accurately. The reminder of this paper is organized as follows. Section 2 reviews the background machine learning theory and related work in applying machine learning techniques for traffic signal timing. In Section 3, *Q*-learning, NN, and FLS are used for timing an isolated traffic signal. Related results of experiments are presented in Section 4. Finally, Section 5 concludes the paper.

2. Background and related work

In this section, the related work and the background useful to understand the material presented in the rest of this paper will be covered.

2.1. Reinforcement learning and Q-learning

As a simple term, an agent in reinforcement learning tries to reach a goal through dynamically interacting with its environment. The agent examines different actions in different situations according to the goal and determines the best action or best sequence of actions. For each action the environment provides a feedback useful for the agent to recognize to what extent the action is beneficial to reach the goal (Abdulhai et al., 2003).

Generally, Markov decision process (MDP) (Puterman, 1994) is regarded as the mathematical foundation for reinforcement learning. A fully observable MDP is a quadruple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, T \rangle$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is the set of actions, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function that describes the probability $p(s'|s, a)$ of ending up in state s' when performing action a in state s, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is reward function that returns a numeric value after taking action a in state s. An agent's policy is a mapping $\pi : \mathcal{S} \rightarrow \mathcal{A}$. γ $(0 \leqslant \gamma \leqslant 1)$ is the discount factor. The agent aim to find an optimal policy π that maximizes the expected sum of discounted rewards. Eq. (1) formulates this definition.

$$V(s, \pi) = \sum_{t=0}^{\infty} \gamma^t E(r_t | \pi, s_t). \tag{1}$$

2.2. The Q-learning algorithm

Q-learning is an incremental reinforcement learning method. It does not need a model of the environment and learning can be performed online (Sutton & Barto, 1998). Fig. 1 explains the *Q*-learning method in more details.

An agent is an entity that interacts with the environment. It chooses an action based on inputs receives from its sensors and learns on the basis of the effects of its action on the environment. The *Q*-learning agent at time t receives a signal from the environment. This signal describes the current state s. Actually the state is composed of a group of characteristics presenting the current situation of the environment relevant to the problem. State information must have Markov property. This information and the action being taken are needed to predict the environment's effect. Although it may not be completely true, it is assumed that the process is Markovian. By Markov property the agent does not need to know the history of previous actions or states in its decision making.

In *Q*-learning the agent selects action a based on the relative value of all possible action in state s. This value is the *Q*-value of undertaking action a in state s and lead to transition to state s'. This value presented by $Q(s, a)$. The *Q*-value is obtained gradually during the learning and by exploring randomly various possible actions in each state. By performing action a in state s, the agent receives reward $r(s, a)$. The obtained reward highly depends on the effect of the action on the environment. During the learning, the agent's objective is to find the optimal policy that maximize the accumulative reward. Related to the problem, punishment can be replace by reward in which the agent aim to minimizes the accumulative

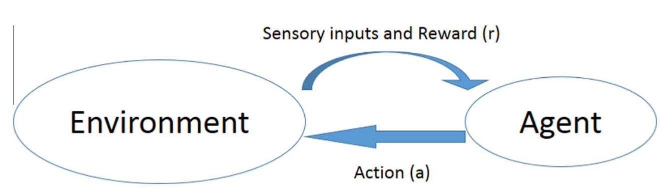

Fig. 1. Interaction of agent and environment in reinforcement learning. Sensory inputs that describing current state of the environment received by agent, the agent choose an appropriate action and receive reward from environment.

punishment over time. One of the other factors that is generally considered in Q-learning is the discount factor γ $(0 \leqslant \gamma \leqslant 1)$. It is applied for bounding the reward, specially in problem domain with continuous episodes, it considers higher value for short term future rewards compared to long term rewards. For updating the Q-value in learning process Eq. (2) is used.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) - Q(s_t, a_t)\right) \tag{2}$$

where α $(0 \leqslant \alpha \leqslant 1)$ is the learning rate and γ $(0 \leqslant \gamma \leqslant 1)$ is the discount factor. The learning rate needs to be decreased in order to guarantee the convergence of Q-function in stochastic environments. It is often required that each state-action pair is visited infinite times. For future use of the Q-values they can be stored in a Q-table which need a high amount of memory. Also, it is possible that the Q-values are used as the inputs of a function approximator designed to generalize the Q-function. In this case, the approximator can estimates the Q-value for not visited state-action pairs by the help of the similar situations. Fig. 2 describes the Q-learning algorithm step by step.

2.3. Neural network

The basic concept of NN were originally obtained from the way biological nervous systems work. A NN is an information processing technique well-known because of its excellent approximation and learning capabilities. A NN is a universal approximator that for any nonlinear mapping can approximate various degree of accuracy (Bishop, 1995), using this feature NNs are able to recognize hidden patterns from imprecise and complicated data. In a better word for a problem that is too complicated to be considered by either traditional data mining methods or humans, NN can be a suitable option. In different fields of engineering and science NNs have been broadly used for control, modeling, prediction and classification problems. In the case of supervised learning, NNs are usually trained by the minimizing an error-based cost function. The parameters of the NN can be optimally adjusted for situations with unknown expected values, by the using and minimizing the cost function. To obtain the optimal set of parameters, the global optimization methods such as simulated annealing (SA) (Dekkers & Aarts, 1991) or genetic algorithm (Mitchell, 1997), are suitable.

2.4. Fuzzy logic systems

In early 1975, fuzzy set theory was proposed by Zadeh (1975a, 1975b, 1975c). Fuzzy logic system (FLS) is a suitable method to represent the vagueness and uncertainties of the linguistic phrases. Actually, it is possible to handle inexact data and uncertain information by fuzzy sets. Using fuzzy theory instead of crisp set theory provides the ability to implement the real-world scenarios in more details. One of the most important feature of FLS is the ability to include an expert's knowledge in their design. Additionally, they are transparent, which makes them more understandable by operators compared to black box NN models. A FLS maps the inputs to the output of the system. In situation of no fuzziness in the definition of a cluster or class of objects, there is just a simple two-valued characteristic function, zero and one, but by fuzzy set this domain is extended to the range of whole numbers between zero and one. To represent some linguistic values the system domain is divided in fuzzy sets, for example we can define low, medium, and high traffic flow. Then, membership functions are used to show the degree of dependency to each fuzzy set. Each input value may belong to more than one fuzzy set. Same situation can be considered about the output space. Associating numerical values to fuzzy sets is fuzzification and defuzzification is the name of the opposite process. The logic of the system is define by if-then rules in fuzzy inference. Fig. 3 shows FLS's structure.

By the structure of the FLS, it is easily possible to recognize the reason of the operating of the system in the way it is. Therefore, changing the system to make it more appropriate and focusing on week points is more feasible. A FLS is flexible and its modification is not difficult. Membership function parameter and their quantify, fuzzy rules, system operators, defuzzification methods and also the other characteristics of a FLS all can be changed with the purpose of achieving better results.

2.5. Machine learning for traffic light control

To improve the traffic control system performance, artificial intelligence and specially machine learning methods have emerged as an effective tools. Machine learning methods by the ability of learning from experience provides special opportunity to manage traffic congestion. Q-learning, NN, and FLS are three of machine learning methods which a review of previous works in these three categories for controlling traffic will be presented in this section. Researchers in some studies applied hybrid machine learning methods. It is attempted to review the previous works in three separate categories, however there may be some overlaps between them.

2.5.1. Q-learning controller for traffic signal timing

Most existing traffic control systems need predefined model of traffic flow to have a short-time prediction of future traffic condition. In Q-learning no prespecified model of the environment is required and relationship between actions, states, and environment are learned by interaction with the environment.

In this regard, for the first time Thorpe and Anderson (1996) studied using reinforcement learning for traffic signal control (Wiering, Vreeken, van Veenen, & Koopman, 2004). Thorpe applied SARSA (Sutton, 1996) to a traffic control problem. He evaluated the performance of SARSA with three different representations of an specific state, and used a NN to estimate the reinforcement value.

```
Initialize Q(s, a) arbitrarily
for all episode do
    Initialize s
    for all step of episode do
        Choose a from s using policy derived from Q (e.g., ϵ-greedy)
        Take action a, observe r, s′
        Q(s, a) ← Q(s, a) + α [r + γ max_a′ Q(s′, a′) − Q(s, a)]
        s ← s′;
    end for
end for
```

Fig. 2. The Q-learning algorithm.

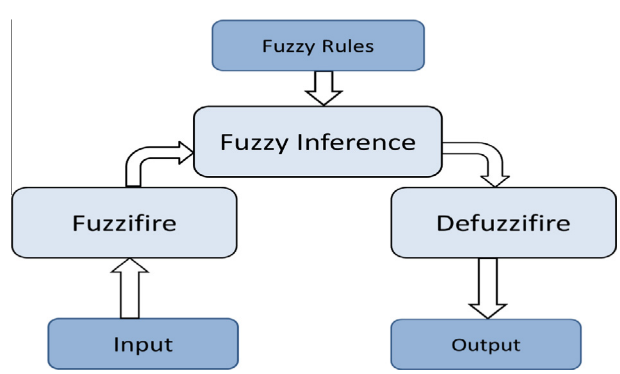

Fig. 3. Structure of a FLS.

In his study, states were defined by the number and position of vehicles in all directions ending to an intersection. Action for each state was set to change the lights' color from red to green and vice versa. These features were combined in three different ways. The first representation called *vehicle count*. In this approach Thorpe considered ten partitions based on the number of cars, and then by considering all pairs of combination of these ten partitions for east–west and north–south directions and considering two possible modes for traffic lights, 200 ($10 \times 10 \times 2$) states were inputs to the learning agent. For the second representation or *fixed distance*, Thorpe divided each lane to 110-foot intervals that led to four partitions at each lane. An *occupied* bit is set to show the absence or presence of vehicles in each partition, which causes having eight components for whole east–west or north–south lane and one component for the traffic light's color. Totally, there were nine components vector as the input to the NN. Third representation was *variable distance*, similar to the previous one but with a variable distance for each partition. The distances were set to 50, 110, 220, and 400 feet, and there were again four partitions for each lane and for traffic light. The learning agent in this representation had nine components input similar to *fixed distance*. Thorpe set the reward to $r = -1$ for each step of trial to reach the goal. Evaluation is done in a 4×4 network, and during the performance evaluation the best result for total simulation steps required to clear vehicles from the simulation environment belonged to *variable partitions*, and for the case of minimum travel time *fixed distance* had the best result.

Wiering (2000), Wiering et al. (2004) proposed a transition model that estimates waiting time for both green and red lights at each intersection. They applied multi-agent reinforcement learning to control traffic signals. Their method was car-centric; each car estimates its own waiting time and communicates it with the nearest traffic light. For state definition they considered position and orientation of vehicles in the queue, and their destination address. The action was set to change between red and green phase, and for the reward function, if a car stayed at the same place $r = 1$ otherwise $r = 0$. In this system the goal was to minimize the overall waiting time, and it learned the assignment function for estimating overall waiting time of vehicles. During their experiments both local and global communication scenarios for better decision making of traffic lights are applied.

Abdulhai et al. (2003), applied *Q*-learning as a traffic controller. They performed the experiment for an isolated intersection, but had some suggestions for the case of multi-agent in this study. In the case of single intersection, states are the length of queues on four approaching links to the intersection and the elapsed phase time. Action was defined as extending the current red or green phase or changing to the next one. In this study, reward was considered as a penalty and it was the total delay time between two successive decision by the vehicles in the queues formed behind stop light of four approaching link of the intersection. In addition, a power function is used to approximate balancing of the queue length in order to modifying the reward, which is directly proportional to the queue length in each 1 s step. This was useful to prevent agent of being indifferent about too long, too short queue or situation of equal-length of queues. For the case of multiple intersection some other states such as the split between two intersection may be added, and the reward would be the weighted summation of all single intersection by considering highly weighted reward for the main road. Abdulhai et al. have shown that reinforcement learning and especially *Q*-learning is a promising approach to build an adaptive traffic signal controller in Abdulhai et al. (2003). The result of experiments for single intersection showed that *Q*-learning outperformed the pre-timed controller for variable traffic flows, and either slightly outperformed or was equal to the pre-timed controller for situations of the constant or uniform flows.

In Wunderlich, Liu, Elhanany, and Urbanik (2008), Longest Queue First (LQF) was proposed as a traffic signal scheduling algorithm for an isolated intersection. The LQF algorithm was designed for a signal control problem and the concepts were employed from the field of packet switching in computer networks. This method utilized a maximal weight matching algorithm to minimize the queue sizes at each approaching link and led to a significantly lower average vehicle delay through the intersection. It was proved that LQF was stable and had strong performance under various traffic scenarios. The authors decided to apply LQF in multi-intersection network in their next study (Arel, Liu, Urbanik, & Kohls, 2010). In a multi-intersection network a phase scheduling decision at one intersection would largely affect the traffic conditions in its neighbor intersections and applying LQF became a more complex task. In this research reinforcement learning is used in order to have the capability to have distributed control as needed for scheduling multiple intersections. In fact, they introduced a novel use of a multi-agent system and reinforcement learning framework to obtain an efficient traffic signal control policy. The focus was at minimizing the average delay, avoiding congestion and intersection cross-blocking. The network contained five intersections and each intersection is governed by an autonomous intelligent agent. A central agent and an outbound agent were two kinds of agents employed in this work. The outbound agents schedule traffic signals by following the LQF algorithm. These agents provide traffic statistics for the central agent. The central agent learned a value function driven by its local and neighbors traffic conditions. The proposed methodology in their work utilized the *Q*-learning algorithm with a feed-forward NN in order to *Q*-value function approximation. In the setting of *Q*-learning method the state is represented as an eight-dimensional feature vector in which each element represented the relative traffic flow at one of the lanes. For the outbound intersection agent, only local traffic statistics is considered, but the central intersection agent had access to all states of its neighboring intersections which causes increasing the number of state space. Action set was defined as he eight different combination of available phase (Wunderlich et al., 2008). The reward was considered in the range from −1 to 1, where positive reward values were obtained if the current delay is lower than the previous time step. The agent gave a penalty (negative value) if an increased average delay is observed. Experimental results revealed the higher performance of multi-agent control based on reinforcement learning against LQF governed isolated single-intersection control.

Prashanth and Bhatnagar, in Prashanth and Bhatnagar (2011) proposed the feature based reinforcement learning for controlling traffic signals. They also claimed that using feature based state-action algorithms made their method appropriate for using in high-dimensional setting of a multi-intersection network. Authors mentioned their work is against the prior work like Abdulhai et al. (2003), that required full state representation and it was not practically possible to implement them. Their method did not require the precise information on elapsed time and queue length. To perform that they divide the queue length in three sets: low, medium, high and put a threshold for elapsed time to check if the detected elapsed time is higher than threshold or less than it takes place in two different groups. They compared the performance of the proposed method against fixed-time, longest queue and also the algorithms proposed in Abdulhai et al. (2003) and Cools, Gershenson, and D'Hooghe (2008), which the proposed feature based algorithms outperformed all the others.

Abdoos et al. in Abdoos, Mozayani, and Bazzan (2011) presented an approach for controlling traffic signals in a network of 50 intersections based on *Q*-learning. Each intersection was considered as an agent and the whole network formed a multi-agent system. In their research, the average length of queue in approaching links was the states of Q-learning and the number of permutations of the approaching links determined states' number. During the experiments, they considered intersections with four approaching links, therefore, state space consists of 24 states and different phase splits of the cycle time were the proposed actions in Q-learning. Phase split refers to the division of the cycle time into a sequence of green signals for each group of approaching links. In addition, cycle time set as a fixed value and a minimum green time is adjusted for each phase. Reward was considered as inversely proportional to the average lengths of the queues in the approaching links, which is normalized to remain between 0 and 1. In their next work (Abdoos, Mozayani, & Bazzan, 2013), they developed a holonic multi-agent system to model a large traffic network. In this study, each intersection had a similar structure to Abdoos et al. (2011) and they presented as homogeneous agents. The result of their research revealed that the performance of the individual Q-learning and holonic Q-learning is almost the same. The average standard deviation of delay time for holonic Q-learning was less than the individual Q-learning, which shows that they are clustered more closely in holonic Q-learning and are more reliable.

Some other works that Abdulhai had contribution and they were about controlling traffic are Abdi, Moshiri, Abdulhai, and Sedigh (2012, 2013), El-Tantawy, Abdulhai, and Abdelgawad (2013). In El-Tantawy et al. (2013), an adaptive traffic signal controller designed which was using a multiagent reinforcement learning approach. Each controller (agent) was responsible to control traffic lights timing around a single traffic junction. El-Tantawy et al. (2013) proposed two possible modes: (1) independent mode, where each intersection controller works independently of other agents; and (2) integrated mode, where each controller coordinates signal control actions with neighboring intersections. They tested the model on a network of 59 intersections in the lower downtown core of the City of Toronto, Canada, for the morning rush hour. Their results showed reduction in the average intersection delay ranging from 27% in mode 1 to 39% in mode 2.

Many others are also applied reinforcement learning and specially Q-learning in developing the traffic controller. For example, in Cools et al. (2008), a self-organizing traffic light control method is presented. The phase of a lane is changed to green if the elapsed time during the red phase hits a certain threshold. This is also useful to recognize that the number of cars on the lane are above the threshold and the queue length is indirectly used for signal configuration. In da Silva, Basso, Perotto, and Engel (2006), a context detection reinforcement learning method was proposed which was able to create a partial model of the environment based on demands. The partial model improved or a new one was constructed through the time to satisfy the demand. Adaptive reinforcement learning controller are proposed in Wen, Qu, and Zhang (2007), Dai, Zhao, and Yi (2010) to signalizing a model free traffic environment. Also, Houli, Zhiheng, and Yi (2010) used reinforcement learning to propose a multi-objective control algorithm. They predict the overall value of given vehicle's states by using reinforcement learning.

The challenge for all Q-learning controller (QLC) is to manage the huge amount of state-action space. One of the solutions to reduce the number of states is categorizing possible states in groups. Although this approach increases the learning rate, limiting the number of states to the number of groups decrease the accuracy of the system. Based on the review, proposed QLCs usually consider the extension of green time as an action in Q-learning. Generally the extension time is a fixed period of time which may repeated until reaching the maximum threshold. This fixed period of time is an assumption causes low efficiency. Considering some predefined numbers as possible green time similar to Abdoos et al. (2011, 2013) studies has the same deficiency. Preparing enough data to train the system is the other issues for QLC. Q-learning without enough training samples cannot converge to the optimal results. However, Q-learning is a beneficial methods to have online learning and it can improve its performance and adopt to the new situations.

2.5.2. Neural network controller for traffic signal timing

Adaptive controllers based on NN are recommended in many other works. For example, Spall and Chin (1997) employed simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992) based gradient estimates with a neural network controller (NNC) for optimizing the system. SPSA was used for modeling the weight update process of a NN. A function was developed to take the current traffic information and generate the signal timings. In their work the current traffic information was used to solve the current instantaneous traffic issue. The system that presented by them named S-TRAC with these advantages: (1) It did not require any system-wide traffic flow model; (2) S-TRAC automatically adapted to long-term changes in the system such as seasonal variations while providing real-time responsive signal commands; and (3) This system was able to work with existing hardware and sensor configurations within the network of interest while additional sensors may help the overall control capability. For S-TRAC they used a feed-forward NN with 42 inputs and two hidden layers. Inputs included: (1) the queue at each cycle termination for 21 traffic queues of the simulation; (2) eleven nodes for per-cycle vehicle arrivals in the system; (3) simulation start time; and (4) the nine outputs from the previous control solution. The output layer of the NN contained nine nodes for each signals split, and for two hidden layers there were 12 and 10 nodes respectively. To evaluate the performance of S-TRAC, a simulation of a nine-intersection network of the central business district of Manhattan, New York was used. They have 10% and 11% improvement for both case of constant arrival rates of and increase in mean arrival respectively against fixed-time method during 90 days.

In study conducted by Chin, Spall, and Smith (1999) they apply S-TRAC in a moderately congested network, in Maryland. The interruptions of the traffic flow caused by the traffic signal was evaluated. The result of their evaluation showed an average increase of 7% with 90% confidence bound equal to $\pm 2.5\%$.

Yin, Wong, Xu, and Wong (2002) developed a fuzzy neural model to predict the traffic flows in an urban street network. Their developed model consists of two modules: a gate network (GN) and an expert network (EN). The first one classified data into a number of clusters through fuzzy approach. The EN module specifies the input–output relationship as in a conventional NN approach. In fact, GN groups traffic patterns of similar characteristic into clusters and EN models the specific relationship within each cluster. The model used an online rolling training procedure. Their fuzzy neural model had 23% and 30% improvement respectively for offline and online schema against a designed NN model.

Among various method to control traffic lights, Choy et al.'s study (Choy, Srinivasan, & Cheu, 2003) is one of the well-known research in this area. In this work, a new hybrid, synergistic approach was proposed that applied computational intelligence concepts to implement a cooperative, hierarchical, multiagent system for real-time traffic signal control of a large-scale traffic network. The problem of controlling the network was divided to various subproblems and each handled with an agent by fuzzy neural decision making capability. At the first, the decision were made by lower-level agents and then they were mediated by higher-level agents. In Choy et al. (2003), a multistage online learning process for each agent was implemented that involved reinforcement learning, weight and learning rate adjustment, in addition of dynamic update of fuzzy relations by evolutionary algorithm. The test bed used for evaluation of the proposed method was a section of the Central Business District of Singapore.

The result of the experiments illustrated that the performance of the proposed multiagent architecture against the one used for real-time adaptive traffic control system of the moment had significant improvements. It caused reducing total vehicle stoppage time by 50% and the total mean delay by 40%.

In another work Srinivasan, Choy, and Cheu (2006), the authors presented an enhanced version of the SPSA-NN system for a multiagent system and they tested that in more complicated scenarios. Authors claimed that although the SPSA algorithms is a useful method for updating the weight online, the model proposed in Spall and Chin (1997) had some limitations influence its performance. Spall et al. used a three-layer NN and relevant traffic variables were used as inputs. Based on Srinivasan et al. (2006), there were two shortcomings for that system: first, the system used heuristic method to identify the general traffic patterns (morning and evening peaks) and assignment of time periods for patterns. This cause the robustness of the system comes into question for not periodic traffic patterns. Second, a NN was considered for each time period, and the weight were updated only whenever the same traffic pattern and time period was arisen. It may not be possible to respond appropriately to changes of the traffic inside the same time period. Srinivasan et al. improved that method and compared it with the hybrid multiagent architecture presented by Choy et al. (2003) and Green Link Determining (GLIDE), which was the excising traffic signal control for the city and is the local version of the SCATS. To evaluate the performance they considered a large traffic network in Singapore Central Business District with 25 intersections. After 15 separate simulations with different seeds, which each was set for three hours, the lowest mean delay belonged to SPSA-NN, Hybrid NN and GLIDE respectively.

In the research done by Teodorovic, Varadarajan, Popovic, Chinnaswamy, and Ramaraj (2006), an intelligent isolated intersection control system was developed. Their model was based on the combination of the NNs and dynamic programming. The proposed system makes real time decisions to extend (and how much) current green time. They conclude from their experimental tests that the outcome (the extension of the green time) of the proposed neural network is nearly equal to the best solution.

Chao, Lee, and Wang (2008) presented an intelligent traffic light control method based on extension neural network (ENN) theory for crossroads. First, the number of passing vehicles and passing time of one vehicle within green light time period were measured in the main-line and sub-line of a selected crossroad. during the next step, the measured data are adopted to construct an estimation method based on ENN for recognizing the traffic flow of a standard crossroad. They claimed their proposed method can discriminate the traffic flow of a standard crossroad rapidly and accurately.

The work done by Nagare and Bhatia (2012) was another attempt to forecast traffic flow for controlling traffic congestion. It was mentioned that NN introduces some flaws such as flow convergence and the obtained solution is usually local optimal. The idea was that by using combined optimization methods better optimization results can be obtained. They applied three different combined optimization methods in their work to have a comparison between them.

NLCs have ability to consider different ranges of green time and their training times are usually lower than QLC, but they are not easily adjustable for new situation. Considering appropriate number of layers and neurons in each layer are the other issues regarding to NLC.

Using NN directly as a controller or in combination to other methods for example as an optimizer have been presented in many works. In some of them both FLS and NN have been used for designing the traffic signal controller, which will be discussed later.

2.5.3. FLS for traffic signal timing

The first attempt in applying FLS for controlling traffic signals at a single intersection was done in 1977 by Pappis and Mamdani (1977). Their controller has three inputs and one output. It was designed for a two-phase intersection with random vehicle arrivals and no turning movements. Seven seconds after starting the green time, every 10 s the controller decided about extension of the phase or changing that. Actually, the fuzzy rules were developed to evaluate and make decision about the suitable extension of current green phase based on different time duration and by measuring "degree of confidence". The extensions were compared with the highest degree of confidence and if none of them had more than 50% of confidence, then the green signal will be terminated immediately. Otherwise, the green time was selected and the process was repeated until maximum acceptable green time is reached. In this work it was assumed that the vehicle detectors were placed upstream from the intersection to inform the controller about the future vehicles that will arrive, which is useful to predict the future queue length of vehicles at the intersection. To evaluate the system it was compared with the efficient vehicle-actuated method and the result of simulation showed the better performance of fuzzy logic controller (FLC).

Nakatsuyama, Nagahashi, and Nishizuka (1985) applied fuzzy logic to control two adjacent intersections with one-way movements. This controller determined the extension or termination of the green signal based on the upstream traffic for the downstream intersection. A FLC for freeway ramp metering is developed by Chiu and Chand (1993). The FLC presented by Lee, Lee, and Leekwang (1995), was for a set of intersections each of which manages the phase length and sequence dynamically according to its own and neighboring traffic situations.

Favilla, Machion, and Gomide (1993) also applied FLS to control an isolated intersection with two-way streets. They considered the number of vehicles that had already passed the intersection and the length of the vehicle queue in the red approach as inputs for the fuzzy rules, and the amount of the extension was the output of the FLS. Some additional strategies were considered for adapting the numerical bounds on the input and output as well. Some other studies that proposed FLC for a single intersection are Wei, Zhang, Mbede, Zhang, and Song (2001), Murat and Gedizlioglu (2005), Hu, Thomas, and Stonier (2007), Zeng, Li, and Lin (2007).

Niittymaki have many works in the field of traffic management (Capek, Pitkanen, & Niittymki, 2011; Granberg & Niittymki, 2001; Granberg, Niittymki, Karppinen, & Kukkonen, 2000; Heltimo, Niittymaki, & Bjork, 2002; Kononen & Niittymaki, 2000; Niittymaki, 2001a, 2001b, 2001c; Niittymaki & Pursula, 1996, 1997; Niittymaki & Sane, 1996 Niittymaki & Pursula, 2000; Niittymaki & Pursula, 2001; Niittymaki & Kikuchi, 1998; Niittymaki & Kononen, 2000; Niittymaki & Nevala, 2001; Niittymaki & Turunen, 2003; Niittymaki, Nevala, Halme, & Aitken, 2008; Niittymaki, Kosonen, & Nevala, 2001; Lehmuskoski, Niittymaki, & Silfverberg, 2000). As a part of Niittymaki and his colleagues work we can mention to developing fuzzy rule bases for both choice and sequencing of signal stages to be used (Niittymaki, 2001b). They presented a systematic approach to fuzzy traffic signal control and prepared fuzzy rule based on experts knowledge. He considered the traffic signal programming in two sets: the choice and sequence of signal stages, and the optimization of the relative length of these stages. In this study, the rule bases for both of these problems were introduced and the result of the experiments for the rule bases was promising. Pedestrian friendly signals, separate signals for cyclists, public transport priorities, heavy vehicle priorities, all other priority systems, environmental sensitivity, and general routing aspect were the factors that Niittymaki considered them as effective factors in traffic and rule bases were prepared based on them. In another research they introduced a lukasiewicz many-valued logic similarity based fuzzy control algorithm which had a good statistical results high density traffic (Niittymaki & Turunen, 2003).

In the beginning of 90's, the first application of FLS in a multi-intersection network was published. Chiu and Chand (1993) presented a fully distributed system with cooperative local controllers to self-organizing traffic signal control. The parameters of each controller were adjusted by a local controller by considering the local traffic condition and the parameters of the adjacent intersections. For adjusting the standard signal timing parameters a set of fuzzy rules were applied by each local controller. In Their proposed system cycle time, phase split and also offset adjustments were considered. Their approach made it possible for the local controllers to have their own cycle time when the coordination is not important.

Lee and Lee-Kwang (1999) presented a FLC for a group of intersections. Each intersection controlled its own traffic while it had cooperation with its neighbors. The controller used these information and obtained the optimal signal's time through fuzzy rules. Each controller had three modules; the green phase observing module, the next phase selection module, and the decision module. The observation module produces the stop degree based on traffic conditions. The stop degree indicates the possibility that the controller should stop the green phase. The next phase module selects one candidate for the next green phase among all phases except the green phase. It observes the traffic conditions and selects the phase which is the most urgent among them, and the decision module decides about the time to switch to green phase. In addition, the controller had capability to manage phase length and phase sequence adaptively to the adjacent intersection as well as its own traffic conditions. To test the performance of the controller they developed a simulator for intersection groups. They compared the proposed method with vehicle actuated method under 18 traffic conditions; six traffic plans and three intersection groups. The intersections were two-ways and turning movements were allowed. The proposed method in that research showed better performance for all cases. It showed from 3.5% to 8.4% improvements over the vehicle actuated method in steady traffic conditions. In time-varying conditions, it had improvements from 4.3% to 13.5% were obtained in total average delay time.

Zhang et al. proposed a FLC for an over-saturated intersection having two-way streets with left-turning movement. This controller decided on whether to extend or terminate the current green phase. In another work Zhang, Wu, and Liu (2007), they proposed a two-layer fuzzy control algorithm for traffic control of the network which is supposed to have large traffic flow and high possibility of congestion.

A FLC for an isolated signalized intersection was proposed in Nair and Cai (2007). The aim of the controller was to ensure smooth flow of traffic by reducing the delay time. Most of the FLCs attempt to optimize the performance of the network by maximizing traffic flows or minimizing traffic delays under typical traffic conditions. As a result of that, these controllers are not optimal for exceptional traffic cases such as roadblocks and road accidents. In this research the authors proposed a FLC able to control traffic flows under both normal and exceptional traffic conditions. Traffic detector sensors were placed at incoming and outgoing links (lanes) and the controller utilized the information received from them to make optimal decisions. They also developed a simulator to evaluate the performance of traffic controllers under different conditions. Results showed that the performance of their proposed traffic controller was similar to that of conventional FLCs under normal traffic conditions and was better than of others under abnormal traffic conditions.

Rahman and Ratrout (2009) reviewed FLC in their study. The review covers: applying fuzzy method for two-way single intersection without turning vehicles, single intersection with all possible movements, multiple intersections, phase sequence and time determination, and congested intersection and network. This paper indicated better performance of fuzzy based controller compared to traditional traffic signal controls, specifically during uneven and heavy traffic conditions. Regarding to the similar situations that they recognized in Saudi Arabia, they found the FLC as a suitable solution for traffic issue there. They predicted the FLS approach will have a significant contribution in the future approach of transportation management system. In addition, they expected the contribution of FLC in the advancement of adaptive traffic signal control by improving the performance of the adaptive controller and the overall decision making process of the transportation management system.

Balaji and Srinivasan (2011) and Sabetghadam, Shabaninia, Vaziri, and Vadhava (2012) used type-2 fuzzy logic in controlling traffic signals. Non-stationary sensor noise, stochastic nature of drivers behavior, use of rules to control vehicles flow and signals, and use of expert knowledge for mining fuzzy rules from opinions are factors worth to be mentioned to make fuzzy type-2 more appropriate to be employed in designing such controllers. In Sabetghadam et al. (2012), it was mentioned that although computational intelligence based method like NN have been used for designing signal controller, a large training data set with all uncertainties they may contain make it difficult to obtain a proper controller. In their research they developed a multiagent distributed architecture signal control system based on type-2 fuzzy sets. All agents were homogeneous and had equal decision making capabilities. An agent calculated the appropriate green time based on averaged flow rate, queue length, and communicated data from the immediate neighbors, gathered by detectors attached to the intersection. Result of experiments against fixed-time method showed around 40% improvement. In Balaji and Srinivasan (2011), they also used a distributed agent architecture with fuzzy type-2 sets for reducing total delay time. In this study, the proposed method was compared with a hybrid NN based hierarchical multiagent system controller and real time adaptive traffic controller (Glide) which is used in Singapore. They showed that the proposed method had a significant improvement against the benchmarks for both dual and multiple peak traffic scenarios. Authors have other studies in applying multi-agent, NN and FLS in this regard (Balaji, 2011; Balaji & Srinivasan, 2009; Balaji & Srinivasan, 2010).

Wenchen, Lun, Zhaocheng, and Lijian (2012) and Chiou and Huang (2013) are two of the recent works in this field. In the work presented by Wenchen et al. (2012), the authors developed two adaptive two-stage fuzzy controllers for traffic signals at isolated intersections. Their controller had online optimization ability. Chiou and Huang (2013) proposed a stepwise genetic fuzzy logic controller. They considered queue lengths and traffic flows of cars and motorcycles as state variables and extension of green time as control variable based on these factors they worked on minimization of total vehicle delays. Through the experimental results they conclude that their proposed signal control model is efficient and robust.

Experts knowledge about traffic is useful to design a FLC. Predefining appropriate membership functions for both inputs and outputs need this knowledge. FLS is a beneficial tool to design a traffic controller as it can describe traffic situations appropriately.

Bi, Srinivasan, Lu, Sun, and Zeng (2014), proposed a multi-agent type-2 FLC. In this paper differential evolution is used to optimize the parameters of FLC membership functions and rule base. The network model in this study was composed of eleven intersections. Each intersection was controlled by one separate type2-FLC and the neighbor intersections communicate with each other. They demonstrated their proposed method can enhance the vehicular throughput rate and reduce delay time and queue length of vehicles.

In this section a review of the most cited and well known previous works in applying machine learning methods for traffic controlling are presented. *Q*-learning NN, and FLS were three popular methods that related works presented. Each of these methods have some strengths and weaknesses.

3. Traffic signal controllers for an isolated intersection

QLC, NNC, FLC, and fixed-time controller are four controllers designed and implemented in this study for controlling signal times for an isolated intersection with four approaching links Fig. 4.

In most of the previous reviewed studies, the timing of signal is done by extending or terminating the current traffic signal phase. In this case it is not possible to have a estimation of the end of a phase at the start of that phase, therefore traffic signals with timer are not suitable for them. Furthermore, usually there are fixed (discrete) rather than continuous extension times. Our proposed intelligent controllers (specially NNC and FLC) are flexible and can produce different range of values as a traffic signal phase duration. These numbers are determined at the start of each cycle. This option gives the opportunity of using timer signal controller that is useful for drivers to know how long they have to stay in traffic. More details about each of the controllers are presented later.

3.1. Design of QLC

We have designed a tabular QLC. States are formed from the average length of queues at approaching links. All lengths of queues are categorized in four ranges: low, medium and high. Different states are made of different combinations of these values for all approaching links. For example, an intersection with k approaching links we will have 4^k members in the state space. Here, we have 81 states for a single intersection with four approaching links.

The QLC is a tabular *Q*-learning method and there are limited action sets for this controller. Action sets are a combination of green times for each phase. The cycle time is flexible and based on the traffic demand, rather than fixed. We consider 10, 20, 30, 50 as all possible green times for a green phase and 256 different action sets exist. For example, for an intersection with four phases, some possible actions are: $\{10, 20, 20, 30\}$, $\{50, 30, 10, 10\}$, or $\{20, 10, 50, 10\}$ in which each member of the sets are the green times for related defined phases.

Reward is defined as inversely proportional to the average delay time at the end of each cycle for all the approaching links. It means there is a higher value for cases with a lower average delay.

The process of interaction between QLC and PARAMICS is presented in Fig. 5. The detected queue lengths form the environment (here this information is obtained from PARAMICS as a simulator) are sent to QLC and the appropriate green time for each phase are proposed. The proposed green times are selected from the predefined action list in *Q*-learning method.

3.2. Design of NNC

The developed NNC uses a feed-forward network. It is Genetic Algorithm (GA) based NN (GA-NN), which uses GA to find the optimal parameters for NNC. It consists of four input neurons, ten neurons in the hidden layer, and four neurons in the output layer. In every cycle the length of queues from PARAMICS are fed to the NN and the proposed green times for each cycle are generated and provided to PARAMICS. After each period of simulation, when a model simulation time is finished, the total average delay time, from the first cycle to the last cycle, is calculated and sent to Matlab as the cost function of the optimization process. Based on the cost function and by using GA optimization method, new weights for the NN are generated and weights are updated accordingly. This process is repeated until there is no further improvement for several iteration or after reaching a maximum number of iteration set in GA options. Upon termination, the NN parameters are set to the optimal set of weights.

It is important to note that the NN model is indirectly trained here, as the desired targets, which are green times in this problem, are unknown during training. For each generation, new populations with a new set of parameters are developed for the NN model. PARAMICS uses these temporary NN models as the brain for controlling traffic lights. At the end of each simulation, it returns the calculated average delay per vehicle as the cost function to Matlab. Decision whether to accept or discard the current solution (the neural network set of weights) is made in the GA optimization method. This process allows for the optimal adjustment of the NN parameters even if the desired targets are unavailable or unknown. Fig. 6 shows the training process.

3.3. Design of FLC

The FLC is also an optimal GA based controller (GA-FLS). FLC obtain its optimal parameters after training. Length of queues at

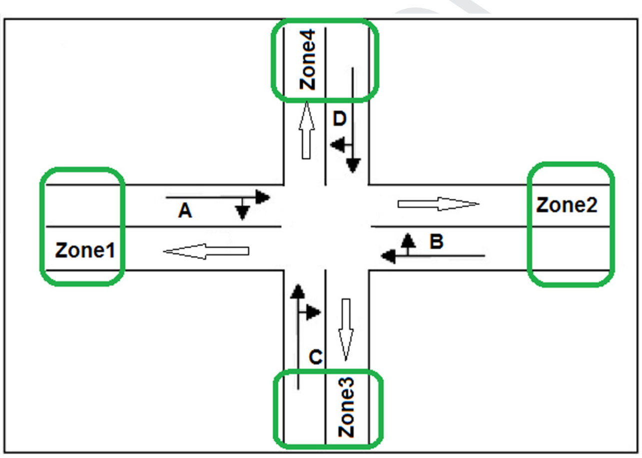

Fig. 4. An intersection with four phases: A, B, C, D. The cycle time is divided between these four phases. Vehicles in each lane can cross the intersection based on the related green time and direction of the phase. Four zones are also specified in the figure as Zone1, Zone2, Zone3, and Zone4.

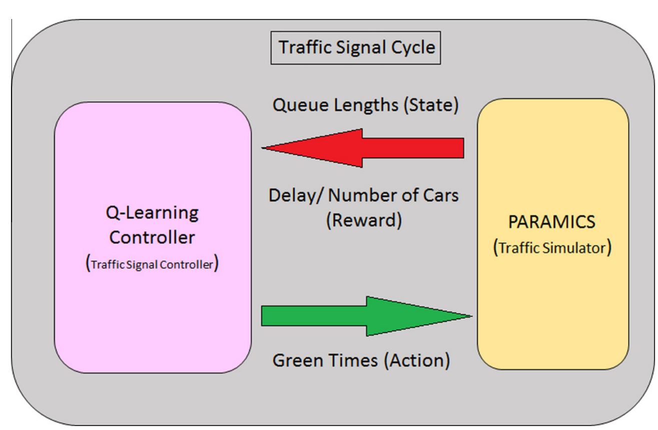

Fig. 5. This figure shows the learning process during each traffic cycle in a QLC. PARAMICS sends the information such as the number of cars and vehicles queue length in each lane in the current cycle to the controller. The controller computes and sends the appropriate green time for each phase of the next cycle to PARAMICS.

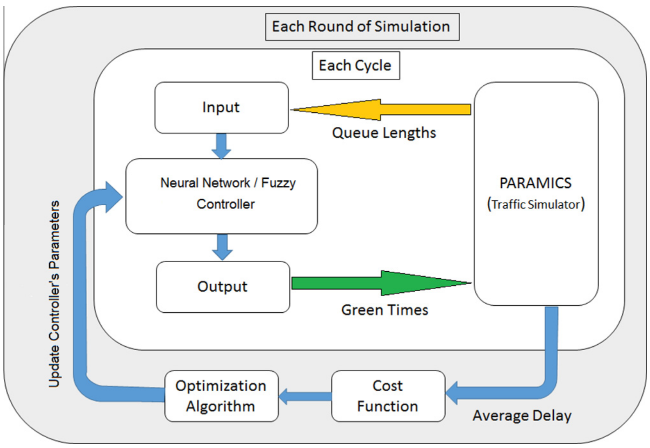

Fig. 6. The process of NN/FLS training. NN/FLS parameters are updated after each simulation run using a GA optimization method.

Table 1
FLC Rule Base. In this table S stands for small, M for medium, L for large, and ~ is for negation (eg., ~S means not-small, which could be either medium or large).

	CL	NL	2NL	3NL	Green time
(1)	S	S/M/L	S/M/L	S/M/L	S
(2)	M	S	S	S	M
(3)	M	~S	~S	~S	S
(4)	L	~L	~L	~L	L
(5)	L	L	~L	~L	M
(6)	L	~L	L	~L	M
(7)	L	~L	~L	L	M
(8)	L	~L	L	L	M
(9)	L	L	~L	L	M
(10)	L	L	L	~L	M
(11)	L	L	L	L	L

Table 2
Parameters of NNC.

Parameters	Value
Maximum number of generation	300
Population size	20
Number of layers	3
Number of inputs	4
Number of neurons in hidden layer	10
Number of outputs	4
Range of inputs	0–50
Rage of output	0–100

approaching links are fed as inputs of the FLC. The output of the FLC is the proposed green time for each phase. Therefore for an intersection with four approaching links we need four FLCs each for estimating the appropriate green time for related link. Membership functions for all inputs and the output of the FLCs are considered identical. Each input and the output has three membership functions named small, medium, large. Membership functions are Gaussian whose sigma and mean are optimized during training. Design of the FLC is similar to the NNC in Fig. 6.

Eleven rules are defined for the FLC. Queue length of vehicles at current link (CL), next link (NL), second next link (2NL), and third next link (3NL) are the factors considered in the rule base definition. For example we define the second rule as this:

*If CL is **medium**, NL is **small**, 2NL is **small**, and 3NL is **small** then Green Time is **medium**.*

Table 1 shows the rule base used for FLC.

Table 3
Parameters of FLC.

Parameters	Value
Maximum number of generation	300
Population size	20
Number of inputs	4
Number of outputs	1
Number of membership function for inputs	3
Number of membership function for output	3
Range of inputs	0–50
Range of output	0–100

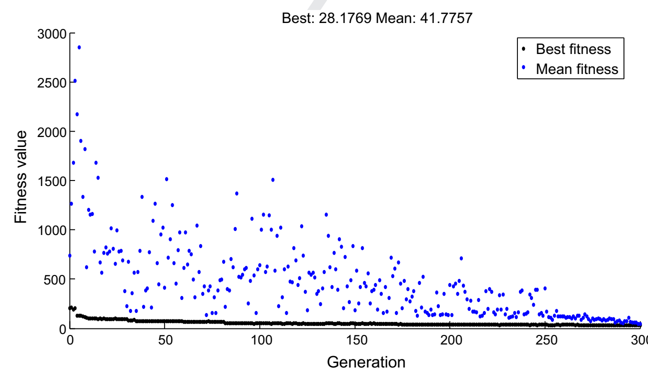

Fig. 7. The convergence profile of cost function for NNC for scenario one.

3.4. Fixed-time controller

Fixed-time controller or pre-timed controller is usually used as a benchmark for evaluating the performance of proposed controllers. In fixed-time controller a constant amount of time is set for each phase. The constant pre-defined time for each phase causes the least flexibility for fixed-time controller to adapt traffic demands. A simple structure is considered for fixed-time controller. In this structure we set equal time for all green phases.

4. Experiments results and discussion

For testing different controllers, an intersection with four approaching links and four phases (A, B, C, D) is considered. The cycle times are divided between aforementioned four phases. Zones are the areas that vehicles are released into the intersection, as shown in Fig. 4. Here, four different entrances to the intersection create four zones. The simulation is modeled in PARAMICS version 6.8. and all controllers are implemented in Matlab R2011b.

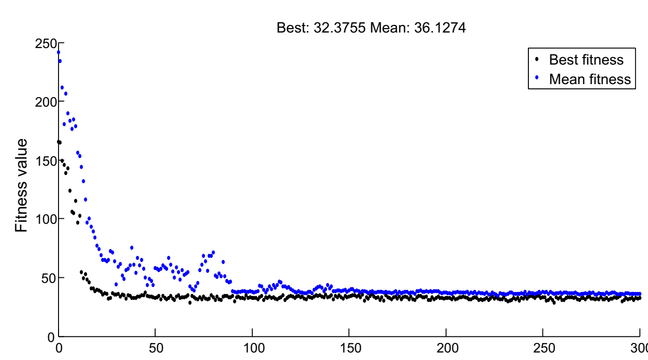

Fig. 8. The convergence profile of cost function for FLC for scenario one.

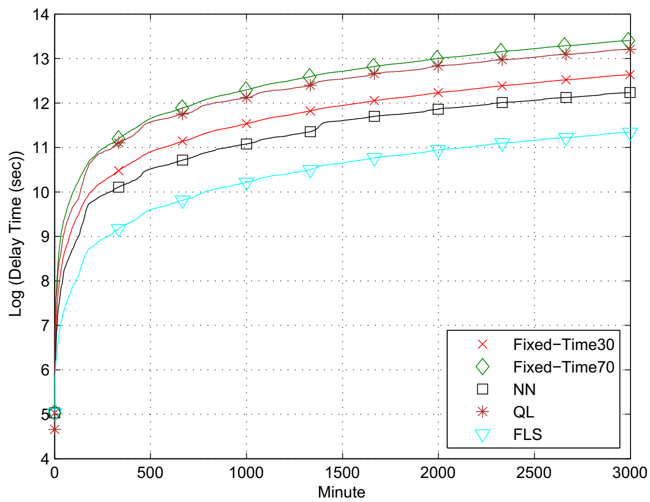

Fig. 9. The results of experiments for scenario one with 5500 vehicles. Accumulative total delay for an intersection for ten simulation runs with different seed numbers in five hours (300 min) simulation.

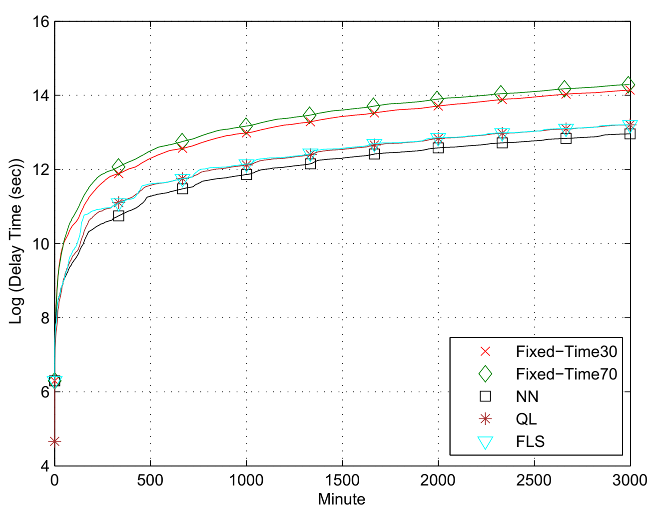

Fig. 10. The results of experiments for scenario two with 3000 vehicles. Accumulative total delay for an intersection for ten simulation runs with different seed numbers in five hours (300 min) simulation.

Two different scenarios are considered for evaluating the performance of the controllers:

1. Five hour simulation with 5500 vehicles (peak load); and
2. Five hour simulation with 3000 vehicles (non-peak load);

The developed Q-learning algorithm has 81 states for a single intersection with four approaching links. The queue length in each approaching link is categorize in three groups; small corresponds to 0–4; medium relates to 5–12; and more than 12 is considered as the large category. In this controller, 10, 20, 30, 50 are acceptable green times. Therefore, we have 256 actions for the intersection with flexible cycle time. The cycle length is calculated by summation of an allocated green time for each green phase plus two-second safety time after each green phase for clearing the intersection. For example, the cycle time is 108 s for a case with 50, 10, 20, 20 as proposed action.

For the QLC with ϵ-greedy method, a decreasing ϵ between 0.9 and 0.1 is considered, α (learning rate) is set to 0.1, and γ (discount factor) is set to 0.9. The reward function is defined as below:

$$Reward = \frac{1}{mean\left(\sum_{i=1}^{4} d_i\right) + 1} \quad (3)$$

where $i = 1, \ldots, 4$ is the number of approaching links, d is the calculated delay time for each link, and +1 is to refuse zero in denominator.

Similar structure are considered for NNC and FLC. Tables 2 and 3 show the parameters set for these controllers during our experiments.

The NNC has four inputs (number of approaching links to the intersection) and four outputs (proposed green time for each phase). For the FLC we have four inputs (number of approaching links to the intersection) and one output in our simulation for a 4-way intersection. We consider four FLCs for controlling the intersection with four phases. Each FLC proposes the appropriate green time for one phase. The proposed green time with both NNC and FLC are in range of 0–100 s.

Figs. 7 and 8 show the NNC and FLC parameters optimization trend during training respectively. Scenario one with 5500 vehicles is applied for training processes for both controllers. Both cost functions quickly converge to their global minimum in less than 50 iterations. The optimal set of parameters for NNC and FLC remains almost unchanged after this iteration.

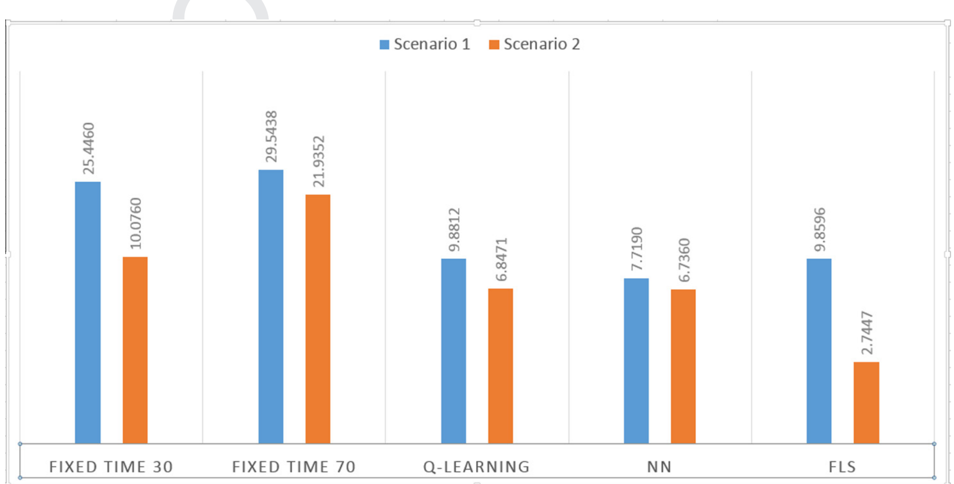

Fig. 11. Average total delay per vehicle for five-hour simulation. Results are the average for ten runs of scenario one and two.

Table 4
Performance order between controllers.

Method	Scenario 1	Rank in Scenario 1	Scenario 2	Rank in Scenario 2
Fixed-Time 30	25.45	4	10.08	4
Fixed-Time 70	29.54	5	21.94	5
Q-learning	9.88	3	6.85	3
NN	7.72	1	6.74	2
FLS	9.86	2	2.74	1

Two fixed-time controller are developed as benchmarks. To have reliable comparisons we consider two different time fixed-time controller. The first one has 30 s green phase time and the second one is set to 70 s for each of the predefined phase.

For both scenarios, we consider different seed numbers for each run of training and testing. Figs. 9 and 10 show the accumulative delay of the intersection in 10 runs for scenario one and two respectively. Each simulation lasts five hours.

The results show better a performance of intelligent controllers compared to both fixed-time controllers. Fixed-time controller has not flexibility to adapt to traffic demand. The best result for scenario one belongs to NNC, while FLC and QLC are the next ones respectively. In this scenario, the performance of the FLC and QLC are very close.

FLC obtains the best performance in scenario two. In this scenario NNC and QLC have almost similar accumulative delays.

FLC and NNC can propose and use all the integer numbers between 0 and 100. However, the difficulty of QLC in handling huge amount of data forced us to limit the action list to 10, 20, 30, and 50 during the experiments. Putting this condition leads reducing the QLC performance.

Fig. 11 shows the average delay per vehicle for ten runs of both scenarios. All presented results are the delay time in seconds. This figure shows the same order of performance among the controllers. Based on numbers presented in Table 4 it can be conclude that FLC, NNC, and QLC respectively have 74%, 71%, and 66% improvement against fixed-time controller.

5. Conclusion and future work

The main purpose of this paper is to present a review of the previous works that applied computational intelligence methods for traffic signal controlling. There are many different intelligent techniques applied for traffic signal timing, but there is not any study to compare the performance of these techniques in a similar platform. In this paper, we present some of the most valuable previous works applied computational intelligence methods; specially, *Q*-learning, NN, and FLS, to traffic signal control. Based on our studies these methods are more popular among other intelligent methods in design of adaptive traffic signal controllers.

In addition, we design and implement four controllers: *Q*-learning, NN, FLS, and fixed-time for an intersection with four approaching links and four different traffic signal phases. The designed NN and FLS controllers are GA based. It means their parameters are not predefined and they obtained the optimal parameters during training time. It is attempted to develop and evaluate all the controllers in similar conditions. This is the first time that a comparison between performances of the aforementioned controllers is made as part of one study.

The implemented intelligent controllers (especially NNC and FLC) in this study are flexible. It means these controllers are able to propose different range of number as traffic signal phase duration and they determine these numbers at the start of each cycle. This option gives the opportunity of using timer signal controller useful for drivers to know how long they have to stay in traffic. However, most of the previous controllers just have the ability to extend or terminate the current traffic phase.

The review is done in this paper reveals the better performance of intelligent controllers compared to traditional ones. The results of experiments for all designed controllers indicate NNC and FLC have the best performance. QLC also has a close performance to NNC and FLC. All intelligent controllers significantly outperform two fixed-time controllers. However, for having accurate conclusion about each of these controllers, there are some factors important to consider: Fixed-time controller may have a good performance in some cases, if the predefined time matches the traffic condition, but it is not useful for unpredictable situations of urban traffic. In addition, it needs experts' knowledge to set the appropriate time for each phase. Appropriately defining of states, actions, and reward function are issues for QLC. Furthermore, *Q*-learning is not useful to handle a huge amount of states and actions. It does not need supervision during learning period which is beneficial for controlling unpredictable urban traffic. NNC and FLC are suitable methods because of their speed and accuracy in learning and approximating hidden patterns in a huge amount of data. The performance of both methods is sensitive to their initialization and training process.

Designing controllers for a network of intersections and comparing the performance in multi-intersection case are considered as parts of the future works. Each of these learning methods has its own strengths and weaknesses. One of the other suggestion as the next study is designing a hybrid method, which leverage the strength of individual algorithms. In this work, we applied genetic algorithm for finding the optimal parameters for NN and FLS, but the rule base of the FLS is composed of predefined rules. Applying optimized rules seems lead to better performance for FLC and will be considered as an option for our next work.

References

Abdi, J., Moshiri, B., Abdulhai, B., & Sedigh, A. (2012). Forecasting of short-term traffic-flow based on improved neurofuzzy models via emotional temporal difference learning algorithm. *Engineering Applications of Artificial Intelligence, 25*, 1022–1042.

Abdi, J., Moshiri, B., Abdulhai, B., & Sedigh, A. (2013). Short-term traffic flow forecasting: Parametric and nonparametric approaches via emotional temporal difference learning. *Neural Computing and Applications, 23*, 141–159.

Abdoos, M., Mozayani, N., & Bazzan, A. (2011). Traffic light control in non-stationary environments based on multi agent q-learning. In *14th International IEEE conference on intelligent transportation systems (ITSC)* (pp. 1580–1585). Washington, DC, USA.

Abdoos, M., Mozayani, N., & Bazzan, A. L. (2013). Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence, 26*, 1575–1587.

Abdulhai, B., Pringle, R., & Karakoulas, G. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering, 129*, 278–285.

Arel, I., Liu, C., Urbanik, T., & Kohls, A. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems, 4*, 128–135.

Balaji (2011). *Distributed multi-agent based traffic management system* (Ph.D. thesis). Singapore: University of Madras.

Balaji, P. G., & Srinivasan, D. (2009). Distributed multi-agent type-2 fuzzy architecture for urban traffic signal control. In *IEEE international conference on fuzzy systems, 2009. FUZZ-IEEE 2009* (pp. 1627–1632). IEEE.

Balaji, P. G., & Srinivasan, D. (2010). Multi-agent system in urban traffic signal control. *IEEE Computational Intelligence Magazine, 5*, 43–51.

Balaji, P., & Srinivasan, D. (2011). Type-2 fuzzy logic based urban traffic management. *Engineering Applications of Artificial Intelligence, 24*, 12–22.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Bi, Y., Srinivasan, D., Lu, X., Sun, Z., & Zeng, W. (2014). Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization. *Expert Systems with Applications, 41*, 7338–7349.

Cai, C. (2010). *Adaptive traffic signal control using approximate dynamic programming* (Ph.D. thesis). London: University College.

Capek, K., Pitkanen, J.-P., & Niittymki, J. (2011). Evaluating the impacts of its applications using microscopic traffic simulators. *Advances in Transportation Studies*, 5–14 [Cited By (since 1996)0].

Chao, K.-H., Lee, R.-H., & Wang, M.-H. (2008). An intelligent traffic light control based on extension neural network. In *Knowledge-based intelligent information and engineering systems*. In I. Lovrek, R. J. Howlett, & L. C. Jain (Eds.). *Lecture notes in computer science* (5177, pp. 17–24). Berlin Heidelberg: Springer.

Chin, D., Spall, J., & Smith, R. (1999). Evaluation of system-wide traffic signal control using stochastic optimization and neural networks. In *American control conference* (pp. 2188–2194). Vol. 3.

Chiou, Y.-C., & Huang, Y.-F. (2013). Stepwise genetic fuzzy logic signal control under mixed traffic conditions. *Journal of Advanced Transportation, 47*, 43–60.

Chiu, S., & Chand, S. (1993). Self-organizing traffic control via fuzzy logic. In *Proceedings of the 32nd IEEE conference on decision and control, 1993* (pp. 1897–1902). Vol. 2.

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

Cools, S.-B., Gershenson, C., & D'Hooghe, B. (2008). Self-organizing traffic lights: A realistic simulation. In M. M. Hon & Mikhail Prokopenko (Eds.), *Advances in applied self-organizing systems. Advanced information and knowledge processing* (pp. 41–50). London: Springer.

Dai, Y., Zhao, D., & Yi, J. (2010). A comparative study of urban traffic signal control with reinforcement learning and adaptive dynamic programming. In *The international joint conference on neural networks (IJCNN)* (pp. 1–7).

Dekkers, A., & Aarts, E. (1991). Global optimization and simulated annealing. *Mathematical Programming, 50*, 367–393.

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 Transactions on Intelligent Transportation Systems, 14*, 1140–1150.

Favilla, J., Machion, A., & Gomide, F. (1993). Fuzzy traffic control: Adaptive strategies. In *Second IEEE international conference on fuzzy systems, 1993* (pp. 506–511). Vol. 1.

Gartner, N. H., Tarnoff, P. J., & Andrews, C. M. (1991). Evaluation of optimized policies for adaptive control strategy. *Transportation Research Record*.

Granberg, M., & Niittymki, J. (2001). Traffic performance-based air quality management. *Advances in Transport, 8*, 399–408 [Cited By (since 1996)1].

Granberg, M., Niittymki, J., Karppinen, A., & Kukkonen, J. (2000). Combined application of traffic microsimulation and street canyon dispersion models, and evaluation of the modelling system against measured data. *Advances in Transport, 6*, 349–358 [Cited By (since 1996)0].

Heltimo, J., Niittymaki, J., & Bjork, E. (2002). A microlevel method for road traffic noise prediction. *Advances in Transport*, 659–668 [Cited By (since 1996)0].

Houli, D., Zhiheng, L., & Yi, Z. (2010). Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network. *EURASIP Journal on Advances in Signal Processing*, 1687–6180.

Hu, Y., Thomas, P., & Stonier, R. (2007). Traffic signal control using fuzzy logic and evolutionary algorithms. In *IEEE congress on evolutionary computation, 2007. CEC 2007* (pp. 1785–1792).

Hunt, P., Robertson, D., & Bretherton, R. (1982). The SCOOT on-line traffic signal optimisation technique (glasgow). *Traffic Engineering & Control, 23*, 190–192.

Kononen, V., & Niittymaki, J. (2000). *New methods for traffic signal control – Development of fuzzy controller, 1*, 358–362 [Cited By (since 1996)1].

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., Bonneson, J., Tarnoff, P., & Urbanik, T. (2008). *Traffic signal timing manual. Technical Report*.

Kronborg, P., & Davidsson, F. (1996). *Development and field trials of the new SOS algorithm for optimising signal control at isolated intersections. Technical Report*.

Kronborg, P., & Davidsson, F. (1993). MOVA and IHOVRA. Traffic signal control for isolated intersections. *Traffic Engineering and Control, 34*, 195–200.

Lee, J.- H., Lee, K.- M., & Leekwang, H. (1995). Fuzzy controller for intersection group. In *International IEEE/IAS conference on industrial automation and control: Emerging technologies, Proceedings* (pp. 376–382).

Lee, J.-H., & Lee-Kwang, H. (1999). Distributed and cooperative fuzzy controllers for traffic intersections group. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 29*, 263–271.

Lehmuskoski, V., Niittymaki, J., & Silfverberg, B. (2000). Microscopic simulation on high-class roads: Enhancement of environmental analyses and driving dynamics: Practical applications. *Transportation Research Record*, 73–81 [Cited By (since 1996)3].

Malej, A., & Brodnik, A. (2007). Fuzzy urban traffic signal control – An overview. *Elektrotehniski Vestnik/Electrotechnical Review, 74*, 291–296.

Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies, 9*, 415–432.

Mitchell, T. M. (1997). *Machine learning* (1st ed.). McGraw-Hill [Science/Engineering/Math].

Murat, Y. S., & Gedizlioglu, E. (2005). A fuzzy logic multi-phased signal control model for isolated junctions. *Transportation Research Part C: Emerging Technologies, 13*, 19–36.

Nagare, A., & Bhatia, S. (2012). Traffic flow control using neural network. In *International journal of applied information systems (IJAIS)* (pp. 50–52). New York, USA. Vol. 1.

Nair, B., & Cai, J. (2007). A fuzzy logic controller for isolated signalized intersection with traffic abnormality considered. In *2007 IEEE intelligent vehicles symposium* (pp. 1229–1233).

Nakatsuyama, M., Nagahashi, H., & Nishizuka, N. (1985). Fuzzy logic phase controller for traffic junctions in the one-way arterial road. In *IFAC proceedings series* (pp. 2865–2870).

Niittymaki, J. (2001a). Fuzzy traffic signal control – Environmentally better choice. *Advances in Transport, 8*, 327–336 [Cited By (since 1996)0].

Niittymaki, J. (2001b). General fuzzy rule base for isolated traffic signal control-rule formulation. *Transportation Planning and Technology, 24*, 227–247 [Cited By (since 1996)18].

Niittymaki, J. (2001c). Installation and experiences of field testing a fuzzy signal controller. *European Journal of Operational Research, 131*, 273–281 [Cited By (since 1996)21].

Niittymaki, J., & Kikuchi, S. (1998). Application of fuzzy logic to the control of a pedestrian crossing signal. *Transportation Research Record*, 30–38 [Cited By (since 1996)27].

Niittymaki, J., & Kononen, V. (2000). *Traffic Signal Controller Based on Fuzzy Logic, 5*, 3578–3581 [Cited By (since 1996)7].

Niittymaki, J., Kosonen, T., & Nevala, R. (2001). *Fuzzy Traffic Signal Control in Major Arterials*, 346–351 [Cited By (since 1996)2].

Niittymaki, J., & Nevala, R. (2001). *Fuzzy Adaptive Traffic Signal Control – Principles and Results, 5*, 2870–2875 [Cited By (since 1996)2].

Niittymaki, J., Nevala, R., Halme, K., & Aitken, S. (2008). Simulation study of the main road network of south-eastern Finland. *Advances in Transportation Studies*, 43–48 [Cited By (since 1996)0].

Niittymaki, J., & Pursula, M. (1996). Saturation flows at signal-group-controlled traffic signals. *Transportation Research Record*, 24–32 [Cited By (since 1996)5].

Niittymaki, J., & Pursula, M. (1997). Saturation flows at signal-group-controlled traffic signals. *Transportation Research Record*, 24–32 [Cited By (since 1996)3].

Niittymaki, J., & Pursula, M. (2000). Signal control using fuzzy logic. *Fuzzy Sets and Systems, 116*, 11–22 [Cited By (since 1996)56].

Niittymaki, J., & Pursula, M. (2001). The 11th mini-euro conference: Artificial intelligence on transportation systems and science. *European Journal of Operational Research, 131*, 229–231 [Cited By (since 1996)4].

Niittymaki, J., & Sane, K. J. (1996). *Tram simulation in Helsinki – A new research method*, 76–80 [Cited By (since 1996)0].

Niittymaki, J., & Turunen, E. (2003). Traffic signal control on similarity logic reasoning. *Fuzzy Sets and Systems, 133*, 109–131 [Cited By (since 1996)20].

Pappis, C. P., & Mamdani, E. H. (1977). A fuzzy logic controller for a traffic junction. *IEEE Transactions on Systems, Man and Cybernetics, 7*, 707–717.

Prashanth, L., & Bhatnagar, S. (2011). Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems, 12*, 412–421.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc..

Rahman, S. M., & Ratrout, N. T. (2009). Review of the fuzzy logic based approach in traffic signal control: Prospects in Saudi Arabia. *Journal of Transportation Systems Engineering and Information Technology, 9*, 58–70.

Sabetghadam, B., Shabaninia, F., Vaziri, M., & Vadhava, S. (2012). Type-2 fuzzy multiagent traffic signal control. In *2012 IEEE 13th international conference on information reuse and integration (IRI)* (pp. 509–512).

da Silva, B.C., Basso, E.W., Perotto, F.S., C. Bazzan, A.L., & Engel, P.M. (2006). Improving reinforcement learning with context detection. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems AAMAS'06* (p. 810–812). New York, NY, USA: ACM.

Sims, A. G., & Dobinson, K. W. (1980). The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Transactions on Vehicular Technology, 29*, 130–137.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control, 37*, 332–341.

Spall, J. C., & Chin, D. C. (1997). Traffic-responsive signal timing for system-wide traffic control. *Transportation Research Part C: Emerging Technologies, 5*, 153–163.

Srinivasan, D., Choy, M. C., & Cheu, R. (2006). Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems, 7*, 261–272.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems* (Vol. 8, pp. 1038–1044). MIT Press.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning an introduction*. MIT Press.

Teodorovic, D., Varadarajan, V., Popovic, J., Chinnaswamy, M. R., & Ramaraj, S. (2006). Dynamic programming neural network real-time traffic adaptive signal control algorithm. *Annals of Operations Research, 143*, 123–131.

Thorpe, T. L., & Anderson, C. W. (1996). *Traffic light control using SARSA with three state representations. Technical Report*. IBM Corporation.

Vincent, R., & Peirce, J. (1988). *MOVA: Traffic responsive, self-optimising signal control for isolated intersections. Technical Report RR 170*, Crowthorne: TRRL Transport and Road Research Laboratory.

Webster, F.V. (1958). *Traffic signal settings. Technical Report*. H.M. Stationery Office.

Wei, W., Zhang, Y., Mbede, J., Zhang, Z., & Song, J. (2001). Traffic signal control using fuzzy logic and moga. In *Proceedings of the IEEE international conference on systems, man and cybernetics* Vol. 2. (pp. 1335–1340).

Wenchen, Y., Lun, Z., Zhaocheng, H., & Lijian, Z. (2012). Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and paramics simulation. In *2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 391–396).

Wen, K., Qu, S., & Zhang, Y. (2007). A stochastic adaptive control model for isolated intersections. In *IEEE international conference on robotics and biomimetics* (pp. 2256–2260).

Wiering, M. (2000). *Multi-agent reinforcement learning for traffic light control*.

Wiering, M., Vreeken, J., van Veenen, J., & Koopman, A. (2004). Simulation and optimization of traffic in a city. In *IEEE Intelligent Vehicles Symposium* (pp. 453–458).

Wunderlich, R., Liu, C., Elhanany, I., & Urbanik, T. (2008). A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection. *IEEE Transactions on Intelligent Transportation Systems, 9*, 536–547.

Yin, H., Wong, S., Xu, J., & Wong, C. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies, 10*, 85–98.

Zadeh, L. A. (1975a). The concept of a linguistic variable and its application to approximate reasoning: Part 1. *Information Sciences, 8*, 199–249.

Zadeh, LA. (1975b). The concept of a linguistic variable and its application to approximate reasoning: Part 2. *Information Sciences, 8*, 301–357.

Zadeh, LA. (1975c). The concept of a linguistic variable and its application to approximate reasoning: Part 3. *Information Science, 9*, 43–80.

Zeng, R., Li, G., & Lin, L. (2007). Adaptive traffic signals control by using fuzzy logic. In *Second international conference on innovative computing, information and control, 2007. ICICIC'07* (pp. 527–527).

Zhang, W. -b., Wu, B. -z., & Liu, W. -j. (2007). Anti-congestion fuzzy algorithm for traffic control of a class of traffic networks. In *IEEE international conference on granular computing, 2007. GRC 2007*. (p. 124).

Zhang, L., Li, H., & Prevedouros, P. (2008). Signal control for oversaturated intersections using fuzzy logic. In *Transportation and development innovative best practices* (pp. 179–184). American Society of Civil Engineers.

Q1 Please cite this article in press as: Araghi, S., et al. A review on machine learning methods for controlling traffic signal timing. *Expert Systems with Applications* (2014), http://dx.doi.org/10.1016/j.eswa.2014.09.003