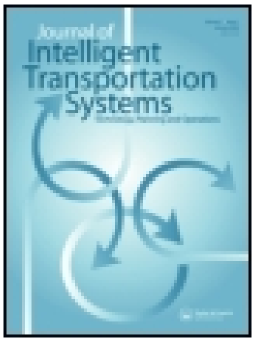


Taylor & Francis
Taylor & Francis Group

Journal of Intelligent Transportation Systems

Technology, Planning, and Operations

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: http://www.tandfonline.com/loi/gits20

Learning based Traffic Signal Control Algorithms with Neighborhood Information Sharing: An Application for Sustainable Mobility

H. M. Abdul Aziz, Feng Zhu & Satish V. Ukkusuri

To cite this article: H. M. Abdul Aziz, Feng Zhu & Satish V. Ukkusuri (2017): Learning based Traffic Signal Control Algorithms with Neighborhood Information Sharing: An Application for Sustainable Mobility, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2017.1387546

To link to this article: http://dx.doi.org/10.1080/15472450.2017.1387546

Accepted author version posted online: 04 Oct 2017.

Learning based Traffic Signal Control Algorithms with Neighborhood Information Sharing: An Application for Sustainable Mobility

H. M. Abdul Aziz, Ph.D.[1], Feng Zhu, Ph.D.[2], Satish V. Ukkusuri, Ph.D. Professor[3]

[1]Urban Dynamics Institute, Oak Ridge National Laboratory, 1 Bethel Valley Road, TN 37830, USA
[2]Lyles School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA
[3]Lyles School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA, Phone : (765)-494-2296, Fax : (765)-496-7996
(Corresponding author) E-mail: aziz.husain.nexus@gmail.com

E-mail: zhu214@purdue.edu

E-mail: sukkusur@purdue.edu

Journal of Intelligent Transportation Systems: Technology, Planning, and Operations

(Final draft, January 2017)

ABSTRACT

This research applies R-Markov Average Reward Technique based reinforcement learning (RL) algorithm, namely RMART, for vehicular signal control problem leveraging information sharing among signal controllers in connected vehicle environment. We implemented the algorithm in a network of 18 signalized intersections and compare the performance of RMART with fixed, adaptive, and variants of the RL schemes. Results show significant improvement in system performance for RMART algorithm with information sharing over both traditional fixed signal timing plans and real time adaptive control schemes. The comparison with reinforcement learning algorithms including Q learning and SARSA indicate that RMART performs better at higher congestion levels. Further, a multi-reward structure is proposed that dynamically adjusts the reward function with varying congestion states at the intersection. Finally, the results from test networks show significant reduction in emissions (CO, CO_2, NO_x, VOC, PM_{10}) when RL algorithms are implemented compared to fixed signal timings and adaptive schemes.

Keywords

Reinforcement learning; Vehicular Emissions; Connected and Automated Vehicles; Traffic Signal Control; Sustainable Transportation.

1. MOTIVATION AND BACKGROUND

Traffic congestion is ubiquitous in the 471 urban areas of the United States and is responsible for 3.1 billion gallons of additional fuel consumption in 2014 (Schrank et al., 2015). The net congestion cost is 160 billion (in 2014 U.S. dollars) with about 6.9 billion hours of delay. Further, the transportation sector alone is responsible for about 76 percent of the total CO emissions and 50 percent of the total NOx (EPA, 2013) emissions in the United States. The U.S. Environmental protection agency (EPA) reports transportation sector as the fastest growing source of greenhouse gas (GHG) emissions indicating 47 percent net increase from 1990 to 2003 (EPA, 2006). The contribution of delay due to inefficient traffic signals is about 5 to 10 percent of the net delay (NTOC Report, 2012). National Traffic Signal Report Card (NTOC Report, 2012) gave C grade for the current traffic signal operations and emphasizes on optimized signal scheme implementation. Clearly, it is important to design signal control systems that can minimize travel delay, intersection delay, and number of stops at the intersection. Moreover, signal systems with reduced number of stops, intersection delay, and low acceleration rates will lead to less vehicular emissions for the network. Accordingly, this research aims to find signal control schemes that lead to sustainable mobility, i.e., reduced delay for the users and less vehicular emissions at network level.

Adaptive signal control schemes such as SCOOT (Hunt et al., 1982), SCATS (Lowrie, 1982), PRODYN (Farges et al., 1983), OPAC (Gartner, 1983), RHODES (Mirchandani & Head, 2001), UTOPIA (Mauro & Taranto, 1989), CRONOS (Boilot et al., 2006), and TUC (Diakaki et al., 2002) are found to perform better than fixed and actuated signal timing plans. Nevertheless, adaptive schemes are often limited in terms of scalability and robustness (Abdulhai & Kattan, 2003; Medina & Benekohal, 2011; El-Tantawy et al., 2013). Many of these control systems (e.g., SCOOT and SCATS) are centralized systems operating based on real time traffic data and some (e.g., OPAC and RHODES) apply dynamic optimization to find control schemes. However, none of them adaptively learns from the environment and the computational complexity increases exponentially with the network size. Further, researchers from the machine learning and artificial intelligence area have also applied algorithms that include neuro-fuzzy networks (Srinivasan et al., 2006), neural networks (Li & Mueck, 2010), Tabu search (Hu & Chen, 2012), self-organizing maps (Li et al., 2011), emotional algorithm (Ishihara & Fukuda, 2001), and genetic algorithms (Stevanovic et al., 2012). Two major limitations with these algorithms are the requirement of large data to calibrate the parameters and exponential complexity for large scale networks (Balaji et al., 2010). To overcome these limitations, researchers also explored data-driven learning techniques as an alternative to real time adaptive algorithms (Abdulhai & Kattan, 2003; Wiering et al., 2004; Bazzan et al., 2010; Medina & Benekohal, 2011; El-Tantawy & Abdulhai, 2010; El-Tantawy et al., 2013).

Since traffic environment is inherently dynamic and changes over time, there is a scope to learn for its elements (e.g., signal controllers) through interaction with the environment. Later, controllers can adjust the actions towards the desired state of the system. Among different learning techniques, reinforcement learning (RL) is one of the widely used control techniques applied to solve the traffic signal control. In RL-based schemes, the agent (i.e., signal controller) learns from interacting with the environment, which is often modeled as Markov Decision Process (MDP). The ability to learn from the environment and scalability are the key advantages of RL in terms of implementation because no direct optimization is generally involved. The interactive nature of reinforcement learning algorithms requires a communication interface where the agents (vehicles and controllers) can send and receive information among each other. This fits well into the paradigm of connected and automated vehicles in transportation networks.

With growing interests and investments of the US federal agencies (including the US DOT and the DOE) and the automobile industry, we expect to see an extensive deployment of Connected and Automated Vehicles (CAV) in near future. The bi-directional communication capability between CAVs and traffic signals can be leveraged to develop control schemes that will lead to a desired state of the system—minimal fuel consumption and greenhouse gas (GHG) emissions in addition to maximizing throughput and minimizing travel delays. CAV deployment allows vehicles talk to each other (Vehicle-to-Vehicle, V2V), to the infrastructure components (Vehicle-to-Infrastructure, V2I), and infrastructure-to-infrastructure communication (I2I). Researchers have used connected vehicle (CV) environment for dynamic traffic control for throughput maximization (Chen et al., 2013), to design adaptive control scheme (Feng et al., 2015), to analyze the impact of CV on work zone safety (Genders & Razavi, 2015), and to design coordinated transit signal priority schemes (Hu et al., 2015). Florin & Olariu (2015) provides a survey on the use of CV for traffic signal optimization.

This research develops and implements a learning-based algorithm for signal control that allows signal controller agents to share information with its neighbor controllers through I2I communication within CV environment. Later we also show that, learning with information sharing improves the performance of the RL algorithm. To summarize, this research applies reinforcement-learning (RL) techniques for signal control (namely, the R-Markov Average Reward Technique or RMART) at network level. The developed control algorithm leverages the communication capabilities in connected vehicle environment through information sharing among the neighborhood controllers. The remainder of the paper is organized as follows: the literature review section describes related works previously done by researchers, the problem definition section states the hypotheses and research questions of this research, the methodology section explains the RL algorithm, the numerical results section reports the results obtained from test networks, and finally, we discuss the important contributions, limitations, and future directions of this research.

2. LITERATURE REVIEW

The implementation of RL in signal control area has been well studied in the last decade. Thorpe (1997) used a neural network to predict waiting time and applied on-policy RL (SARSA) for signal control. Miakami & Kakazu (1994) proposed cooperative signal control scheme with a combination of evolutionary algorithm and reinforcement learning techniques. Bingham (2001) proposed rules based on fuzzy-logic that allocates green times based on the number of vehicles. Other than signal control researchers have also used it for other problems in transportation (Dai et al., 2005; Desjardins & Chaib-draa, 2011; Abdoos et al., 2011; Wei-Song & Jih-Wen, 2011).

Abdulhai et al. (2003) applied off policy (Q-learning) algorithm to optimize signal control in an isolated intersection. Application to larger networks was challenging due to exponential increase in the joint state-action space. Later, Wiering et al. (2004) proposed co-learning algorithms at network level accounting for the waiting time for the vehicles and used car-based value function that reduced the state space to a reasonable number. However, the prediction of waiting time is not accurate and the traffic simulator lacks important modules such as lane changing and dynamic route choice. Researchers (Kuyer et al., 2008; Bazzan, 2005; Bazzan et al., 2010) have also studied cooperative multi agent system for urban traffic control. More recently, El-Tantawy et al. (2013) proposed neighborhood coordinated RL based signal control and described a joint decision framework to present multi agent framework. Although Q-learning and SARSA are most widely used temporal difference techniques, researcher also applied other algorithms like actor-critic temporal difference (Xie, 2007), Q-learning with function approximation (Prashanth & Bhatnagar, 2011) and action dependent adaptive dynamic programming (Li et al., 2008). Although commonly used in long-term average reward specific algorithms, R-Markov Average Reward Technique (RMART) has not been applied potentially in the context of vehicular traffic control. Recognizing the potential to address long term average reward this research applies RMART technique for signal control.

The RL algorithms applied for signal control vary greatly with the definitions of state and reward. El-Tantawy et al. (2013) provides an excellent discussion on the variations in state and reward definitions in context of signal control. The most common definitions of state include number of arriving vehicles, queue lengths, average delay, and so on. Most of them do not consider information sharing among neighborhood controllers. Neighborhood information provides us with congestion status of the surrounding controllers. Including this information will help the controller to learn better. Consider a case when the adjacent intersections of a particular intersection are heavily loaded and in near future this intersection will experience heavy load. Using only local information, the agent does not have any idea of the immediate congestion that will appear. When the state definition includes congestion status of the adjacent intersections the agent learns to adjust signal settings when the nearby intersections are congested. Based on this idea, this research adds congestion information of the neighborhood intersections to the definition of state in the RL algorithm. This idea is different from the multi-agent coordination research (El-Tantawy & Abdulhai, 2010; El-Tantawy et al., 2013, Wiering et al., 2004; Kuyer et al., 2008; and Bazzan et al., 2010) because multi-agent cooperative learning deals with the joint state-action space optimality. This research focuses on adding neighborhood information in the state definition without explicit coordination among the controllers.

Rewards in a RL algorithm can take different forms including number of stops made, intersection delay, and throughput for the intersection. The reward is defined in a static manner and the definition does not change over time. However, rewards can be dynamic as a response to the current state of the traffic network and multiple reward structure can be used (Ngai & Yang, 2011). Houli et al. (2010) defined different reward functions such as stops, delay, and so on, for different congestion levels: free flow, saturated condition, etc. However, their approach is not truly dynamic because the congestion level is always known beforehand and rewards are predefined for different time of analysis. This study takes a different approach where the reward takes a dynamic form in the sense that reward definitions changes based on the congestion level in real-time. In addition to static reward structure, we also examine the performance of RL algorithms with this kind of dynamic structure.

Finally, most research works relevant to learning techniques for signal control do not evaluate the benefits in terms of reducing vehicular emissions. Although connected vehicle research area has some potential works (Lee & Park, 2012; Huang et al., 2012; Kwak et al., 2012) that evaluate benefits in terms of reducing emissions, however not in the context of applying learning techniques.

We address these limitations and the key contributions of this work are as follows:

a) Implementation of the novel RMART technique for traffic signal control that allows for neighborhood congestion information sharing within the CV environment and compare with fixed, adaptive, and other learning algorithms,

b) Evaluation of multi-reward RL algorithm that accounts for the dynamic variation of traffic demand,

c) Demonstration of the benefits of learning based control algorithms in terms of reducing emissions from the traffic network.

3. PROBLEM DEFINITION

3.1 Solving signal control problem through RL technique

Optimization of vehicular traffic control requires the determination of signal timing parameters: scheduling and allocation of green time to specific set of movements. A set of non-conflicting allowable movements is defined as phase or stage. In context of RL, traffic network is the environment and the traffic controllers act as agents. We define the action of an agent as the activation of a particular phase (predefined) at the decision interval. Thus, the traffic signal control problem has all the elements of MDP. Each time the agent takes an action that influences the current environment, the state of the environment changes. The problem is to find the optimal policy (mapping between the phase-activation and traffic states) that gives the largest reward that is commonly defined in terms of average delay, number of stops, etc. in the long-term.

The key idea of RL comes from DP and artificial intelligence (AI) based learning techniques. A detailed description can be found in Sutton & Barto (1998) and Gosavi (2003). The two key elements of MDP are the reward and state transition probability. The RL technique is most appropriate when these elements are not deterministic. The solution methodology should contain components that determine the transition probabilities and rewards as a feedback from the environment. However, a simulator of the real environment can provide us with the reward and the transition of the states can be observed. This research uses VISSIM (PTV, 2012) as a traffic simulator that provides rewards and other performance.

3.2 Research hypotheses

The research hypotheses are as follows:

H1: *The learning technique based algorithm will perform better than both fixed signal timing plans and adaptive control schemes [in our case the MWM algorithm (Wunderlich et al., 2008)] in terms of travel delay, intersection delay, and number of stops*.

H2: *Information sharing through the CV environment (I2I) will improve the learning algorithm (e.g., higher reduction of delay, intersection delay).*

H3: *The multi-reward structure will perform better than the single reward RL algorithm.*

H4: *Learning based algorithm will yield lower emission for the traffic network than the fixed timing plans and adaptive controls.*

4. METHODOLOGY

Reinforcement learning techniques follow sampling based approaches to solve the optimal control problems. RL systems contain basic components: the state, action, and reward. These components are specific to the problem at hand. Next, we define the state, action, and reward for the proposed RL algorithm.

4.1 State of the system (traffic environment)

Before defining state, we need to define the *Normalized-queuing-index* for each lane group served by the signal phases at the intersection. The notion of *Normalized-queuing-index* captures the capacity of a certain lane. It tells us how much capacity a certain lane is left. Traditional concept of queue length does not account for the length of the lane—30 feet queue in 100 feet lane is apparently more congested compared with a 30 feet queue in a 150 feet lane. This is why we introduce the concept of *Normalized-queuing-index*. Using jam density and number of vehicles in the queue, we compute the *Normalized-queuing-index*. Jam density is usually expressed in number of passenger car equivalent (PCE) per lane-mile. It refers to the density of a lane when speed equals to zero for all the vehicles on that lane. Highway Capacity Manual (HCM) suggests using 190 PCE per lane-mile for freeway facilities. Note that, one should choose the value that is consistent with the existing network conditions. Now, the *Normalized-queuing-index* for a lane *i*, is defined as:

$$\omega_i^t = \frac{q_i^t}{J} \times \frac{1}{l_i} \qquad (1)$$

$\omega_i^t =$ *Normalized-queuing-index*

$q_i^t =$ Queue length (PCE units) for lane *i* at step *t*

$J =$ Jam density (PCE units per lane-mile)

$l_i =$ Length of lane (in miles)

Further, the *Normalized-queuing-index* for the lane group served by phase *p*, can be estimated by taking the average over all the lanes.

$$\pi_p^t = \frac{\sum_{i \in \text{lane group}, p} \omega_i}{\sum_{i \in \text{ lane group}, p} i} \qquad (2)$$

$\pi_p^t =$ Average *Normalized-queuing-index* for the lane group serving phase *p* at step *t*.

It can be seen that π_p^t is continuous in nature and can take any value between 0 and 1. Next, the average *Normalized-queuing-index* (*NQ-index*) for a particular phase, *p* is labeled as low, high, or medium using the following conditions:

$$\Pi_p^t = \left\{ \begin{matrix} L, \text{if } \pi_p^t < \tau_1 \\ M, \text{if } \tau_1 \le \pi_p^t < \tau_2 \\ H, \text{if } \pi_p^t \ge \tau_2 \end{matrix} \right\}; L = \text{low}, H = \text{high}, M = \text{medium}. \qquad (3)$$

$\Pi_p^t =$ label of π_p^t.

In our experimental settings we have used $\tau_1 = 0.4$, and $\tau_2 = 0.7$.

The *NQ-index* of the intersection μ_p^t is computed using the *NQ-index* values of the phases. Different values are assigned to the labels of *NQ-index* of a particular phase.

$$\mu_p^t = \begin{Bmatrix} \lambda_1, \text{if } \Pi_p^t = L \\ \lambda_2, \text{if } \Pi_p^t = M \\ \lambda_3, \text{ if } \Pi_p^t = H \end{Bmatrix} \qquad (4)$$

We assume, $\lambda_1 = 1, \lambda_2 = 3, \lambda_3 = 5$

Now, the labels for *NQ-index* values of the intersection are defined as follows:

$$\Omega_j^t = \begin{Bmatrix} \text{Free flow;if} \sum_{p \in \text{Phases}} \mu_p^t < \Theta_1 \\ \text{Average flow;if } \Theta_1 \le \sum_{p \in \text{Phases}} \mu_p^t < \Theta_2 \\ \text{Saturated flow;if} \sum_{p \in \text{Phases}} \mu_p^t \ge \Theta_2 \end{Bmatrix} \qquad (5)$$

The simulation uses $\Theta_1 = 10$ and $\Theta_2 = 16$

Note that these values in (3), (4), and (5) are arbitrary and depends on the judgment of the analyst and scope of the problem. The threshold values we have used are based on experimental observations. Our primary objective in this research is to demonstrate the capability of learning based signal control schemes that can leverage the connected-vehicle paradigm. We chose the traffic flow parameters such that congestion states ranging from low to saturated can be tested with our developed algorithms. Using recommended saturation rates, jam density and other parameters as suggested in the highway capacity manual, we designed the scenarios. The values 0.4, 0.7, etc.

are chosen after multiple experiments to match the jam density and flow state that define low-medium-high congestion states.

4.2 System state for RL algorithm

At any step of the RL algorithm, the state of the system is represented by three elements:

(a) The average label of *NQ-index* values of the phases in signal timing plan

(b) The phase number with maximum queue length for the intersection

(c) The adjacent intersection number with maximum queue length

The state at step t for signal controller j can be represented as:

$$s_j^t = \left\{ \begin{array}{l} \text{average}\left(\Pi_p^t, \forall p \in P \right) \\ \arg\max_p \left(\Pi_p^t, \forall p \in P \right) \\ \arg\max_{\tilde{j}} \left(\Omega_{\tilde{j}}^t, \forall \tilde{j} \in \Gamma(j) \right) \end{array} \right\} \qquad (6)$$

An example for the state: {medium, phase 3, adjacent intersection no. 104} is interpreted as:

a) The intersection has a medium congestion label in terms of *NQ-index*.

b) Phase 3 corresponds to the maximum queue length.

c) Intersection 104 (straight northbound neighbor) has the maximum queue length.

Where,

$P =$ Set of phases in the signal timing plan for $\text{int}\,\text{er}\,\text{sec}\,\text{tion}\; j.$
$\Gamma(j) =$ The set of adjacent intersection for intersection $j.$

4.3 Action selection strategies

A signal controller agent takes action by switching on one of the phases in the timing plan. We do not restrict the sequence of the phases. However, the signal scheme imposes maximum and minimum green constraints. Flexible sequence in signal timing plan has been implemented in real world signalized intersections and adopted in the previous literature as well. Action selection strategy in RL algorithms involves an art of balancing exploration and exploitation. Actions can be entirely *greedy*—selecting the action with the maximal reward or can be exploratory by selecting random actions with assigned probability values. Sutton & Barto (1998) suggests two potential methods for action selection: (a) $\varepsilon - \text{greedy}$ method: Each agent behaves greedily in most cases by choosing the action that offers the maximum reward. However, at some cases it chooses a random action with a probability of ε assigned beforehand. The advantage of $\varepsilon - \text{greedy}$ methods over the greedy methods is highly dependent on the type of problem. For instance, with higher variance in the reward values the $\varepsilon - \text{greedy}$ methods might perform better, and (b) Softmax method: One limitation with the $\varepsilon - \text{greedy}$ method is that it gives equal priority to all actions while exploring. It is possible to choose the worst action instead of choosing the next best action. To resolve this, Softmax algorithms vary the action probabilities as a graded function of estimated value. Although, the greedy action has the highest selection probability the other are ranked and weighted according to the value estimates. In general, Gibbs or Boltzman distribution is used to define the probability. The probability for choosing action a in state s,

$$P(a\,|\,\text{state}=s) = \frac{\exp\left(\dfrac{Q(s,a)}{\tau}\right)}{\sum_{b=1}^{\text{all actions}} \exp\left(\dfrac{Q(s,b)}{\tau}\right)} \quad (7)$$

τ = Positive parameter called the *temperature*

Higher values for the temperature can make the probability of choosing any of the actions nearly equal. On the other hand, lower value of the temperature will create a higher difference in the action selection probabilities.

Another commonly used action strategy is the combination of the above mentioned strategies that is referred to as $\varepsilon-\text{Softmax}$. The agent behaves greedily with the probability of $(1-\varepsilon)$ and the rest of the cases it selects an action using the probability computed from Softmax selection process.

4.4 Reward Function

Three separate reward functions have been used: Queue length (R1), average delay experienced by the intersection since previous action (R2), and *NQ-index* (R3). In addition, we propose the multi reward structure that defines queue length as reward at free flow, average delay as reward over the time interval at medium level congestion, and *NQ-index* as reward at near saturated condition.

4.4.1 Multi-reward structure

The multi reward structure dynamically changes the reward function type based on the traffic congestion in real time. We consider the three categories of congestion states: (a) free flow to low congestion, (b) low to medium congestion and (c) medium congestion to high congestion (saturated condition). The algorithm identifies the congestion state in real time and uses the proper reward function in response. This research defines queue length as reward at free flow (to reduce the number of stops), average delay as reward over the time interval at medium level congestion, and *NQ-index* as reward at near saturated condition (to avoid the gridlock and spill back condition).

5. ALGORITHM DESCRIPTION

Like most RL based schemes, the proposed algorithm has two phases: learning phase and implementation phase. The learning takes place before the implementation. During the learning phase the agents update the state-action value through interacting with the environment. Balancing the exploration and exploitation is important at this phase. Initially, the algorithm starts with using higher probability for exploration. Then, gradually the value is decreased and at the end of the learning phase we implement the Softmax method. During the implementation period, the algorithm emphasizes on exploitation with very small value.

5.1 Terminologies

ρ = The average reward per time step.

$Q(s,a)$ = The value of state-action pair (s,a).

$\mathrm{r}(\mathrm{s},\mathrm{a},\mathrm{s}') =$ Observed reward when the agent takes action a in state s, and moves to state s'.

$\alpha^{(k)}$ = Learning rate for the $Q-$values (scalar) at $k-th$ iteration. $\beta^{(k)}$ = Learning rate for the average reward at step, k.

N = Maximum number of iterations allowed in the learning phase. γ = Discount factor for reward value.

5.2 RMART description

RMART is a novel temporal-difference based method where the value functions are defined with respect to the average expected reward. At any time step under the policy κ we define the average expected reward as:

$$\rho^{\kappa} = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} E_{\kappa}(r_t) \qquad (8)$$

Because RMART follows ergodic process, the average expected reward ρ^{κ} does not depend on the initial state. The long term average value should be identical for any initialized state. RMART involves transient due to which from some states better-than-average rewards can be received for a while and from other worse-than-average rewards are received. One clear distinction from other temporal difference technique is the use of relative value functions. It can be shown that as the discount factor approaches to one, the value function by discounted technique i.e., Q-learning approaches the differential value function by average reward technique i.e., RMART (Tsitsikilis & Roy, 2002). Figure 1 and 2 present the pseudo codes for Q-Learning and SARSA, and RMART.

6. IMPLEMENTATION AND NUMERICAL RESULTS

The RL-based algorithms are implemented in VISSIM using COM interface. Fig. 3 shows two networks: network-I (inside the rhombus) with eight junctions and network-II, with 18 junctions. VISSIM follows the psychophysical car-following models (Wiedemann74 and Wiedemann99) by Wiedemann (1974) and its variants developed later. Rakha & Gao (2010) describe methodologies to calibrate VISSIM for experiments. In our case, the networks are hypothetical with assumed geometry and flow conditions that resemble real world traffic networks. Therefore, we do not have to calibrate using field data. Further, we assume the same set of parameters for all simulation scenarios for all algorithms to have consistency in the results. For instance, RMART and Q-Learning have identical set of car following parameters.

Congestion level variation at intersection level

Algorithms are evaluated at three different congestion levels: low, medium, and high (near saturation). The trip rates for distinct origin-destination pairs are varied to create low to high congestion level. However, this is not the exact representation of the congestion experienced by the intersections. Two intersections can experience varying level of congestion state, even though the demand (network congestion level is same). Intersection-8 (*IS-8*) and intersection-11(*IS-11*) in figure 3 experience different patterns of congestion, although the network congestion level is same. Table 1 shows the distribution of experienced states for these intersections.

6.1 Statistical tests for the sample obtained from simulation

We obtain a sample from 10 simulation runs (for each scenario) in VISSIM using a different random seed each time. The sample size is N = 10 with unknown standard deviation and accordingly, we use the Student t distribution for tests. First, we determine the mean value of the performance metrics (i.e., travel time, delay, and stops) at 95% confidence interval. The resulting values indicate the range of population mean at desired confidence interval (in our case 95%). For instance, the system delay of adaptive controller (Table 1) $156.2<\mu<172.7$ indicates that we are 95% confident that the population mean for the system delay with adaptive controller lies between 156.2 × 104 and 172.7 × 104 seconds. Second, we test the validity of the claim that RMART performs better than other controller using statistical tests. For instance, with a predefined scenario we test whether the mean values of system delay for adaptive controller and RMART controller are significantly (statistically) different. Table 2 shows the results for high demand at network-II.

6.2 Performance comparison: Average Delay (H1)

Average delay, stopped delay, number of stops and network wide delay are chosen as the measures of effectiveness (MOE). Table 3 shows the sample comparison of average delay for different RL algorithms with different reward functions at different congestion levels for network-I. The results are reported for intersection-8, however the trend is same for other intersections. At low congestion, Q-learning exhibits best performance with R1 and R2, and RMART performs better only with R3. The results are similar for both network-I and network-II. Due to space limitation, we report the results for network-I only. At high congestion, RMART outperforms all other algorithms with all types of reward functions. We made the following conclusions:

a) SARSA performs worse than the other two algorithms

b) At low congestion, Q-learning is a good choice. Note that, *NQ-index* (R3) is a more appropriate reward, when the congestion level is higher, aiming at avoiding gridlock, however not directly related with delay.

c) At high congestion, RMART is the best choice that yields the minimal average delay.

6.3 Performance comparison: Stopped Delay (H1)

Table 4 reports the comparison of intersection delay along with percentage of improvement compared to fixed signal control for intersection-8. At low congestion, Q-learning performs best with R1 and R2. At high congestion, RMART yields the best results with all reward functions. Similar to previous results, RMART is the best choice to reduce stopped delay at signalized intersection at high congestion level of the network.

6.4 Comparison of system wide performance (H1)

Table 5 compares the system delays for different algorithms. Q-learning and RMART perform significantly better than the fixed control at all congestion levels. SARSA has better performance at medium and high congestion levels. At high congestion level, RMART shows the highest percentage of improvement compared to fixed control.

6.5 Comparison of multi-reward algorithms (H3)

Table 6 compares the results for algorithms using multi-reward structure and single reward structure. The results from the multi-reward case are compared with the best and worst cases from single reward algorithms. For instance, the best case of a single reward Q-learning algorithm is the algorithm-reward combination that yields the minimum delay. Table 6 shows that, the multi-reward scheme only for RMART performs better than the worst case in single reward in most cases. The performances are not improved for other cases. The multi-reward structure requires comprehensive analysis before reaching any insightful conclusion. Table 5 presents a single test case only for network-I.

6.6 Comparison with real time adaptive control (H1)

The RL algorithms are compared with a real time adaptive signal control, namely the Enhanced-Longest-Queue-First (ELQF) algorithm. The ELQF algorithm is based on a routing algorithm in data communication network and has been implemented by researchers in traffic control context (Arel et al., 2010; Wunderlich et al., 2008). Wunderlich et al. (2008) proposed a variant of this algorithm, namely the Maximal Weight Matching algorithm. For the test purpose, we modified the algorithm to make it more efficient. The changes include provision for minimum and maximum green in the signal-timing plan and adjusting for repetitive phases for the case when a particular approach is highly congested compared to all other approaches. The LQF algorithm uses real time information to make signal control decision. ELQF activates the phase with longest queue size within the defined constraints. Queue size is defined as the number of stopped vehicles at the intersection on red. Table 7 reports the comparison of RL algorithms with LQF algorithm. The RL algorithms perform better than the LQF algorithm in terms of both average delay and stopped delay (not reported due to space limitation). The results are similar for both intersection-3 and intersection-6. Note that both RL and ELQF algorithms use real time traffic information to make signal control decision. The key difference is that, the RL based algorithm have the notion of learning i.e., the controllers learn to make the better decision with training. Similar results are found for network-II.

6.7 Value of information sharing among neighbors (H2)

Sharing traffic information among neighborhood controllers has been mentioned as one of the distinct feature of the proposed RL algorithm in this research. To justify the impact of information sharing we compare the results from two test cases: with and without information sharing. Table 8 shows the comparison results for different congestion levels. For Q-Learning, we see improvements at all congestion levels. For RMART, we see improvement for higher congestion and for SARSA negligible deterioration is observed at higher congestion level. It can be concluded that, sharing of neighborhood information helps to improve the performance of RL algorithms. Table 9 shows the results for network-II.

7. EMISSION COMPARISON USING MOVES2010 (H4)

To evaluate the control algorithms from sustainability consideration, we estimate and compare emissions (CO, CO_2, NO_X, VOC, PM_{10}) for the major seven roads. Since all the roads are two-way, in total we have 14 road links. MOtor Vehicle Emission Simulator (MOVES2010), developed by the U.S. EPA (EPA, 2012), is used to estimate the emission. EPA has regulated to use MOVES2010 for emission conformity analysis (except, California) for states in the U.S. MOVES2010 have the capability of estimating emissions with time dependent speed profiles (Lin et al., 2011; Xie et al., 2012). We simulate the morning peak hour (8:00 am to 9:00 am) with higher level of congestion assuming the geographical and meteorological details of Tippecanoe County, Indiana for the year 2012. Only passenger cars with Gasoline type of fuel are used in the analysis. Although the values obtained are only for an hour, generally the analysis is done for the entire day or for the week. If we assume 4 hours of peak congestion each day and assume the conditions prevail as we simulate through MOVES2010, then we see significant reduction in the emission level for a typical work week. Similar results are found for network-II.

8. CONCLUSIONS AND FUTURE WORKS

The research presents and implements RL based signal control algorithm, namely RMART, which adapts with the traffic dynamics through learning from the stochastic environment. Our empirical analysis shows that we cannot reject the research hypotheses H1, H2, and H4. Table II shows that, the total system delay and stopped delay are lower for RMART compared to fixed control and adaptive control (i.e., the variant of longest-queue-first). The obtained results are statistically significant ($p < 0.001$). Adaptive controllers are quite different from the RL based controllers in terms of principle and implementation (Mirchandani & Head, 2001; Abdulhai et al., 2003). The results from our empirical tests show that, learning based controls can perform better than the adaptive control. In addition, the RL controllers perform much better than the fixed timing plan. This implies that, learning is a useful and potential feature in the real time signal control algorithms and can improve the performance of the controllers (H1).

The inclusion of neighborhood information sharing in the RL algorithms is found to improve the performance (H2) in most cases for the RL algorithms (Table 8 and 9 report the results). Information sharing facilitates the controller to make decisions based on the overall congestion states of its neighborhoods. Without neighborhood information, the controller makes decision based on local information. However, the state of traffic flow of the neighbors affects the future traffic conditions (simplest example would be the arrival rate at the downstream intersection). This is particularly important at higher congestion level (near saturation).

To assess the benefits in terms of reducing vehicular emissions we compare the levels of emissions of RMART with other algorithms (see Table 10 for the results). The levels of emissions (CO, NO_X, PM_{10}, CO_2, VOC) are significantly lower for the network when RMART is implemented (H4). The reduction in number of stops and average stopped delays at the intersections with RMART can be the major reasons behind lower level of emissions. Further, we also observe lower energy consumption for RMART compared to other algorithms. Therefore, RMART performs better than the other algorithms in terms of reducing vehicular emissions and energy consumption for the entire network.

Note that, our test cases do not indicate a better performance of the multi-reward structure (H3). The random nature of on-road traffic (e.g., drastic changes from low congestion to high congestion due to an incident) is the key motivation behind the implementation of multi-reward structure. However, the multiple reward structure did not have better performance in our tests compared to the other learning algorithms. This phenomenon can occur due to the basic principle of the algorithms, i.e., learning over time. To test the algorithms, we applied varying demand over time and report the results for the last 15 minutes of the simulation. The algorithms with single reward structure learn over the simulation period. Accordingly, these algorithms are also able to take the best decision even in the most random environment (sudden change from low to high congestion). As a result, we do not observe significant improvements for the multi-reward based RL algorithm.

To conclude, the RMART algorithm as illustrated by the results has shown higher potential to reduce delay at highly congested states. In addition, this research shows the advantages of information sharing and potential of emissions reduction of the RL based algorithms.

8.1 Limitations and future directions

This research reports the results from hypothetical networks. We designed the experiments (car-following parameters, lane changing behavior, and so on) so that the experiments are as close as the real world traffic networks. However, to strengthen our conclusions we must test the algorithms with real world networks. Further, the complexity of the algorithm should be tested extensively with different parameters of the algorithms. The learning rate parameter and discount factor are assumed arbitrarily and a sensitivity analysis can provide us with useful information. We plan to implement the algorithm on a larger network to show the benefits of the learning to facilitate sustainably mobility in traffic networks.

REFERENCES

Abdoos, M., Mozayani, N., & Bazzan, A. L. C. (2011). Traffic light control in non-stationary environments based on multi agent Q-learning. Proc., *Intelligent Transportation Systems (ITSC)*, 14th International IEEE Conference on, 1580--1585.

Abdulhai, B., & Kattan, L. (2003). Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*, 30(6), 981--991.

Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement Learning for True Adaptive Traffic Signal Control. *Journal of Transportation Engineering*, 129(3), 278--285.

Arel, I., Liu, C., Urbanik, T., & Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *Intelligent Transport Systems, IET*, 4(2), 128--135.

Balaji, P. G., German, X., & Srinivasan, D. (2010). Urban traffic signal control using reinforcement learning agents. *Intelligent Transport Systems, IET,* 4(3), 177--188.

Bazzan, A. L. C. (2005). A Distributed Approach for Coordination of Traffic Signal Agents. *Autonomous Agents and Multi-Agent Systems*, 10(2), 131--164.

Bazzan, A. L. C., de Oliveira, D., & da Silva, B. C. (2010). Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence*, 23(4), 560--568.

Bingham, E. (2001). Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research*, 131(2), 232--241.

Boillot, F., Midenet, S., & Pierrelée, J.-C. (2006). The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, 14(1), 18--38.

Chen, L.-W., Sharma, P. & Tseng, Y.-C. (2013). Dynamic Traffic Control with Fairness and Throughput Optimization Using Vehicular Communications. *Sel. Areas Commun. IEEE J.* 31(9)**,** 504–512.

Desjardins, C., & Chaib-draa, B. (2011). Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4), 1248--1260.

Diakaki, C., Papageorgiou, M., & Aboudolas, K. (2002). A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2), 183--195.

El-Tantawy, S., & Abdulhai, B. (2010). Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers (MARLIN-OTC). *Transportation Letters: the International Journal of Transportation Research*, 2(2), 89--110.

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Design of Reinforcement Learning Parameters for Seamless Application of Adaptive Traffic Signal Control. *Journal of Intelligent Transportation Systems*, DOI: 10.1080/15472450.15472013.15810991.

EPA (2013). U. S. EPA, National Emissions Inventory (NEI) Air Pollutant Emissions Trends Data (1970-2012). URL:http://www.epa.gov/ttn/chief/trends/index.html, accessed: July, 2012.

EPA (2006). U. S. EPA, Greenhouse Gas Emissions from the US Transportation Sector, 1990–2003. Transportation GHG Emissions Report, Office of Transportation and Air Quality.

EPA (2012). US Environmental Protection Agency, MOVES2010b Technical Guidance.

Farges, J., Henry, J., & Tufal, J. (1983). The PRODYN real-time traffic algorithm. *Proceedings of the fourth IFAC symposium on transportation systems*, 307--312.

Feng, Y., Head, K. L., Khoshmagham, S. & Zamanipour, M. (2015). A real-time adaptive signal control in a connected vehicle environment. *Transp. Res. Part C Emerg. Technol.* 55, 460–473.

Florin, R. & Olariu, S. (2015) A survey of vehicular communications for traffic signal optimization. *Veh. Commun.* 2(2), 70–79.

Gartner, N. H. (1983). OPAC: A DEMAND-RESPONSIVE STRATEGY FOR TRAFFIC SIGNAL CONTROL. *Transportation Research Record*, 906, 75--81.

Genders, W., Razavi, S.N, (2015). Impact of Connected Vehicle on Work Zone Network Safety through Dynamic Route Guidance. J. Comput. Civ. Eng. 4015020. doi:10.1061/(ASCE)CP.1943-5487.0000490

Gosavi, A. (2003). Simulation-Based Optimization: Parametric Optimization Techniques & Reinforcement Learning. Springer, United States.

.Houli, D., Zhiheng, L., & Yi, Z. (2010). Multi-objective reinforcement learning for traffic signal control using vehicular ad hoc network. *EURASIP J. Adv. Signal Process*, 1--7.

Hu, T., & Chen, L. (2012). Traffic Signal Optimization with Greedy Randomized Tabu Search Algorithm. *Journal of Transportation Engineering*, 138(8), 1040--1050.

Hu, J., Park, B. B. & Lee, Y.-J. (2015) .Coordinated transit signal priority supporting transit progression under Connected Vehicle Technology. *Transp. Res. Part C Emerg. Technol.* 1, 393–408

Hunt, P. B., Robertson, D. I., Bretherton, R. D., & Royle, M. C. (1982). The SCOOT On-line Traffic Signal Optimisation Technique. *Traffic Engineering and Control*, 23(4), p. 190--192.

Huang, S., Sadek, A. W., & Zhao, Y. (2012). Assessing the Mobility and Environmental Benefits of Reservation-Based Intelligent Intersections Using an Integrated Simulator. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3), 1201--14.

Ishihara, H., & Fukuda, T. (2001). Traffic signal networks simulator using emotional algorithm with individuality. *Intelligent Transportation Systems*, 2001. Proceedings. 2001 IEEE, 1034--1039.

Kuyer, L., Whiteson, S., Bakker, B., & Vlassis, N. (2008). Multiagent Reinforcement Learning for Urban Traffic Control Using Coordination Graphs. *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, eds., Springer Berlin / Heidelberg, 656--671.

Kwak, J., Park, B., & Lee, J. (2012). Evaluating the impacts of urban corridor traffic signal optimization on vehicle emissions and fuel consumption. *Transportation Planning and Technology*, 35(2), 145--160.

Lee, J., & Park, B. (2012). Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1), 81--90.

Li, Y., & Mueck, J. (2010). A Recurrent Neural Network Approach to Network-wide Traffic Signal Control. Transportation Research Board 89th Annual Meeting, Washington,DC,USA, 23p.

Li, Y., Yang, J., Guo, X., & Abbas, M. M. (2011). Urban Traffic Signal Control Network Partitioning Using Self-Organizing Maps. Transportation Research Board 90th Annual MeetingTransportation Research Board, 20p.

Lin, J., Yi-Chang, C., Vallamsundar, S., & Song, B. (2011). Integration of MOVES and dynamic traffic assignment models for fine-grained transportation and air quality analyses. *Proc., Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, 176--181.

Lowrie, P. R. (1982). SCATS:The Sydney coordinated adaptive traffic system principles, methodology, algorithms. *Proceedings of the IEE international conference on road traffic signaling*, 66--70.

Mauro, V., & Taranto, D. (1989). UTOPIA. Proceedings of the sixth IFAC/IFIP/IFORS symposium on control, computers, communications on transportation, 245--252.

Medina, J., & Benekohal, R. (2011). Reinforcement Learning Agents for Traffic Signal Control in Oversaturated Networks. Transportation and Development Institute Congress 2011, American Society of Civil Engineers, 132--141.

Mikami, S., & Kakazu, Y. (1994). Genetic reinforcement learning for cooperative traffic signal control." Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, 1, 223--228.

Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6), 415--432.

Ngai, D. C. K., & Yung, N. H. C. (2011). A Multiple-Goal Reinforcement Learning Method for Complex Vehicle Overtaking Maneuvers. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2), 509--522.

NTOC Report (2012). The National Transportation Operations Coalition (NTOC).2012 National Traffic SignalReport Card – Executive Summary, 2012.

Prashanth, L. A., & Bhatnagar, S. (2011). Reinforcement Learning With Function Approximation for Traffic Signal Control. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2), 412--421.

PTV (2012), PTV AMERICA.VISSIM Version 5.30-04 User Manual. Innovative Transportation Concepts. PTV Planung Transport Verkehr AG, Karlsruhe, Germany.

Rakha, H., Gao, Y., & Center, M.-A. U. T. (2010). Calibration of steady-state car-following models using macroscopic loop detector data. Mid-Atlantic Universities Transportation Center.

Schrank, D., Eisele, B., & Bak, J. (2015). 2015 Urban Mobility Scorecard. Texas A&M Transportation Institute and INRIX.

Srinivasan, D., Min Chee, C., & Cheu, R. L. (2006). "Neural Networks for Real-Time Traffic Signal Control. *Intelligent Transportation Systems, IEEE Transactions on*, 7(3), 261--272.

Stevanovic, A., Stevanovic, J., Jolovic, D., & Nallamothu, V. (2012). Retiming Traffic Signals to Minimize Surrogate Safety Measures on Signalized Road Networks. Transportation Research Board 91st Annual Meeting Transportation Research BoardWashington, DC, USA, 15p.

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction, Cambridge Univ Press.

Tadepalli, P., & Ok, D. (1998). Model-based average reward reinforcement learning. *Artificial Intelligence*, 100(1–2), 177--224.

Tao, L., Dongbin, Z., & Jianqiang, Y. (2008). Adaptive Dynamic Programming for Multi-intersections Traffic Signal Intelligent Control. Proc., Intelligent Transportation Systems, ITSC 2008. 11th International IEEE Conference on, 286--291.

Thorpe, T. (1997). Vehicle traffic light control using SARSA. Master’s Project Rep., Computer Science Department, Colorado State University, Fort Collins, Colorado.

Tsitsiklis, J. N., & Roy, B. V. (2002). On Average Versus Discounted Reward Temporal-Difference Learning. *Machine Learning*, 49(2-3), 179--191.

Watkins. C. (1989). Learning from delayed rewards. PhD Thesis, University of Cambridge, England.

Watkins, C. & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3), 279--292.

Wei-Song, L., & Jih-Wen, S. (2011). “Metro Traffic Regulation by Adaptive Optimal Control.” Intelligent Transportation Systems, IEEE Transactions on, 12(4), 1064--1073.

Wiedemann, R. (1974). Simulation des Strassenverkehrsflusses. Schriftenreihe des Instituts für Verkehrswesen Heft, Universität Karlsruhe, Ph.D. Thesis.

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

Wunderlich, R., Cuibi, L., Elhanany, I., & Urbanik, T. (2008). A Novel Signal-Scheduling Algorithm With Quality-of-Service Provisioning for an Isolated Intersection. *Intelligent Transportation Systems, IEEE Transactions on*, 9(3), 536--547.

Xie, Y. (2007). Development and evaluation of an arterial adaptive traffic signalcontrol system using reinforcement learning, Dissertation in Civil Engineering.TexasA&M University: College Station.

Xie, Y., Chowdhury, M., Bhavsar, P., & Zhou, Y. (2012). An integrated modeling approach for facilitating emission estimations of alternative fueled vehicles. *Transportation Research Part D: Transport and Environment*, 17(1), 15--20.

Xiaohui, D., Chi-Kwong, L., & Rad, A. B. (2005). An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control. *Intelligent Transportation Systems, IEEE Transactions on*, 6(3), 285--293.

Table 1 Description Of congestion variation for experiments

Experienced State	Q-learning		SARSA		RMART	
	IS-8	***IS-11***	***IS-8***	***IS-11***	***IS-11***	***IS-11***
Low congestion state (%)	45.74	45.74	22.87	37.67	27.81	78.92
Medium congestion state (%)	39.91	49.78	45.3	57.85	43.49	20.18
High congestion state (%)	14.35	4.48	31.83	4.48	2	0.90

Table 2 Comparing System performance (μ = population mean value) at high demand for network-II (*All the improvements are tested and we find $p < 0.001$ with $n = 10$*)

Performance metrics		**Fixed control**	**Adaptive control**	**RMART**
System delay	Mean (in seconds × 10^4)	173	164 (4.9% reduction)	130 (24.6% reduction)
	Population mean range (at 95% confidence)	$167<\mu<178$	$156<\mu<172$	$121<\mu<139$
Stopped delay	Average stopped delay (in seconds × 10^4)	118	114 (3.14% reduction)	85 (27.9% reduction)
	Population mean range (at 95% confidence)	$114<\mu<122$	$106<\mu<121$	$77<\mu<92$

Table 3 Average Delay (in Seconds) Comparison (Network-I) at Intersection-8

Reward definition	Congestion level	Fixed control	Q-Learning	SARSA	RMART
Queue length (R1)	Low	155	135	142	139
		Decrease by	13%	8%	10%
	Medium	236	193	191	179
		Decrease by	18%	19%	24%
	High	353	265	291	227
		Decrease by	25%	18%	36%
Average delay (R2)	Low	155	136	149	145
		Decrease by	12%	4%	7%
	Medium	236	171	187	195
		Decrease by	28%	21%	17%
	High	353	297	290	277
		Decrease by	16%	18%	22%
Residual queue (R3)	Low	155	140	148	139
		Decrease by	10%	5%	10%
	Medium	236	167	201	176
		Decrease by	29%	15%	25%
	High	353	260	276	226
		Decrease by	26%	22%	36%

Table 4 Comparison of Stopped Delay (In Seconds) for (Network-I)

Reward definition	Congestion	Fixed control	Q-Learning	SAR	RMART
Queue length (R1)	Low	113	91	112	96
		Decrease by	20%	1%	15%
	Medium	183	137	156	123
		Decrease by	25%	15%	33%
	High	284	189	208	156
		Decrease by	34%	27%	45%
Average delay (R2)	Low	113	94	105	104
		Decrease by	17%	7%	8%
	Medium	183	118	131	140
		Decrease by	36%	28%	24%
	High	284	216	207	201
		Decrease by	24%	27%	29%
Residual queue	Low	113	95	93	96
		Decrease by	16%	18%	15%
	Medium	183	110	118	121
		Decrease by	40%	36%	34%
	High	284	182	216	151
		Decrease by	36%	24%	47%

Table 5 Comparisons of System Delay In Seconds × 10^3 (Network-I)

Congestion level	Fixed Control		Reduction* (%)		Reduction* (%)		Reduction* (%)
Low	235	229	2%	226	4%	235	0%
Medium	434	369	15%	376	14%	402	7%
High	774	622	20%	619	20%	638	18%

* Delay reduction with respect to Fixed Control

Table 6 Comparison For Multi-Reward Structure

Algorithm	Congestion	Average delay* (Multi reward)	Average delay*(Best-sing	Change from best case	Average delay* (Worst-single reward)	Change from worst case
Q-Learning	Low	140	135	-4%	140	0
	Medium	171	167	-2%	193	11%
	High	338	260	-30%	297	-11%
SARSA	Low	149	148	0%	149	0%
	Medium	201	187	-7%	201	0%
	High	352	276	-27%	291	-20%
RMART	Low	140	139	0%	145	3%
	Medium	159	176	10%	179	11%
	High	233	226	-3%	277	16%

Table 7 Comparison with Adaptive (ELQF) Controllers

Congestion	**Average Delay (seconds)**			
	Fixed Timing Plan	**Adaptive (ELQF)**	**Off-Policy**	**RMART**
Intersection-3				
Low	144	177	132	132
Medium	203	207	160	175
High	357	294	270	232
Intersection-6				
Low	155.13	198.338	145.86	134.53
Medium	236.42	223.947	176.79	177.42
High	353.47	279.32	263.95	228.06

Table 8 With And Without Information Sharing (Network-I)

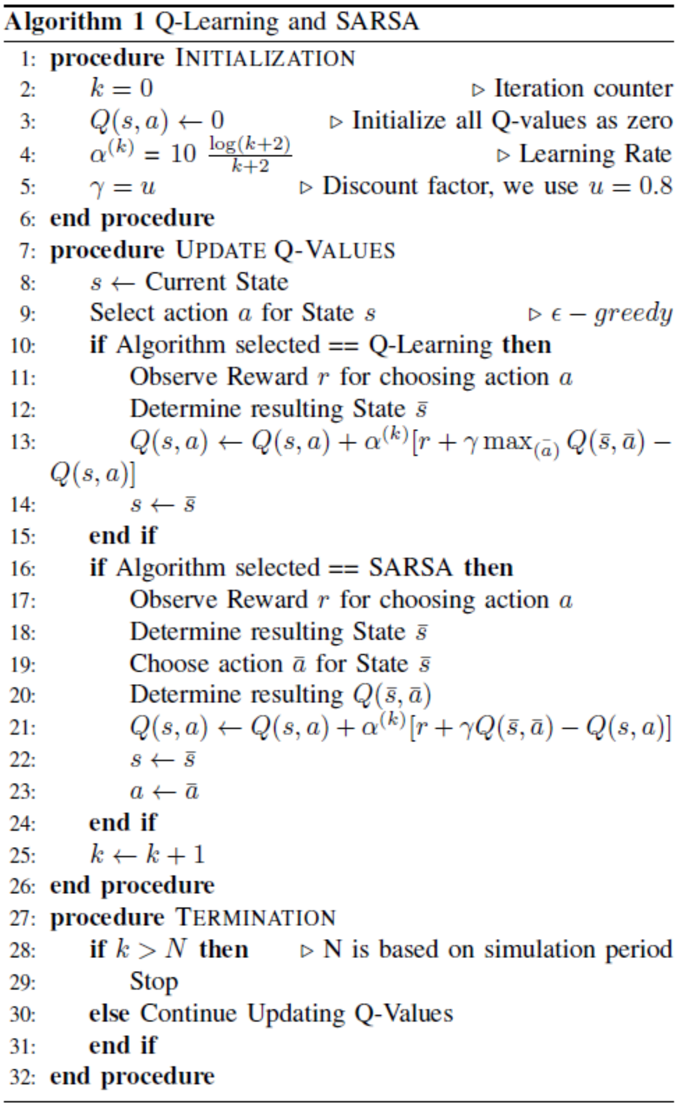
Test Case		Average Delay (in seconds)			Stopped Delay (in seconds)		
		With Info.	Without Info.	Delay Reduction	With Info.	Without Info.	Delay Reduction
Off-policy (Q-Learning)	Low	132.39	133.58	0%	92.80	93.50	0%
	Medium	160.13	168.56	5%	111.30	118.29	6%
	High	270.12	270.37	0%	204.70	204.06	0%
RMART	Low	131.89	131.89	0%	93.09	93.09	0%
	Medium	174.98	172.52	-1%	123.81	121.31	-2%
	High	231.55	284.15	23%	168.42	218.03	30%
On Policy (SARSA)	Low	152.78	152.78	0%	111.43	111.43	0%
	Medium	184.55	192.88	5%	133.75	140.46	5%
	High	320.82	319.31	0%	247.58	245.92	0%

Table 9 With And Without Information Sharing (Network-II)

	Average delay			Stopped delay		
Intersection	Without Info.	With Info.	Reduction	Without Info.	With Info.	Reduction
1	94.12	50.3288	-46.53%	55.6178	21.5541	-61.25%
2	22.5464	22.0252	-2.31%	11.2202	10.588	-5.63%
3	495.967	337.01	-32.05%	402.064	272.205	-32.30%
4	41.9977	44.5992	6.19%	17.8508	19.3465	8.38%
5	379.611	378.179	-0.38%	295.69	286.574	-3.08%
Sum	1034.2421	832.1422	-19.54%	782.4428	610.2676	-22.00%

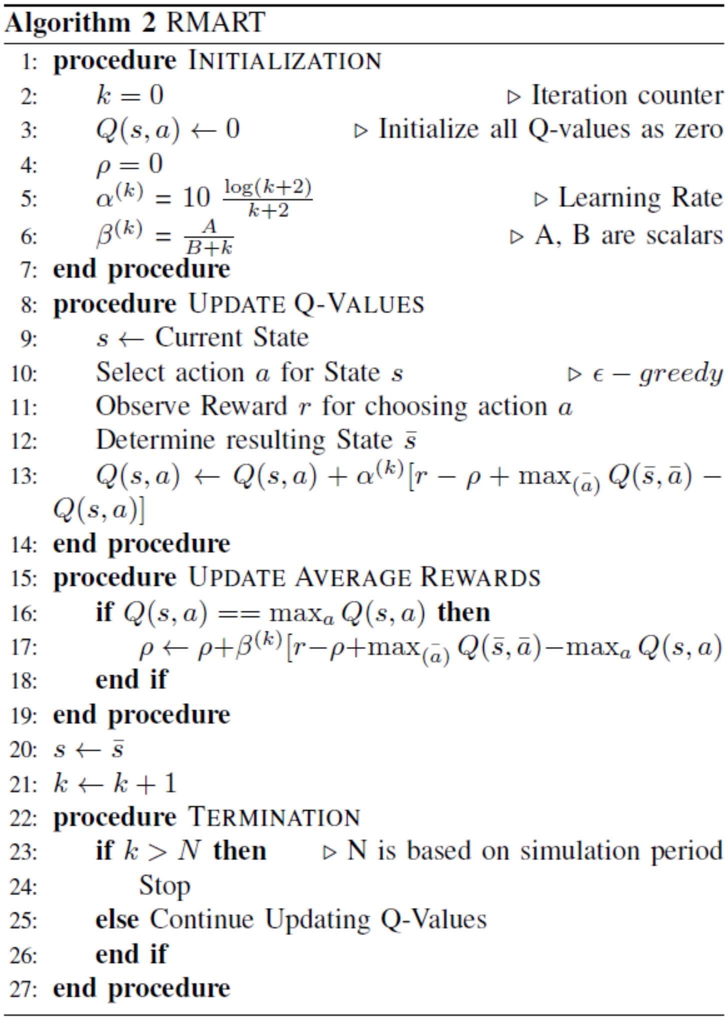
Table 10 Comparison of Emission for Different Algorithms (Network-I)

Pollutant	Emissions			
	Fixed Timing Plan	Adaptive (ELQF)	Off-Policy	RMART
CO (g/hour)	10683	4318	4394	4247
Weekly total (g)	84935	86357	87883	213656
NOX (g/hour)	732	312	308	302
Weekly total (g)	14632.34	6246.612	6155.406	6037.2
VOC (g/hour)	264	108	106	104
Weekly total (g)	5290	2151	2112	2079
PM10 (g/hour)	38.1	16.33	16.2	15.78
Weekly total (g)	762.03	326.55	320.23	315.6072
CO2 (g/hour)	841070.6	345631.6	339527.5	334037.4
Weekly total (g)	16821412	6912632	6790550	6680748
Energy consumption	117032	48093	47244	46480
Equivalent fuel (in gallons per week)	1777	730	717	705



```
Algorithm 1 Q-Learning and SARSA
procedure INITIALIZATION
    $k = 0$                                        ▷ Iteration counter
    $Q(s,a) \leftarrow 0$                          ▷ Initialize all Q-values as zero
    $\alpha^{(k)} = 10\,\frac{\log(k+2)}{k+2}$     ▷ Learning Rate
    $\gamma = u$                                   ▷ Discount factor, we use $u = 0.8$
end procedure
procedure UPDATE Q-VALUES
    $s \leftarrow$ Current State
    Select action $a$ for State $s$                ▷ $\epsilon - greedy$
    if Algorithm selected == Q-Learning then
        Observe Reward $r$ for choosing action $a$
        Determine resulting State $\bar{s}$
        $Q(s,a) \leftarrow Q(s,a) + \alpha^{(k)}[r + \gamma \max_{(\bar{a})} Q(\bar{s},\bar{a}) - Q(s,a)]$
        $s \leftarrow \bar{s}$
    end if
    if Algorithm selected == SARSA then
        Observe Reward $r$ for choosing action $a$
        Determine resulting State $\bar{s}$
        Choose action $\bar{a}$ for State $\bar{s}$
        Determine resulting $Q(\bar{s},\bar{a})$
        $Q(s,a) \leftarrow Q(s,a) + \alpha^{(k)}[r + \gamma Q(\bar{s},\bar{a}) - Q(s,a)]$
        $s \leftarrow \bar{s}$
        $a \leftarrow \bar{a}$
    end if
    $k \leftarrow k + 1$
end procedure
procedure TERMINATION
    if $k > N$ then                                ▷ N is based on simulation period
        Stop
    else Continue Updating Q-Values
    end if
end procedure
```

Figure 1 Pseudo-code for Q-learning and SARSA



```
Algorithm 2 RMART
procedure INITIALIZATION
    $k = 0$                                        ▷ Iteration counter
    $Q(s, a) \leftarrow 0$                         ▷ Initialize all Q-values as zero
    $\rho = 0$
    $\alpha^{(k)} = 10\,\frac{\log(k+2)}{k+2}$     ▷ Learning Rate
    $\beta^{(k)} = \frac{A}{B+k}$                  ▷ A, B are scalars
end procedure
procedure UPDATE Q-VALUES
    $s \leftarrow$ Current State
    Select action $a$ for State $s$               ▷ $\epsilon - greedy$
    Observe Reward $r$ for choosing action $a$
    Determine resulting State $\bar{s}$
    $Q(s, a) \leftarrow Q(s, a) + \alpha^{(k)}[r - \rho + \max_{(\bar{a})} Q(\bar{s}, \bar{a}) - Q(s, a)]$
end procedure
procedure UPDATE AVERAGE REWARDS
    if $Q(s, a) == \max_a Q(s, a)$ then
        $\rho \leftarrow \rho + \beta^{(k)}[r - \rho + \max_{(\bar{a})} Q(\bar{s}, \bar{a}) - \max_a Q(s, a)$
    end if
end procedure
$s \leftarrow \bar{s}$
$k \leftarrow k + 1$
procedure TERMINATION
    if $k > N$ then                                ▷ N is based on simulation period
        Stop
    else Continue Updating Q-Values
    end if
end procedure
```

Figure 2 Pseudo-code for RMART

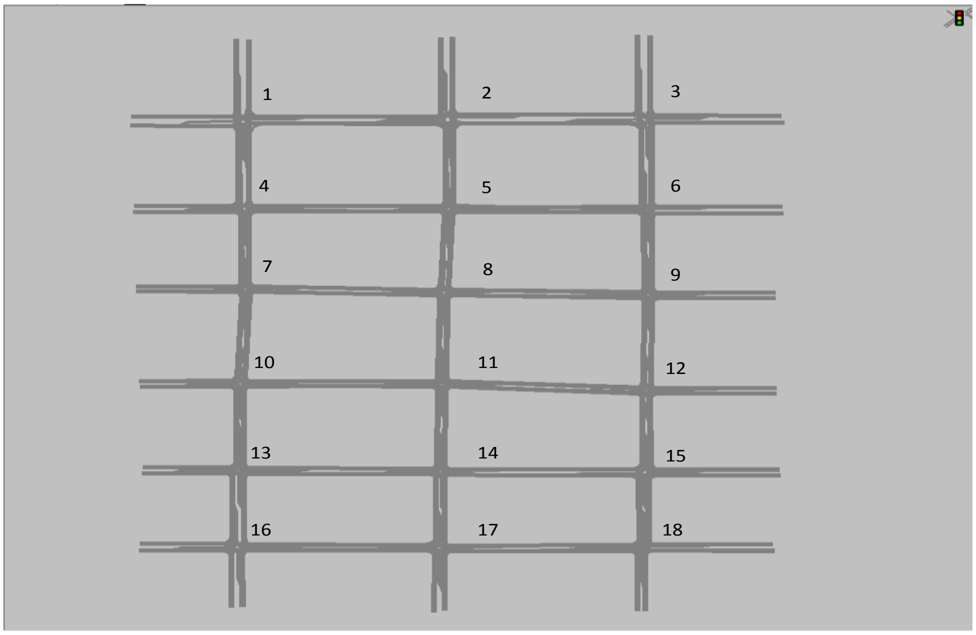


Figure 3 Test network for evaluating the signal control algorithms