Transportation Research Interdisciplinary Perspectives 1 (2019) 100020

Contents lists available at ScienceDirect

Transportation Research Interdisciplinary Perspectives

journal homepage: https://www.journals.elsevier.com/transportation-research-interdisciplinary-perspectives

Decentralized network level adaptive signal control by multi-agent deep reinforcement learning

Yaobang Gong [*], Mohamed Abdel-Aty, Qing Cai, Md Sharikur Rahman

Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, FL 32816, USA

ARTICLE INFO

Article history:
Received 31 March 2019
Received in revised form 31 May 2019
Accepted 8 June 2019
Available online xxxx

Keywords:
Deep reinforcement learning
Multi-agent reinforcement learning
Adaptive signal control

ABSTRACT

Adaptive traffic signal control systems are deployed to accommodate real-time traffic conditions. Yet travel demand and behavior of the individual vehicles might be overseen by their model-based control algorithms and aggregated input data. Recent development of artificial intelligence, especially the success of deep learning, makes it possible to utilize information of individual vehicles to control the traffic signals. Several pioneering studies developed model-free control algorithms using deep reinforcement learning. However, those studies are limited to isolated intersections and their effectiveness was only evaluated in ideal simulated traffic conditions by hypothetical benchmarks. To fill the gap, this study proposes a network-level decentralized adaptive signal control algorithm using one of the famous deep reinforcement methods, double dueling deep Q network in the multi-agent reinforcement learning framework. The proposed algorithm was evaluated by the real-world coordinated actuated signals in a simulated suburban traffic corridor which emulates the real-field traffic condition. The evaluation results showed that the proposed deep-reinforcement-learning-based algorithm outperforms the benchmark. It is able to reduce 10.27% of the travel time and 46.46% of the total delay.

1. Introduction

Booming economy and growing population result in increasing travelling demand often beyond the capacity of the current traffic system, which leads to inevitable congestion. Rather than building more and more roadways, a cost-effective approach to address the issue is improving the efficiency of traffic management, such as the traffic signal control. Even frequently re-timed, the traditional pre-timed signal controllers, whose timing is determined by historical traffic information, is not necessarily suitable to the dynamic traffic demand. To overcome such limitations, many adaptive traffic signal control systems (ATCS) were developed, such as Split Cycle Offset Optimization Technique (SCOOT), Sydney Coordinated Adaptive Traffic System (SCATS) and Real-time Hierarchical Optimizing Distributed Effective System (RHODES). Those systems use loop or video detectors to extract current traffic flow information, then feed them into models to estimate future traffic flow profiles and finally adjust the timings according to the prediction. Those systems were seen successful in multiple deployments over the years.

Aforementioned ATCSs are fully based on "human-crafted" aggregated traffic features such as traffic flow, queue length, delay or travel time; "human-crafted" traffic flow models; and "human-crafted" signal control elements such as cycle length, offset, splits, etc. Admittedly, all those "human-crafted" features are valuable human knowledge. Yet the complex, discrete and heterogeneous travel demand and behaviour of the individual vehicles might be overseen by the model-based decision making and aggregated input traffic data. Especially with the development of connected vehicle (CV) technology, vehicles are able to provide their information to signal controllers via vehicle-to-infrastructure (V2I) communication. Therefore, the vehicle-based data will be largely available in the foreseen future. Hence, many researchers proposed ATCSs based on one of the artificial intelligence (AI) algorithms, reinforcement learning (RL), to let the "machines" learn how to control traffic signals (El-Tantawy et al., 2014). Although, the "human-crafted" traffic flow models and most of "human-crafted" signal control elements are no longer needed in such ATCSs, unfortunately, due to the limited computational power of conventional RL algorithms, most of studies are still using "human-crafted" aggregated traffic features as the input to the RL algorithms.

Recently, thanks to the rapid development of deep learning, the so-called deep reinforcement learning (DRL) algorithms incorporated with deep neural networks (DNNs) shows its ability to handle high-dimensional disaggregated input data. Hence, ATCSs based on DRL could get rid of "human-crafted" traffic information and high-resolution traffic data such as the position, speed or even the origin and destination of individual vehicles could directly be used. Li et al. (2016) proposed an ATCS algorithm of an isolated intersection using Deep Q Network (DQN, which will be illustrated later). Compared with a conventional RL algorithm

[*] Corresponding author.
E-mail addresses: gongyaobang@knights.ucf.edu, (Y. Gong), M.Aty@ucf.edu, (M. Abdel-Aty), qingcai@knights.ucf.edu, (Q. Cai), sharikur@knights.ucf.edu. (M.S. Rahman).

http://dx.doi.org/10.1016/j.trip.2019.100020

incorporated with a shallow neural network (will be illustrated later), the algorithm could reduce the average delay of a simulated intersection by 14%. However, the study oversimplified the signal control problem. Their algorithm was trained and evaluated by an intersection which forbids turning movements, and the signal does not configure the yellow and all-red clearance. The algorithms proposed by several other studies (Mousavi et al., 2017; Muresan et al., 2018; Van Der Pol and Oliehoek, 2016) were also evaluated by the similar simplified isolated intersection (no turning movement and no clearance time). Gao et al. (2017) proposed an algorithm considering the clearance time and turning movements. However, in their algorithm, the left-turning signal was always activated after the through signal and its length was fixed at 10 s. Therefore, even though their algorithms were proven to outperform the fixed-timing or actuated signals in the traffic simulation, whether their algorithm works for realistic intersections remains unknown. Aiming at realistic intersection representation, Gendes and Razavi's algorithm (Genders and Razavi, 2016) allows protected left-turning phases and dedicated left-turning lanes. The algorithm takes the position and speed of individual vehicles as well as the latest signal state as the input and the appropriate signal is determined every 2 s. Through simulation, the algorithm outperforms an algorithm with a shallow neural network in terms of average cumulative delay, average queue length and average travel time. However, it was not evaluated by comparison to traditional fixed-timing or actuated signals.

While those studies have shown a good potential, there are two major limitations:

(1) ATCSs based on DRL are shown to be effective for isolated intersections, its effectiveness of network-wide signal control, especially for coordinated signals, is not proven. Although Casas (2017) tests a DRL algorithm for a network-level signal control, the algorithm does not converge.

(2) To the best of the authors' knowledge, all the studies about ATCSs based on DRL are trained and evaluated by ideal simulated traffic scenarios, which are not related to any real-world application. The intersection geometric design such as length of the turning storage bays is not considered. The traffic demand is hypothetical and the route choice behavior is simplified. The benchmark signal control system is naïve fixed-timing signal, signal controlled by conventional RL algorithm incorporated with a shallow neural network which is never applied in the field, or even random signals (the only exception is the study conducted by Muresan et al. (2018) which used a calibrated timing by Synchro based on hypothetical traffic demand). As a consequence, the effectiveness of those algorithms on real-world application is hard to be proven.

To fill the gap of the early research, this study proposes a network-level decentralized adaptive signal control algorithm using one of the famous DRL methods, double dueling deep Q network in the Multi-Agent Reinforcement Learning (MARL) framework. It allows the coordination among nearby intersections by information sharing. The proposed algorithm was trained by a simulated AM peak scenario of a suburban traffic corridor calibrated by real-world data. Finally, it was evaluated by real-world coordinated actuated signal system whose configuration is provided by the local jurisdiction.

2. Background

2.1. Reinforcement learning

Deep Reinforcement Learning is a family of Reinforcement learning (RL) algorithms incorporated with Deep Neural Networks (DNNs). RL (Sutton and Barto, 2018) is a goal-oriented machine learning algorithm. It learns to achieve a complex *goal* over many discrete steps by interacting with the environment. For a control problem, in every discrete control step, an RL control *agent* (e.g. signal controller) iteratively observes the *state s* of the environment (e.g. roadway network), takes an *action a* (e.g. directly change the signal phase or change the duration of the signal phase) accordingly based on its underlying behavior *policy* π, receives a feedback reinforce *reward r* (e.g. waiting time, delay or travel time) for the action taken, which will be accumulated to its long-run *goal* (minimizing delay, decreasing travel time or minimizing stops), from the environment, and transits to the *next state s′* according to the environment dynamics and state *transition probability P*. The RL agent optimizes the *policy*, which is the mapping from the set of the all possible states *S* to the set of all possible actions *A*, by learning from the accumulated discounted long term *reward*, with a *discount factor* γ, of applying different action sequences. During the learning process, it keeps adjusting its *policy* by maximizing the expectation of the long term *reward* until it converges to the *optimal policy* π□. Fig. 1 gives an illustration of the setting of the reinforcement learning problem.

The *value function* of the RL problem is the estimation of the long term reward of each state or state-action pair. The state value is the expected long term discounted reward for following *policy* π from *state s*, which is defined as:

$$V_\pi(s) = E[R_t | s_t = s] \tag{1}$$

and it decomposes into the Bellman equation:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} P(s',r|s,a)[r + \gamma V_\pi(s')] \tag{2}$$

The Q value, or action value, refers to the expected long term discounted reward for selecting *action a* and in *state s* and then following the *policy* π, which is defined as:

$$Q_\pi(s,a) = E[R_t | s_t = s, a_t = a] \tag{3}$$

and it also decomposes into the Bellman equation:

$$Q_\pi(s,a) = \sum_{s',r} P(s',r|s,a)\left[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right] \tag{4}$$

One of the famous RL algorithms is Q-learning (Watkins and Dayan, 1992). Q-learning is an off-policy algorithm, which means its policy being followed (the actioned chosen) is independent with its learning process. In Q-learning, the agent chooses the action *a* □ *A* with the highest Q-value (greedy action) based on a matrix called Q-table. The Q-table is a mapping table of all discrete state value *s* □ *S* to all discrete action value *a* □ *A*. At every discrete step, Q-learning improve its policy greedily. The adjusted Q-value is learned by

$$Q_k(s,a) = (1-\alpha)Q_{k-1}(s,a) + \alpha\left(r + \gamma \max_{a' \in A} Q_{k-1}(s',a')\right) \tag{5}$$

where Q_k is the new Q-value after the adjustment at learning step *k*; Q_{k-1} is the current Q-value stored in the Q-table; *s*, *a*, *r* are current state, action and reward at step k; *s*', *a*' are the next state and action; α is the learning rate controlling the adjusting size.

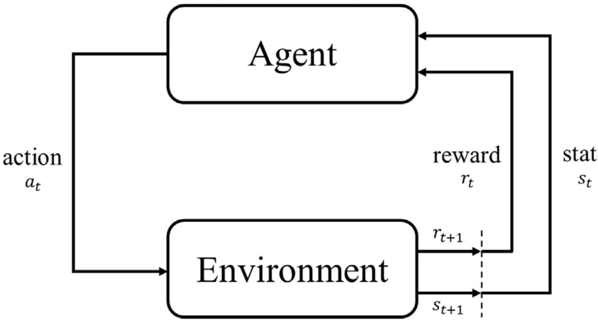
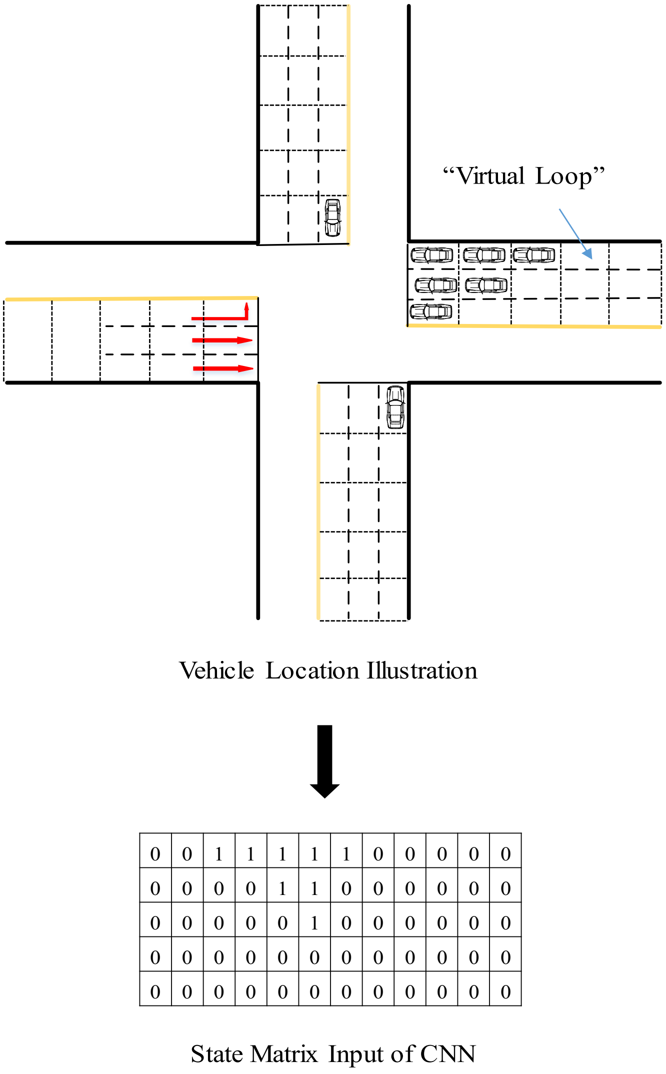


Fig. 1. Illustration of the reinforcement learning problem setting (Sutton and Barto, 2018).



0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. Traffic state representation.

2.2. Double dueling deep Q network (3DQN)

2.2.1. Deep Q network

Conventional tabular form Q-learning requires a Q-table to store Q-values for all (s, a) pairs. However, when state set S (e.g. detailed representation of traffic states) are growing, the Q-table becomes extremely large, which makes the learning process intractable (curse of dimensionality). As a consequence, function approximation $Q(s,a;\theta)$ (where θ is the hyper-parameter of the approximator) of the Q-table was introduced. Function approximation aims to generalize a "function" from example to estimate the mapping. It is usually a concept of supervised learning. Linear function approximation is a popular choice before the age of deep learning. However, the neural network was utilized dated back a long time ago (Bertsekas and Tsitsiklis, 1996).

Since the Q-values are estimated by a function, the learning process is no longer directly updating Q-value, but updating the hype-parameters θ. However, diverge and instability might occur when the neural network approximators are applied, especially for high-dimensional continuous state-action spaces (Tsitsiklis and Van Roy, 1997). It is because of the highly correlated consecutive states inputs and frequently changing policy due to slight changes in Q values.

Recently, Deep Q Network (DQN), one of the earliest DRL algorithm using deep neural networks (DNNs) as the functional approximator, shows its capability to deal with the large state-action space in RL problems (Mnih et al., 2015; Mousavi et al., 2018). Besides, it tackles the aforementioned instability and diverge issue by experience replay (LIN, 1993) and target network. Experience replay breaks the temporal correlation of states in consecutive learning process. In every training step, the agent stores its 'experience' (s,a,r,s') into the experience replay memory θ and then randomly samples a minibatch from it to updated the θ. In this way, the strong correlations between consecutive states are eliminated.

Target network also helps to mitigate the correlation issue. A target network is a DNN which has the same structure with primary network (or value network) yet updates less frequently. It is used to generate target-Q-value to update the hype-parameters θ of the primary network:

$$J = \sum_{s} P(s)\left(Q_{target}(s,a;\theta^{-}) - Q(s,a;\theta)\right) \tag{6}$$

where $P(s)$ denotes the probability of the state s in the minibatch; Q and Q_{target} are Q-values estimated by primary network and target network respectively. J is used as a loss function to update θ in backpropagation optimization. Every certain steps, the hyper-parameters of target network θ^{-} is updated by the hyper-parameters of primary network θ. As the result, the loss function of the current training step is evaluated by an earlier snapshot, decorrelate the target and primary Q-values/state and increase stability of learning algorithm.

2.2.2. Deep Q network

Double DQN (van Hasselt et al., 2015) is an improved version of DQN. In standard DQN, the selection of the greedy action and the evaluation of the Q-value are using the same action value (Q-value), which might lead to overoptimistic value estimates. van Hasselt et al. (2015) proposed to evaluate the greedy action by the online network but to estimate its value by the target network. This is achieved by modifying the calculation of target Q value:

$$Q_{target}(s,a) = r + \gamma Q\left(s', \ \arg\ \max_{a'}(Q(s',a';\theta)); \theta^{-}\right) \tag{7}$$

Double DQN is proved to be able to found better policies than standard DQN.

2.2.3. Dueling DQN

To get a faster converge, Wang et al. (2015) proposed the dueling DQN. The dueling DQN estimate the state value function and associated advantage function and then confine them to get the Q-value:

$$Q(s,a) = V(s;\alpha) + \left[A(s,a;\beta) - \frac{1}{|A|}\sum_{a'} A(s,a';\beta)\right] \tag{8}$$

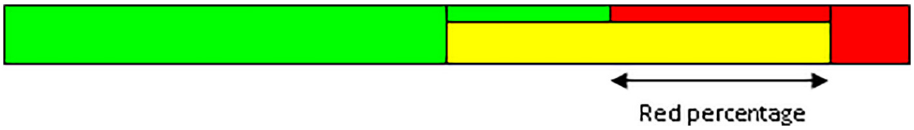


Fig. 3. "Red percentage" of the yellow time (Aimsun, 2018).

where $V(s;\alpha)$ is value function which indicates the expected overall rewards of a specific state s; $A(s,a;\beta)$is state-dependent advantage of a specific action a over other actions and it is then normalized by the average value of all actions $a' \in A$. The performance of dueling DQN is shown better than the standard DQN especially when there exist several similar-valued actions.

3. Decentralized adaptive signal control algorithm based on 3DQN

In this study, a decentralized signal control problem is formulated into the standard MARL setting: each individual signal controller acts as a RL agent; the agent observes the condition of the intersections as the state; the agent directly selects the appropriate phase every step as its action and the discrete signal performance metrics act as the reward (actually is the penalty) of the state-action pair; the long run goal of the system is reducing the cumulative delay. The signal control agent coordinate with agents controlling its upstream and downstream intersections by sharing their state vectors with each other. Convolutional neural network (CNN) was used to build the 3DQN of the algorithm. The details of the algorithm are elaborated in this section.

3.1. State representation

The state matrices are collected every control step. In order for the controller to coordinate with other controllers, not only the condition of the intersection it controlled but also those of the upstream and downstream intersections are considered (3 intersections in total). Two aspects of the intersection condition are collected as the state: the current traffic state which the controller should "adaptive to" and the current signal phase.

As mentioned in the previous sections, the detailed data of individual vehicle could provide additional heterogeneous travel demand and behavior information than aggregated traffic parameters. For example, the blockage of left-turning-storage bay could be captured using detailed location of individual vehicles. The proposed algorithm uses the detailed location of every vehicle occupying the roadway within certain distance from the stop line to represent the traffic state. It is important not to use the locations of ALL vehicles within the network. Since in the field, traffic cameras used in many ATCSs are only able to capture vehicles close to the intersection. The widely used "virtual loop" (see Fig. 2) concept in video detection is applied as well. The length of the virtual loop detectors in this study is 15 ft and the maximum number of loop detectors for each lane is 20 (due to the length of turning storage bays, the number of "virtual loop" could be less than 20). Then the algorithm converts the "virtual loop" actuations to a traffic state matrix. (Fig. 2) The traffic state matrices of all three intersections are stacked into one big matrix as an input of 3DQN.

The phase of the signal controller is defined as the legal combination of two or more non-conflicting vehicular movements, such as southbound through and southbound left-turning or southbound through and northbound through. Interphases are added between two phases as the clearance time including yellow time and all-red clearance. Note that the behavior of vehicles under yellow time has two phases depending on the "red percentage" defined. It indicates the percentage of yellow time the vehicles will consider as red light (Fig. 3). Therefore, the representation of the current signal phase is coded as a vector with length of n + 1, where n is the number of phases of the signal. For example, when phase 1 is activated as green or the signal is not at the "red percentage" of the yellow time behind phase 1, the first element of the vector is set to 1 and all others are set to 0; when all-red phase was activated or the signal is at the "red percentage" of the yellow time, the last element is set to 1 and all others are set to 0. The traffic state matrices of all three intersections are also stacked into one big vector.

Not that this particular algorithm did not modal the pedestrian phase explicitly, however pedestrian movements could be considered implicitly by changing the action set. The details is depicted in the next section.

3.2. Action definition

The action set of the algorithm is all legal phases of the particular signal. When the signal is not in the "interphase", the algorithm selects an appropriate phase and sends the phase to the controller. If a phase changing occurs, the controller will activate the "interphase" to clear the intersection.

In addition, the action set of is subject to be partially or fully "frozen" in order to maintain the safety and/or to serve the special needs of signal system operators. If the action set is "fully frozen", the agent could only take the immediate last action until the action set is "defreezed", which means the phase of the signal remains unchanged during the "fully frozen" period. The "fully frozen" mode is used to enforce a minimum duration of each phase in consideration of the reaction time, to call a fixed-length pedestrian phase when there is a pedestrian call and to serve a preemption. "Partially frozen" means some actions within the action set are disabled during the "frozen" period. The operator could utilize this mode to ensure the cyclic traffic signal operation while the proposed algorithm is designed to be acyclic.

3.3. Reward definition

For a RL problem, the reward should accumulate to the ultimate goal of the agent with temporal discount. As for the adaptive signal controller, the

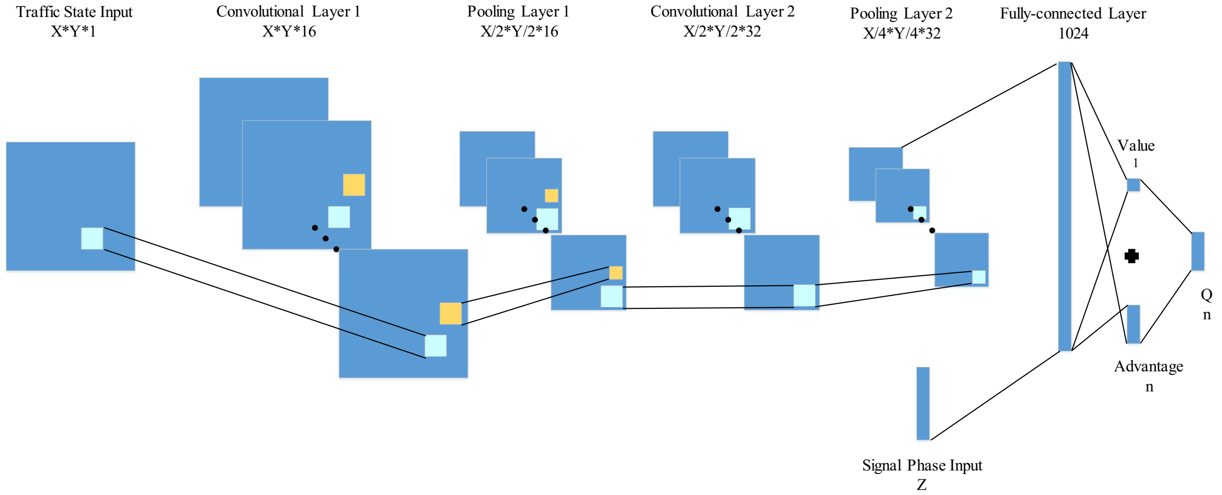


Fig. 4. Structure of the CNN used in the algorithm.

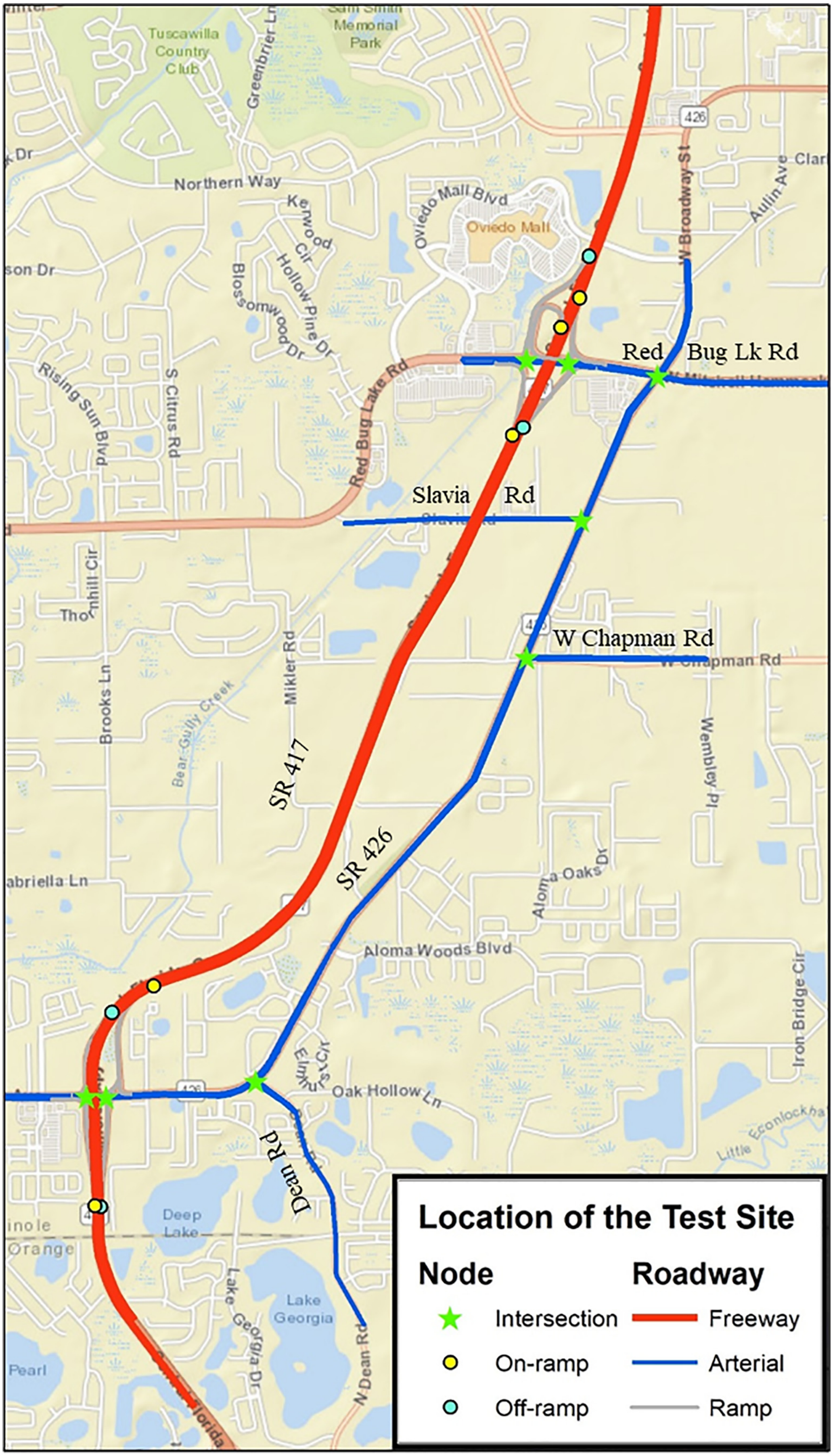


Fig. 5. Location of the simulation test site.

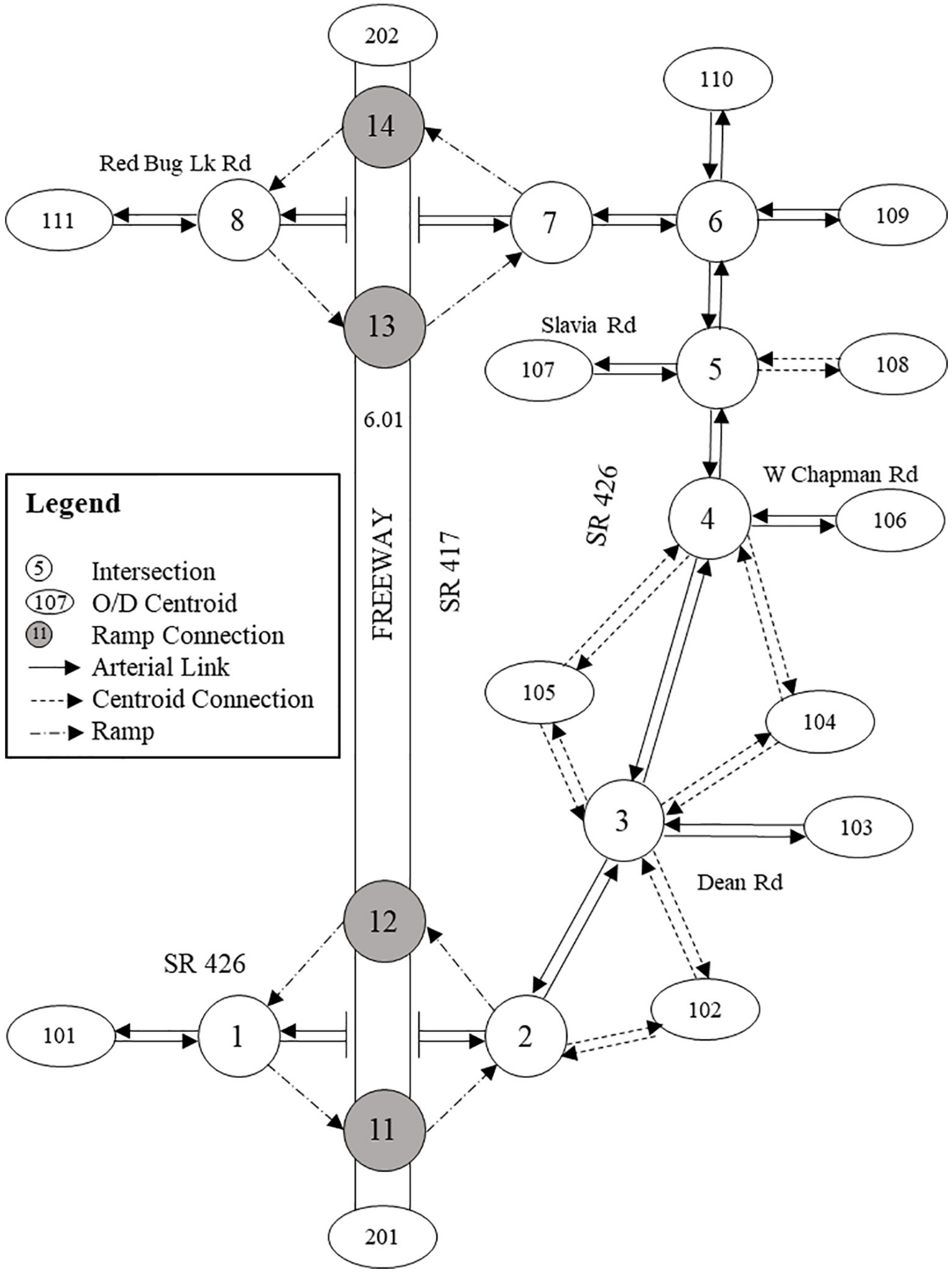


Fig. 6. High-level abstract of the test corridor.

goals could be minimizing the travel time of a vehicle, or the total delay, which theoretically is defined as the difference between the actual and expected travel times. However, it is unfeasible to get the travel time or total delay every discrete control step (e.g. every one second) since those measures are only available when the vehicle reaches its destination (e.g. every several minutes). Therefore, the cumulative waiting time of the vehicles in the queue (the dominant part of total delay), which could be monitored every control step, rather than the total delay was utilized as the goal indicator. And the reward is defined as the difference between the current and previous waiting times of all vehicles:

$$r_t = -(W_{t+1} - W_t) \tag{9}$$

where W_{t+1}, W_t are the waiting time of step $t+1$ and t. When the vehicle is queued, the agent will be penalized; when the vehicle waiting in the queue is discharged, the agent will be rewarded.

However, getting the cumulative waiting time of every vehicle within a relatively large road network is computationally expensive. Hence, the reward is estimated for each step directly:

$$r_t = wn_t^d - n_t^q t_s \tag{10}$$

where n_t^q is the number of the queued vehicles at step t; t_s is the step length; n_t^d is the number of the vehicles discharged during step t; w is an average waiting time of discharged vehicles.

3.4. Deep Q network structure

As mentioned in the last section, DQN uses DNNs as the functional approximator. The input of the DNN is the state and the output of it is the action. To extract valuable information from the big traffic state matrix,

Table 1
Information regarding signals under control of the algorithm.

Intersection ID	Connected with ramp	Lanes of approaches	Number of phases	Legal movement of phases	*Aberrations*
1	Yes	SB: 1*LR + 1*R	3	ST (SL + SR)	*Approaches*:
		WB: 1*L + 2*T		WL + WT	*NB*: *Northbound*
		EB: 3*T		ET + WT	*SB*: *Southbound*
2	Yes	NB: 2*L + 2*R	3	NT (NL + NR)	*WB*: *Westbound*
		WB: 3*T		ET + WT	*EB*: *Eastbound*
		EB: 1*L + 2*T		EL + ET	*Movement of a lane*:
3	No	NB: 2*L + 1*R	3	NT (NL + NR)	*T*: *Through*
		WB: 2*L + 2*T		WL + WT	*L*: *Left turning*
		EB: 2*T + 1*R		ET + WT	*R*: *Right turning*
4	No	NB: 2*T + 1*R	3	NT + ST	*LR*: *Shared left-right turning*
		SB: 2*L + 2*T		SL + ST	*Legal Movement of a Phase*:
		WB: 2*L + 1*R		WT(WL + WR)	*First digit*: *Approach*
5	No	NB: 1*L + 2*T	3	NL + NT	*Second digit*: *Turning Movement*
		SB: 2*T		ST + SL	
		EB: 1*L + 1*R		ET(EL + ER)	
6	No	NB: 2*L + 2*T + 1*R	4	NL + NT	
		SB: 2*L + 2*T + 1*R		SL + ST	
		WB: 2*L + 2*T + 1*R		WL + WT	
		EB: 2*L + 2*T + 1*R		EL + ET	
7	Yes	NB: 2*L + 1*R	3	NT (NL + NR)	
		WB: 3*T		WT + WL	
		EB: 1*L + 3*T		WT + ET	
8	Yes	SB: 2*L + 1*R	2	ST (SL + SR)	
		WB: 3*T		WT + ET	
		EB: 3*T			

convolutional neural network (CNN), which is widely used in many pattern recognition problems such as image-processing, is used in this algorithm.

CNN is a deep neural network composed by a sequence of three kinds of layers: convolutional layer, pooling layer and fully-connected layer. Convolutional layer acts a "window" "scanning" across the big input matrix and extracts information of a local region; pooling layer performs a down sampling operation and fully-connected layer is a regular neural network layer which is typically one of the last few layers. Compared to the regular neural network, the CNN takes advantage of local spatial coherence in the input which allows it to have much fewer parameters, and therefore overcomes the overfitting issue and saves computational resources.

The structure of CNN used in the proposed algorithm is shown in Fig. 4. The traffic state matrix is filtered into a vector by two convolutional layers and two pooling layers. Then it is combined with the signal phase state vector as the input of the fully-connected layer. The output vector of the fully connected layer is used to estimate state value and the advantage of all actions. Finally, they are added up to get the Q-value. The Leaky Rectifier Nonlinearity Units (Leaky ReLU) (Maas et al., 2013) was applied as the activation function. Noticed that the size of layers varies for different intersections due to the different input sizes.

3.5. Supervised pre-training

In order to speed up the training process, the parameters of DNN θ are initialized by a supervised learning process instead of random initialization and exploration. During the pre-training, the algorithm is forced to record the policy of a reasonable fixed-timing traffic signal, then updates θ based on learned policy. The main difference between the pre-training and training are: (1) the agent observes the action taken by the fixed-timing traffic signal rather than execute its own; (2) the reward is set to a constant value in order to update the θ.

3.6. Overall algorithm

The pseudocode of the proposed algorithm is summarized in Algorithm 1. Note that during training, the number of steps to update the target network (frozen period) is increasing by episodes to ensure the convergence. The algorithm is coded by Python programming language using deep learning modules Tensorflow (Abadi et al., 2016).

Algorithm 1. 3DQN for decentralized adaptive signal control.

Table 2
Two-hour aggregated O/D matrix.

O\D	101	102	103	104	105	106	107	108	109	110	111	201	202	Total
101	0.00	0.00	0.00	149.98	157.33	226.87	0.00	84.43	90.20	45.75	23.93	839.48	297.32	1915.28
102	0.00	0.00	4.18	2.19	4.25	15.01	0.00	22.63	1.87	1.49	1.32	0.00	24.27	77.21
103	0.00	1.94	0.00	1.05	2.11	0.00	0.00	11.75	0.00	0.00	0.00	0.00	0.00	16.84
104	121.92	1.01	9.91	0.00	0.72	5.81	16.21	6.26	12.12	24.46	0.00	122.68	11.10	332.21
105	131.80	2.07	2.19	0.76	0.00	0.00	101.12	14.41	0.00	23.25	0.00	239.74	92.57	607.91
106	241.04	20.30	10.48	7.42	0.00	0.00	219.88	0.00	0.00	8.75	0.00	0.00	129.16	637.03
107	0.00	0.00	0.00	16.07	64.06	127.57	0.00	133.32	0.00	0.00	0.00	0.00	0.00	341.01
108	139.11	3.09	4.33	1.79	3.93	0.00	53.76	0.00	0.00	6.11	0.00	324.96	39.59	576.67
109	184.33	82.21	58.60	24.22	0.00	0.00	0.00	0.00	0.00	20.96	658.66	468.06	444.42	1941.45
110	38.31	37.93	43.73	34.32	30.96	7.77	0.00	20.77	47.63	0.00	97.76	104.18	81.88	545.26
111	15.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	193.43	68.33	0.00	71.30	238.01	586.83
201	516.35	42.32	0.00	37.07	53.08	0.00	0.00	141.52	146.68	112.16	101.13	0.00	1782.95	2933.27
202	291.61	0.00	0.00	29.38	71.51	91.16	0.00	115.26	188.36	68.13	274.68	1534.16	0.00	2664.24
Total	1680.23	190.87	133.43	304.25	387.95	474.20	390.97	550.34	680.27	379.39	1157.47	3704.56	3141.26	13,175.20

Table 3
Parameters used in algorithm.

Parameter	Initial value	Fine-tuned value
Discount factor γ	0.99	0.999
Ending greedy □$_e$	0.1	0.01
Greedy decrement □' (if applicable)	0.001	0.0001
Replay memory size M	2000	20,000
Minibatch size B	64	64
Number of episodes N	20	20
Number of the steps in an episode T	7200	7200
Starting number of steps to update target network N_{us}	200	200
Target network updating increment N_{ui}	0	200
Learning rate α	0.001	0.0001
Leaky ReLU β	0.01	0.01

4. Experiment and evaluation

The proposed algorithm was implemented in a simulated traffic network using a commercial traffic simulator Aimsun Next 8.2.3. The simulator provides Python application programming interface (API) to get access to the coded ATSC algorithm. The algorithm gets the information from the simulated traffic and implements its control command to the simulated signal controllers. The simulated RL signal control agents were trained for certain episodes. After the training, its performance is evaluated by the real-world signal timings.

4.1. Simulation set up

To better examine the performance of the algorithm, the simulation scenario was built based on a real-world suburban traffic corridor in Seminole County, Florida. There is one limited-access freeway (SR 417) and several parallel or connected arterials (SR 426, Red Bug Lake Road, Slavia Road, West Chapman Road and Dean Road) within the corridor. Fig. 5 shows the location of the corridor and Fig. 6 illustrates the high-level abstract of the corridor including nodes IDs (signalized intersections, Origin/Destination centroids and ramp connections), connectivity between nodes and the length in miles of the links. The authors did their best effort to match the geometric design of simulation corridor with that in the real-world, although some inevitable minor modifications were done to fit in the available demand and traffic data.

Along with the corridor, there are eight signals in total under control of the proposed algorithm. The detailed information about number of lanes of each approach, whether there are dedicated left-turning or right-turning lanes and the phase configuration is provided in Table 1. Note that there are four signals among them which are connected with the on/off ramp of the freeway, and therefore interact with the freeway traffic flow. In this particular case study, the “fully frozen” model of the action set is utilized only to accommodate the minimum length of the phases. The pedestrian phase is not used due to the extreme low pedestrian activities within the test site and the “partially frozen” mode is not activated. All eight signals are using coordinated actuated signal controller and their timings are provided by Seminole County. These actuated signal controllers are used as the benchmark to evaluate the performance of the proposed algorithm. In addition, the length of the “interphases” used in the algorithm for each signal were the same as the summation of yellow and red clearance time of the real-world signal timing.

In this study, the demand data of the AM peak hours (7:00–9:00) were used as the input to simulate the recurrent congestion situation. The daily Origin-Destination (O/D) matrices were extracted from Orlando Urban Area Transportation Study (OUATS) with base year 2009, which is the most recent regional planning model available when the authors conducted the study. The 15-minute aggregated real traffic count data of January 18th, 2018 were utilized to estimate peak-hour O/D matrix from the raw daily O/D matrices. The real dataset was extracted from Microwave Vehicle Detection System where (MVDS) and Automated Traffic Signal Performance Measures (ATSPM) for the freeway and arterials, respectively. The two-hour aggregated matrix is provided in Table 2 for the readers' reference. Static origin-destination and departure adjustment were applied to convert the two-hour O/D matrix to eight 15-minute time-dependent matrices based on the aforementioned real dataset. The simulation scenario is further calibrated by the traffic counts from the real dataset. One of the most important calibration metrics, the root mean square error between simulated counts and the real counts, is 4.7 vehicles per hour, which means the scenario is well calibrated.

4.2. Training

The algorithm is trained in episodes. One episode is one simulation replication with the aforementioned two-hour AM congested traffic. The goal of the algorithm is to minimize the total delay of the episode. The length of training and control step is one second. Therefore, there are 7200 training steps in one episode. The simulation replications were warmed up by 15-min real-world demand (6:45–7:00). To reduce the overfitting issue, even though the traffic demand of every episode is the same, its random state was set differently.

The signal timing used in the pre-training was generated by the maximum green time of the real-world actuated signal timing. The initial and fined-tuned input hyper-parameters of the algorithm are shown in Table 3.

Fig. 7 visualizes the average waiting time per vehicle per mile for each training episodes. The training episode 0 indicates the value of the pre-training. The curve shows that the average waiting time dropped dramatically after the first training episode and it was continuously dropping until the 9th episode. During the last 10 episodes, the average waiting time fluctuate in a very small range which indicates the optimal policy was learnt.

4.3. Training evaluation results and discussion

To evaluate the performance of the proposed ATSC algorithm, both the well-trained algorithm and the real-world benchmark signal control were implemented in a simulation replication with the same random state. Three kinds of performance measures were observed in 5-min aggregation intervals: average travel time per vehicle of all the vehicles travelling in the road network; average total delay (the difference between the actual and expected travel times) per vehicle per mile of the whole network. Fig. 8 shows the performance measures by simulation time.

During the whole simulation episode, the average travel time and average delay of the simulated network controlled by the proposed algorithm are less than those of the network controlled by the benchmark signal. On average, the proposed algorithm reduced 10.27% (4.26 min versus 4.75 min) of travel time and 46.46% (11.27 s/mile versus 21.06 s/mile) of the total delay. According to Fig. 7, at the beginning of the simulation, there is no significant difference of performance between the two signal controls. However, as the time goes by, on the one hand, the performance of the benchmark is getting worse than that of the proposed algorithm. On the other hand, the performance of the benchmark fluctuates a lot while that of the proposed algorithm remains stable. One possible reason is that at the beginning of the AM peak period the traffic volume remains at a low level and then it fluctuates a lot (see flow rate in Figs. 3–7). This confirms that the algorithm well adapts to the changing traffic demand.

However, given that the number of the stops remains at a low level, the proposed ATSC algorithm tends to increase the number of stops by 11.29% on average (0.31 versus 0.28). Interestingly, compared with the benchmark, the number of the stops remains at a relatively stable level when the network is controlled by the proposed algorithm. In previous studies, only Li et al. (2016) evaluated the specific performance metrics but it was compared with the signal control based on conventional RL algorithm incorporating with a shallow neural network. Thus, it is not appropriate to use the result as the guidance of the field application. According to the careful examination of the algorithm during the simulation process, the length of the phases allowing through movement of major approaches are

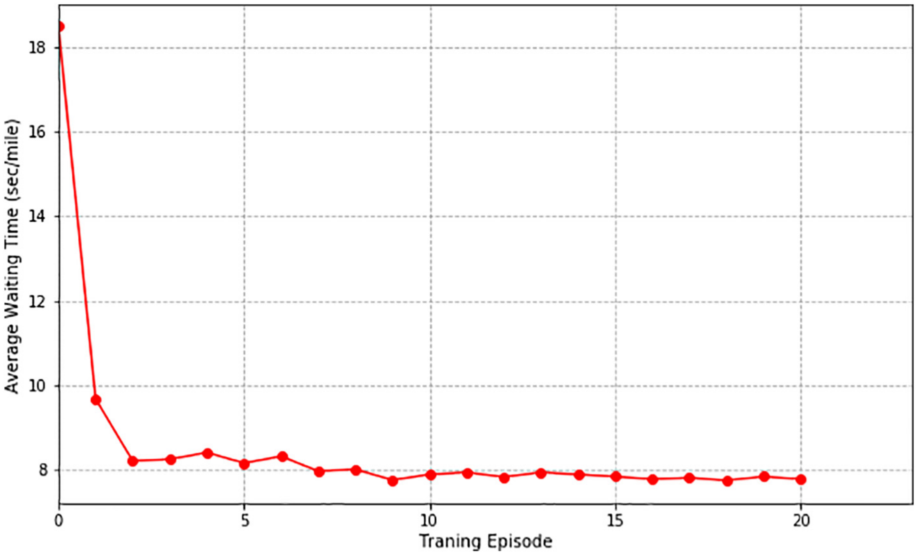


Fig. 7. Location of the simulation test site.

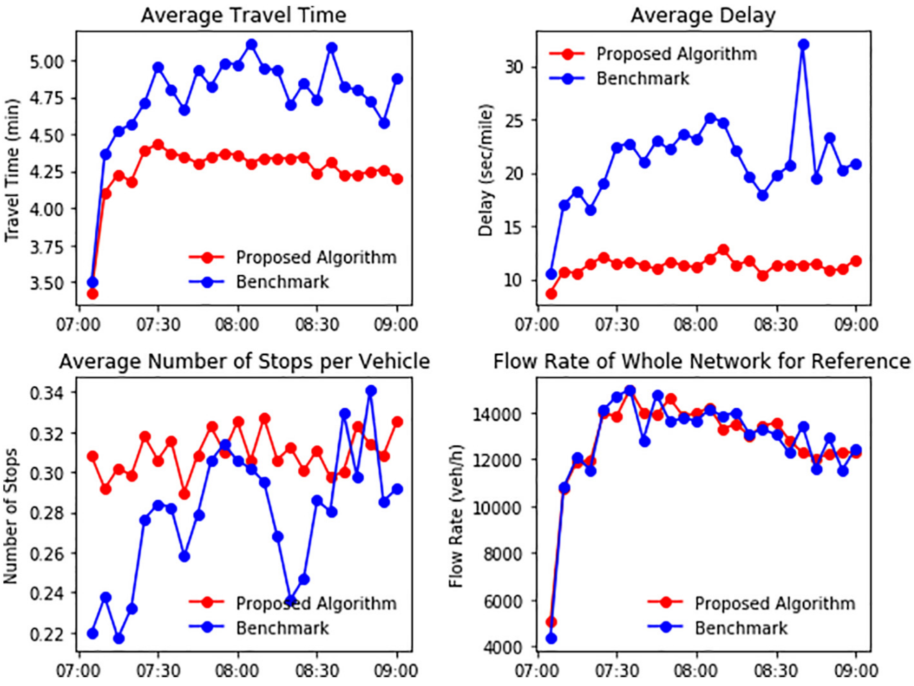


Fig. 8. Performance of proposed ATSC algorithm and benchmark.

typically shorter than the benchmark. This ensures the fair travelling rights between major and minor approaches as well as through and turning movements by reducing the waiting time of vehicles queueing the minor approaches or left-turning bay. However, the side effect is that the vehicles travelling through the major approaches are forced to stop to yield the right of way. Although the result is always reasonable since reducing the number of stop is never the goal the algorithm. This trade-off should be taken into account by practitioners.

5. Conclusion

This study proposes a decentralized adaptive signal control algorithm, in the context of network level control based on multi-agent framework, using one of the famous deep reinforcement learning methods, double dueling deep Q network. In the algorithm, for every individual intersection, a reinforcement learning agent controls the signals based on high-resolution real-time traffic data. It captures detailed locations of vehicles as the input, extracts relevant information by convolutional neural networks, and selects appropriate signal phases every second to reduce vehicles' waiting time. To achieve better network performance, control agents are coordinated with each other by sharing the information regarding the traffic and signal state.

The proposed algorithm was trained and evaluated in a simulated real-world suburban traffic corridor with eight signalized intersections in Seminole County, Florida. Travel demand of AM peak period was used in the simulation for the algorithm to reduce the recurrent congestion. The performance of well-trained algorithm was compared with the real-world coordinated actuated signals of the corridor provided by local jurisdiction. The evaluation results showed that the algorithm adapts well to the changing traffic demand. And it reduces 10.27% of network travel time and 46.46% of network delay compared with the real-world benchmark. Meanwhile, it ensures the fair travelling right for all movements.

The flexibility of the proposed ATCS based on shed light upon further extensions of the work. Firstly, although the algorithm is able to ensure the fair travelling right, in some situations, giving priority for some specific movement is important. Therefore, the design, goal and reward could be altered to accommodate the priority requirement. Secondly, the flexibility of the objective definition in DRL algorithm also enlightens the possibility to develop a multi-objective ATCSs, for example, both of the number of stops and the delay could be considered as the objectives concurrently and even safety-related measures (Yuan et al., 2018, 2019; Yuan and Abdel-aty, 2018) could also be integrated. In addition, although the proposed algorithm shows promising performance for the test roadway network, additional work might be needed when it is applied to a large network due to the constraint of computational resources. Furthermore, with the development of connected and automated vehicles (CAV) technology, more information about individual vehicles such as their travel origin-destination, speed and acceleration could also be collected through the vehicle to infrastructure (V2I) communication to enrich the state representation. As the control agent takes more information as the input, its performance is expected to be improved.

Acknowledgment

The authors are grateful to the Florida Department of Transportation (FDOT) for supporting this study. All opinions are those of the authors and not necessarily reflect FDOT's position. The authors also appreciate the data provided by FDOT and Seminole County.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I.J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D.G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P.A., Vanhoucke, V., Vasudevan, V., Viégas, F.B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems. CoRR abs/1603.0.

Aimsun, 2018. Aimsun Next 8.2 User's Manual.

Bertsekas, D., Tsitsiklis, J., 1996. Neuro-dynamic Programming, Third World Planning Review - THIRD WORLD PLAN REV. https://doi.org/10.1007/978-0-387-74759-0_440.

Casas, N., 2017. Deep Deterministic Policy Gradient for Urban Traffic Light Control.

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.

Gao, J., Shen, Y., Liu, J., Ito, M., Shiratori, N., 2017. Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm With Experience Replay and Target Network. arXiv.

Genders, W., Razavi, S., 2016. Using a deep reinforcement learning agent for traffic signal control. arXiv Prepr. arXiv1611.01142.

Li, L., Lv, Y., Wang, F.-Y., 2016. Traffic signal timing via deep reinforcement learning. IEEE/CAA J. Autom. Sin. https://doi.org/10.1109/JAS.2016.7508798.

LIN, L.-J., 1993. In: Carnegie Mellon Univ (Ed.), Reinforcement Learning for Robots Using Neural Networks Ph. D. thesis.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. Proc. Icml, p. 3.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529.

Mousavi, S.S., Schukat, M., Howley, E., 2017. Traffic light control using deep policy-gradient and value-function based reinforcement learning. doi:https://doi.org/10.1049/iet-its.2017.0153.

Mousavi, S.S., Schukat, M., Howley, E., 2018. Deep reinforcement learning: an overview. In: Bi, Y., Kapoor, S., Bhatia, R. (Eds.), Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016. Springer International Publishing, Cham, pp. 426–440.

Muresan, M., Fu, L., Pan, G., 2018. Adaptive traffic signal control with deep reinforcement learning – an exploratory investigation. 97th Annual Meeting of the Transportation Research Board. Washington, D.C.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT press.

Tsitsiklis, J.N., Van Roy, B., 1997. An analysis of temporal-difference learning with function approximation. IEEE Trans. Autom. Control 42 (5), 674–690. https://doi.org/10.1109/9.580874.

Van Der Pol, E., Oliehoek, F.A., 2016. Coordinated deep reinforcement learners for traffic light control. NIPS'16 Work. Learn. Inference Control Multi-Agent Syst.

van Hasselt, H., Guez, A., Silver, D., 2015. Deep reinforcement learning with double Q-learning. CoRR abs/1509.0.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N., 2015. Dueling network architectures for deep reinforcement learning. arXiv Prepr. arXiv1511.06581.

Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3), 279–292. https://doi.org/10.1007/BF00992698.

Yuan, J., Abdel-aty, M., 2018. Approach-level real-time crash risk analysis for signalized intersections. Accid. Anal. Prev., 274–289 https://doi.org/10.1016/j.aap.2018.07.031 119 April.

Yuan, J., Abdel-Aty, M., Wang, L., Lee, J., Yu, R., Wang, X., 2018. Utilizing bluetooth and adaptive signal control data for real-time safety analysis on urban arterials. Transp. Res. Part C Emerg. Technol. 97, 114–127. https://doi.org/10.1016/j.trc.2018.10.009.

Yuan, J., Abdel-Aty, M., Gong, Y., Cai, Q., 2019. Real-time crash risk prediction using long short-term memory recurrent neural network. Transp. Res. Rec. https://doi.org/10.1177/0361198119840611.

Update

Transportation Research Interdisciplinary Perspectives

Volume 9, Issue , March 2021, Page

DOI: https://doi.org/10.1016/j.trip.2021.100306

Contents lists available at ScienceDirect

Transportation Research Interdisciplinary Perspectives

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

Erratum

Erratum regarding missing Declaration of Competing Interest statements in previously published articles

Declaration of Competing Interest statements were not included in the published version of the following articles that appeared in previous issues of "Transportation Research Interdisciplinary Perspectives".

The appropriate Declaration/Competing Interest statements, provided by the Authors, are included below.

1. "Learning to build strategic capacity for transportation policy change: An interdisciplinary exploration" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100006] https://doi.org/10.1016/j.trip.2019.100006
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
2. "Comparison of Waze crash and disabled vehicle records with video ground truth" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100019] https://doi.org/10.1016/j.trip.2019.100019
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
3. "Autonomous, connected, electric shared vehicles (ACES) and public finance: An explorative analysis" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100038] https://doi.org/10.1016/j.trip.2019.100038
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
4. "Airline CEOs: Who are they, and what background and skill set are most commonly chosen to run the world's largest airlines?" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100054] https://doi.org/10.1016/j.trip.2019.100054
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
5. "Assessing service and price sensitivities, and pivot elasticities of public bikeshare system users through monadic design and ordered logit regression" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100015] https://doi.org/10.1016/j.trip.2019.100015
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
6. "Automated speed control on urban arterial road: An experience from Khon Kaen City, Thailand" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100032] https://doi.org/10.1016/j.trip.2019.100032
   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.
7. "Findings from a visibility survey in the construction industry" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100056] https://doi.org/10.1016/j.trip.2019.100056
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
8. "Decentralized network level adaptive signal control by multi-agent deep reinforcement learning" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100020] https://doi.org/10.1016/j.trip.2019.100020
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
9. "Multiobjective integrated signal-control system calibration in urban areas: Application of response surface methodology" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100011] https://doi.org/10.1016/j.trip.2019.100011
   Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
10. "Relationships between aggressive driving behaviors, demographics and pareidolia" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100037] https://doi.org/10.1016/j.trip.2019.100037
    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
11. "Practitioners' perspective on user experience and design of cycle highways" [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100010] https://doi.org/10.1016/j.trip.2019.100010

DOI of original article: https://doi.org/10.1016/j.trip.2019.100006. https://doi.org/10.1016/j.trip.2019.100019. https://doi.org/10.1016/j.trip.2019.100038. https://doi.org/10.1016/j.trip.2019.100054. https://doi.org/10.1016/j.trip.2019.100015. https://doi.org/10.1016/j.trip.2019.100032. https://doi.org/10.1016/j.trip.2019.100056. https://doi.org/10.1016/j.trip.2019.100020. https://doi.org/10.1016/j.trip.2019.100011. https://doi.org/10.1016/j.trip.2019.100037. https://doi.org/10.1016/j.trip.2019.100010. https://doi.org/10.1016/j.trip.2019.100036. https://doi.org/10.1016/j.trip.2019.100039. https://doi.org/10.1016/j.trip.2019.100043. https://doi.org/10.1016/j.trip.2019.100007. https://doi.org/10.1016/j.trip.2019.100052.

https://doi.org/10.1016/j.trip.2021.100306
Available online 13 February 2021
2590-1982/© 2021 Published by Elsevier Ltd.

Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.

12. “Scenario-based analysis for intermodal transport in the context of service network design models” [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100036] https://doi.org/10.1016/j.trip.2019.100036

    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
13. “Driver brake response to sudden unintended acceleration while parking” [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100039] https://doi.org/10.1016/j.trip.2019.100039

    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
14. “Analysis of severe and non-severe traffic crashes on wet and dry highways” [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100043] https://doi.org/10.1016/j.trip.2019.100043

    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
15. “Antecedents of flight delays in the Australian domestic aviation market” [Transportation Research Interdisciplinary Perspectives, 2019; Volume 1: 100007] https://doi.org/10.1016/j.trip.2019.100007

    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.
16. “Evaluation of traffic management strategies for special events using probe data” [Transportation Research Interdisciplinary Perspectives, 2019; Volume 2: 100052] https://doi.org/10.1016/j.trip.2019.100052

    Declaration of competing interest: The authors were contacted after publication to request a Declaration of Interest statement.