Accident Analysis and Prevention 146 (2020) 105713

Contents lists available at ScienceDirect

Accident Analysis and Prevention

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

Self-learning adaptive traffic signal control for real-time safety optimization

Mohamed Essa*, Tarek Sayed

Department of Civil Engineering, University of British Columbia, 6250 Applied Science Lane, Vancouver, BC, V6T 1Z4, Canada

ARTICLE INFO

Keywords:
Adaptive traffic signal control
Real-time safety optimization
Connected vehicles
Reinforcement learning
Traffic simulation
Real-time safety models

ABSTRACT

Adaptive traffic signal control (ATSC) is a promising technique to improve the efficiency of signalized intersections, especially in the era of connected vehicles (CVs) when real-time information on vehicle positions and trajectories is available. Numerous ATSC algorithms have been proposed to accommodate real-time traffic conditions and optimize traffic efficiency. The common objective of these algorithms is to minimize total delay, decrease queue length, or maximize vehicle throughput. Despite their positive impacts on traffic mobility, the existing ATSC algorithms do not consider optimizing traffic safety. This is most likely due to the lack of tools to evaluate safety in real time. However, recent research has developed various real-time safety models for signalized intersections. These models can be used to evaluate safety in real time using dynamic traffic parameters, such as traffic volume, shock wave characteristics, and platoon ratio. Evaluating safety in real time can enable developing ATSC strategies for real-time safety optimization. In this paper, we present a novel self-learning ATSC algorithm to optimize the safety of signalized intersections. The algorithm was developed using the Reinforcement Learning (RL) approach and was trained using the simulation platform VISSIM. The trained algorithm was then validated using real-world traffic data obtained from two signalized intersections in the city of Surrey, British Columbia. Compared to the traditional actuated signal control system, the proposed algorithm reduces traffic conflicts by approximately 40 %. Moreover, the proposed ATSC algorithm was tested under various market penetration rates (MPRs) of CVs. The results showed that 90 % and 50 % of the algorithm's safety benefits can be achieved at MPR values of 50 % and 30 %, respectively. To the best of the authors' knowledge, this is the first self-learning ATSC algorithm that optimizes traffic safety in real time.

1. Introduction

Real-time optimization of traffic signals has recently received increasing interest among researchers and practitioners, especially with the availability of real-time traffic data from emerging technologies such as connected vehicles (CVs) (U.S. Department of Transportation, 2015) and innovative video detection techniques (Bramberger et al., 2004; Zhang et al., 2007; Redmon et al., 2016; Ke et al., 2018; Formosa et al., 2020). Over the past few decades, adaptive traffic signal control (ATSC) systems have shown considerable advances. Several ATSC algorithms have been developed and implemented (e.g., Sims, 1979; Hunt et al., 1981; Gartner, 1983; Head et al., 1992; Luyanda et al., 2003), while numerous have been proposed (e.g., Abdulhai et al., 2003; Camponogara and Kraus, 2003; Salkham et al., 2008; Balaji et al., 2010; Goodall et al., 2013; El-Tantawy et al., 2013; Guler et al., 2014; Yang et al., 2016; Shabestary and Abdulhai, 2018; Gong et al., 2019). The common objective of these algorithms is to accommodate real-time traffic conditions and optimize traffic efficiency by maximizing throughput capacity, minimizing traffic delay, and/or reducing queue lengths. Compared to the traditional fixed-time or actuated signals, ATSC algorithms have shown a significant improvement in traffic efficiency at signalized intersections.

However, despite the aforementioned mobility benefits, the safety impact of the existing ATSC algorithms remains unclear. Some studies showed that mobility-oriented ATSC algorithms can improve safety and significantly reduce traffic collisions (Fink et al., 2016; Ma et al., 2016; Khattak et al., 2018) or traffic conflicts (Stevanovic et al., 2011; Fyfe and Sayed, 2017). Meanwhile, other studies indicated that implementing ATSC algorithms either leads to insignificant reduction in traffic collisions (Dutta et al., 2010; Lodes and Benekohal, 2013) or increases traffic conflicts significantly and worsens traffic safety (Tageldin et al., 2014). This inconsistency in the safety impact of existing ATSC algorithms maybe related to that these algorithms do not consider optimizing traffic safety as a primary objective. More importantly, optimizing mobility does not necessarily mean optimizing safety (Sabra et al., 2010). For example, an ATSC algorithm might tend to minimize the total delay by generating many stops, each with a short duration. Although this might lead to improved mobility, generating

* Corresponding author.
E-mail addresses: m.a.essa@civil.ubc.ca (M. Essa), tsayed@civil.ubc.ca (T. Sayed).

https://doi.org/10.1016/j.aap.2020.105713
Received 15 March 2020; Received in revised form 20 July 2020; Accepted 29 July 2020
Available online 18 August 2020

many stops can increase the potential risk of collision and deteriorate safety.

A few studies (Sabra et al., 2013; Stevanovic et al., 2013, 2015) have attempted to optimize safety of signalized intersections using traffic simulation and the Surrogate Safety Assessment Model (SSAM) (Gettman et al., 2008). The safety optimization process comprises tuning various signal timing parameters (e.g., cycle length, offset, and phase change interval) to minimize the number of traffic conflicts. Multiple signal designs were tested offline and their corresponding safety levels were evaluated using SSAM. However, the optimization algorithms in these studies are not as practically effective as self-learning ATSC algorithms, in terms of responding instantaneously to real-time traffic changes and covering all possible traffic conditions. Besides, using SSAM to evaluate traffic safety has generally been criticized due to several concerns. First, vehicles in simulation models follow specific rules that aim to produce a crash-free environment. Using these safe-moving vehicles to evaluate conflicts and near-misses may lead to inaccurate results. Second, the SSAM results can vary significantly depending on the assumed values of the simulation model parameters and the approach used in modelling. Finally, unrealistic crashes and unusual movements are often recorded in traffic simulations, most likely due to an insufficient minimum gap size, a failure to yield to a priority rule, an abrupt lane change of a vehicle, or an irregular queuing up at left/right turn bay tapers (Gettman and Head, 2003a; Gettman et al., 2003a,b; 2008; Essa and Sayed, 2015a).

Despite the importance of the real-time safety optimization, it has generally been disregarded in existing ATSC algorithms, most likely due to the lack of tools to evaluate safety of signalized intersections in real time. Unlike vehicle delay and travel time, the safety level of signalized intersections cannot be directly estimated from real-time traffic data. However, various real-time safety models for signalized intersections have recently been developed and validated (Essa and Sayed, 2018, 2019; Essa et al., 2019; Zheng et al., 2019a, 2019b). These models relate the number of traffic conflicts or the risk of collisions to various dynamic traffic parameters at a very short time period (i.e., a few seconds). The dynamic traffic parameters include traffic volume, shock wave area, shock wave speed, queue length, and platoon ratio. These models can be used to evaluate safety in real time; subsequently, they can enable developing ATSC strategies for real-time safety optimization.

This paper presents a novel self-learning adaptive traffic signal control algorithm to optimize traffic safety in real time using CVs data. The algorithm is referred to as RS-ATSC (Real-time Safety-optimized Adaptive Traffic Signal Control). The RS-ATSC algorithm has several advantages. First, the safety evaluation is not based on simulated conflicts which were shown not to well represent actual-field conflicts and crashes (Essa and Sayed, 2015a,b; Zheng et al., 2019c). Rather, the optimization is based on real-time safety models that were originally developed and validated using real-world traffic data. Second, the algorithm is developed using the Reinforcement Learning (RL) technique as an efficient approach to solve the ATSC problem considering real-time and stochastic traffic changes (Abdulhai and Kattan, 2003; Abdulhai et al., 2003; El-Tantawy et al., 2014). Third, the algorithm is practical since it respects all traffic signal operation standards, including the phasing sequence, the minimum/maximum green time, and the intersection clearance time. Fourth, the algorithm is validated using real-world traffic data obtained from two signalized intersections. Fifth, the presented algorithm is found to be effective and feasible under low market penetration rates of CVs. Lastly, to the best of the authors' knowledge, this is the first self-learning ATSC algorithm that optimizes traffic safety in real time (i.e., safety is evaluated and optimized over a very short time period, a few seconds).

2. Previous work

2.1. Implemented ATSC algorithms

Over the past few decades, several ATSC algorithms have been implemented around the world. The earliest two algorithms were the Sydney Coordinated Adaptive Traffic System (SCATS) (Sims, 1979), and the Split Cycle Offset Optimization Technique (SCOOT) (Hunt et al., 1981). After that, Federal Highway Administration (FHWA) adaptive control systems were developed and used, including the Optimization Policies for Adaptive Control (OPAC) (Gartner, 1983), the Real Time Hierarchical Optimized Distributed Effective System (RHODES) (Head et al., 1992), and, more recently, the ACS Lite (Luyanda et al., 2003). These algorithms differ in operation, but they share a common objective of accommodating current traffic demands to maximize throughput capacity and minimizing traffic delays (Sabra et al., 2010). However, these ATSC systems generally suffer from several operational limitations, such as handling several intersections at the same time, using a centralized control system, and relying on loop detectors for detection and estimation (Abdulhai and Kattan, 2003; El-Tantawy et al., 2014). More importantly, these systems do not consider optimizing traffic safety as an objective.

2.2. Traffic signal control algorithms using CVs data

With the increasing emergence of the CVs technology, numerous traffic signal control algorithms have recently been proposed to optimize traffic efficiency using CVs real-time data. Some studies, for example, proposed various algorithms to optimize and coordinate traffic movement in road intersections without using any traffic lights, assuming that all vehicles are connected and autonomous (e.g., Lee and Park, 2012; Lee et al., 2013a; Kamal et al., 2015; Mirheli et al., 2019). More realistically, other studies assumed various market penetration rates of CVs to develop and test ATSC algorithms. The developed algorithms generally aim at minimizing the total delay (e.g., Lee et al., 2013b; Guler et al., 2014; Yang et al., 2016; Liang et al., 2020; Rafter et al., 2020). Several studies have also considered multiple objectives, such as minimizing the total delay and the number of stops (Goodall et al., 2013), or minimizing the total delay and the queue length (Feng et al., 2015). Most of the existing algorithms optimize the traffic signal timing based on real-time vehicle information, assuming one-way vehicle-to-infrastructure (V2I) communications. Some algorithms, however, optimize both the traffic signal timing and vehicle trajectories, assuming a specific percentage of autonomous vehicles and bidirectional V2I communications (e.g., Yang et al., 2016; Al Islam and Hajbabaie, 2017; Jiang et al., 2017; Xu et al., 2017; Guo et al., 2019a). While the majority of previous studies has mainly focused on adapting traffic signals to improve mobility, a limited number of studies have considered optimizing traffic signals to reduce traffic emissions and fuel consumption (e.g., Xu et al., 2017; Jiang et al., 2017; Guo et al., 2019a). On the other hand, optimizing traffic safety has generally been disregarded. More details and a systematic review of research on using CVs real-time data for urban traffic signal control can be found in a recent study by Guo, et al. (Guo et al., 2019b).

2.3. Self-learning ATSC algorithms

Self-learning ATSC algorithms are emerging methods that rely on learning the control policy from the direct interaction with the traffic environment without needing a predefined model for the environment nor human intervention. A significant amount of research has been conducted on developing self-learning ATSC algorithms with the goal of improving traffic efficiency and optimizing mobility using real-time traffic data. The Reinforcement Learning (RL) technique seems to be the most attractive approach in the literature to develop self-learning ATSC algorithms. Several RL methods have been applied, including model-based Q-learning (Wiering, 2000), Q-learning (Abdulhai et al., 2003; Camponogara and Kraus, 2003; Shoufeng et al., 2008; Salkham et al., 2008; Balaji et al., 2010; Arel et al., 2010; El-Tantawy et al., 2014), State-Action-Reward-State-Action (SARSA) (Thorpe and Anderson, 1996; El-Tantawy et al., 2014; Brys et al., 2014), Multiagent Reinforcement Learning (Wiering, 2000; El-Tantawy et al., 2013), and, more recently, Deep Q-Network (DQN) (Shabestary and Abdulhai, 2018; Gong et al., 2019). Various objectives have been considered to optimize mobility, including minimizing queue length, minimizing travel time, minimizing total delay, and maximizing vehicle throughput.

Although these RL-based ATSC algorithms have shown a significant improvement in traffic mobility, they have not considered evaluating or optimizing traffic safety. The safety evaluation in these studies is limited to avoiding crashes between simulated vehicles, providing standard signal times (e.g., yellow time, all-red time, minimum green time), and prohibiting conflicting signal phases from being operated simultaneously.

2.4. Real-time safety evaluation of signalized intersections

The safety of signalized intersections has often been evaluated at an aggregate level (i.e., usually years) relating collisions to annual traffic volume and the geometric characteristics of the intersection (AASHTO, 2010). However, for many safety issues, it is essential to understand how real-time changes in traffic parameters and signal control affect safety. Therefore, a few studies have considered the real-time safety analysis for signalized intersections and urban arterials (e.g., Theofilatos, 2017; Theofilatos et al., 2017; Yuan et al., 2018; Yuan and Abdel-Aty, 2018). Most of these studies investigated the relationship between the potential crash risk and real-time traffic and weather characteristics at signalized intersections in a time period shorter than one hour, generally 5 − 15 min. Although this time period is considerably shorter than the analysis period in the traditional safety performance functions, evaluating safety for a 5-minute time period does not capture the safety effects of real-time variations in traffic conditions and signal timing. More recent studies (Essa and Sayed, 2018, 2019; Essa et al., 2019; Zheng et al., 2019a, 2019b) have considered evaluating real-time safety over a time period shorter than 5 min. Specifically, the time period was considered to be the signal cycle length. Several real-time safety models were developed to relate the number of traffic conflicts (Essa and Sayed, 2018, 2019; Essa et al., 2019) or the risk of collision (Zheng et al., 2019a, 2019b) to various dynamic traffic characteristics such as the traffic volume, the shock wave characteristics, and the platoon ratio.

3. The proposed RS-ATSC algorithm

This section includes the methodology of developing the proposed RS-ATSC algorithm. First, the real-time safety models, on which the algorithm is based, are described. Second, an overview of both the reinforcement learning (RL) technique and the selected method to solve the RL problem is provided. Third, the formulation of the RL problem for the proposed RS-ATSC is presented. This includes the state, action, and reward definitions; the learning and discount rates; and the tradeoff between exploration and exploitation. Lastly, details on modelling the environment and training the algorithm are elaborated.

3.1. Real-time safety models

In this paper, the proposed RS-ATSC algorithm is based on the real-time safety models developed in (Essa and Sayed, 2018). These models relate various dynamic traffic parameters to the number of rear-end conflicts at the signal-cycle level. The Time-to-Collision (TTC) (Hayward, 1972) was used as a traffic conflict indicator. The traffic parameters (Fig. 1) include traffic volume (V), shock wave area (A), shock wave speed (S_{12}), queue length (Q), and platoon ratio (P). The models were originally developed using real-world traffic data obtained from several signalized intersections. The models have good fit and all the explanatory variables are statistically significant (Table 1) (Essa and Sayed, 2018). It is also worth noting that these models have been further validated and their transferability was investigated in subsequent studies (Essa and Sayed, 2019; Essa et al., 2019).

3.2. Reinforcement learning

The Reinforcement Learning (RL) technique was applied to develop the proposed RS-ATSC algorithm. RL is an area of machine learning that has widely been applied in the literature for self-learning ATSC algorithms (e.g., Wiering, 2000; Abdulhai et al., 2003; Camponogara and Kraus, 2003; Richter et al., 2007; Shoufeng et al., 2008; Salkham et al., 2008; Balaji et al., 2010; Arel et al., 2010; El-Tantawy et al., 2014; Gong et al., 2019; Shabestary and Abdulhai, 2018). In RL, the agent or the decision-maker (e.g., the signal controller) dynamically interacts with its surrounding environment (e.g., the traffic network). The agent iteratively observes the state of the environment, takes an action accordingly (e.g., determining which signal phase will be green), and receives a reward or an evaluative feedback (Fig. 2). Unlike the supervised machine learning paradigm, the RL agent is not told which actions to take. Instead, it learns and discovers which actions yield the maximum reward over time. In other words, RL is a goal-directed learning, in which, the agent learns how to map states and actions to achieve a specific goal (i.e., maximizing the total cumulative reward). This state-action mapping is called the control policy. The agent tries to learn the optimal control policy by iteration (i.e., trial-and-error search). It should also be noted that actions may affect not only the immediate reward but also the next state and subsequently the future rewards. Thus, RL has two main distinguishing characteristics: trial-and-error search and delayed reward (Sutton and Barto, 1998).

3.3. Q-learning

Solving the RL problem requires computing the optimal control policy. However, it should be noted that the expression “optimal control policy” came from the theoretical definition of the RL technique. Practically, there is no solution that is optimal under all conditions, and the optimality case cannot be defined. Therefore, in this research, the control policy is optimized but not necessarily optimal.

There are numerous methods to solve the RL problem. Generally, the RL methods can be classified into three main classes: dynamic programming (DP) methods, Monte Carlo (MC) techniques, and temporal difference (TD) learning methods. TD learning methods are recommended as the most relevant to the ATSC problem (El-Tantawy et al., 2014; Abdulhai and Kattan, 2003). TD methods have an advantage over DP methods. Unlike DP methods, TD methods do not require a model of the environment dynamics. Instead, the agent learns directly through interaction with the environment. TD methods also have an advantage over MC methods. While MC methods require waiting until the end of an episode to find out the return, TD methods require waiting for only one time-step (Sutton and Barto, 1998; Abdulhai and Kattan, 2003).

There are several TD methods, including the SARSA method, the Q-learning method, and the n-step difference learning method. A previous study (El-Tantawy et al., 2014) has compared the performance of these methods in solving the ATSC problem. The results showed that SARSA and Q-learning lead to the same results and outperform the n-step difference method. This outperformance may be attributed to the nature of the ATSC problem. The control task in ATSC algorithms is a continuing task (i.e., not a finite episode) with a discounted reward in which looking ahead to future steps is less important compared to a finite episodic task with undiscounted reward (El-Tantawy et al., 2014). Numerous studies have succeeded in solving the ATSC problem using

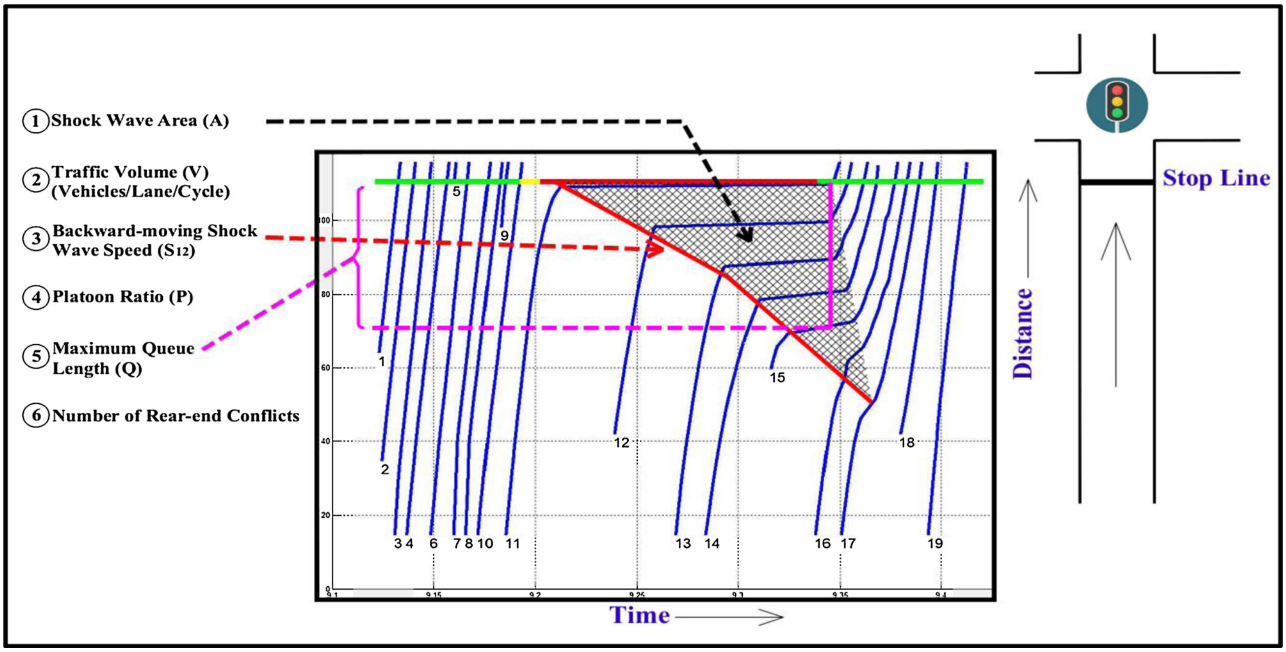

Fig. 1. Cycle-related traffic parameters for real-time safety models.
**Edited from the original figure provided in (Essa and Sayed, 2019).*

Table 1
Conflict-based Safety Models at the Cycle Level (Essa and Sayed, 2018).

#	Model	Variables	Error Structure	K	SD	df	χ^2	AIC
1	$\boldsymbol{E(Y)} = V^{1.563}exp(-3.231)$	V	NB	3.05	249	220	356	775
2	$\boldsymbol{E(Y)} = V^{0.706}exp(-1.797 + 0.501\,A)$	V, A	NB	14.9	244	219	241	702
3	$\boldsymbol{E(Y)} = V^{0.65}exp(-2.046 + 0.0122\,Q)$	V, Q	NB	8.73	243	219	253	716
4	$\boldsymbol{E(Y)} = V^{1.637}exp(-3.316 + 0.05\,S_{12})$	V, S_{12}**	NB	3.10	248	219	347	775
5	$\boldsymbol{E(Y)} = V^{1.571}exp(-1.768 - 1.266\,P)$	V, P	Poisson	—	276	219	281	706
6	$\boldsymbol{E(Y)} = V^{1.239}exp(-1.624 + 0.294\,A - 0.828\,P + 0.119\,S_{12})$	V, A, P, S_{12}	Poisson	—	240	217	215	674

E(Y): Predicted number of rear-end conflicts per cycle with time-to-collision (TTC) equal or less than.1.50 s
K: Shape parameter for Negative binomial (NB) family.
All variables are significantly different from zero at 95 % confidence level.
SD, df, χ^2, AIC: Scaled deviance, Degree of freedom, Pearson chi-squared, Akaike's Information Criterion.
** Significantly different from zero at 90 % confidence level.

the Q-learning method (Abdulhai et al., 2003; Camponogara and Kraus, 2003; Shoufeng et al., 2008; Salkham et al., 2008; Balaji et al., 2010; Arel et al., 2010; El-Tantawy et al., 2014). Therefore, this method was selected in this research to develop the proposed RS-ATSC algorithm.

The Q-learning (Watkins, 1989; Watkins and Dayan, 1992) is an off-policy TD method, in which the algorithm uses the experience of each state transition to update one element of a table called Q-table. The Q-table is a matrix in which each row represents a specific state and each column represents a specific action. Each cell in this matrix represents a Q-value for a specific state-action pair $Q(s, a)$ (Sutton and Barto, 1998). The Q-value in general is used to compare various actions at a specific state. Given a specific state (specific row), the best action (column) is that with the highest Q. To train the algorithm, the Q-table is usually initiated with all values set to zero. Then, the Bellman's equation (Eq. 1)

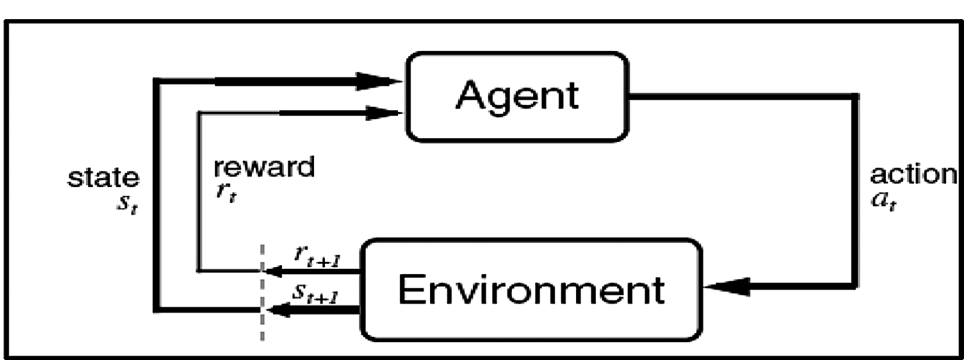

Fig. 2. The agent–environment interaction in reinforcement learning (Sutton and Barto, 1998).

is used to update these values every time-step (Δt). The Q-value for a specific state-action pair $Q(s, a)$ depends on the reward of taking that action at that state. The Q-value also depends on the new state that is achieved as a result of taking this action. Two factors are also considered when updating the Q-value: the discount rate (γ) and the learning rate ($\propto^{t+1}$). The unit of the Q-value is the same as that for the reward, since all other factors are unitless. When the agent performs action a^t at state s^t, leading to a new state s^{t+1} and a reward r^{t+1}, the Q-learning algorithm improves its policy by updating the Q-table according to Bellman's equation as follows:

$$Q^{t+1}(s^t,\ a^t) = Q^t(s^t,\ a^t) + \propto^{t+1}[r^{t+1} + \gamma \max_{a \in A} Q^t(s^{t+1},\ a^{t+1}) - \ Q^t(s^t,\ a^t)] \tag{1}$$

Where:

s^t, a^t: the current state and the selected action at the current state;
Q^{t+1}, Q^t: the updated and the old Q-value;
r^{t+1}: the reward of applying action a^t at state s^t;
s^{t+1}, a^{t+1}: the new state and the best action at the new state;
$\propto^{t+1}$: the learning rate;
γ: the discount rate;
A: the action's space.

3.4. State representation

One of the main challenges in the Q-learning method is the use of the tabular form of the Q-matrix to represent realistic environments that have a very large or infinite number of states. Including large number of states in the Q-matrix can result in most states being not experienced by the agent. This issue exists in the ATSC problem, where the continuous and stochastic nature of traffic leads to an infinite number of possible states (i.e., various vehicle positions and speeds). To overcome this issue, there are typically two ways. The first is to enable generalization among states by representing Q-values not as a table but as a trainable parameterized function. Such a generalization is called *"function approximation"* because it takes examples from a desired function and attempts to generalize from them to construct an approximation of the entire function. There are many methods for the function approximation, such as artificial neural networks and statistical curve fitting (Sutton and Barto, 1998). However, due to its imperfect value estimations, the function approximation can have many consequences that can affect the quality of the solution, such as the divergence of Q-estimates (Abdulhai and Kattan, 2003). Another simpler way, that overcomes the problem of having a very large number of states, is to discretize all possible states into ranges and define only these ranges in the Q-matrix. Since Q-matrix with discretized ranges of states was successfully applied in previous studies for the ATSC problem (e.g., Wiering, 2000; Abdulhai et al., 2003; Camponogara and Kraus, 2003; Richter et al., 2007; Shoufeng et al., 2008; Salkham et al., 2008; Balaji et al., 2010; Arel et al., 2010; El-Tantawy et al., 2014), this method was selected for the state representation in the proposed RS-ATSC.

In the proposed RS-ATSC, the state is represented by the current green phase as well as the status of existing vehicles in each approach within the *V2I Dedicated Short-Range Communications (DSRC)* domain upstream the stop line. Specifically, the state vector consists of 5 elements, assuming a 4-approach intersection. The first element is the current green phase represented by a phase index (the length of the current phase is not included), while the other four elements represent the current traffic status of each approach.

Representing the current traffic status at each approach took several forms in the literature. This includes the number of existing vehicles (Camponogara and Kraus, 2003; Salkham et al., 2008), the queue length (Abdulhai et al., 2003; El-Tantawy et al., 2014), the number of arriving vehicles to the current green phase and the queue length at the red phase (El-Tantawy et al., 2014), the cumulative delay (El-Tantawy et al., 2014; Shoufeng et al., 2008), the relative delay (Arel et al., 2010), and the detectors status (Richter et al., 2007). In this paper, the objective of the RS-ATSC algorithm is to improve safety by minimizing the rear-end conflict rate at the intersection's approaches. Therefore, the current traffic status for an approach is represented by the number of rear-end conflicts per second (i.e., the current rear-end conflict rate) at that approach.

The number of rear-end conflicts for each lane at the signal cycle level can be estimated from the real-time safety models presented in Table 1, using dynamic traffic variables. Of the six real-time safety models shown in Table 1, **model 6** is used since it shows the best statistical fit with all variables being significant (i.e., V, A, S_{12}, P) (Essa et al., 2019). The predicted number of conflicts at each lane at the signal cycle level is then normalized by the cycle length (C) to obtain the conflict rate (conflicts/second). Since signal cycles can have different lengths, the cycle length (C) is dynamically updated in the algorithm every time-step (Δt) to be the length of the last cycle. Lastly, the conflict rate at each approach is calculated as the summation of conflict rates from all lanes, as follows:

$$C_{r(App)} = \sum_{i=1}^{N} \frac{Y_i}{C} \tag{2}$$

Where:

$C_{r(App)}$: the rear-end conflict rate for the approach (number of conflicts/second);
Y_i: the number of rear-end conflicts at the signal cycle level for lane i;
N: the number of lanes at the approach;
C: the signal cycle length in seconds.

To obtain a discretized Q-matrix with all possible states, the calculated conflict rate per approach is discretized into specific ranges. The discretization method involves determining the minimum and the maximum value of the conflict rate as well as specifying the range width. The minimum conflict rate was set to zero (i.e., no vehicles exist at the cycle). On the other hand, the maximum conflict rate was calculated given: (1) the maximum number of vehicles that can exist within the V2I DSRC domain upstream the stop line; (2) the number of lanes per approach; and (3) the minimum cycle length (i.e., corresponding to the maximum conflict rate given a specific number of conflicts). The minimum cycle length equals the summation of the minimum green times plus yellow and all-red times for all phases. The range width was set to be increasing uniformly with the range number. This means the first range (i.e., the range that starts from the minimum conflict rate) has the lowest width, while the last range (i.e., the range that ends with the maximum conflict rate) has the largest width. Hypothetical simulation runs were performed for several hours considering many different scenarios (various traffic volumes and various cycle lengths) to validate the reasonability of the discretized ranges before training the Q-learning algorithm.

3.5. Action representation

In RL-based ATSC algorithms, the action, taken by the agent at each decision point, is to determine the next green phase. The size of the action space (i.e., the number of possible actions) depends on the phasing sequence. If the phasing sequence is variable, the action space typically includes n actions; where n is the number of phases. If the phasing sequence is fixed, the action space has only two actions: (1) extending the current green phase, and (2) switching the green light to the following phase. Several previous studies have applied the variable phasing sequence (e.g., Wiering, 2000; Richter et al., 2007; Salkham et al., 2008; Arel et al., 2010; El-Tantawy et al., 2014; Gong et al., 2019; Shabestary and Abdulhai, 2018), while others have used the fixed phasing sequence (e.g., Abdulhai et al., 2003; Camponogara and Kraus, 2003; Shoufeng et al., 2008; Balaji et al., 2010; El-Tantawy et al., 2014).

The variable phasing sequence could theoretically lead to a better performance, since it gives the RL agent more actions to investigate. However, it is generally recommended to prohibit ATSC systems from changing the phase sequence (NCHRP, 2015) for several safety and mobility concerns. The variable phasing sequence may confuse road-users, leading to unsafe traffic movements. For example, at 4-leg intersections with protected-permissive left turns, varying the phasing sequence could cause a yellow trap (i.e., a condition that leads the left-turning user into the intersection believing the opposing user is seeing a yellow). In addition, when the next green phase is not expected, road-users tend not to react quickly to the green indication. This could increase the start-up lost time causing additional delays (NCHRP, 2015). The fixed phasing sequence, on the other hand, meets the road-users' expectation, providing a safer traffic environment without unnecessary start-up delays. Furthermore, having only two possible actions in the fixed phasing sequence, instead of n actions, decreases the Q-matrix size dramatically, which enables faster convergence of the RL algorithm to the optimized policy.

Therefore, we adopted the fixed phasing sequence for the proposed RS-ATSC algorithm. The RS-ATSC agent performs one of the two following actions: (1) extending the current green phase (A1); and (2) switching the green light to the next phase (A2). If action A1 is selected, the current green phase will be extended by a specific time interval (assumed to be 5 s). On the other hand, if action A2 is selected, the yellow (Y) and the all-red (AR) times will be applied before switching the green light to the next phase and applying its minimum green time (G_{min}). Thus, the update time interval Δt (i.e., the time between decision points) for the RS-ATSC algorithm can be expressed as follows:

$$\Delta t = \begin{Bmatrix} 5 & if\ a^t = A1 \\ Y + AR + G_{min} & if\ a^t = A2 \end{Bmatrix} \tag{3}$$

Where:

Δt: the update time interval in seconds for the RS-ATSC algorithm;

Y: the yellow time in seconds;

AR: the all-red time in seconds;

G_{min}: the minimum green time in seconds.

The proposed RS-ATSC algorithm also applies the maximum green time as a fundamental constraint. This constraint defines the maximum length of time that a phase can be green in the presence of a conflicting call. If the maximum green time is reached, the RS-ATSC agent is prohibited from extending the green time of the current phase (i.e., action A1 cannot be applied). A typical value of 70 s is assumed for the maximum green time (NCHRP, 2015).

3.6. Reward representation

Since the main objective of the proposed RS-ATSC is to optimize traffic safety, the reward for each state-action pair in the RS-ATSC algorithm is defined by the rear-end conflict rate estimated from all approaches as a penalty. The conflict rate for each lane at each approach is estimated at every decision point. Then, the reward value r^{t+1} for performing action a^t at state s^t can be defined as follows:

$$r^{t+1} = -\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{Y_{ij}}{C} \tag{4}$$

Where:

r^{t+1}: the reward value for the state-action pair (a^t, s^t) estimated at state s^{t+1};

N: the number of lanes per approach;

M: the number of approaches at the signalized intersection;

Y_{ij}: the number of rear-end conflicts at the signal cycle level for lane j in approach i (**model 6** in Table 1);

C: the signal cycle length in seconds.

3.7. Learning rate and discount rate

The learning rate α^{t+1} in Eq. (1) is determined every time-step as the reciprocal of the number of visits by the agent to the state–action pair (s^t, a^t) as follows (Sutton and Barto, 1998; El-Tantawy et al., 2014):

$$\alpha^{t+1} = \frac{1}{V^{t+1}(s^t,\ a^t)} \tag{5}$$

Where: $V^{t+1}(s^t,\ a^t)$: the number of visits by the agent to the state–action pair (s^t, a^t).

In addition, the discount rate γ in Eq. (1), which considers the long-run reward, is assumed to be 0.5.

3.8. Exploration versus exploitation

The trade-off between exploration and exploitation is one of the main challenges in RL. While the agent must exploit the most effective experienced actions to obtain a lot of reward, it must also explore new actions in order to make better action selections in the future. To obtain the optimized policy, neither exploitation nor exploration can be followed exclusively. Rather, an action selection strategy should be applied to balance the exploration and exploitation. The typical action selection strategies used in the literature are ∈-greedy and softmax (Sutton and Barto, 1998).

In this paper, the ∈-greedy method is adopted as the action selection strategy. This means the RS-ATSC agent selects, in each iteration, the greedy action most of the time except for ∈ amount of time, when it selects a random action uniformly. The rate of exploration ∈ is assumed to decrease gradually with the number of iterations (i.e., the age of the agent). The highest exploration occurs at the beginning of the learning, since the agent does not have much experience. At the end of the learning, the lowest exploration occurs, and more exploitation takes place as the agent converges to the optimized policy (Sutton and Barto, 1998). The gradual decreasing rate of exploration ∈ can be represented as follows (El-Tantawy et al., 2014):

$$\varepsilon = e^{-En} \tag{6}$$

Where: E is a constant and n is the iteration number.

3.9. Modeling the environment

The traffic microsimulation model VISSIM (v7.00.16) (PTV, 2015a) was utilized in this study. VISSIM is a time-step and behavior-based model developed to simulate traffic and depends on a psycho-physical car-following model that is based on Wiedemann's model (Wiedemann, 1974; PTV, 2015a). The Wiedemann model assumes that the driver can have one of four driving modes: free driving, approaching, following, and braking (PTV, 2015a).

An isolated signalized intersection was modeled in VISSIM, representing a connected-vehicle environment for the proposed RS-ATSC algorithm. The modeled intersection has four approaches with two through lanes and a single left-turn lane. The traffic control unit, the agent, receives real-time V2I information from all CVs that exist within a specific distance from the stop lines. This distance virtually represents the standard V2I DSRC domain. Since the standard V2I DSRC domain roughly ranges from 150 to 300 m (U.S. Department of Transportation, 2015), the distance was assumed to be 225 m (i.e., the average). Furthermore, various market penetration rates of CVs were represented in the VISSIM model by creating a new vehicle class called *"connected vehicle"* and varying traffic composition percentages of each traffic input point. In addition to CVs, loop detectors installed at each lane are assumed to provide real-time traffic information to the traffic controller. Two types of loop detectors were considered: traffic counting detectors at trough lanes and left-turn detectors at the beginning and the end of each left-turn bay (Fig. 3).

To simulate the CVs environment and the RS-ATSC algorithm, an

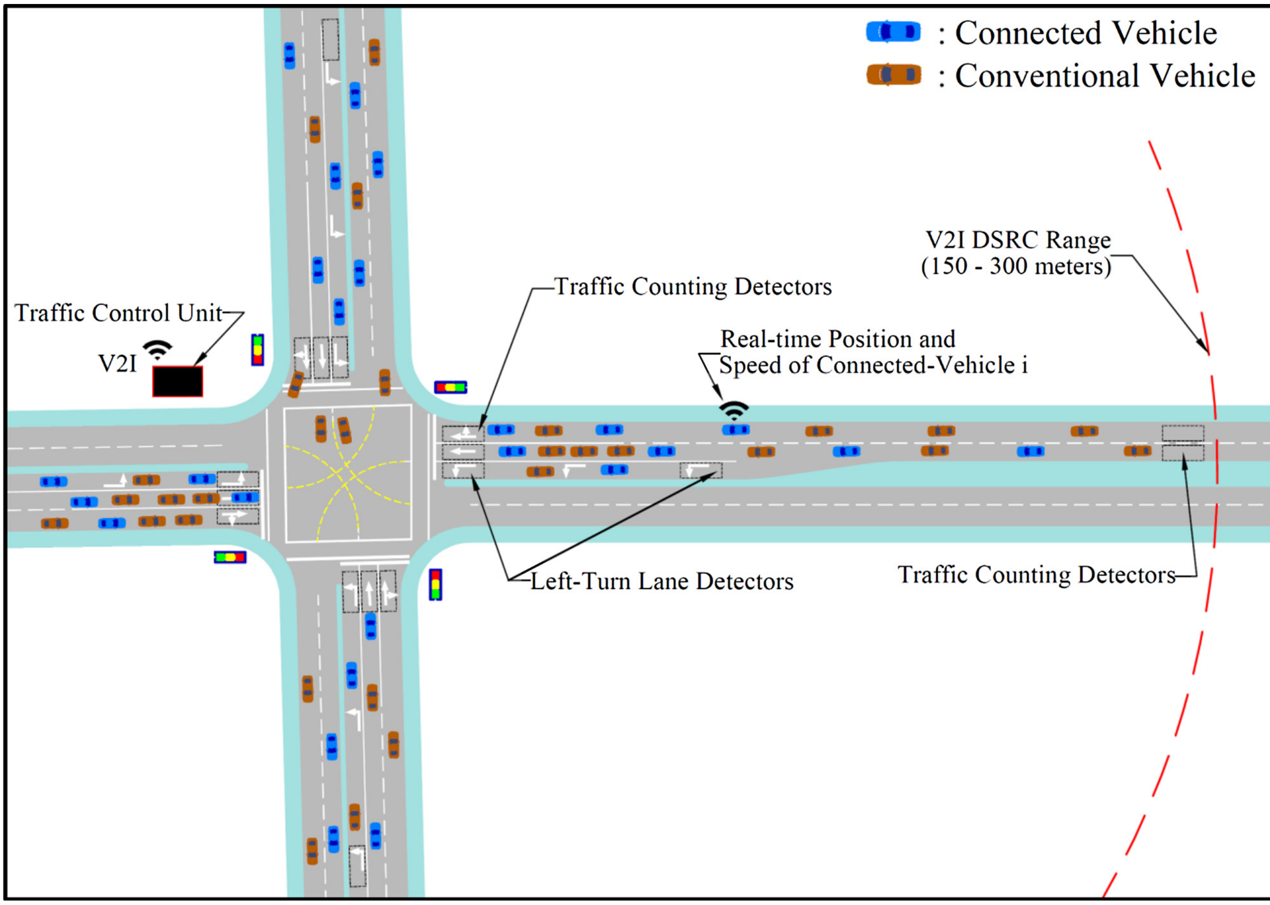

Fig. 3. Modelling an isolated signalized intersection with connected-vehicles in a simulation platform for the proposed RS-ATSC algorithm.

external MATLAB code was developed to control the VISSIM model through its COM interface (PTV, 2015b). The MATLAB code can run/pause simulation at any time using the *"sim-break-at"* function, record detailed information on traffic signals and vehicles (i.e., vehicle identification number, class, position, and speed), and apply any required real-time changes to traffic signal heads in VISSIM. Thus, this code represents the agent (i.e., the traffic controller) for the Q-learning, since it is able to receive the environment's state and take various actions.

3.10. Training the algorithm

The RS-ATSC algorithm was trained by running the VISSIM simulation model of the isolated intersection depicted in Fig. 3. The simulation was run for 420 episodes. Each episode included 20,000 simulation seconds divided into 1500s as a warming-up period, 500 s as a cooling-down period, and 18,000 s (i.e., 5 h) to train the algorithm. During the training period of each episode, the simulation was paused every Δt seconds (as per Eq. 3), the state was defined, the next action was selected and applied, the reward was calculated, and lastly, the Q-matrix was updated. To account for the stochastic nature of traffic, various random seeds were considered in VISSIM. Additionally, to allow the algorithm to visit as many states as possible, the traffic volume entering each approach was uniformly randomized between 200 vehicle/hour to 1600 vehicle/hour. Traffic volumes were determined by assuming random values, from 0.1 to 1, of the volume to capacity ratio (V/C). These values correspond to undersaturation flow conditions where the real-time safety models presented in Table 1 (Essa and Sayed, 2018) can be applied.

When training such a RL-based algorithm, observing the learning progress of the agent and assuring the algorithm's convergence to the optimized policy are essential. In general, the theoretical definition of convergence to the optimal policy is that the agent visited each state-action pair an infinite number of times. Since this is not feasible, we observed the learning progress of the agent after each episode using two measures: (1) the number of visited state-action pairs and (2) the minimum conflict rate (conflicts/exposure) resulted from all episodes. Fig. 4 illustrates the learning progress of the agent represented by the number of traffic conflicts normalized by the traffic volume (conflicts/exposure). After about 200 episodes, most state-action pairs were visited many times by the agent and the minimum conflict rate was not changed throughout the next episodes (i.e., episodes 201–420). Therefore, we considered the agent converged to the optimized policy. The conflict rate was reduced from approximately 0.18 conflict/vehicle at the beginning of the learning process to 0.11 conflict/vehicle when the convergence was reached.

4. Validation using real-world traffic data

The proposed RS-ATSC algorithm was validated using real-world traffic data obtained from two signalized intersections in the city of Surrey, British Columbia, Canada. For both intersections, the real-world signal control is a typical fully actuated signal control with stop-line and extension detectors. The real-world actuated signal control was set as a benchmark to evaluate the effectiveness of the RS-ATSC. For each intersection, both the trained RS-ATSC and the real-world benchmark actuated signal control (ASC) were implemented in a calibrated VISSIM model. The safety and operational performances of both signal controllers were then observed and compared.

It should be noted that, over the past few decades, research has shown that actuated signal controllers outperform pre-timed (fixed) signal controllers in terms of traffic operation and safety, especially when traffic volumes are not predictable and vary significantly.

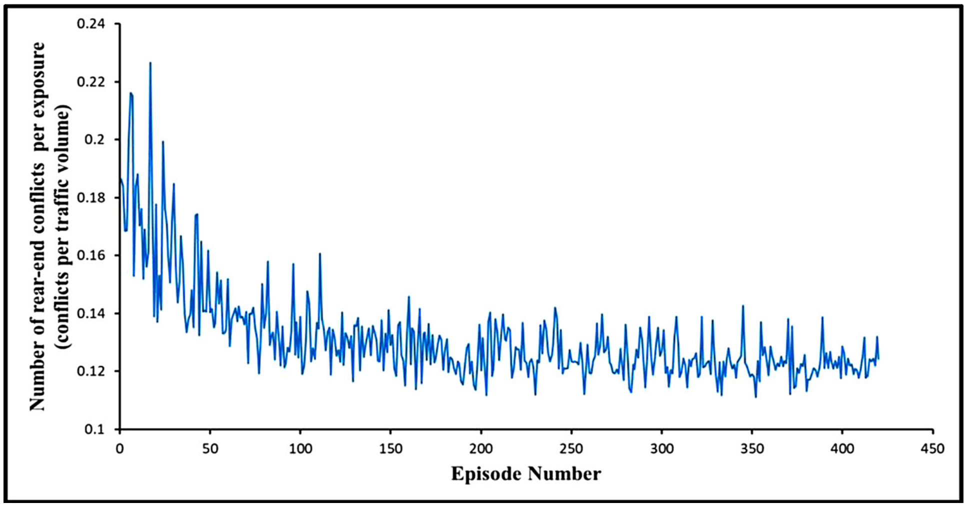

Fig. 4. Learning progress of the proposed RS-ATSC algorithm.

Compared to pre-timed signal controllers, actuated signal controllers can better accommodate widely fluctuating volumes, decrease delays, reduce driver frustration and red-light running, and improve safety. That's why converting the signal controller from pre-timed to actuated has recently become a common upgrading measure to improve the performance of signalized intersections. Thus, in this study, the actuated signal controller (i.e., the state-of-art controller) was used as a benchmark to validate the performance of the proposed RS-ATSC algorithm.

4.1. Real-world traffic data

The first selected intersection is 72nd Avenue and 128 Street, while the second selected intersection is 72nd Avenue and 132 Street. At each intersection, video data were collected using 8 high-resolution cameras (29.97 frames per second) distributed to cover the four approaches (i.e. two cameras for each approach). The data were collected during a weekday from 9:00 am to 6:00 pm, to cover both peak and off-peak hours. Thus, the total amount of the collected data is 144 video-hours (8 cameras * 9 h * 2 intersections). Fig. 5 shows the location of the selected intersections, the selected approaches, and the recorded video scenes.

Detailed real-world traffic data at each approach for each hour were extracted from the video recordings. These data include the actual signal program, the traffic volume of all movements, the number of vehicles arriving on green, the average platoon ratio, the average delay time, the desired speed distribution, and the traffic composition. The traffic composition includes percentages of passenger cars, trucks, and buses. Motorcycles were neglected as they were rarely found in the videos.

4.2. Calibrated simulation models

VISSIM models of the two selected intersections came from previous studies (Essa and Sayed, 2015a,b; 2016). The VISSIM models were built accurately to match actual field conditions in terms of intersection geometry, traffic volumes, traffic composition, and traffic signal settings (i.e., the actuated signal controller). The real-world ASC was defined in VISSIM using the Ring Barrier Controller (RBC) module (PTV, 2015a). Visual inspection was also performed to ensure that there are no abnormal movements of the simulated vehicles. In addition, the VISSIM models were precisely calibrated in (Essa and Sayed, 2015a,b) using a comprehensive two-step calibration procedure. The first

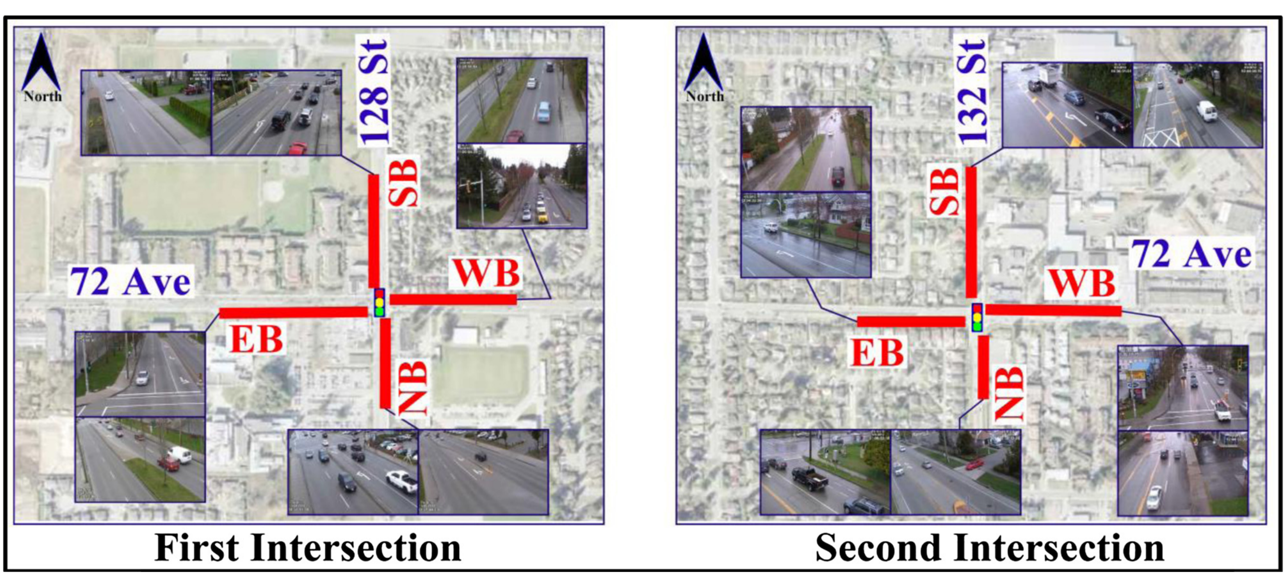

Fig. 5. Study locations and video scenes.
(EB: eastbound, WB: westbound, NB: northbound, SB: southbound)

Table 2
Calibrated VISSIM Parameters (Essa and Sayed, 2015a, 2015b).

Parameter	Description	Unit	Default Value	Calibrated Value (128 St &72 Ave)	Calibrated Value (132 St & 72 Ave)
Standstill distance	The desired distance between stopped vehicles	m	1.50	2.50	2.10
Headway time	The time that a driver wants to keep	s	0.90	1.3	1.30
Following thresholds	The thresholds which control the speed differences during the 'Following' state	—	±0.35	±0.25	±1.10
Reduction factor for safety distance closed to stop line	This reduction factor defines the vehicle behavior close to stop line at signalized intersections	—	0.60	0.75	0.60
Start upstream of stop line	Distance upstream of the stop line of signalized intersection	m	100	110	100
Desired deceleration	Desired deceleration is used as the maximum for: the deceleration caused by a desired speed decision; the deceleration in case of Stop & Go traffic, when closing up to a preceding vehicle; the deceleration toward an emergency stop position (route); and for co-operative braking	m/s^2	−2.80	−2.80	−2.80

calibration step aimed to match the simulated delay times with the field-observed delay times. This was achieved by matching the arrival pattern and the desired speed to the field conditions. The second calibration step aimed at enhancing the correlation between field-observed and simulated traffic conflicts by calibrating the VISSIM parameters. Firstly, important VISSIM parameters that had the most significant effect on the simulated conflicts were determined through a sensitivity analysis. Subsequently, a Genetic Algorithm was applied to estimate the best values of these parameters with the objective of enhancing the correlation between field-observed and simulated conflicts. Table 2 shows the selected VISSIM parameters and their calibrated values at each intersection (Essa and Sayed, 2015a,b).

4.3. Safety and operational performance of the proposed algorithm

To validate the proposed RS-ATSC, its performance was compared to the benchmark ASC. The number of rear-end traffic conflict occurred at the intersection approaches was considered as a measure of safety performance. Other measures of operational performance were also considered, including the number of stops, the maximum queue length, the 95th percentile of queue length, and the average delay time per vehicle. To evaluate these measures, the calibrated VISSIM model for each intersection was run for a 9 -h period (i.e., 9:00 am to 6:00 pm). For each hour, two signal controllers in VISSIM were simulated separately: (1) the RBC module that represents the real-world benchmark ASC, and (2) an external real-time MATLAB code that represents the trained RS-ATSC. For each signal controller, 10 different random seeds were applied, and the results were then averaged. The minimum required number of random seeds to compare the performance measures of the two alternatives (i.e., the proposed RS-ATSC and the ASC benchmark) was estimated, following the methodology provided by Dowling et al. (2004). The statistical analysis showed that 10 simulation runs are sufficient to reject the null hypothesis at 95 % confidence level. This means the differences in the performance measures are caused by using two different alternatives and not just a result of using different random seeds.

During each simulation run, detailed simulated traffic data were continuously recorded at a very short time step (e.g., every second of simulation). These data included position and speed of every vehicle, vehicle types, and status of all signal heads. The data recording was obtained using an external program that controls the simulation model via the VISSIM COM interface (PTV, 2015b). After running the simulation and recording detailed traffic data, several steps were applied to estimate dynamic traffic parameters (i.e., V, A, S_{12}, P) and evaluate safety (i.e., predict the number of conflicts) (Essa and Sayed, 2020). First, signal cycles for each approach of the intersection were determined using the recorded status of the approach signal head. Second, recorded vehicle trajectories were filtered by time and space to specify vehicle trajectories for each lane per each signal cycle. Third, for each lane, the space-time diagram for each signal cycle (Fig. 1) was obtained using both the filtered trajectories and the cycle timing. This space-time diagram was then used to calculate various traffic parameters at the signal-cycle level, including traffic volume, shock wave area, shock wave speed, and platoon ratio. Lastly, the estimated cycle-related parameters were inputted into a real-time safety model (i.e., **model 6** in Table 1) to predict the number of rear-end conflicts at the cycle level. The model predicts the number of conflicts per a signal cycle using the traffic volume, the shock wave area, the shock wave speed, and the platoon ratio of this cycle.

4.4. Validation results

The aforementioned dynamic traffic parameters as well as the number of rear-end conflicts were extracted from simulation at each of the selected intersections for the 9 -h analysis period (9:00 am to 6:00 pm). The safety performance of the trained RS-ATSC was compared with that of the real-world benchmark ASC. Overall, the RS-ATSC led to positive safety impacts in terms of reducing rear-end conflicts. Fig. 6 shows the conflict rate for each hour at the two selected intersections with both ASC and RS-ATSC. As shown in the figure, when the RS-ATSC was implemented instead of the ASC, the average conflict rate was decreased from 0.165 to 0.08 conflict/vehicle/hour at the first intersection and from 0.17 to 0.11 conflict/vehicle/hour at the second intersection.

The real-time variation of traffic conflicts was also investigated at each approach of both intersections. The number of rear-end conflicts were estimated for each lane per each signal cycle from **model 6** in Table 1. The conflict rate (conflict/second) was then estimated by dividing the number of conflicts at each cycle by the cycle length. Figs. 7 and 8 show the real-time variation of the conflict rate for each approach at the first and the second intersection, respectively. Moreover, the cumulative numbers of rear-end conflicts throughout the 9 -h analysis period for both intersections are shown in Figs. 9 and 10. Compared to the benchmark ASC, the proposed RS-ATSC reduced the number of rear-end conflicts significantly at both intersections. The magnitude of reduction in the rear-end conflicts was not the same for all approaches. Some approaches experienced higher reduction in the number of conflicts, such as the southbound approach at the first intersection (Figs. 7 and 9) and the westbound approach at the second intersection (Figs. 8 and 10). At the same time, some approaches showed less reduction in the number of conflicts, such as the southbound approach at the second intersection (Figs. 8 and 10). More importantly, the results do not indicate any increase in the cumulative number of conflicts at any approach. This means that the RS-ATSC not only improved the overall safety level of each intersection, but also it did not deteriorate the safety level of any individual approach.The overall comparison of the performance of the proposed RS-ATSC to the benchmark ASC is reported in Table 3. For the 9 -h analysis period, the RS-ATSC led to a significant improvement in the safety level of both analyzed intersections. The overall rate of rear-end conflicts (i.e., the total number of conflicts

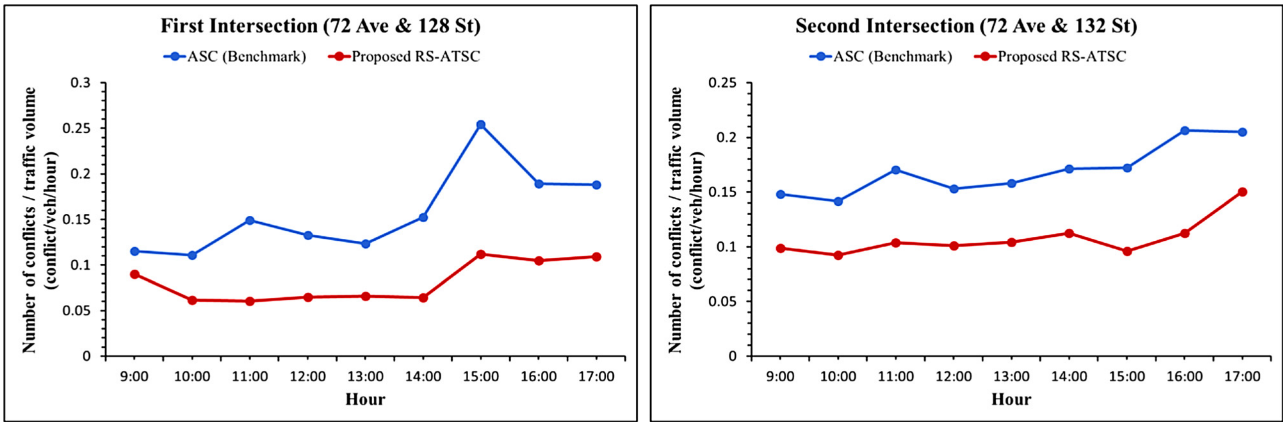

Fig. 6. Traffic conflicts at the selected intersections before and after implementing the proposed RS-ATSC.

normalized by the exposure) was reduced by 49 % and 37 % at the first and the second intersection, respectively. The proposed algorithm also improved the operational performance of the analyzed intersections. Compared to the benchmark ASC, the RS-ATSC reduced the average delay time by 12 % and 23 % for the first and the second intersection, respectively. The number of stops, the maximum queue length, and the 95th percentile of queue length were also reduced by 47 %, 23 %, and 51 % at the first intersection; and by 27 %, 17 %, and 28 % at the second intersection, respectively.

It is noteworthy that the performance results provided in Table 3 were derived based on the geometric and traffic characteristics of the selected intersections. These results can vary if the algorithm is implemented to other intersections with different characteristics. It should also be noted that the V2I DSRC domain was assumed to be 225 m. Using a higher value of this domain can potentially improve the algorithm's performance. In addition, the algorithm's performance was evaluated with the assumption that the V2X communication system is ideal. Practically, several sources of error may exist in connected systems, including position error, packet delay, and packet loss. These connectivity error sources can impact the algorithm's performance.

The reduction in the conflict rate confirms the positive safety impact of the proposed RS-ATSC algorithm. In addition, the improvement in the average delay time, the number of stops, and the queue length indicates that the algorithm has a positive mobility impact. Thus, the proposed RS-ATSC algorithm improves both the safety and the mobility performance of the analyzed intersections. In other words, the algorithm optimizes safety without deteriorating mobility.

Given the aforementioned validation results, the proposed RS-ATSC algorithm can be implemented in real-world to optimize the safety of signalized intersections using CVs real-time data. When implemented to a specific intersection, the RS-ATSC algorithm can be designed to continue learning itself using real-world traffic and geometric data of this intersection. The Q-table should be re-trained to properly consider the site characteristics as well as the local driving behavior. Minor

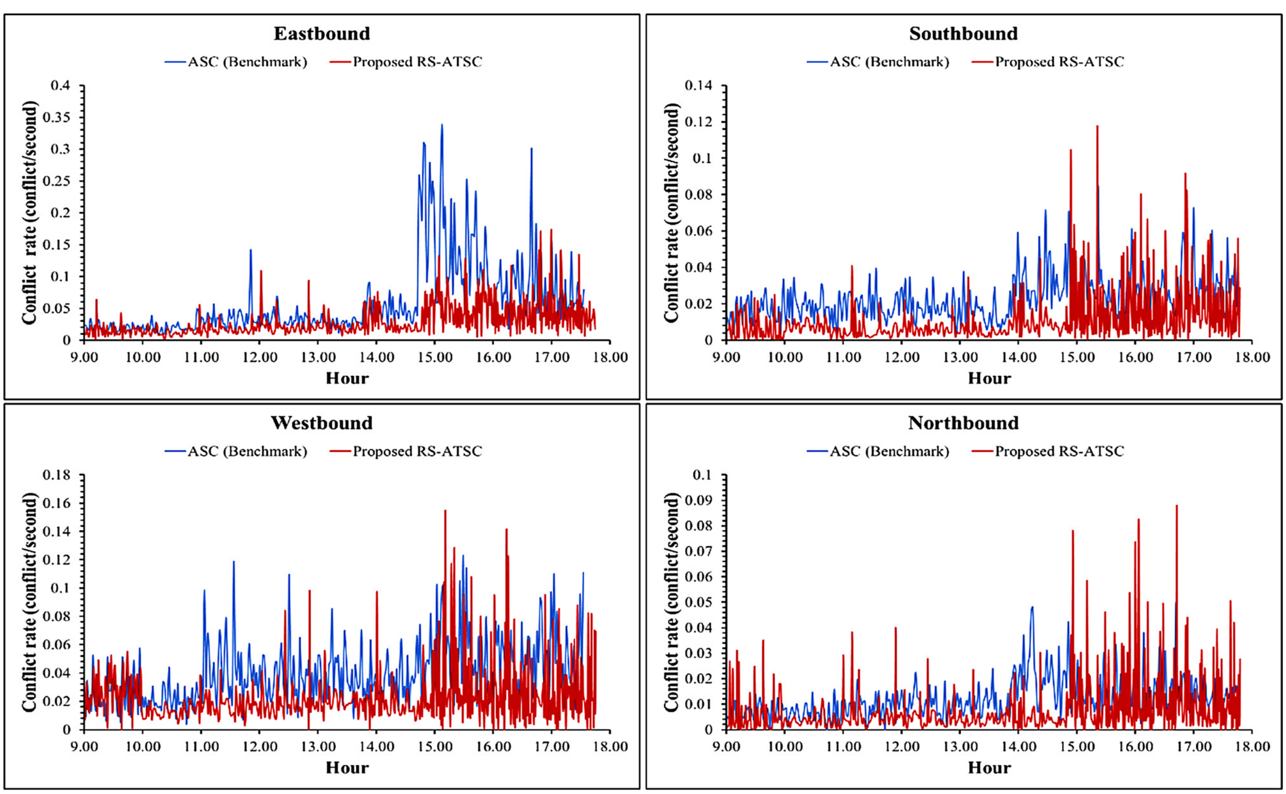

Fig. 7. Real-time variation of the conflict rate at each approach of the first intersection (72 Ave and 128 St) before and after implementing the proposed RS-ATSC.

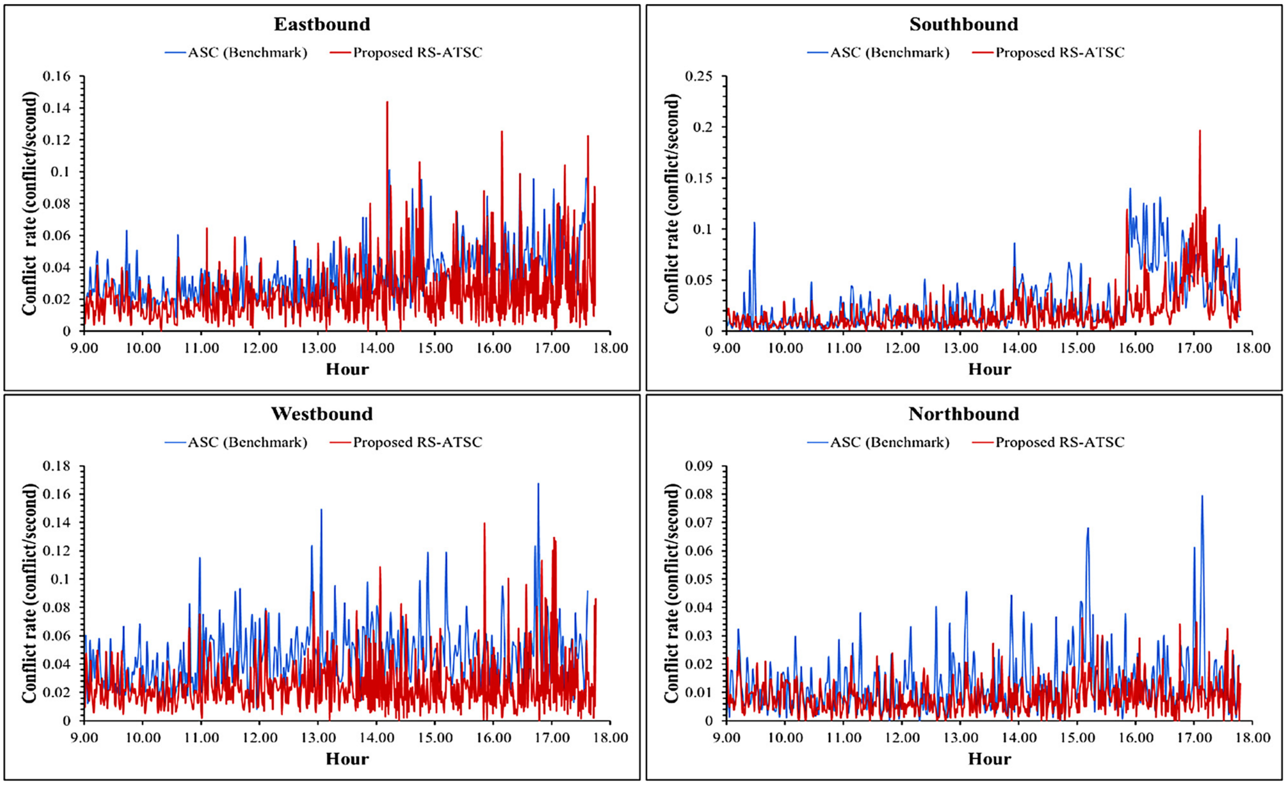

Fig. 8. Real-time variation of the conflict rate at each approach of the second intersection (72 Ave and 132 St) before and after implementing the proposed RS-ATSC.

modifications should also be applied in the algorithm to match the intersection specifications, such as the number of approaches, the number of lanes, the number of phases, the phase sequence, and the signal timing constraints (e.g., minimum/maximum green time, yellow time, all-red time). Considering these site-specific data can potentially lead to better safety and mobility performances.

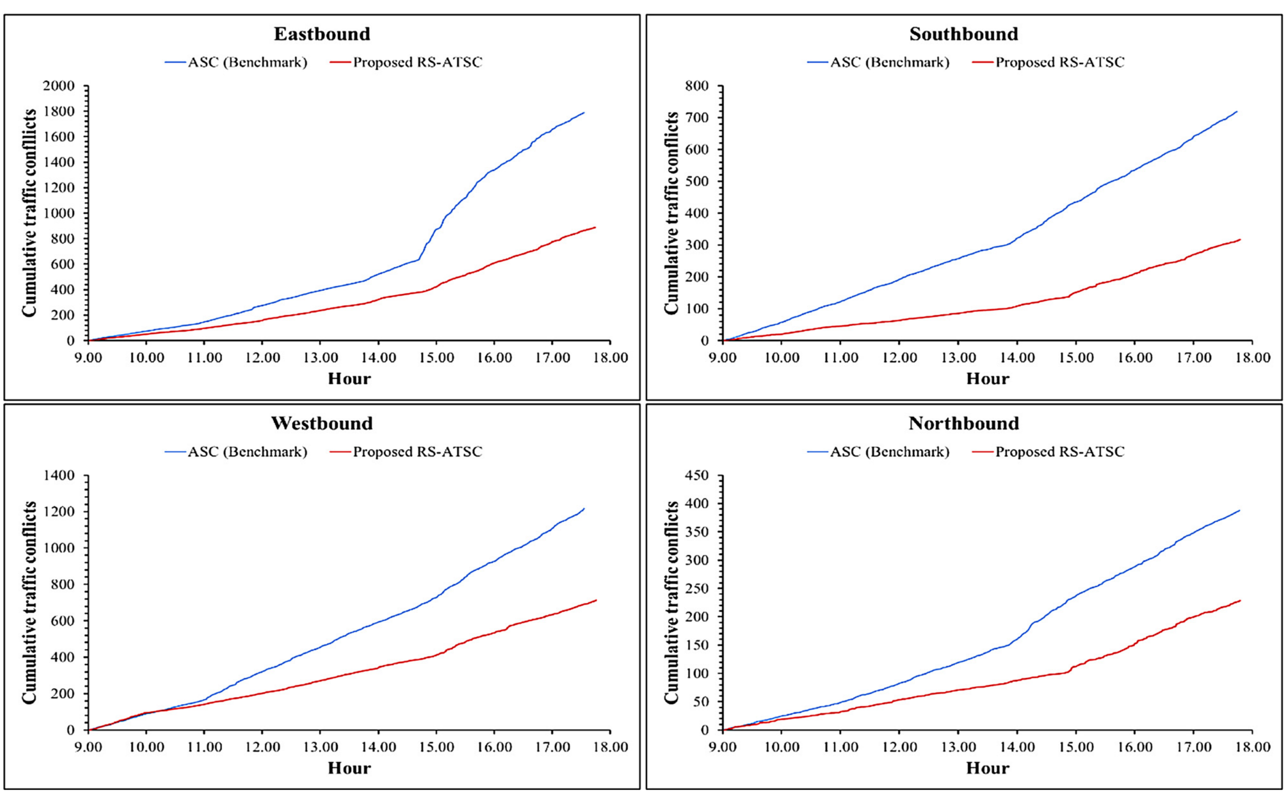

Fig. 9. Cumulative traffic conflicts each approach of the first intersection (72 Ave and 128 St) before and after implementing the proposed RS-ATSC.

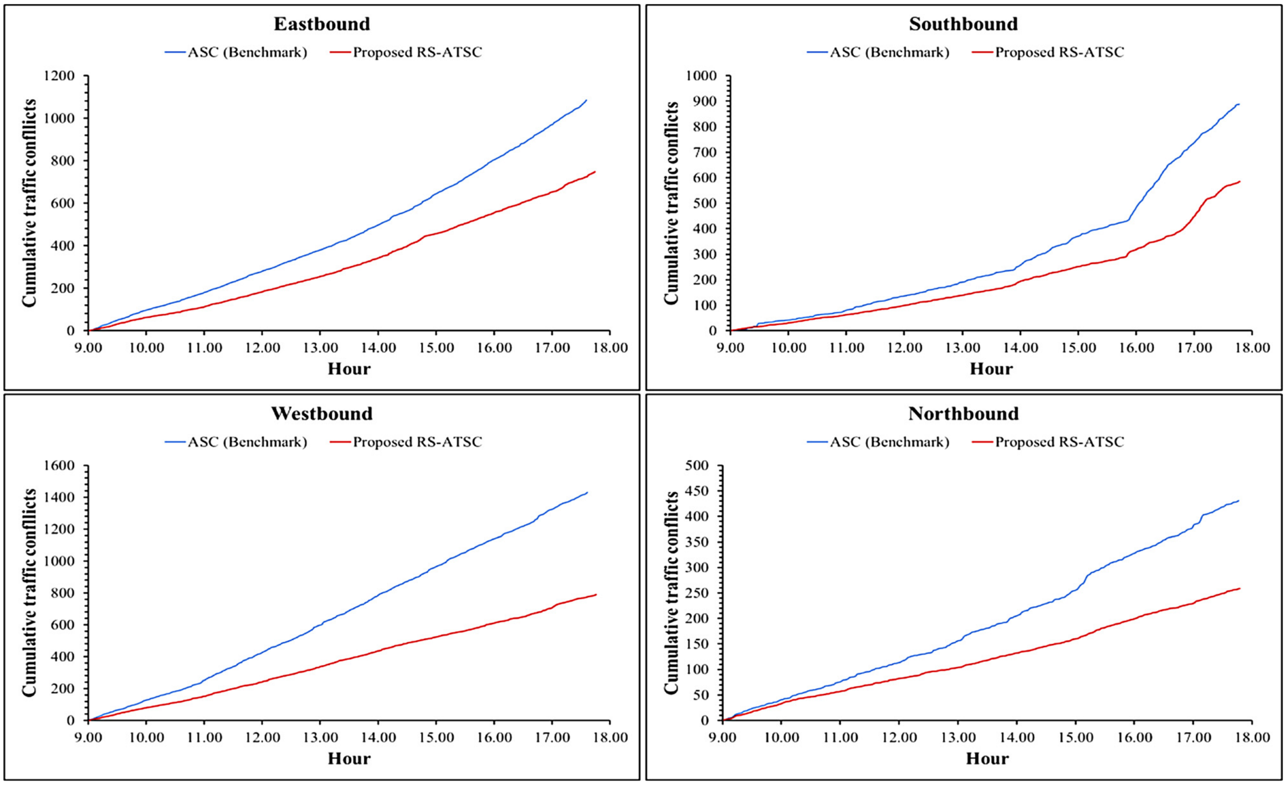

Fig. 10. Cumulative traffic conflicts each approach of the second intersection (72 Ave and 132 St) before and after implementing the proposed RS-ATSC.

4.5. Effect of CVs market penetration rate

The CVs technology is supposed to be deployed gradually. A mix of CVs and conventional vehicles is expected to exist in road networks during the transition period that predates the full deployment of CVs technology. Subsequently, it is not feasible to validate any ATSC algorithm assuming that all vehicles are CVs. Rather, various market penetration rates (MPRs) of CVs should be considered. Therefore, in this study, we investigated the performance of the proposed RS-ATSC at the two selected intersections under various MPRs of CVs, ranging from 10 % to 100 %.

Various MPRs of CVs were represented in the VISSIM model by creating a new vehicle class called "connected vehicle" and varying traffic composition percentages of each traffic input point. Under imperfect MPRs, detailed trajectories are available only for CVs. Therefore, when implementing the RS-ATSC algorithm with a specific MPR, the algorithm captures instantaneous vehicle information (i.e., vehicle trajectory) from vehicles with the "connected vehicle" class only. To define the real-time state (i.e., the conflict rate at each approach) and select the best action, the cycle-related parameters V, A, S_{12}, and P (Fig. 1) were estimated using the captured CVs trajectories and were inputted into the real-time safety model (i.e., **model 6** in Table 1). The estimated traffic volume per cycle per lane (V) was corrected (multiplied by a magnification factor) before be inputted in the real-time safety model. This factor equals the reciprocal of the MPR value. The exact MPR value can be estimated in real time, given the number of CVs from the V2I communications and the total traffic counts from the counting detectors upstream each approach of the intersection. Unlike the traffic volume, the estimated values of A, S_{12}, and P per cycle per lane are not in a linear relationship with the MPR. Therefore, these values were applied directly (i.e., without correction) in the real-time safety model to estimate the conflict rate. However, it should be noted that the lower the MPR, the less accurate are the estimated values of these cycle-related parameters. After running the simulation for the 9 -h analysis period, all vehicle trajectories (connected and non-connected) were analyzed to evaluate the performance. The traffic conflict rate and the average delay time were estimated under each MPR and were compared to the benchmark ASC.

Figs. 11 and 12 show the average conflict rate and the average delay time of the analyzed intersections when the RS-ATSC is applied under various MPRs. The benchmark ASC is also illustrated for comparison. As shown in Fig. 11, the maximum safety benefit of the RS-ATSC is corresponding to the MPR of 100 % (i.e., all vehicles are connected). At this MPR, the conflict rate was reduced from 0.165 to 0.084 at the first intersection and from 0.173 to 0.109 at the second intersection. However, it should be noted that 90 % of these benefits can be achieved when the MPR is 50 %. Moreover, the MPR of 30 % seems sufficient to achieve about 50 % of the maximum safety benefit. On the other hand, the results of the average delay time (Fig. 12) emphasize that the RS-ATSC algorithm has a positive mobility impact. This means the algorithm optimizes safety without deteriorating mobility.

Overall, the proposed RS-ATSC algorithm can be very effective when the MPR of CVs is equal to or higher than 30 %. The higher the MPR value, the more the safety effectiveness of the algorithm. MPR values less than 20 % may not lead to significant safety benefits, since the algorithm cannot define the environment state with a reasonable accuracy due to the lack of real-time information on vehicle positions and speeds.

5. Summary and conclusions

In this paper, we present a novel adaptive traffic signal control algorithm (i.e., RS-ATSC) that optimizes safety of signalized intersections in real time using CVs data. The algorithm is based on real-time safety models developed in recent research (Essa and Sayed, 2018, 2019). The models use dynamic traffic parameters, such as the platoon ratio and the shock wave characteristics, to predict traffic conflicts and evaluate safety of signalized intersections in real time. To the best of our knowledge, the presented RS-ATSC is the first self-learning ATSC algorithm that uses CVs data to optimize traffic safety in real time (i.e.,

Table 3
Safety Optimization Results of the Proposed RS- ATSC Algorithm Compared to the ASC.

First intersection (128 St & 72 Ave)
Analysis period of 9 h in total (9:00 am to 6:00 pm)

Approach*		EB	SB	WB	NB	Overall
Traffic Volume		7886	5562	7201	4239	24,888
Rear-end conflicts per exposure (conflict/vehicle)	1- Benchmark ASC	0.226	0.128	0.169	0.093	0.165
	2- Proposed RS-ATSC	0.110	0.058	0.095	0.053	0.084
	% Reduction/increase**	−51%	−55%	−44%	−43%	−49%
Total number of stops	1- Benchmark ASC	6414	4488	5760	2491	19,153
	2- Proposed RS-ATSC	3853	2482	2451	1428	10,214
	% Reduction/increase**	−40%	−45%	−57%	−43%	−47%
Maximum queue length (vehicle)	1- Benchmark ASC	18.5	12.8	14.5	9.3	18.5
	2- Proposed RS-ATSC	14.3	10.8	11	7.5	14.3
	% Reduction/increase**	−23%	−16%	−24%	−19%	−23%
95th percentile of queue length (vehicle)	1- Benchmark ASC	15.2	6.6	9.5	5.1	14.3
	2- Proposed RS-ATSC	7.2	5	6.1	3	7.1
	% Reduction/increase**	−52%	−24%	−36%	−41%	−51%
Average delay time per vehicle (second/vehicle)	1- Benchmark ASC	24.7	17.7	21.6	15	20.5
	2- Proposed RS-ATSC	18.7	19.8	13.3	21.6	18
	% Reduction/increase**	−24%	+12 %	−38%	+44 %	−12%

Second intersection (132 St & 72 Ave)
Analysis period of 9 h in total (9:00 am to 6:00 pm)

Approach*		EB	SB	WB	NB	Overall
Traffic Volume		7526	3527	7650	3077	21,780
Rear-end conflicts per exposure (conflict/vehicle)	1- Benchmark ASC	0.142	0.234	0.187	0.141	0.173
	2- Proposed RS-ATSC	0.099	0.163	0.104	0.086	0.109
	% Reduction/increase**	−30%	−30%	−44%	−39%	−37%
Total number of stops	1- Benchmark ASC	5178	5899	5985	2321	19,383
	2- Proposed RS-ATSC	3505	5242	3697	1705	14,149
	% Reduction/increase**	−32%	−11%	−38%	−27%	−27%
Maximum queue length (vehicle)	1- Benchmark ASC	14	23	15.3	16.5	23
	2- Proposed RS-ATSC	11	19	10.8	9.5	19
	% Reduction/increase**	−21%	−17%	−30%	−42%	−17%
95th percentile of queue length (vehicle)	1- Benchmark ASC	8.5	22	8.5	10.6	20.3
	2- Proposed RS-ATSC	6.1	16	5.1	4.2	14.5
	% Reduction/increase**	−29%	−27%	−40%	−61%	−28%
Average delay time per vehicle (second/vehicle)	1- Benchmark ASC	15.4	64.2	20.7	32.5	28.5
	2- Proposed RS-ATSC	16	46.7	15.5	20.3	22
	% Reduction/increase**	+4%	−27%	−25%	−37%	−23%

* EB: eastbound, WB: westbound, NB: northbound, SB: southbound.
** Positive values indicate increase and negative values indicate reduction.

safety is evaluated and optimized over a very short time period, a few seconds).

The RS-ATSC algorithm was developed using the RL technique. Specifically, the Q-learning off-policy TD method was applied. In the developed Q-learning algorithm, the state is defined using the rate of rear-end conflicts (number of conflicts per second) upstream each approach within a specific V2I DSRC domain. The action space includes only two actions representing the fixed phasing sequence. Thus, every time step, the RL agent decides whether to extend the current green time or to switch the green light to the next phase. The reward function is defined by the summation of conflict rates estimated from all approaches as a penalty. In addition, several constraints are considered to ensure the feasibility of implementing the proposed algorithm in real-world. This includes accommodating the yellow time, the all-red time, and the minimum/maximum green time, whenever they are necessary.

To train the RS-ATSC algorithm, an isolated intersection was modelled in the simulation platform VISSIM. The VISSIM model was controlled by an external program to emulate the CVs environment as well as real-time signal changes. In the learning process, the simulation model was run using random traffic volumes for 420 episodes, each includes 20000 s. The RS-ATSC agent converged to the optimized policy after about 200 episodes.

The trained RS-ATSC algorithm was validated using real-world traffic data of two signalized intersections in the city of Surrey, British Columbia. The algorithm's performance was compared with the performance of the existing fully actuated traffic signal control (ASC). Overall, the validation results showed that the proposed RS-ATSC algorithm outperforms the real-world ASC. When implementing the RS-ATSC, the overall rate of rear-end conflicts (i.e., the total number of conflicts normalized by the exposure) was decreased by 49 % and 37 % at the first and second intersection, respectively. In addition to these safety benefits, the RS-ATSC has positive mobility impacts. Compared to the benchmark ASC, the RS-ATSC reduced the average delay time by 12 % and 23 % for the first and second intersection, respectively. The number of stops, the maximum queue length, and the 95th percentile of queue length were also reduced by 47 %, 23 %, and 51 % at the first intersection; and by 27 %, 17 %, and 28 % at the second intersection, respectively.

The RS-ATSC algorithm was also tested under various market penetration rates of CVs. Although the maximum safety benefit is corresponding to the MPR of 100 %, the results showed that 90 % of this benefit can be achieved when the MPR is 50 %. Moreover, the MPR of 30 % seems sufficient to achieve about 50 % of the maximum safety benefit. MPR values less than 20 % may not lead to significant safety benefits, since the algorithm cannot define the environment state with a reasonable accuracy due to the lack of real-time information on vehicle positions and speeds.

In conclusion, the proposed RS-ATSC is a promising and feasible algorithm that can adapt traffic signals to optimize real-time safety in the CVs environment. The algorithm outperforms the traditional

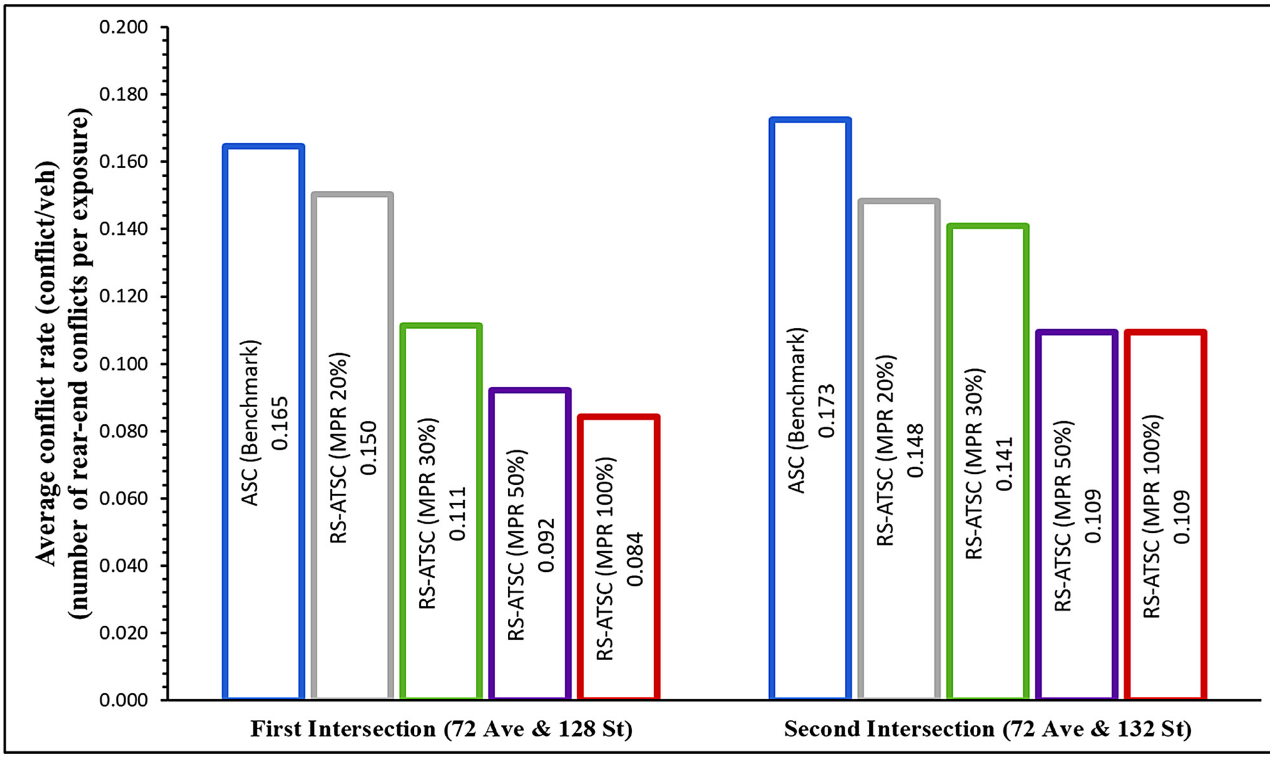

Fig. 11. The effect of the CVs MPR value on the average conflict rate at the selected intersections when implementing the proposed RS-ATSC.
*For clarity, only MPRs with a considerable change in the conflict rate are shown. For example, MPRs 60–90 % almost have the same conflict rate as MPR 50; therefore, they are not shown here.

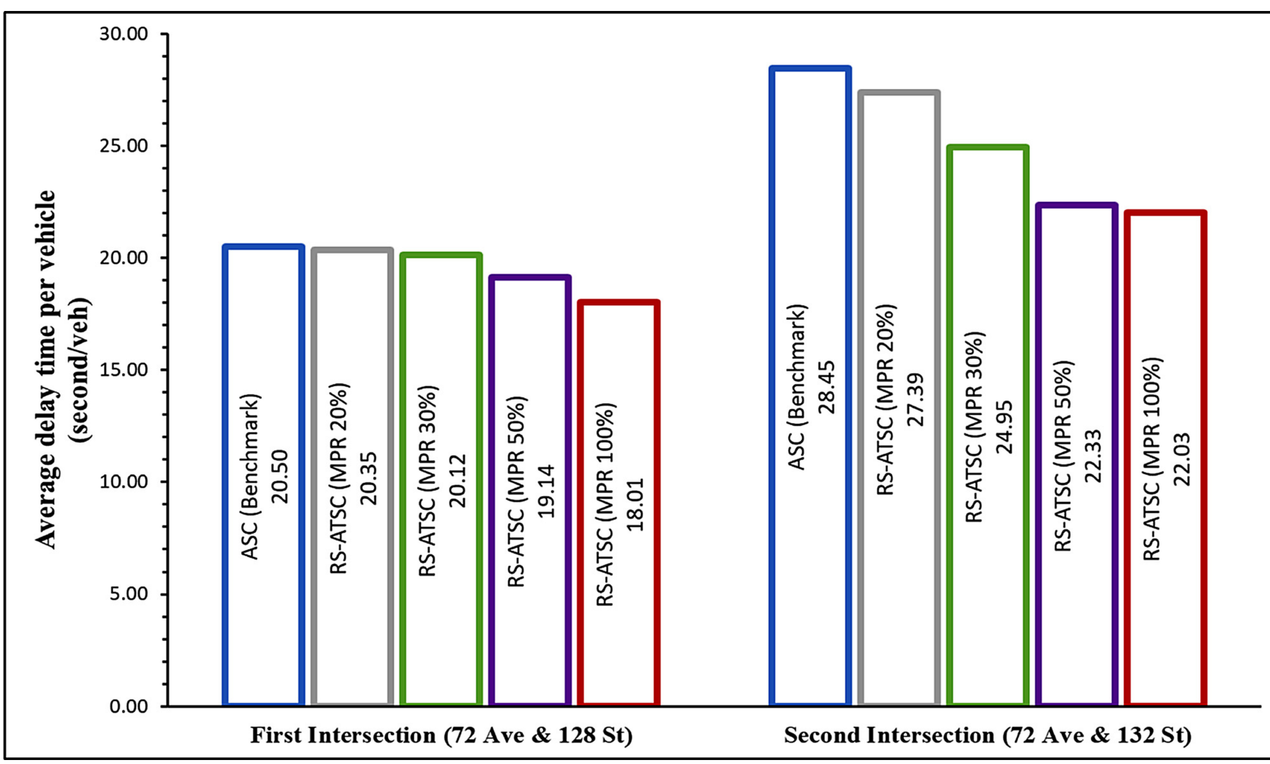

Fig. 12. The effect of the CVs MPR value on the average delay time at the selected intersections when implementing the proposed RS-ATSC.

actuated traffic signal control in terms of the produced number of rear-end traffic conflicts. The proposed RS-ATSC algorithm can be very effective when the MPR of CVs is equal to or higher than 30 %. The higher the MPR value, the more the safety effectiveness of the algorithm. More importantly, when implemented to a specific intersection, the RS-ATSC algorithm can be designed to continue learning itself using real-world traffic and geometric data of this intersection. The Q-table should be re-trained to properly consider the site characteristics as well as the local driving behavior. Minor modifications should also be applied in the algorithm to match the intersection specifications, such as the number of approaches, the number of lanes, the number of phases, the phase sequence, and the signal timing constraints. Considering these site-specific data can potentially lead to better safety and mobility performances.

It should be noted that although the RS-ATSC algorithm reduces delay times, its mobility performance cannot be considered the optimum. The reason is that the RS-ATSC is a safety-oriented algorithm whose optimized policy is based on minimizing the number of traffic conflicts to optimize safety. Other ATSC algorithms that consider minimizing delay times as a primary objective can lead to a better mobility performance. In fact, traffic delays are an essential issue since congestion occurs more frequently and leads to significant economic and environmental cost. Meanwhile, traffic safety as well is a fundamental issue due to high collision frequencies and severities at signalized intersections and their associated enormous social and economic costs. Therefore, both mobility and safety should be considered as fundamental optimization objectives in ATSC algorithms. Since previous research has been focused on optimizing delays, the main contribution of this research is to present a new algorithm that optimizes safety without deteriorating mobility (without increasing delays). Based on the results of this research, safety and mobility of signalized intersections seem to be non-conflicting objectives, although their optimum designs may not be the same. The developed RS-ATSC algorithm can further be modified to incorporate both safety and mobility in a multi-objective optimization problem. In such a problem, a weight can be assigned to each objective based on its associated cost (e.g., savings resulted from decreasing delays or collisions). These weights can vary among different locations and jurisdictions.

Several future areas of research are suggested to address some limitations of the proposed RS-ATSC algorithm. First, the algorithm's performance needs to be tested under non-ideal V2X communication systems. The results' sensitivity to several sources of error in connected systems (e.g., position error, packet delay, and packet loss) should be investigated. Second, the algorithm can be extended to model multiple intersections (e.g., a corridor or a network) instead of one isolated intersection. In this case, the signal coordination must be considered. Third, the algorithm's performance should be investigated under extremely oversaturated conditions where the queue length exceeds the V2I DSRC domain. Fourth, the state space in the algorithm can be expanded to be a continuous state space by converting the Q-matrix to a neural network (i.e., deep reinforcement learning). Fifth, it is suggested to investigate the results' sensitivity to various parameters, such as the discount factor, the update time-interval, and the V2I DSRC domain. Sixth, incorporating other conflict types, such as crossing and merging conflicts, is recommended. Moreover, safety measures other than traffic conflicts, such as the risk of collision or the predicted number of crashes (Zheng et al., 2019a, 2019b), can be used to represent real-time traffic safety and define the reward function in the RS-ATSC. Lastly, it is worthwhile to develop a multi-objective RL ATSC algorithm that includes both safety and mobility as two primary objectives in the real-time signal optimization.

CRediT authorship contribution statement

Mohamed Essa: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Tarek Sayed:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge the contributions of TransLink (the South Coast British Columbia Transportation Authority) and the Insurance Corporation of British Columbia (ICBC) for their support to this research.

References

AASHTO, 2010. Highway Safety Manual. Transportation Research Board, Washington, D.C.
Abdulhai, B., Kattan, L., 2003. Reinforcement learning: introduction to theory and potential for transport applications. Can. J. Civ. Eng. 30 (6), 981–991.
Abdulhai, B., Pringle, R., Karakoulas, G., 2003. Reinforcement learning for true adaptive traffic signal control. J. Transp. Eng. 129 (3), 278–285.
Al Islam, S., Hajbabaie, A., 2017. Distributed coordinated signal timing optimization in connected transportation networks. Transp. Res. Part C Emerg. Technol. 80, 272–285.
Arel, I., Liu, C., Urbanik, T., Kohls, A., 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4 (2), 128–135.
Balaji, P., German, X., Srinivasan, D., 2010. Urban traffic signal control using reinforcement learning agents. IET Intell. Transp. Syst. 4 (3), 177–188.
Bramberger, M., Brunner, J., Rinner, B., Schwabach, H., 2004. Real-time video analysis on an embedded smart camera for traffic surveillance. Proceedings. RTAS10th IEEE Real-Time and Embedded Technology and Applications Symposium 174–181.
Brys, T., Pham, T., Taylor, M., 2014. Distributed learning and multi-objectivity in traffic light control. Conn. Sci. 26 (1), 65–83.
Camponogara, E., Kraus, W., 2003. Distributed learning agents in urban traffic control. Portuguese Conference on Artificial Intelligence 324–335.
Dowling, R., Skabardonis, A., Alexiadis, V., 2004. Traffic Analysis Toolbox, Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software (No. FHWA-HRT-04-040). Federal Highway Administration. Office of Operations, United States.
Dutta, U., Lynch, J., Dara, B., Bodke, S., 2010. Safety Evaluation of the SCATS Control System (No. RC-1545K). Michigan Ohio University Transportation Center, Detroit.
El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. IEEE Trans. Intell. Transp. Syst. 14 (3), 1140–1150.
El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2014. Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. J. Intell. Transp. Syst. 18 (3), 227–245.
Essa, M., Sayed, T., 2015a. Simulated traffic conflicts: do they accurately represent field-measured conflicts? Transp. Res. Rec. 2514 (1), 48–57.
Essa, M., Sayed, T., 2015b. Transferability of calibrated microsimulation model parameters for safety assessment using simulated conflicts. Accid. Anal. Prev. 84, 41–53.
Essa, M., Sayed, T., 2016. A comparison between PARAMICS and VISSIM in estimating automated field-measured traffic conflicts at signalized intersections. J. Adv. Transp. 50 (5), 897–917.
Essa, M., Sayed, T., 2018. Traffic conflict models to evaluate the safety of signalized intersections at the cycle level. Transp. Res. Part C Emerg. Technol. 89, 289–302.
Essa, M., Sayed, T., 2019. Full Bayesian conflict-based models for real time safety evaluation of signalized intersections. Accid. Anal. Prev. 129, 367–381.
Essa, M., Sayed, T., 2020. Comparison between surrogate safety assessment model and real-time safety models in predicting field-measured conflicts at signalized intersections. Transp. Res. Rec. 2674 (3), 100–112.
Essa, M., Sayed, T., Reyad, P., 2019. Transferability of real-time safety performance functions for signalized intersections. Accid. Anal. Prev. 129, 263–276.
Feng, Y., Head, K., 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.
Fink, J., Kwigizile, V., Oh, J., 2016. Quantifying the impact of adaptive traffic control systems on crash frequency and severity: Evidence from Oakland County, Michigan. J. Saf. Res. 57, 1–7.
Formosa, N., et al., 2020. Predicting real-time traffic conflicts using deep learning. Accid. Anal. Prev. 136, 105429.
Fyfe, M., Sayed, T., 2017. Safety evaluation of connected vehicles for a cumulative travel time adaptive signal control microsimulation using the surrogate safety assessment model. Transportation Research Board 96th Annual Meeting 17-01651.
Gartner, N.H., 1983. OPAC: a demand-responsive strategy for traffic signal control. Transp. Res. Rec. 906, 75–81.
Gettman, D., Head, L., 2003a. Surrogate Safety Measures From Traffic Simulation Models-Final Report-FHWA-RD-03-050. s.l.. FHWA.
Gettman, Douglas, Head, L., 2003b. Surrogate safety measures from traffic simulation models. Transp. Res. Rec. 1840, 104–115.
Gettman, et al., 2008. Surrogate Safety Assessment Model and Validation: Final Report. s.l.: FHWA-HRT-08-051. .
Gong, Y., Abdel-Aty, M., Cai, Q., Rahman, M., 2019. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. Transp. Res. Interdiscip. Perspect. 1, 100020.
Goodall, N., Smith, B., Park, B., 2013. Traffic signal control with connected vehicles.

Transp. Res. Rec. 2381, 65–72.
Guler, S., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transp. Res. Part C Emerg. Technol. 46, 121–131.
Guo, Y., et al., 2019a. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: combined dynamic programming and shooting heuristic approach. Transp. Res. Part C Emerg. Technol. 98, 54–72.
Guo, Q., Li, L., Ban, X., 2019b. Urban traffic signal control with connected and automated vehicles: a survey. Transp. Res. Part C Emerg. Technol. 101, 313–334.
Hayward, J.C., 1972. Near-miss determination through use of a scale of danger. Highway Res. Rec. 384.
Head, K.L., Mirchandani, P.B., Sheppard, D., 1992. Hierarchical framework for real-time traffic control. Transportation Research Record. J. Transp. Res. Board 1360, 82–88.
Hunt, P., Robertson, D., Bretherton, R., Winton, R., 1981. SCOOT-A Traffic Responsive Method of Coordinating Signals (No. LR 1014 Monograph). Transport and Road Research, Crowthorne, England.
Jiang, H., et al., 2017. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment. Transp. Res. Part C Emerg. Technol. 79, 290–307.
Kamal, M., et al., 2015. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. IEEE Trans. Intell. Transp. Syst. 16 (3), 1136–1147.
Ke, R., et al., 2018. Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow. IEEE Trans. Intell. Transp. Syst. 20 (1), 54–64.
Khattak, Z., Magalotti, M., Fontaine, M., 2018. Estimating safety effects of adaptive signal control technology using the Empirical Bayes method. J. Saf. Res. 64, 121–128.
Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intell. Transp. Syst. 13 (1), 81–90.
Lee, J., Park, B., Malakorn, K., So, J., 2013a. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. Transp. Res. Part C Emerg. Technol. 32, 193–206.
Lee, J., Park, B., Yun, I., 2013b. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. J. Transp. Eng. 139 (10), 1020–1029.
Liang, X., Guler, S., Gayah, V., 2020. An equitable traffic signal control scheme at isolated signalized intersections using Connected Vehicle technology. Transp. Res. Part C Emerg. Technol. 110, 81–97.
Lodes, M., Benekohal, R., 2013. Safety Benefits of Implementing Adaptive Signal Control Technology: Survey Results (FHWA-ICT-12-020). Illinois Center for Transportation, Urbana, Illinois.
Luyanda, F., et al., 2003. ACS-Lite algorithmic architecture: applying adaptive control system technology to closed-loop traffic signal control systems. Transp. Res. Rec. 1856 (1), 175–184.
Ma, J., et al., 2016. Estimation of crash modification factors for an adaptive traffic-signal control system. J. Transp. Eng. 142 (12), 04016061.
Mirheli, A., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2019. A consensus-based distributed trajectory control in a signal-free intersection. Transp. Res. Part C Emerg. Technol. 100, 161–176.
NCHRP, 2015. Signal Timing Manual - Second Edition. The National Academies Press, Washington DC.
PTV, 2015a. PTV VISSIM 7 User Manual. PTV AG., Karlsruhe, Germany.
PTV, 2015b. PTV VISSIM 7 INTRODUCTION TO THE COM API. PTV Planung Transport Verkehr AG, Karlsruhe, Germany.
Rafter, C., Anvari, B., Box, S., Cherrett, T., 2020. Augmenting traffic signal control systems for urban road networks with connected vehicles. IEEE Trans. Intell. Transp. Syst. 21 (4), 1728–1740.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 779–788.
Richter, S., Aberdeen, D., Yu, J., 2007. Natural actor-critic for road traffic optimisation. Adv. Neural Inf. Process. Syst. 1169–1176.
Sabra, Z., et al., 2010. Balancing Safety and Capacity in an Adaptive Signal Control System—Phase 1 (No. FHWA-HRT-10-038). Federal Highway Administration, Mclean, United States.
Sabra, Z., Gettman, D., Henry, R., Nallamothu, V., 2013. Enhancing Safety and Capacity in an Adaptive Signal Control System—Phase 2. Rep. No. FHWA-PROJ-10-0037. Federal Highway Administration, Washington, DC.
Salkham, A., Cunningham, R., Garg, A., Cahill, V., 2008. A collaborative reinforcement learning approach to urban traffic control optimization. Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. IEEE Computer Society 560–566.
Shabestary, S., Abdulhai, B., 2018. Deep learning vs. discrete reinforcement learning for adaptive traffic Signal control. 21st International Conference on Intelligent Transportation Systems (ITSC) IEEE 286–293.
Shoufeng, L., Ximin, L., Shiqiang, D., 2008. Q-learning for adaptive traffic signal control based on delay minimization strategy. IEEE International Conference on Networking, Sensing and Control 687–691.
Sims, A., 1979. SCAT-the Sydney coordinated adaptive traffic system. In: Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems. Pacific Grove, California, USA.
Stevanovic, A., Kergaye, C., Haigwood, J., 2011. Assessment of surrogate safety benefits of an adaptive traffic control system. Indianapolis. 3rd International Conference on Road Safety and Simulation.
Stevanovic, A., Stevanovic, J., Kergaye, C., 2013. Optimization of traffic signal timings based on surrogate measures of safety. Transp. Res. Part C Emerg. Technol. 32, 159–178.
Stevanovic, A., Stevanovic, J., So, J., Ostojic, M., 2015. Multi-criteria optimization of traffic signals: mobility, safety, and environment. Transp. Res. Part C Emerg. Technol. 55, 46–68.
Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. s.l.. The MIT Press.
Tageldin, A., Sayed, T., Zaki, M., Azab, M., 2014. A safety evaluation of an adaptive traffic signal control system using computer vision. Adv. Transp. Stud.: Int. J. 2 (Special Issue), 83–96.
Theofilatos, A., 2017. Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. J. Saf. Res. 61, 9–21.
Theofilatos, A., Yannis, G., Vlahogianni, E., Golias, J., 2017. Modeling the effect of traffic regimes on safety of urban arterials: the case study of Athens. J. Traffic Transp. Eng. 4 (3), 240–251.
Thorpe, T., Anderson, C., 1996. Traffic Light Control Using Sarsa With Three State Representations. s.l.: Technical Report, Citeseer. .
U.S. Department of Transportation, 2015. ITS Research 2015-2019: Connected Vehicle (Connected Vehicles: Benefits, Roles, Outcomes). [Online] Available at: https://www.its.dot.gov/research_areas/WhitePaper_connected_vehicle.htm [Accessed 12 May 2017]. .
Watkins, C., 1989. Learning From Delayed Rewards (Doctoral dissertation). King's College, Cambridge.
Watkins, C., Dayan, P., 1992. Q-learning. Mach. Learn. 279–292.
Wiedemann, R., 1974. Simulation des Straßenverkehrsflusses. Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft, pp. 8.
Wiering, M., 2000. Multi-agent reinforcement learning for traffic light control. Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000) 1151–1158.
Xu, B., et al., 2017. V2I based cooperation between traffic signal and approaching automated vehicles. In: Intelligent Vehicles Symposium (IV). IEEE. pp. 1658–1664.
Yang, K., Guler, S., Menendez, M., 2016. Isolated intersection control for various levels of vehicle technology: conventional, connected, and automated vehicles. Transp. Res. Part C Emerg. Technol. 72, 109–129.
Yuan, J., Abdel-Aty, M., 2018. Approach-level real-time crash risk analysis for signalized intersections. Accid. Anal. Prev. 119, 274–289.
Yuan, J., et al., 2018. Real-time crash risk analysis of urban arterials incorporating bluetooth, weather, and adaptive signal control data. In: Proceedings of the Transportation Research Board 97th Annual Meeting Transportation Research Board. Washington, D.C.
Zhang, G., Avery, R., Wang, Y., 2007. Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. Transp. Res. Rec. 1993 (1), 138–147.
Zheng, L., Sayed, T., Essa, M., 2019a. Bayesian hierarchical modeling of the non-stationary traffic conflict extremes for crash estimation. Anal. Methods Accid. Res. 23, 100100.
Zheng, L., Sayed, T., Essa, M., 2019b. Validating the bivariate extreme value modeling approach for road safety estimation with different traffic conflict indicators. Accid. Anal. Prev. 123, 314–323.
Zheng, L., Sayed, T., Essa, M., Guo, Y., 2019c. Do Simulated Traffic Conflicts Predict Crashes? An Investigation Using the Extreme Value Approach. IEEE Intelligent Transportation System Conference, Auckland, New Zealand.