Intelligent Traffic Signal Control: Using Reinforcement Learning with Partial Detection

Rusheng Zhang *, Akihiro Ishikawa *, Wenli Wang *, Benjamin Striner †, and Ozan Tonguz *
* Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA
† Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA

***Abstract*—Intelligent Transportation Systems (ITS) have attracted the attention of researchers and the general public alike as a means to alleviate traffic congestion. Recently, the maturity of wireless technology has enabled a cost-efficient way to achieve ITS by detecting vehicles using Vehicle to Infrastructure (V2I) communications.**

Traditional ITS algorithms, in most cases, assume that every vehicle is observed, such as by a camera or a loop detector, but a V2I implementation would detect only those vehicles with wireless communications capability. We examine a family of transportation systems, which we will refer to as 'Partially Detected Intelligent Transportation Systems'. An algorithm that can act well under a small detection rate is highly desirable due to gradual penetration rates of the underlying wireless technologies such as Dedicated Short Range Communications (DSRC) technology. Artificial Intelligence (AI) techniques for Reinforcement Learning (RL) are suitable tools for finding such an algorithm due to utilizing varied inputs and not requiring explicit analytic understanding or modeling of the underlying system dynamics.

In this paper, we report a RL algorithm for partially observable ITS based on DSRC. The performance of this system is studied under different car flows, detection rates, and topologies of the road network. Our system is able to efficiently reduce the average waiting time of vehicles at an intersection, even with a low detection rate.

***Index Terms*—Reinforcement Learning, Artificial Intelligence, Intelligent Transportation Systems, Partially Detected Intelligent Transportation Systems, Vehicle-to-Infrastructure Communications**

I. Introduction

[1] Traffic congestion is a daunting problem that affects the daily lives of billions of people in most countries across the world [1]. Over at least the past 30 years, many attempts to alleviate this problem in the form of intelligent transportation systems have been designed and demonstrated [2]–[9]. Among these different approaches, some use real time traffic information measured or collected by video cameras or loop detectors and optimize the cycle split of a traffic light accordingly [10]. Unfortunately, such intelligent traffic control schemes are expensive and, therefore, they exist only at a small percentage of intersections in the United States, Europe, and Asia.

Recently, several cost-effective approaches to implement intelligent transportation systems were proposed by leveraging the fact that Dedicated Short-Range Communication (DSRC) technology will be mandated by US Department of Transportation (DoT) and will be implemented in the near future [11]–[13]. DSRC technology is potentially a much cheaper technology for detecting the presence of vehicles on the approaches of an intersection. However, at the early stages of deployment, only a small percentage of vehicles will be equipped with DSRC radios. Since this adoption stage could last several years due to increasing vehicle life [14], new control algorithms that can handle partial detection of DSRC-equipped vehicles are required.

One potential AI algorithm is deep reinforcement learning (DRL), which has recently been explored by several groups [15], [16]. These results showed an improvement in terms of waiting time and queue length experienced at an intersection in a fully-observable environment. Hence, in this paper, we investigate this promising approach **given a partially observable environment**. As expected, we observe an asymptotically improving result as we increase the penetration rate of DSRC-equipped vehicles.

In this paper, we explore the capability of DRL to solve the DSRC-based partially detected intelligent traffic signal control systems. Though we mainly consider DSRC detection in this context, the scheme described here is generic enough to be used for any other possible forms of partially detected intelligent traffic signal control systems, such as vehicle detection based on RFID, Bluetooth Low Energy 5.0 (BLE 5.0), and LTE. We perform extensive simulations to analyze different aspects of the RL method. Our results clearly show that AI, in general, and reinforcement learning, in particular, is capable of providing an excellent traffic management scheme that is able to reduce the waiting time of commuters at a given intersection, even at a low penetration rate.

II. Related Works

Traffic signal control using Artificial Intelligence (AI), especially reinforcement learning (RL), has been an active field of research for the last 20 years. In 1994, Mikami, et al. proposed distributed reinforcement learning (Q-learning) using a Genetic Algorithm to present a traffic signal control scheme that effectively increased throughput of a road network [17]. Due to the limitations of computational power in 1994, however, it could not be implemented at that time.

Bingham proposed RL for parameter search of a fuzzy-neural traffic signal controller for a single intersection [18],

[1] The research reported in this paper was partially funded by King Abdulaziz City of Science and Technology (KACST), Riyadh, Kingdom of Saudi Arabia

while Choy et al. adapted RL on the fuzzy-neural system in a cooperative scheme, achieving adaptive control for a large area [19]. These algorithms are based on RL, but the major goal of RL is parameter tuning of the fuzzy-neural system. Abdulhai et al. proposed the first true adaptive traffic signal which learns to control the traffic signal dynamically based on a Cerebellar Model Articulation Controller (CMAC), as a Q-estimation network [20]. Silva, et.al. and Oliveira et.al. then proposed a context-detector (CD) in conjunction with RL to further improve the performance under non-stationary traffic situations [21], [22]. Several researchers have focused on multi-agent reinforcement learning for implementing it on a large scale [23]–[26].

Recently, with the development of GPU and computation power, Deep Reinforcement Learning has become an attractive method in several fields. Several attempts have been made using Deep Q-learning for ITS, including [15], [16], [27], [28]. These results show that a DQN based Q-learning algorithm is capable of optimizing the traffic in an intelligent manner.

Traditional intelligent traffic signal systems use loop detectors, magnetic detectors and cameras for improving the performance of traffic lights. In the past few decades, various adaptive traffic systems were developed and implemented. Some of these traffic systems such as SCOOT [4], SCATS [3], are based on dynamic traffic coordination [5], and can be viewed as a traffic-responsive version of TRANSYT [2]. These systems optimize the offsets of traffic signals in the network, based on current traffic demand, and generate 'green-wave' for major car flow. Meanwhile, some other model-based systems has been proposed, including OPAC [6], RHODES [7], PRODYN [8]. These systems use both the current traffic arrivals and the prediction of future arrivals, and choose a signal phase planning which optimize the objective functions. While these systems work efficiently, they do have some significant shortcomings. The cost of these systems are generally high [29]. Considering SCATS, for example, the initial cost of the system is $20,000 to $30,000 per intersection, and $28,800 per mile per year, not to mention that the installation will cost an extra $20,000 per intersection [30]. The cost is due to the fact that these systems use loop detectors and video cameras to detect vehicles. They are generally expensive and hard to install and maintain.

Even though RL yields impressive results for these cases, it does not outperform current systems. Hence, the progress of these algorithms, while interesting, is of limited impact, since traditional ITS systems perform comparably.

Meanwhile, as Dedicated Short-Range Communications start to be installed on vehicles in the United States, traffic signal control schemes based on such technology have become a rising field, as the cost is significantly lower than a traditional ITS [11]–[13]. Within these schemes, a system known as Virtual Traffic Lights (VTL) is very attractive, as it proposes an infrastructure-free DSRC-based solution, by installing traffic control devices in vehicles and having the vehicles decide the right-of-way at an intersection locally. Different aspects of VTL technology, including algorithm design, system simulation, deployment policy, and carbon emission have been studied by different research groups in the last few years [11], [31]–[41]. However, a VTL system requires all vehicles in the road network to be equipped with DSRC device, therefore, a transition scheme for the current transportation systems to smoothly transition to VTL system is needed.

On the other hand, several methods have been proposed for floating vehicle data gathered from Global Position System (GPS) that are used to detect, estimate and predict traffic states based on fuzzy logic, Genetic Algorithm (GA), Support Vector Mechine (SVM) and other statistical learning algorithms [42]–[47]. These works show that it is possible to optimize traffic control based on partially observed data (such a system is formally introduced in section III).

There are a few systems currently available using partial detection. For example, COLOMBO is one of the projects in the European Union (EU) that focuses on low-penetration rate of DSRC-equipped vehicles [48]–[50]. The system uses information provided by V2X technology and feed the information to a traffic management system. Since COLOMBO cannot directly react to real-time traffic flow (the detected and undetected vehicles have the same performance), under low to medium car flow it will NOT achieve optimum performance as the optimal strategy under low-to-medium car flow has to react according to detected car arrivals. Another very recent system is DSRC-Actuated Traffic Lights, which is one of our previous implementations using DSRC radio for traffic control. The designed prototype of this system was publicly demonstrated at an intersection in Riyadh, Saudi Arabia, only 5 months ago, in July 2018 [51], [52]. DSRC-Actuated Traffic Lights, however, is based on the arrival of each vehicle, and hence works well under low to medium car flow rates, but it does not work well under high car flow rate.

The main contributions of this paper are:

1) Explore a new kind of intelligent system that is based on **partial detection** of DSRC-equipped vehicles, which is a cost-effective alternative to current ITS and an important problem not addressed by traditional ITS.
2) Propose a transition scheme to VTL. Not only do we reduce the average commute time for all end users, but those users with DSRC have much lower commute time, which attracts additional users to have DSRC capability.
3) Design a new RL-based traffic control algorithm and system design that performs well under low penetration ratio and detection rates.
4) Provide a detailed performance simulation and analysis. Our simulation and analysis show that, under a low detection rate, the system can perform almost as good as an ITS that employs full detection. This is a promising solution considering its cost-effectiveness.

III. Problem Formulation

The rapid development of the Internet of Things (IoT) has created new technology applicable for sensing vehicles for intelligent transportation systems. Other than DSRC, applicable technologies include, but are not limited to, RFID, Bluetooth, Ultra-Wide Band (UWB), Zigbee, and even cellphone apps such as Google Map [53]–[55]. All these systems are more economical than traditional loop detectors or cameras.

Performance-wise, most of these systems are able to track vehicles in a continuous way, while loop detectors can only detect the presence of vehicles, suggesting that a system based on wireless communications would be able to utilize finer-grained information.

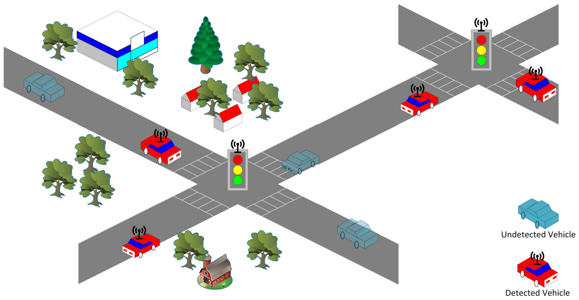


Fig. 1. Illustration of Partially Detected Intelligent Transportation System

Unfortunately, the transportation systems mentioned above have a critical shortcoming: they are not able to detect vehicles unequipped with the communication device. Within these systems, only a portion of all vehicles are detectable, unlike a traditional ITS. As this is a common characteristic for several aforementioned traffic signal control systems, we denote these traffic systems collectively as *Partially Detected Intelligent Traffic Signal Control System* (PD-ITSCS).

Figure.1 gives an illustration of a PD-ITSCS. There are two kinds of vehicles in the system: the red vehicles are equipped with a communication device which is able to communicate with the corresponding device on the traffic lights, so that the traffic lights are able to detect these vehicles; the blue vehicles, on the other hand, are not equipped with a communication device, and hence undetectable by the traffic lights. In a PD-ITSCS, both kinds of vehicles co-exist in the system. The traffic lights, based on the information from the detected vehicles, decide the current phase at the intersections, in order to minimize the delay at the intersection for both detected vehicles and undetected vehicles.

This paper aims to build a traffic control scheme that:

1) performs well even with a low detection rate;
2) accelerates the transition to a higher adoption rate and therefore a higher detection rate.

In the rest of the paper, for notational convenience, we choose one of the typical PD-ITSCS, the transportation system based on DSRC radios, as an example. The detected vehicles are vehicles equipped with DSRC radios, and the undetected vehicles are those unequipped with DSRC radios. Observe that other kinds of PD-ITSCS are analogous, thus making the methodologies described in this paper adaptable for them as well.

IV. Methodology

A. Q-Learning Algorithm

We refer to Watkins [56] for a detailed explanation of general reinforcement learning and Q-learning but we will provide a brief review in this section.

The goal of reinforcement learning is to train an agent that interacts with the environment by selecting the action in a way that maximizes the future reward. At every time step, the agent gets the state (the current observation of the environment) and reward information (the quantified indicator of performance from the last time step) from the environment and makes an action. During this process, the agent tries to optimize (maximize/minimize) the cumulative reward for its action policy. The beauty of this kind of algorithm is the fact that it doesn't need any supervision, since the agent observes the environment and tries to optimize its performance without human intervention.

RL algorithms come in two categories: policy based algorithms such as Trust Region Policy Optimization (TRPO) [57], Advantage Actor Critic (A2C) [58], Proximal Policy Optimization (PPO) [59] that optimize the policy that maps from states to actions; and value based algorithms such as Q-learning [56], double Q-Learning [60] , and soft Q-learning [61] that directly maximize the cumulative rewards. While policy based algorithms have achieved good results and will potentially be applicable for the problem proposed in this paper [62], [63], in this paper, we choose deep Q-learning algorithm.

In the Q-learning approach, the agent learns a 'Q-Value', denoted $Q(s_t, a_t)$, which is a function of observed state s_t and action a_t that outputs the expected cumulative discounted future reward. Here, t denotes the discrete time index. The cumulative discounted future reward is defined as:

$$Q(s_t, a_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ...$$

Here, r_t is the reward at each time step, the meaning of which needs to be specified according to the actual problem, and $\gamma < 1$ is the discount factor. At every time step, the agent updates its Q function by an update of the Q value:

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max Q(s_{t+1}, a_t) - Q(s_t, a_t))$$

In most cases, including the traffic control scenarios of interest, due to the complexity of the state space and action space, deep neural networks can be used to approximate the Q function. Instead of updating the Q value, we use the value:

$$Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max Q(s_{t+1}, a_t) - Q(s_t, a_t))$$

as the output target of a Q network and do a step of back propagation on the input of s_t, a_t.

We utilized two known methods to stabilize the training process [64], [65]:

1) Two Q-networks are maintained, a target Q-network and an on-line Q network. Target Q-network is used to approximate the true Q-values, and the on-line Q-network is back-propagated every step. In the training period, the agent makes decision with the target Q-network, the results from each time instance are used to update the on-line Q-network. At periodic intervals, on-line Q networks weights are synchronized with the target Q-network. This will keep the agent's decision network relatively stable, instead of changing at every step.

2) Instead of training after every step an agent has taken, past experience was stored in a memory buffer and training data was sampled from the memory for a certain batch size. This experience replay aims to break the time correlation between samples [66].

In this paper, we train the traffic lights agents using a Deep Q-network (DQN) [66]. With the Q-learning algorithm described above, our work focuses on the definition of agents' actions and the assignment of the states and rewards, which is discussed in the the following subsection IV-B.

B. Parameter Modeling

We consider a traffic light controller, which takes reward and state observation from the environment and chooses an action. In this subsection, we introduce our design of actions, rewards, and states for the aforementioned PD-ITSCS problem.

1) Agent action: In our context, the relevant action of the agent is either to keep the current traffic light phase, or to switch to the next traffic light phase. At every time step, the agent makes an observation and takes action accordingly, achieving intelligent control of traffic.

2) Reward: For traffic optimization problems, the goal is to decrease the average traffic delay of commuters in the network, by using traffic light phasing strategy $\mathcal{S}$. Specifically, find the best traffic light phasing strategy $\mathcal{S}$, such that $t_{\mathcal{S}} - t_{\min}$ is minimum, where $t_{\mathcal{S}}$ is the average travel time of commuters in the network, under the traffic control scheme $\mathcal{S}$, and $t_{\min}$ is the physically possible lowest average travel time. Consider traveling the same distance d,

$$d = \int_0^{t_{\mathcal{S}}} v_{\mathcal{S}}(t)dt = t_{\min}v_{\max}$$

Here, $v_{\max}$ is some maximum reasonable speed for the vehicle, such as the speed limit of the road in concern. Therefore,

$$t_{\min} = \frac{1}{v_{\max}} \int_0^{t_{\mathcal{S}}} v_{\mathcal{S}}(t)dt$$

$$\begin{aligned} t_{\mathcal{S}} - t_{\min} &= \int_0^{t_{\mathcal{S}}} 1dt - \frac{1}{v_{\max}} \int_0^{t_{\mathcal{S}}} v_{\mathcal{S}}(t)dt \\ &= \frac{1}{v_{\max}} \int_0^{t_{\mathcal{S}}} v_{\max} - v_{\mathcal{S}}(t)dt \end{aligned}$$

Therefore, to get minimum delay $t_{\mathcal{S}} - t_{\min}$ is equivalent to minimizing at each step t, for each vehicle:

$$\frac{1}{v_{\max}}[v_{\max} - v_{\mathcal{S}}(t)] \tag{1}$$

We note that this is equivalent to maximizing $v_{\mathcal{S}}(t)$, if the $v_{\max}$ on all roads for all cars are the same. If different vehicles have different $v_{\max}$, the reward function is taken as the arithmetic average of the function for all vehicles.

We define the statement in (1) as the penalty of each step. Our goal is to minimize the penalty of each step. Since reinforcement learning tries to maximize the reward (minimize penalty), we define the opposite number of the loss as the reward for the reinforcement learning problem:

$$r_t = -\frac{1}{v_{\max}}[v_{\max} - v_{\mathcal{S}}(t)] \tag{2}$$

In some cases, especially when the traffic flow is heavy, one can shape the rewards to guide the agent's action, such as avoiding big traffic jams [67]. This is certainly an interesting direction for future research.

3) State representation: For optimal decision making, a system should consider as much relevant information about traffic processes as possible. Traditional ITS only typically detect simple information such as the presence of vehicles. In partially detected ITS, only a portion of the vehicles are detected, but more specific information about these vehicles such as speed and position are available due to the capabilities of DSRC.

Reinforcement learning enables experimentation with many possible choices of inputs and input representations. Further research is required to determine the experimental benefits of each option and that goes beyond the scope of this paper. Based on initial experiments, for the purpose of this paper, we selected a state representation including the distance to the nearest vehicle at each approach, number of vehicles at each approach, amber phase indicator, current traffic light phase elapsed time and current time, as detailed in Table I. Note that current traffic light phase (green or red) is represented by a sign change in the per-lane detected car count and distance rather than by a separate indicator. In initial experiments, we observed slightly faster convergence using this distributed representation (sign representation) than a separate indicator (shown in Figure 5). We hypothesize that, in combination with Rectified Linear Unit (ReLU) activation, this encoding biases the network to utilize different combinations of neurons for different phases. ReLU units are active if the output is positive and inactive if the output is negative, so our representation may encourage different units to be utilized during different phases, accelerating learning. There are many possible representations and our experimentation with different representations is not exhaustive, but we found that Reinforcement Learning was able to handle several different representations with reasonable performance.

TABLE I
DETAILS OF STATE REPRESENTATION

Information	**Representation**
Detected car count	Number of detected vehicles in each approach
Distance to nearest detected vehicle	Distance to nearest detected vehicle on each approach; if no detected vehicle, set to lane length (in meters)
Current phase time	Duration from start of current phase to now (in seconds)
Amber phase	Indicator of amber phase; 1 if currently in amber phase, otherwise 0
Current time	Current time of day (hours since midnight), normalized from 0 to 1 (divided by 24)
Current phase	Detected car count and distance to nearest detected vehicle is negated if red, positive if green

C. System Design

We provide here one of the possible system realizations for the proposed scheme, based on Dedicated Short-Range

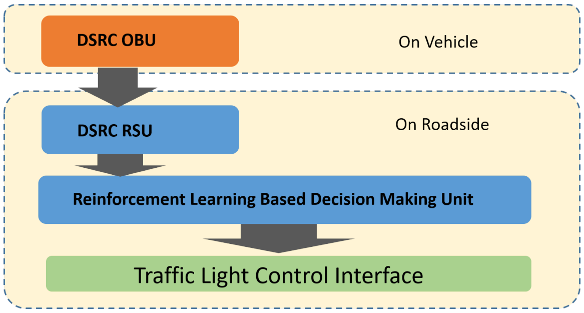

DSRC OBU
On Vehicle
DSRC RSU
On Roadside
Reinforcement Learning Based Decision Making Unit
Traffic Light Control Interface

Fig. 2. One possible system design for the proposed scheme

Communications (DSRC). The system has an 'On Roadside' unit and an 'On Vehicle' unit, as shown in Figure 2. DSRC RoadSide Unit(RSU) senses the Basic Safety Message (BSM) broadcast by the DSRC OnBoard Unit (OBU), parse the useful information out, and send them to the Reinforcement Learning Based Decision Making Unit. This unit will then make a decision based on the information provided by the RSU.

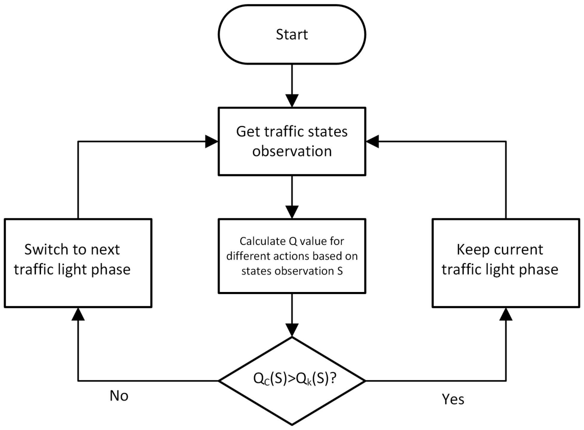

Start
Get traffic states observation
Calculate Q value for different actions based on states observation S
Switch to next traffic light phase
Keep current traffic light phase
$Q_C(S)>Q_k(S)$?
No
Yes

Fig. 3. Control logic of RL based decision making unit

Figure 3 gives a flow chart on how the RL based control unit makes the decision. As shown in the figure, control unit gets the state representation from the DSRC RSU every second, calculates the Q-value for all the possible actions and if the action of keeping the current phase has bigger Q-value, it retains the phase, otherwise, switches to the next phase.

Other than the main logic discussed above, a sanity check is performed on the agent: a mandatory maximum and minimum phase. If the current phase duration is less than the minimum phase time, the agent will keep the current phase no matter what action the DQN is choosing; similarly, if phase duration is larger or equal to maximum phase time, the phase will be forced to switch.

D. Implementation

In this section, we describe the design of the proposed scheme at the system level. The implementation of the system contains two phases, the training phase and the deployment phase. As shown in Figure 4, the agent is first trained with a simulator, which is then ported to the intersection, connected to the real traffic signal, after which it starts to control the traffic.

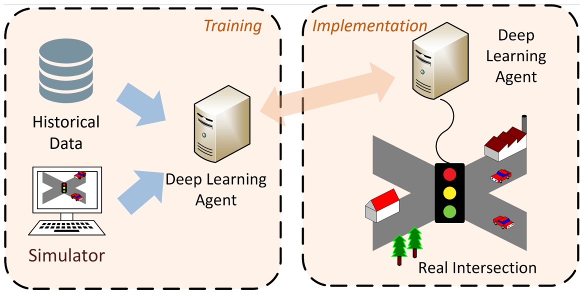


Fig. 4. The deployment scheme

1) Training phase: The agent is trained by interacting with a traffic simulator. The simulator randomly generates vehicle arrivals, then determines whether each vehicle can be detected by drawing from a Bernoulli distribution parameterized by p, the detection rate. In the context of DSRC-based vehicle detection systems, the detection rate corresponds to the DSRC penetration rate. The simulator obtains the traffic state s_t and calculates the current reward r_t accordingly, and feeds it to the agent. Using the Q-learning updating formula cited in previous sections, the agent updates itself based on the information from the simulator. Meanwhile, the agent chooses an action a_t, and forwards the action to the simulator. The simulator will then update, and change the traffic light phase according to agents indication. These steps are done repeatedly until convergence, at which point the agent is trained.

The performance of an agent relies heavily on the quality of the simulator. To obtain similar arrival pattern as the real world, the simulator generates car flow by the historical record of vehicle arrival rate on the same map of the real intersection. To address the variance in car flow in different parts of the day, current time of the day is also specified in the state representation, so that after training the agent is able to adapt to different car flow in different time of the day. Other factors that affect car flow, such as day of the week, could also be parameterized in the state representation.

The goal of training is to have the traffic control scheme achieve the shortest average commute time for all commuters. In the training period, the machine tries different control schemes and eventually converges to an optimal scheme which yields a minimum average commute time.

2) Deployment phase: In the deployment phase, the software agent is installed to the intersection for controlling the traffic light. Here, the agent will not update the learned Q-function, but simply control the traffic signal. Namely, the detector will feed the agent's current detected traffic state s_t; based on s_t, the agent chooses an action based on the trained Q-network and directs the traffic signal to switch/keep phase accordingly. This step is performed in real-time, thus enabling continuous traffic control.

V. Simulation and Performance Analysis

In this section, we give several scenarios of simulations to evaluate various aspects of the performance of the proposed scheme. The simulations are performed with SUMO, a microscopic traffic simulator [68]. Different scenarios are considered, in order to give a comprehensive analysis for the proposed scheme.

Qualitatively speaking, we see the performance of the agent reacting to the traffic in an intelligent way from the GUI. It makes reasonable decisions for the arriving vehicles. We demonstrate the performance of the agent after different periods of training in a video available here [69].

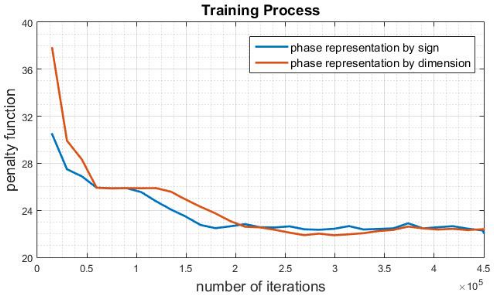


Fig. 5. Penalty function decreasing with number of iterations in training, the situation shown in the figure is plotted from training with dense car flow at a single intersection

Figure 5 shows typical training curves. Both phase representations have similar trends, but we do observe that the sign representation had a slightly faster convergence rate in every experiment (see section IV-B3).

We provide a quantitative analysis in the following subsections. Though currently there are no analytical results for PD-ITSCS, we can predict what will be observed by considering the following two extreme cases:

- When the car flow rate is extremely low, vehicles come to the intersection independently. For detected vehicles, the optimal traffic signal should switch phases on their arrival to yield zero waiting time, for the undetected vehicles, the traffic agent won't be able to do anything. In this case, vehicles can be considered as independent 'particles', and the optimal traffic agent react for each or their arrivals independently. Therefore, we should observe much better performance for the detected vehicles than those undetected vehicles, which corresponds to the cases shown in Figure. 7b.
- When the car flow rate is extremely heavy (at the point of saturation), the optimal traffic agent should take a completely different strategy, instead of only taking care of the detected vehicles, the agent should be aware of the fact that the detected vehicles are only representatives of the car flow, and react in a way that maximizes the overall waiting time. The waiting time of detected vehicles and undetected vehicles should be similar, because they are of the same car flow. The vehicles here should be considered as 'liquid' instead of 'particles' from the previous case. This can be seen in Figure 7a.

The rest of the section is organized as follows: subsection V-A evaluates the performance of the system under different detection rates. One should expect different performance for different car flow rates for the reasons mentioned above. Subsection V-B shows the results for different types of road topology, thus providing evidence that the agent trained is able to function with different arrival patterns. SubsectionV-C gives an estimate on the benefit of the designed agent during different times of the day. Finally, subsection V-D and V-E show that when the implementation scenario is slightly different from the training scenario, the performance of the designed agent is still reasonably good.

A. Performance for different detection rates

In this subsection, we present performance results under different detection rates, to qualify the performance of a partially observable ITS as the detection rate increases from 0% to 100%. We compare to the performance of a typical pre-timed signal with green phase duration of 24 seconds, shown in dashed lines as a simple reference.

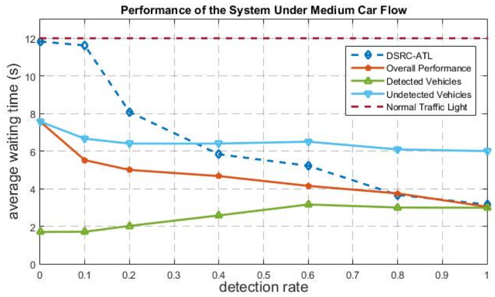


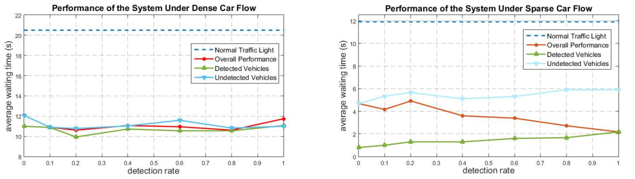
Fig. 6. Waiting time under different detection rate under medium car flow

Figure 6 shows a typical trend we obtained in simulations. The figure shows the waiting time of vehicles at a single intersection under the car flow from north, east, south, west to be 0.02 veh/s, 0.1 veh/s, 0.02 veh/s, 0.05 veh/s, respectively, with vehicles arriving as a Poisson process. One can make several interesting observations from this figure. First of all, the system under AI control is much better than the traditional pre-timed traffic signal, even under low detection rate. We can also observe that the overall waiting time (red line) within this system decreases as the detection rate increases. This is intuitive, since as more vehicles are detected, the more information the system has and thus the system is able to optimize the car flow better.

Additionally, from the figure one can observe that approximately 80% of the benefit happens in the first 20% of transition. This finding is quite significant in that we find a transition scheme that asymptotically gets better as the system gradually evolves to a 100% detection rate, and will be able to receive much of the benefit of the final stage system during the initial transition.

Another important observation is that during the transition, although the agent is rewarded for optimizing the overall average commute time for both detected and undetected vehicles, the detected vehicles (green line in Figure 6) have a lower commute time than undetected vehicles (blue line in Figure 6). This provides an interesting 'potential' or 'incentive' to the system, to transition from no vehicles equipped with the IoT device, to all vehicles equipped with the device. Drivers of those vehicles not yet equipped with the device now have a good reason and strong incentive to install one.

Here, we also compare with our previous designed system known as DSRC-ATL [51], which is an algorithm designed for dealing with partial detection under sparse to medium car flow. We see that though the algorithms exhibit similar trends, RL agents have better performance during the whole transition from 0 to 1 detection rate.



(a) Performance under dense flow (b) Performance under sparse flow

Fig. 7. Waiting time under different detection rate under dense and sparse car flow

Figure 7 shows the performance under the other two cases: when the car flow is very sparse (0.02 veh/s at each lane) or very dense (0.5 veh/s at each lane). For the sparse situation in Figure 7b, the trend is similar to the medium flow case shown in Figure 6.

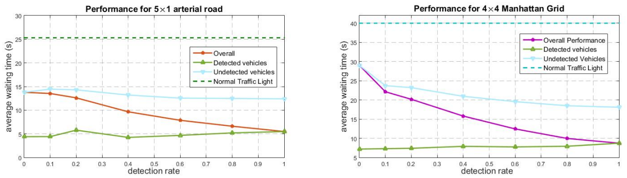
One can see from Figure 7a that under the dense situation, the curve becomes quite flat. This is because when car flow is high, detecting individual vehicles become less important. When many cars arrive at the intersection, car flow has 'liquid' qualities, as opposed to 'particle' qualities in the previous two situations. The trained RL agent is able to seamlessly transition from a 'particle arrival' optimization agent which handles random arrivals to a 'liquid arrival' optimization agent which handles macroscopic flow. This result shows that RL is able to capture the main factors that affect traffic system's performance and performs differently under different car arrival rates. Hence, RL provides a much desired adaptive behavior.

B. *Performance under different network topology*

Figure 6 shows a typical situation for the system at a single intersection with Poisson arrival; however, in most intersections, vehicles form platoons because of previous intersections. We also present results under other topology that create more complicated arrival patterns: arterial road topology and grid network topology.

We use a 5×1 arterial road structure to test performance under arterial road topology, where an arterial road from north to south crosses 5 intersections. The arrival rate on arterial road is 0.2 veh/s from north and 0.1 veh/s from south, the arrival rates on the other roads are all set to 0.05 veh/s. The vehicles on the arterial road, after going through one intersection, will automatically form clusters, and form a more realistic arrival pattern than Poisson arrival. For grid network topology, we choose 4x4 Manhattan Grid road structure for our simulations. This 2-dimensional structure will form more complicated arrival patterns at each intersection.

At each intersection, an independent RL agent is assigned with an independent Q-network. Each agent aims to optimize its own intersection separately, within the same traffic simulation.



(a) Performance for 5x1 arterial road (b) Performance for 4x4 Manhattan Grid

Fig. 8. Expected performance for arterial and network topology under medium car flow

Figure 8 shows, for two types of topology, the performance for medium car flow. Notice that the trend of the two figures are both similar to what we obtained in Figure 6. This indicates that the reinforcement learning agent is capable of handling different arrival patterns and achieves good performance under bulk arrivals.

C. *Performance of a whole day*

Section V-A examines the effect of flow rate on system performance. Since the car flow differs at different times of the day, we simulate an entire day of traffic. To generate realistic car flow of a day, we refer to the whole day car flow reported in [70]. To adapt the reported arrival rate to the simulation system, we multiply the car flow in [70] with a factor so that the peak volume matches the saturation flow rate of the simulated roads. Figure 9 shows the car flow rate we used for the simulation, the car flow reach peak on 8 am in the morning and 6 pm in the afternoon of 1.2 vehicles/s, the car flow of the regular hours is around 0.7 vehicles/s. It is worth mentioning that the car flow of different intersections in the real world might be very different, so the result presented here is just a reference of what the performance looks like under a typical traffic volume of a whole day.

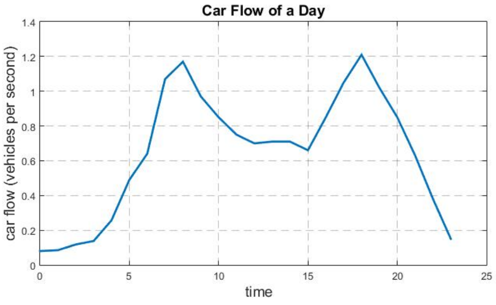


Fig. 9. Typical car flow in a day

Figure 10 shows the performance of different vehicles in a whole day. One can observe from this figure that the performance of 20% detection rate (red line) is very close to the performance of 100% detection rate (green line), at most times of the day (from 5am to 9pm). During rush hours, the system with 100% detection rate is almost the same as the system with 20% detection rate. Though a traffic system under 100% detection rate performs visibly better at midnight, the performance at that time is not as critical as the performance during the busier daytime. This result indicates

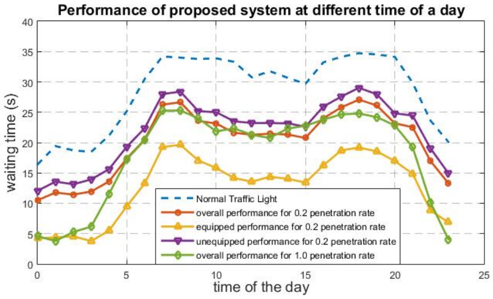


Fig. 10. Expected Performance by Time

that by detecting 20% of vehicles, we can perform almost the same as detecting all vehicles. But those detectable vehicles (yellow lines) will have a benefit against those undetectable vehicles (dash line).

These results are intuitive. With a large volume of cars, a low detection rate should still provide a relatively low-variance estimate of traffic flow. If there are few cars and a low detection rate, the estimate of traffic flow can have very high-variance. Late at night with only a single detected car, an ITS can give that car a green immediately, which would not be possible with an undetected car.

D. Sensitivity Analysis

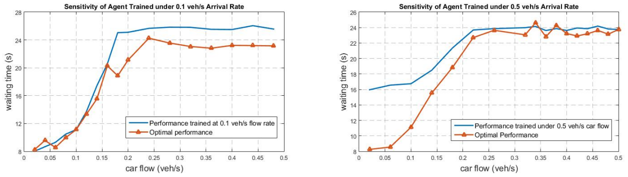
The results obtained above used agents trained and evaluated under the same environmental parameters, since traffic patterns only fluctuate slightly from day to day.

Below, we evaluate the sensitivity of the agents to two environmental parameters: the car flow and the detection rate.

1) Sensitivity of car flow: Figure 11 shows the agents' sensitivity to car flow. Figure 11a shows the performance of an agent trained under 0.1 veh/s car flow, operating at different flow rates. Figure 11b shows the sensitivity of an agent trained under 0.5 veh/s car flow. The blue curve in the figure is the trained agent's performance, while the red one is the performance of the optimal agent (the agent trained under that situation and tested under that situation). Both agents perform well over a range of flow rates. The agent trained under 0.1 veh/s flow can handle flow rates from 0 to 0.15 at near-optimal levels. At higher flow rates, it still performs reasonably well. The agent trained on 0.5 veh/s flow will perform reasonably from 0.25 veh/s to 0.5 veh/s, but under 0.25 veh/s, the agent will start to perform substantially worse than the optimal agent. Since traffic patterns are not expected to heavily fluctuate, these results give a strong indication that the agent trained by the data will be able to adapt to the environment even when the trained situation is slightly different.

2) Sensitivity of detection rate: In most situations, the detection rate can only be approximately measured. It is likely that an agent trained under one detection rate needs to operate under a slightly different detection rate, so we test the sensitivity of agents to detection rates.

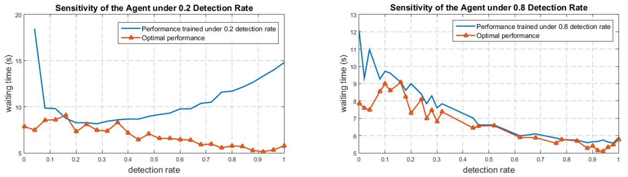
Figure 12 shows the sensitivity of two cases. Figure 12a shows the sensitivity of low detection rate (0.2), figure 12b shows the sensitivity under high detection rate (0.8).



(a) Sensitivity of agent trained under 0.1 veh/s flow rate

(b) Sensitivity of agent trained under 0.5 veh/s flow rate

Fig. 11. Sensitivity analysis of flow rate



(a) Sensitivity of agent trained under 0.2 detection rate

(b) Sensitivity of agent trained under 0.8 detection rate

Fig. 12. Sensitivity analysis of detection rate

We observe that the agent trained under 0.2 detection rate performs at an optimal level from 0.1 to 0.4 detection rate. The sensitivity upward is better than downward. This indicates that at early deployment of this system, it's better to under-estimate detection rate, since the agent's performance is more stable for the higher detection rate.

Figure 12b shows the sensitivity of the agent trained under high detection rate (0.8). We can see that the performance of this agent is at optimal level when detection rate is from 0.5 to 1. Though the sensitivity performance for an agent under low detection rate is different than the sensitivity under high detection rate, for both cases, the agent shows a level of stability, which means that as long as the detection rate used for training is not too different from the actual detection rate, the performance of the agent will not be affected a lot.

E. Robustness between training and deployment scenario

There are many difference between the training and the actual deployment scenario, as the simulator, though sufficiently sophisticated, will never able to take all the factors in the real scenario into account. This simulation aims to evaluate and verify that those minor factors, such as stop-and-go vehicles, arrival patterns and other factors won't affect the system in a major way. We choose a newly published realistic scenario known as Luxembourg SUMO Traffic (LuST) [71]. The scenario is generated on the real map of Luxembourg, the activity of vehicles are generated according to the demographic data published by the government. The authors of this scenario compared the generated traffic with a data set collected between March and April 2015 in Luxembourg, which contains 6,000,000 floating vehicles sample and achieved similar speed distributions, hence the LuST scenario has a high degree of reality.

In our simulation, we don't directly train the traffic light on the scenario; instead, we use this scenario as ground truth to evaluate the trained traffic light. The simulation steps we performed are as follows:

1) Choose a certain intersection from LuST with high rate of car flow (intersection -12408)
2) Measure the hourly traffic volume of that intersection
3) Build a simple intersection in a separate simulator and train a traffic agent with car flow generated by the new simulator, according to the hourly traffic volume measured in step 2.
4) Train an agent on the simplified scenario we built in step 3.
5) After training, we evaluate the performance on the original LuST scenario, by substituting the traffic agent of that intersection to the new traffic agent we trained.

It is worth mentioning here that this simulation follows the steps of actual implementation in real world (described in section IV-D), so the performance here can be considered as a reference for the performance of actual deployment when the simulator and real world have major differences in details.

Other than the difference in the map and car flow, there are more differences between training and evaluation, the scenario used for evaluation is rich in details. In the Table II, we list all the differences between the Lust scenario (for evaluation) and the simulator used for training.

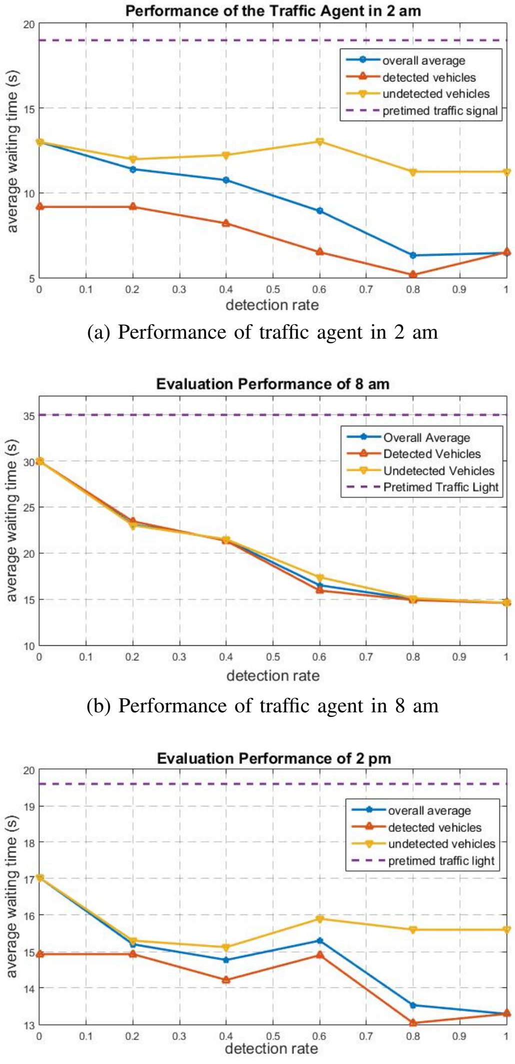
TABLE II
DEFERENCE IN TRAINING AND EVALUATION SCENARIO

	training	Evaluation (LuST)
Map topology	Simple straight street intersection	Real world map
Street length	125m for each approach	Different length for each approach
Car arrival pattern	Poisson	Bulk arrival when vehicle go through intersections
Car speed	Constant	Gaussian mixture distribution
Stop-and-go	No stop-and-go vehicles	Bus stops
U-turn vehicles	No U-turn	A small proportion of U-turn
Location where vehicle generated	End of the road	Anywhere of the road
Location of destination	End of the road	anywhere of the road, some might not even go through the intersection
Buses	No buses	Regular buses arrival with a bus stop close to the intersection
Vehicle passing	Almost no passing due to constant speed	Some vehicle passing due to the randomness of the speed

Notice that the simulator is sophisticated enough to take all the factors listed in the table into account. Here we intentionally introduce differences between training and evaluation. This is a judicious choice on our part. Our goal is to give a reasonable estimate of the performance in the real-world implementation where the simulation scenario is slightly different than the real-world scenario.

We choose three different times of the day to present the results:

1) *Midnight*: 2 AM in the morning, in this case, the car flow at intersection is sparse
2) *Rush-hours*: 8 AM in the morning, this is a situation where car flow is dense
3) *Regular hours*: 2 PM in the afternoon, this is the situation during regular hours, the car flow is in between of midnight car flow and rush hours car flow (medium car flow).



(a) Performance of traffic agent in 2 am

(b) Performance of traffic agent in 8 am

(c) Performance of traffic agent in 2 pm

Fig. 13. Performance of the agent in LuST scenario

Figure 13 shows the performance of the agent in the LuST scenario. We can clearly see that even though the evaluated situation is different from the training situation, we still observe: the performance improves asymptotically as the detection rate grows, which exhibits the same trend as we observed in V-A.

VI. Discussion

As the simulation results show, while all vehicles will experience a shorter waiting time under an RL-based traffic controller, detected vehicles will have a shorter commute time than undetected vehicles. This property makes it possible for hardware manufacturers, software companies, and vehicle manufacturers to help push forward the scheme, other than the Department of Transportation (DoT) alone, for the simple reason that all of them can profit from this system. For example, it would be valuable for a certain navigation app to advertise that their customers can save 30% on commute time.

Therefore, we view this technology as a new generation of intelligent transportation system, as it inherently comes with a lucrative commercial business model. The burden of spreading penetration rate of this system is distributed to a lot of companies, as opposed to the traditional ITS which puts the burden on the DoT alone. This makes it financially possible to have the system installed on most of the intersections in the city, as opposed to the current situation where only a small proportion of intersections are installed with ITS.

The mechanism of the system solution described will also make it possible to have dynamic pricing for different vehicles. Dynamic pricing refers to reserving certain roads during rush hour exclusively for paid users. This method has been scuttled by public or political opposition and only a few cities have implemented dynamic pricing [72], [73]. The method depends hugely on road topologies and public opinion. Those few successful examples, however, cannot be easily copied or adapted to other cities. In our solution, we can accomplish dynamic pricing in a more intelligent way, by simply providing vehicle detection as a service, since detected vehicles experience reduced commute times. There is no requirement to reserve roads, which makes the scheme extremely easy to deploy. For the end-users, they also have a choice; when they are in a hurry, they can pay more for lower commute time; if they aren't in a hurry and wouldn't mind to wait longer, they simply don't pay. The scheme itself, unlike the traditional congestion pricing scheme, will therefore not hurt the nonpaying users significantly. By enabling vehicle detection, the user receives slightly preferential treatment at traffic lights, instead of entirely reserving a road for paid users.

It is also worth mentioning that the system proposed in the paper is the first detailed attempt to show that the proposed approach has merit and significant benefits. However, further research is needed to make this AI-based Intelligent Traffic Control System more practical. First of all, currently, the system and simulation don't take pedestrians and pedestrian phases into account. Even though the pedestrian phase can be considered as a fixed time all-red transition phase, it will be interesting to train an agent to consider the waiting time of both pedestrians and drivers. Secondly, the system currently needs to be fully trained in a simulator; under the partial observation setup, the system will not be able to observe the reward, hence, it won't be able to do any incremental training after deployment. Clearly, this is a drawback or shortcoming of the proposed system. While extensive efforts have been made in this paper to show that the designed traffic agent trained in the simulator is robust and can adapt to a similar real environments, if the environment is significantly different from the training environment the performance of the traffic agent might be sub-optimal. While this is expected to be highly unlikely, it is a problem that requires further investigation. In the future, we want to overcome this difficulty by training the agent only using the partially observed reward. Another future direction would be to further develop the system to achieve multi-agent coordination so that, with the help of DSRC radios (or other forms of communications), traffic lights will be able to communicate with each other. Clearly, designing such a system will significantly improve the performance of PD ITSCS.

VII. Conclusion

In this paper, we have proposed reinforcement learning, specifically deep Q-learning, for traffic control with partial detection of vehicles. The results of our study show that reinforcement learning is a promising new approach to optimizing traffic control problems under partial detection scenarios, such as traffic control systems using DSRC technology. This is a very promising outcome that is highly desirable since the industry forecasts on DSRC penetration process seems gradual as opposed to abrupt.

The numerical results on a single intersection with sparse, medium, and dense arrival rates suggest that reinforcement learning is able to handle all kinds of traffic flow. Although the optimization of traffic on sparse arrival and dense arrival are, in general, very different, results show that reinforcement learning is able to leverage the particle property of the vehicle flow, as well as the liquid property, thus providing a very powerful overall optimization scheme.

We have shown promising results for single agent case that were subsequently extended to 5 intersections that the car arrival distribution will no longer be a Poisson process. The agents are able to deal with different arrival patterns, which shows a sense of robustness.

Acknowledgment

The authors would like to thank to Dr. Hanxiao Liu from Language Technology Institute, Carnegie Mellon University for informative discussions and a lot of suggestions to the methods reported in the paper. The authors would also like to thank Dr. Laurent Gallo from Eurecom, France and Mr. Manuel E. Diaz-Granados of Yahoo, US, for the initial attempt to solve this problem in 2016.

References

[1] "Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation," https://ops.fhwa.dot.gov/congestion_report/executive_summary.htm, 2017, [Online; accessed 19-Aug-2017].

[2] D. I. Robertson, "'tansyt'method for area traffic control," *Traffic Engineering & Control*, vol. 8, no. 8, 1969.

[3] P. Lowrie, "Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic," 1990.

[4] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The scoot on-line traffic signal optimisation technique," *Traffic Engineering & Control*, vol. 23, no. 4, 1982.

[5] J. Luk, "Two traffic-responsive area traffic control methods: Scat and scoot," *Traffic engineering & control*, vol. 25, no. 1, 1984.

[6] N. H. Gartner, *OPAC: A demand-responsive strategy for traffic signal control*, 1983, no. 906.

[7] P. Mirchandani and L. Head, "A real-time traffic signal control system: architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

[8] J.-J. Henry, J. L. Farges, and J. Tuffal, "The prodyn real time traffic algorithm," in *Control in Transportation Systems*. Elsevier, 1984, pp. 305–310.

[9] R. Vincent and J. Peirce, “'mova': Traffic responsive, self-optimising signal control for isolated intersections,” Tech. Rep., 1988.

[10] “Traffic light control and coordination,” https://en.wikipedia.org/wiki/Traffic_light_control_and_coordination, 2016, [Online; accessed 23-Mar-2016].

[11] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, “Self-organized traffic control,” in *Proceedings of the seventh ACM international workshop on VehicularAr InterNETworking*. ACM, 2010, pp. 85–90.

[12] N. S. Nafi and J. Y. Khan, “A vanet based intelligent road traffic signalling system,” in *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*. IEEE, 2012, pp. 1–6.

[13] V. Milanes, J. Villagra, J. Godoy, J. Simo, J. Pérez, and E. Onieva, “An intelligent v2i-based traffic management system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 49–58, 2012.

[14] “Average age of cars on u.s.” https://www.usatoday.com/story/money/2015/07/29/new-car-sales-soaring-but-cars-getting-older-too/30821191/, [Online; accessed 21-Aug-2017].

[15] W. Genders and S. Razavi, “Using a deep reinforcement learning agent for traffic signal control,” *arXiv preprint arXiv:1611.01142*, 2016.

[16] E. van der Pol, “Deep reinforcement learning for coordination in traffic light control,” Ph.D. dissertation, Masters thesis, University of Amsterdam, 2016.

[17] S. Mikami and Y. Kakazu, “Genetic reinforcement learning for cooperative traffic signal control,” in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 223–228.

[18] E. Bingham, “Reinforcement learning in neurofuzzy traffic signal control,” *European Journal of Operational Research*, vol. 131, no. 2, pp. 232–241, 2001.

[19] M. C. Choy, D. Srinivasan, and R. L. Cheu, “Hybrid cooperative agents with online reinforcement learning for traffic control,” in *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1015–1020.

[20] B. Abdulhai, R. Pringle, and G. J. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[21] A. B. C. da Silva, D. de Oliveria, and E. Basso, “Adaptive traffic control with reinforcement learning,” in *Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2006, pp. 80–86.

[22] D. de Oliveira, A. L. Bazzan, B. C. da Silva, E. W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva, and L. Lamb, “Reinforcement learning based control of traffic lights in non-stationary environments: A case study in a microscopic simulator.” in *EUMAS*, 2006.

[23] M. Abdoos, N. Mozayani, and A. L. Bazzan, “Traffic light control in non-stationary environments based on multi agent q-learning,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1580–1585.

[24] J. C. Medina and R. F. Benekohal, “Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 596–601.

[25] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, “Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[26] M. A. Khamis and W. Gomaa, “Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework,” *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 134–151, 2014.

[27] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[28] E. van der Pol, F. A. Oliehoek, T. Bosse, and B. Bredeweg, “Video demo: Deep reinforcement learning for coordination in traffic light control,” in *BNAIC*, vol. 28. Vrije Universiteit, Department of Computer Sciences, 2016.

[29] “Intelligent traffic system cost,” http://www.itscosts.its.dot.gov/ITS/benecost.nsf/ID/C1A22DD1C3BA1ED285257CD60062C3BB?OpenDocument&Query=CApp/, 2016, online; accessed 23-November-2017.

[30] “Scats system cost,” https://www.itscosts.its.dot.gov/ITS/benecost.nsf/0/9E957998C8AB79A885257B1E0049CAFF?OpenDocument&Query=Home, 2016, online; accessed 13-May-2018.

[31] T. Neudecker, N. An, O. K. Tonguz, T. Gaugel, and J. Mittag, “Feasibility of virtual traffic lights in non-line-of-sight environments,” in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM, 2012, pp. 103–106.

[32] M. Ferreira and P. M. d'Orey, “On the impact of virtual traffic lights on carbon emissions mitigation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 284–295, 2012.

[33] M. Nakamurakare, W. Viriyasitavat, and O. K. Tonguz, “A prototype of virtual traffic lights on android-based smartphones,” in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*. IEEE, 2013, pp. 236–238.

[34] W. Viriyasitavat, J. M. Roldan, and O. K. Tonguz, “Accelerating the adoption of virtual traffic lights through policy decisions,” in *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*. IEEE, 2013, pp. 443–444.

[35] A. Bazzi, A. Zanella, B. M. Masini, and G. Pasolini, “A distributed algorithm for virtual traffic lights with ieee 802.11 p,” in *Networks and Communications (EuCNC), 2014 European Conference on*. IEEE, 2014, pp. 1–5.

[36] F. Hagenauer, P. Baldemaier, F. Dressler, and C. Sommer, “Advanced leader election for virtual traffic lights,” *ZTE Communications, Special Issue on VANET*, vol. 12, no. 1, pp. 11–16, 2014.

[37] O. K. Tonguz, W. Viriyasitavat, and J. M. Roldan, “Implementing virtual traffic lights with partial penetration: a game-theoretic approach,” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 173–182, 2014.

[38] J. Yapp and A. J. Kornecki, “Safety analysis of virtual traffic lights,” in *Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on*. IEEE, 2015, pp. 505–510.

[39] A. Bazzi, A. Zanella, and B. M. Masini, “A distributed virtual traffic light algorithm exploiting short range v2v communications,” *Ad Hoc Networks*, vol. 49, pp. 42–57, 2016.

[40] O. K. Tonguz and W. Viriyasitavat, “A self-organizing network approach to priority management at intersections,” *IEEE Communications Magazine*, vol. 54, no. 6, pp. 119–127, 2016.

[41] R. Zhang, F. Schmutz, K. Gerard, A. Pomini, L. Basseto, S. B. Hassen, A. Ishikawa, I. Ozgunes, and O. Tonguz, “Virtual traffic lights: System design and implementation,” *arXiv preprint arXiv:1807.01633*, 2018.

[42] J. Lu and L. Cao, “Congestion evaluation from traffic flow information based on fuzzy logic,” in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1. IEEE, 2003, pp. 50–53.

[43] B. Kerner, C. Demir, R. Herrtwich, S. Klenov, H. Rehborn, M. Aleksic, and A. Haug, “Traffic state detection with floating car data in road networks,” in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE, 2005, pp. 44–49.

[44] W. Pattara-Atikom, P. Pongpaibool, and S. Thajchayapong, “Estimating road traffic congestion using vehicle velocity,” in *ITS Telecommunications Proceedings, 2006 6th International Conference on*. IEEE, 2006, pp. 1001–1004.

[45] C. De Fabritiis, R. Ragona, and G. Valenti, “Traffic estimation and prediction based on real time floating car data,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 197–203.

[46] Y. Feng, J. Hourdos, and G. A. Davis, “Probe vehicle based real-time traffic monitoring on urban roadways,” *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 160–178, 2014.

[47] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, and B. Zhang, “Urban traffic congestion estimation and prediction based on floating car trajectory data,” *Future Generation Computer Systems*, vol. 61, pp. 97–107, 2016.

[48] P. Bellavista, F. Caselli, and L. Foschini, “Implementing and evaluating v2x protocols over itetris: traffic estimation in the colombo project,” in *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications*. ACM, 2014, pp. 25–32.

[49] D. Krajzewicz, M. Heinrich, M. Milano, P. Bellavista, T. Stützle, J. Härri, T. Spyropoulos, R. Blokpoel, S. Hausberger, and M. Fellendorf, “Colombo: investigating the potential of v2x for traffic management purposes assuming low penetration rates,” *ITS Europe*, 2013.

[50] P. Bellavista, L. Foschini, and E. Zamagni, “V2x protocols for low-penetration-rate and cooperative traffic estimations,” in *Vehicular technology conference (VTC Fall), 2014 IEEE 80th*. IEEE, 2014, pp. 1–6.

[51] R. Zhang, F. Schmutz, K. Gerard, A. Pomini, L. Basseto, B. H. Sami, A. Jaiprakash, I. Ozgunes, A. Alarifi, H. Aldossary *et al.*, “Increasing traffic flows with dsrc technology: Field trials and performance evaluation,” *arXiv preprint arXiv:1807.01388*, 2018.

[52] O. K. Tonguz, “Red light, green light — no light: Tomorrow's communicative cars could take turns at intersections,” *IEEE Spectrum Magazine*, vol. 55, no. 10, pp. 24–29, October 2018.

[53] A. Chattaraj, S. Bansal, and A. Chandra, “An intelligent traffic control system using rfid,” *IEEE potentials*, vol. 28, no. 3, 2009.

[54] M. R. Friesen and R. D. McLeod, “Bluetooth in intelligent transportation systems: a survey,” *International Journal of Intelligent Transportation Systems Research*, vol. 13, no. 3, pp. 143–153, 2015.

[55] F. Qu, F.-Y. Wang, and L. Yang, “Intelligent transportation spaces: vehicles, traffic, communications, and beyond,” *IEEE Communications Magazine*, vol. 48, no. 11, 2010.

[56] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[57] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[58] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, 2016, pp. 1928–1937.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[60] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning.” in *AAAI*, vol. 2. Phoenix, AZ, 2016, p. 5.

[61] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, “Reinforcement learning with deep energy-based policies,” *arXiv preprint arXiv:1702.08165*, 2017.

[62] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, “Expert level control of ramp metering based on multi-task deep reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1198–1207, 2018.

[63] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, “Flow: Architecture and benchmarking for reinforcement learning in traffic control,” *arXiv preprint arXiv:1710.05465*, 2017.

[64] L.-J. Lin, “Reinforcement learning for robots using neural networks,” Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1993.

[65] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[66] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.

[67] A. Y. Ng, D. Harada, and S. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *ICML*, vol. 99, 1999, pp. 278–287.

[68] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of sumo–simulation of urban mobility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.

[69] “Reinforcement Learning for Traffic Optimization,” https://youtu.be/HkXriL9SOW4, [Online; accessed 12-May-2018].

[70] “Traffic Monitoring Guide,” https://ops.fhwa.dot.gov/freewaymgmt/publications/frwy_mgmt_handbook/chapter1_01.htm, 2014, online; accessed 5-13-2018.

[71] L. Codecá, R. Frank, S. Faye, and T. Engel, “Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.

[72] A. de Palma and R. Lindsey, “Traffic congestion pricing methodologies and technologies,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1377–1399, 2011.

[73] B. Schaller, “New york citys congestion pricing experience and implications for road pricing acceptance in the united states,” *Transport Policy*, vol. 17, no. 4, pp. 266–273, 2010.

Rusheng Zhang was born in Chengdu, China in 1990. He received the B.E. degree in micro electrical mechanical system and second B.E. degree in Applied Mathematics from Tsinghua University, Beijing, in 2013, and the M.S. degree in electrical and computer engineering from Carnegie Mellon University, in 2015. He is a Ph.D. candidate at Carnegie Mellon University. His research areas include vehicular networks, intelligent transportation systems, wireless computer networks, artificial intelligence and intra vehicular sensor networks.

Akihiro Ishikawa was an MS student in the Electrical and Computer Engineering Department of Carnegie Mellon University until he received his MS degree in 2017. His research interests include vehicular networks, wireless networks, and artificial intelligence.

Wenli Wang has obtained an M.S. degree in the Electrical and Computer Engineering Department of Carnegie Mellon University in 2018. Prior to Carnegie Mellon University, she received B.S. in Statistics and B.A. in Fine Arts from University of California, Los Angeles in 2016. Her research interests include machine learning and it applications in wireless networks and computer vision.

Benjamin Striner is a master’s student in the Machine Learning Department at Carnegie Mellon University. Previously, he was a patent expert witness and engineer, especially in wireless communications. He received a B.A. in neuroscience and psychology from Oberlin College in 2005. Research interests include reinforcement learning, generative networks, and better understandability and explainability in machine learning.

Ozan Tonguz is a tenured full professor in the Electrical and Computer Engineering Department of Carnegie Mellon University (CMU). He currently leads substantial research efforts at CMU in the broad areas of telecommunications and networking. He has published about 300 research papers in IEEE journals and conference proceedings in the areas of wireless networking, optical communications, and computer networks. He is the author (with G. Ferrari) of the book *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective (Wiley, 2006)*. He is the inventor of 15 issued or pending patents (12 US patents and 3 international patents). In December 2010, he founded the CMU startup known as Virtual Traffic Lights, LLC, which specializes in providing solutions to acute transportation problems using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications paradigms. His current research interests include vehicular networks, wireless ad hoc networks, sensor networks, self-organizing networks, artificial intelligence (AI), statistical machine learning, smart grid, bioinformatics, and security. He currently serves or has served as a consultant or expert for several companies, major law firms, and government agencies in the United States, Europe, and Asia.