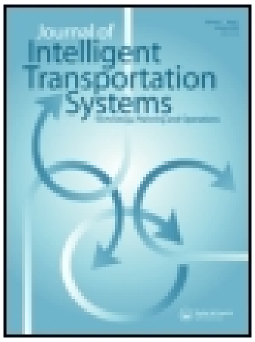

Journal of Intelligent Transportation Systems

Technology, Planning, and Operations

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

Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning

Ming Xu, Jianping Wu, Ling Huang, Rui Zhou, Tian Wang & Dongmei Hu

JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS
https://doi.org/10.1080/15472450.2018.1527694

Taylor & Francis
Taylor & Francis Group

Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning

Ming Xu[a], Jianping Wu[a], Ling Huang[b], Rui Zhou[c], Tian Wang[d], and Dongmei Hu[a]

[a]Department of Civil Engineering, Tsinghua University, Beijing, China; [b]School of Civil Engineering and Transportation, South China University of Technology, Guangzhou, China; [c]Telecommunications Engineering with Management, International School, Beijing University of Posts and Telecommunications, Beijing, China; [d]E-commerce and Law, International School, Beijing University of Posts and Telecommunications, Beijing, China

ABSTRACT
To improve the traffic efficiency of city-wide road networks, we propose a traffic signal control framework that prioritizes the optimal control policies on critical nodes in road networks. In this framework, we first use a data-driven approach to discover the critical nodes. Critical nodes are identified as nodes that would cause a dramatic reduction in the traffic efficiency of the road network if they were to fail. This approach models the dynamic of road networks using a tripartite graph based on the vehicle trajectories and can accurately identify the city-wide critical nodes from a global perspective. Second, for the discovered critical nodes, we introduce a novel traffic signal control approach based on deep reinforcement learning; this approach can learn the optimal policy via constantly interacting with the road network in an iterative mode. We conduct several experiments with a transportation simulator; the results of experiments show that the proposed framework reduces the average delay and travel time compared to the baseline methods.

ARTICLE HISTORY
Received 29 November 2017
Revised 18 September 2018
Accepted 20 September 2018

KEYWORDS
critical node; deep reinforcement learning; traffic signal controller; tripartite graph

Introduction

The optimization of traffic signal control is an important and interesting problem in the community of intelligent transportation system. Numerous studies have been done to develop various traffic signal strategies for improving traffic efficiency. A major branch of existing methods is to utilize reinforcement learning (RL) that attempt to learn an optimal policy function using a trial and error process (Abdulhai, Pringle, & Karakoulas, 2003; Aziz, Zhu, & Ukkusuri, 2018; Balaji, German, & Srinivasan, 2010; El-Tantawy, Abdulhai, & Abdelgawad, 2014; Grégoire, Desjardins, Laumônier, & Chaib-Draa, 2007; Lu, Liu, & Dai, 2008; Richter, Aberdeen, & Yu, 2007; Steingrover, Schouten, Peelen, Nijhuis, & Bakker, 2005; Walraven, Spaan, & Bakker, 2016). However, applying RL to traffic signal control in a large-scale urban area still poses two difficulties: (1) a large number of the traffic states and (2) a number of signal control actions that expands exponentially with the number of considered intersections.

For the first difficulty, deep reinforcement learning (DRL) (Casas, 2017; Gao, Shen, Liu, Ito, & Shiratori, 2017; Genders & Razavi, 2016; Jeon, Lee, & Sohn, 2017; Li, Lv, & Wang, 2016; Mnih et al., 2015; Mousavi, Schukat, Corcoran, & Howley, 2017; van der Pol & Oliehoek, 2016) that uses a deep neural network trained by RL to approximate the optimal policy function is proposed. DRL can learn a traffic state space representation, which can improve the generalization in dealing with the complexity of road network dynamics. However, DRL does not consider how to reduce the scale of the action space. For the second difficulty, several studies (Khamis & Gomaa, 2014; Kuyer, Whiteson, Bakker, & Vlassis, 2008; Pham et al., 2013; Ozan, Baskan, Haldenbilen, & Ceylan, 2015) propose multi-agents reinforcement learning (MARL) methods to reduce the size of signal control action space. With these methods, each agent controls a signalized intersection, and each agent can communicate with its neighboring agents to coordinate their control actions. However, most of their communications are based on a pre-determined protocol with a message passing process; that is, to obtain an optimal coordination, agents need to communicate with other agents many times until their local messages are fully

CONTACT Ling Huang ✉ hling@scut.edu.cn 🖃 School of Civil Engineering and Transportation, South China University of Technology, Guangzhou, China

exchanged. This leads to a slower computation as the number of agents increases. Another disadvantage of these methods is the enormous economic cost. Therefore, MARL is difficult to expand to a large-scale road network.

Several studies (Lämmer, Gehlsen, & Helbing, 2006; Li, Jiang, Rui, & Havlin, 2014; Qian, Wang, Xue, Zeng, & Wang, 2015) reveal that the phenomenon of cascading failures in road networks is ubiquitous. This finding means that when several critical nodes fail, a large number of nodes will also fail as a result of the connection between the nodes, and the entire road network may collapse. From this phenomenon, the following questions naturally arise. If only a small number of nodes can be managed and controlled for limited resources and capital, which nodes do we select and what control strategies are applied on the selected nodes to achieve a better performance of the road network?

To answer this question and to expand on previous studies, we propose a novel traffic signal control framework that can be applied to a large-scale area, even a city-wide road network. First, this framework uses a data-driven method to identify the critical nodes in the road network. The advantage of this method over previous methods is that it integrates four important factors to rank the node importance, including (1) the traffic capacity, (2) centrality, (3) irreplaceability of a node and (4) spatial relationships among neighboring nodes. From an economic perspective, we give the priority to optimal signal control strategies of the several critical nodes under the condition of limited capital. Second, motivated by studies of deep recurrent Q-networks (DRQN) (Hausknecht & Stone, 2015; Justesen, Bontrager, Togelius, & Risi, 2017; Lample & Chaplot, 2017), the proposed framework introduces a novel method based on DRQN to control the signals of critical nodes. This method uses a long short term memory (LSTM) to learn an optimal policy by interacting with the "SUMO", an open-source traffic simulator. Compared with DRL, DRQN can effectively sense the road network dynamics by inputting a time series of traffic information. Experiments validate the effectiveness of this framework.

Framework

As shown in Figure 1, the proposed framework can be divided into two phases: discovery of critical nodes and signal control policy learning.

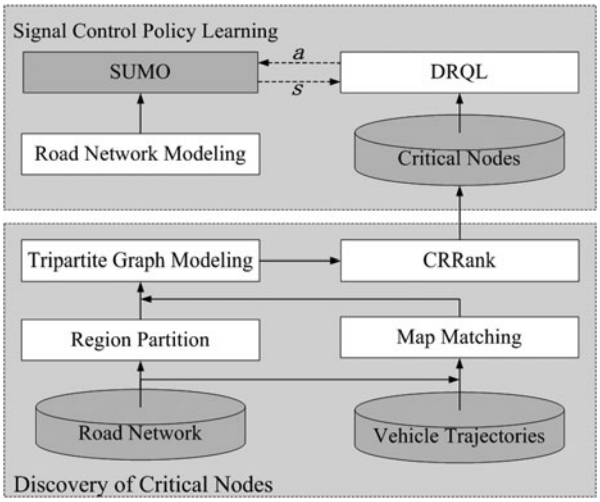

Figure 1. Framework of the proposed method.

Discovery of critical nodes

In this phase, CRRank (Xu et al., 2018) is used. Differing from other methods based on topology analysis (Batista & Bazzan, 2015; Niu, Li, Niu, Zhang, & Liu, 2015; Park & Yilmaz, 2010; Scardoni & Laudanna, 2013; Wu, Gao, & Sun, 2007), CRRank exploits the geographic information of the road network and the real massive vehicle trajectories, which can be easily collected via vehicle-mounted GPS devices. Due to the noise, the GPS points often deviate from the real road segments, the component of map matching is needed to transform each GPS reading in each trajectory to its corresponding road segment. In addition, to obtain the origin-destination-pair (OD-pair) information, the urban map is partitioned into disjoint regions with high-level road segments using a component of region partition. Based on the regions and processed trajectories, the trip information of all of the vehicles is modeled using a tripartite graph, which consists of three types of nodes: OD-pair, path and intersection. Next, CRRank is conducted on the tripartite graph to rank importance of each intersection. However, in the experimental phase, the framework uses a simulated trajectory dataset generated by SUMO, instead of real trajectories.

Signal control policy learning

According to the geographic information and investigated traffic flow distribution, a simulated road network is modeled and fed into SUMO. Each critical intersection is controlled by a DRQN model independently. Meanwhile, other intersections are configured with fixed interval signal timing. In the training phase, Each DRQN model continuously interacts with

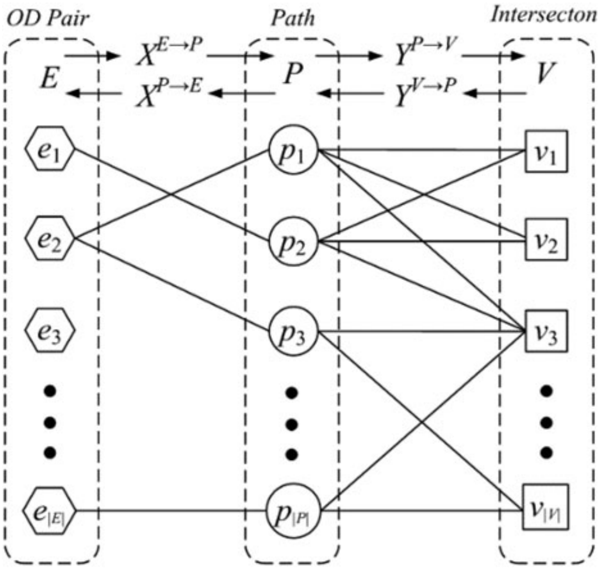

Figure 2. Tripartite graph of the trip network.

SUMO to learn the optimal policy of signal control via a trial and error process. In each turn, DRQN receives a sequence of traffic states and sends a signal control action to SUMO and receives the reward of an action from SUMO.

Methodologies

CRRank

An intuitive view is that the nodes with the high flow are important. However, recent studies (Wu et al., 2007; Zou, Wu, Gao, & Xu, 2014) found that the importance of a node is determined by the combination of its flow and betweenness, rather than any one of them. In addition, we found the irreplaceability of the paths and the spatial relationship between neighboring nodes were also useful. Specifically, the congestion of the downstream node has the potential to cause the congestion of the upstream node. Therefore, to accurately estimate the importance of nodes, we use CRRank that considers all the aforementioned factors. First, CRRank extracts information of flow distribution and path selection of drivers from vehicle trajectories and introduces a novel concept, the trip network, which reflects the dynamics of road network. The trip network can be modeled using a bidirectional tripartite graph represented formally by $G'(E \cup P \cup V, X \cup Y)$, as shown in Figure 2, where E, P and V represent the sets of nodes corresponding to the OD-pairs, paths, and intersections, respectively; the weighted matrixes are represented by X and Y, which are described in greater detail later. Second, CRRank ranks the node importance via score propagation over the tripartite graph.

To implement the score propagation, each type of nodes is assigned a meaning score that is updated during each iteration. In particular, we use OD-load vector L to represent the load on the road network caused by each OD-pair; vector H represents the popularity of each path, and vector C represents the importance of each intersection. Intuitively, the OD pair with a larger flow will likely place a heavier load on the road network, the path with a larger flow is likely more popular, and the intersection linking the higher-level roads is likely more important. To integrate such prior knowledge, CRRank defines three profile vectors $L^{(0)}$, $H^{(0)}$ and $C^{(0)}$, which represent the initial score lists of the OD-load, path popularity and intersection importance, respectively. These profile vectors are given as

$$L^{(0)} = \left[\frac{f_1}{\sum_i f_i} \quad \frac{f_2}{\sum_i f_i} \quad \cdots \quad \frac{f_{|E|}}{\sum_i f_i} \right], \tag{1}$$

$$H^{(0)} = \left[\frac{\delta_1}{\sum_i \delta_i} \quad \frac{\delta_2}{\sum_i \delta_i} \quad \cdots \quad \frac{\delta_{|P|}}{\sum_i \delta_i} \right], \tag{2}$$

$$C^{(0)} = \left[\frac{d_1}{\sum_i d_i} \quad \frac{d_2}{\sum_i d_i} \quad \cdots \quad \frac{d_{|V|}}{\sum_i d_i} \right], \tag{3}$$

where f_e is the traffic flow of the OD-pair e, δ_p is the size of flow on the path p and $d_i = \sum_j r_{ij}$; in this equation, r_{ij} is the level score of the jth road segment linking the intersection i. If the road segment is an expressway, freeway or arterial road, the value of r is set to 1.1, if the road segment is a sub-arterial road, r is set to 1.0 and if the road segment is a bypass, r is set to 0.9. The final scores for all node types can be synchronously calculated with the weighted matrixes iteratively. The weighted matrixes between E and P, denoted by $X^{E\longrightarrow,\mathrm{P}}$ and $X^{P\longrightarrow,\mathrm{E}}$, are given as

$$X_{ep}^{E\to P} = \begin{cases} \dfrac{\delta_p}{f_e} & \textit{if OD pair e select path p} \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

$$X_{pe}^{P\to E} = \begin{cases} 1 & \textit{if path p is selected by OD pair e} \\ 0 & \textit{otherwise} \end{cases} \tag{5}$$

Also, the weighted matrixes between P and V, denoted by $Y^{P\longrightarrow,\mathrm{V}}$and $Y^{V\longrightarrow,\mathrm{P}}$, are given as

$$Y_{pv}^{P\to V} = (u_{pv} + \sigma_{pv})c_v^{(0)}, \tag{6}$$

$$Y_{vp}^{V\to P} = u_{vp}h_p^{(0)}, \tag{7}$$

where $c_v^{(0)}$ is the vth entry of the vector $C^{(0)}$, $h_p^{(0)}$ is the pth entry of the vector $H^{(0)}$ and u_{pv} denotes the connectivity between p and v. Specifically, if p

traverses v, u_{pv} equals 1; otherwise u_{pv} equals 0; σ_{pv} is used to model the impact on the upstream intersection from the downstream intersection. The σ of the first intersection in each path is set to "0". For simplicity in simulation, CRRank sets $\sigma_{pv} = \sigma_{pv'} + 0.01$, in this equation, v' is the last upstream intersection in p. However, in fact, σ_{pv} should be modeled as a function of distance between the downstream and upstream intersection. The score propagation in CRRank consists of a forward phase and a reverse phase. In the forward phase, H and C are updated as the following:

$$H = \alpha\left[X^{E\rightarrow P}\right]^T L + (1-\alpha)H^{(0)}, \tag{8}$$

$$C = \alpha\left[Y^{P\rightarrow V}\right]^T H + (1-\alpha)C^{(0)}, \tag{9}$$

where α is the damping factor and is set to 0.85 like PageRank (Brin and Page, 1998). In the reverse phase, H and L are updated as the following:

$$H = \alpha\left[Y^{V\rightarrow P}\right]^T C + (1-\alpha)H^{(0)}, \tag{10}$$

$$L = \alpha\left[X^{P\rightarrow E}\right]^T H + (1-\alpha)L^{(0)}. \tag{11}$$

The forward phase and the reverse phase are alternated until convergence, which reflects the mutually reinforcing relationships among the OD-load, path popularity and intersection importance as follows: an OD-pair with a heavy load is likely to choose several or only one popular path. A popular path is likely to be chosen by an OD-pair with a heavy load; similarly, a popular path is likely to traverse many important intersections. An important intersection is likely to be traversed by many popular paths. To ensure convergence, L, H and C are normalized by$L = L/L$, $H = H/H$ and $C = C/C$ after each iteration. In CRRank, the significant factors of critical nodes are naturally integrated into the calculation process, including traffic capacity, centrality, irreplaceability of nodes and impact among the neighboring nodes.

Proposed DRQN

On each critical intersection discovered by CRRank, an approximation of optimal policy function for signal control is obtained using a deep recurrent Q-network. In this subsection, we describe in detail the components of the proposed DRQN.

Model

In RL, at each time step t, an agent observes a state $s^{(t)} \in S$, and chooses an action $a^{(t)} \in A$ according to a policy π, which needs to be learned through interacting with the environment. The objective of π is to maximize its expected discount reward

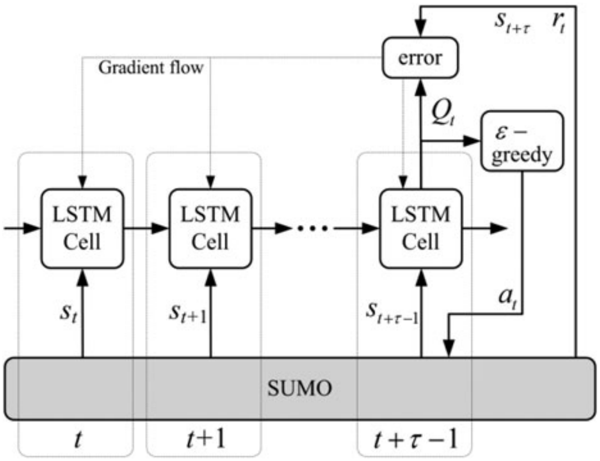

Figure 3. Framework of the proposed DRQN.

$$R^{(t)} = \sum_{t=0}^{+\infty} \gamma^t, \tag{12}$$

where $R^{(t)}$ is the reward that the agent receives after performing action $a^{(t)}$; γ is the discount factor. The Q-function is used to estimate how good it is to perform an action in a given state under a policy π; therefore, it is the action-value function for policy π.

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r^{(t+1)} + \gamma \max_{a'} Q^*\left(s', a'\right)|s,a\right]. \tag{13}$$

In DQN, the neural network with parameters θ, denoted by $Q(s, a; \theta)$, is used to approximate to Q-function. In addition, it is natural to think that $Q(s, a; \theta)$ can be learned by minimizing the following squared TD-error $l(\theta)$ with an iterative update,

$$\ell_i(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q^*\left(s', a'; \theta_{i-1}\right) - Q(s,a;\theta_i)\right)^2\right], \tag{14}$$

where i is the serial number of current iteration. At each iteration, the parameters θ_i are updated as

$$\theta_i = \theta_{i-1} + \varepsilon \nabla_{\theta_{i-1}} \ell_{i-1}(\theta_{i-1}). \tag{15}$$

To optimize loss function $\ell(\theta)$, the parameters θ_{i-1} from previous iteration should be kept unchanged at the current iteration i. Using DQN in the signal control problem, a disadvantage is that a single observation or a fixed number of observations cannot accurately reveal the dynamics of traffic situation. This shortcoming is observed because the trend in the traffic situation of a road is usually determined by that of its neighboring roads at an earlier time and DQN cannot remember the historical information beyond the current input. To solve this problem, we introduce a DRQN (presented in Figure 3) that uses an LSTM to approximate the Q function. DRQN captures the view of traffic flow at current time step with a hidden state and integrates hidden states across multiple time steps to make decision on the next action. Specifically, at each time step, DRQN receives a traffic observation as input. At each τ time steps, DRQN outputs an action and later receives a reward. Here, we set $\tau = 8$.

State space

Due to the advantage of a deep neural network in learning representation of data, the state space should reflect the view of road network dynamics as completely as possible. In this paper, we assume that an agent can observe the actions and local traffic situation of its neighboring intersections, as well as its local traffic situation. The observations of an agent at each time step is composed of five vectors $\langle m_g\ w_g\ \ m_n\ w_n\ z\rangle$, where m_g and w_g are the vectors for the number of vehicles and average speed on each lane linking the controlled intersection, respectively, m_n and w_n are the vectors of the number of vehicles and average speed on each lane linking each neighbor of the controlled intersection, respectively, and z is the signal state vector of the controlled intersection and its neighbors.

Action space

An agent chooses an action from the set of available actions at a regular interval according to its observations. In general, traffic signals refer to three color phases, green signal allows the vehicles to go through the intersection, yellow signal warns the vehicles to slow down for the coming red signal, and red signal prohibits any vehicle from proceeding. In general, the possible settings of signal phases are East-West Green (EWG), East-West Left-turn Green (EWLG), North-South Green (NSG) and North-South Left-turn Green (NSLG). A naïve method is to consider each color setting of signal phases as an action. However, this step does not conform to the typical traffic signal control conventions and may cause traffic chaos. To be safe, the sequence of setting signal phases should be in loop mode, i.e., $\cdots \rightarrow$ EWG $\rightarrow$ EWLG $\rightarrow$ NSG $\rightarrow$ NSLG $\rightarrow$ EWG $\rightarrow \cdots$. Therefore, we give only two possible actions as {N, A}, where N represents keeping current signal phase unchanged; A represents making the transition from current signal phase to the next. In addition, a yellow phase spanning three seconds is added before the signal transition from green to red, which is not treated as an action for DRQN and automatically executes by traffic signal control system.

Reward function

The reward is to reflect the effect of an action on the environment. To achieve optimal signal control, a naïve approach is to define the reward $r(t)$ as the average delay of all of the vehicles on the involved road segments, given by

$$r^{(t)} = \frac{1}{N_t}\sum_{i=1}^{N_t} k_i, \tag{16}$$

where N_t is the number of vehicles on the linked road segments at current time t, and k_i is the waiting time of vehicle i. However, such a reward may lead to excessive awaiting time of minority drivers. That possibility means that, to minimize the vehicle delays from a global perspective, DRQN may tend to keep the signal of the lanes with heavy traffic as green, while the red signal phases of other lanes with small traffic are extended for a longer time. To prevent this situation, the reward $r(t)$, inspired by the well-known link congestion function in transportation planning developed by U.S. Bureau of Public Roads (Wu et al., 2007; Wu, Gao, Sun, & Huang, 2006), is defined as

$$r^{(t)} = \frac{1}{N_t}\sum_{i=1}^{N_t} \eta\left[1-\left(\frac{k_i}{C}\right)^{\rho}\right], \tag{17}$$

where C represents tolerable waiting time; η and ρ are constants. Here, we set $C = 60$, $\eta = 0.15$ and $\rho = 2$.

Algorithm 1. DRQN for signal control.

Input: Replay memory D with size M the number of training episodes N
Output: the parameters θ of Q-network

```
Initialize the parameters $\theta$ of Q-network, $\tilde{\theta}$ of target network
Initialize replay memory D with size M
for each episode $e$ do
    Initialize a simulation episode
    for t:=1 to MAXTIME do
        select an action $a_t$ with $\varepsilon$ –greedy strategy
        execute $a_t$ in SUMO and get reward $r_t$ and next state $s_{t+1}$
    end for
    for t := 1 to MAXTIME - $\tau$ do
        for k := t to t + $\tau$ do
            append ($s_t$, $a_t$, $r_t$, $s_{t+1}$) to sequence $l$
        end for
        put sequence $l$ into $D$
    end for
    sample random mini-batch of sequences from $D$
    for k := $\tau + 1$ to 1, $-1$ do
```

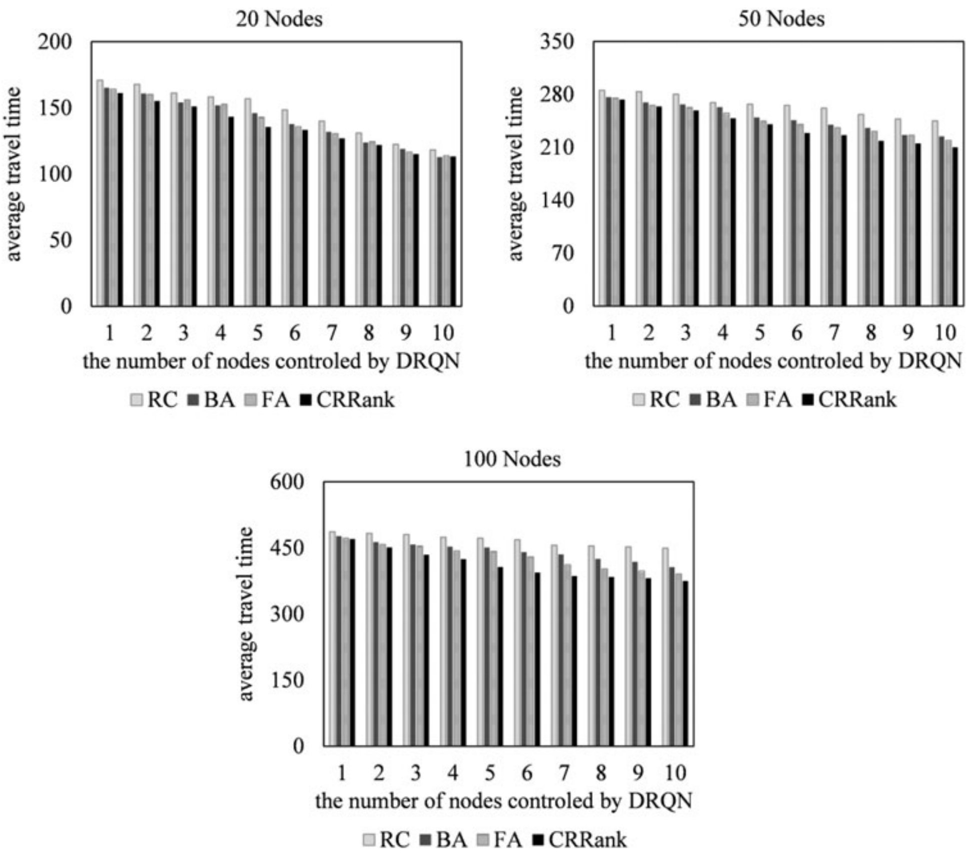

Figure 4. The relation between the average travel time and the cumulative number of critical nodes.

```
        if $s_k$ is terminal then
            $y_k := r_k$
        else $y_k := r_k + \gamma \max_{a'} Q(s_{k+1}, a'; \theta)$
        end if
        accumulate gradients for $\left(y_k - Q\left(s_k, a_k; \widetilde{\theta}\right)\right)^2$
    end for
    $\theta_{i+1} := \theta_i + \varepsilon \nabla_\theta \ell(\theta)$
    $\widetilde{\theta}_{i+1} := \widetilde{\theta}_i + \widetilde{\varepsilon}\left(\theta_{i+1} - \widetilde{\theta}_i\right)$
end for
```

Experiments and evaluation

In this section, we describe the experiments conducted in SUMO to evaluate the performance of the proposed approach. SUMO is deployed on a 64-bit server with 32-cores 3.2 GHz CPU and a 128 GB memory. We develop a proxy layer to be in charge of intersection between deep neural networks and SUMO. Specifically, the proxy layer receives the action commands from the DRQNs, carries out these actions in batches, then reads the states and rewards from SUMO and returns them to DRQNs. The deep neural networks are developed using TensorFlow library and deployed on a workstation with an NVIDIA Geforece GTX 1080 Ti of 11 GB memory.

We generate four distinct scales synthetic road networks with 20 nodes, 50 nodes and 100 nodes, respectively, in SUMO. The duration of each simulation episode is 3600 time steps. We assume that the probability of vehicles arriving follows a Poisson process.

Discovery of critical nodes

We trace each vehicle in each simulation and record its trajectories in the database. Based on the trajectory data, we build the tripartite graph to identify the critical nodes.

To evaluate the effectiveness of prioritizing an optimal control strategy of critical nodes, we compare CRRank with three baseline algorithms, the flow-based algorithm (FA) calculating the importance score of a node according only to its traffic flow, the betweenness-based algorithm (BA) calculating the importance score of a node according only to its betweenness centrality (Kirkley, Barbosa, Barthelemy, & Ghoshal, 2018), and randomly choosing nodes (RC). We implement DRQN-based signal control algorithm on the top ten nodes in the results of RC, FA, BA and CRRank, respectively. Figure 4 shows the results of this experiment. Overall, the average travel time is declining with the increase of the numbers of the nodes controlled by DRQN. In the 20-node

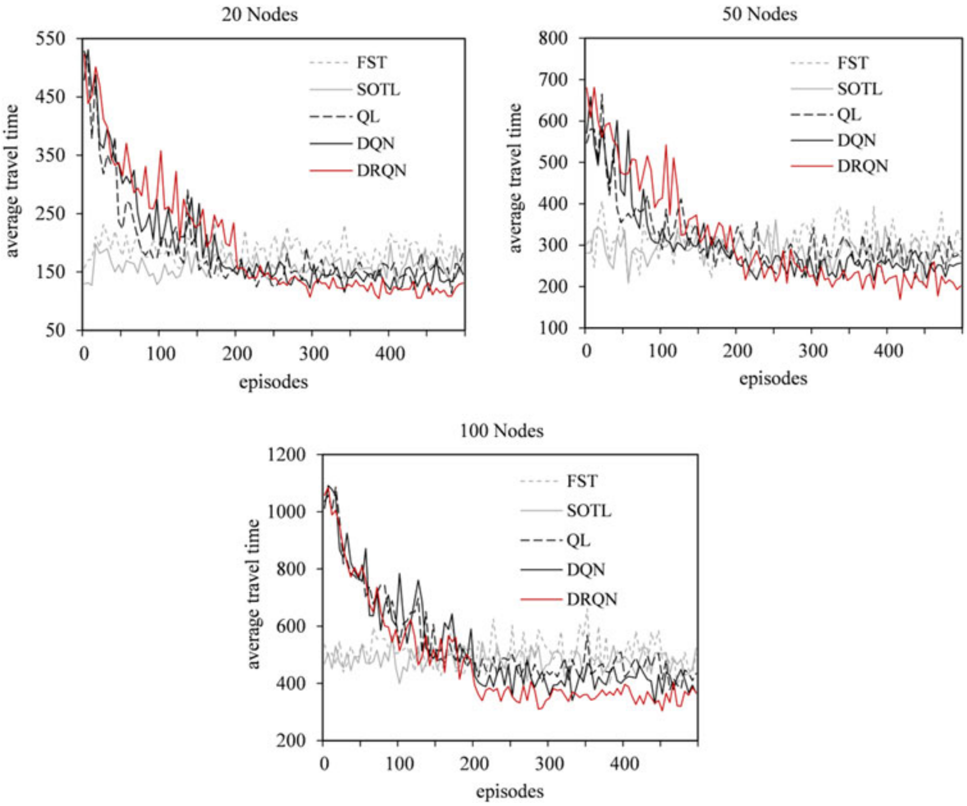

Figure 5. The relation between the average travel time and the number of episodes.

Table 1. The average travel time of different scale networks controlled by different algorithms (after convergence).

	FST	SOTL	QL	DQN	DRQN
20 Nodes	196.3	166.4	173.8	146.8	117.5
50 Nodes	304.7	286.2	278.9	263.8	221.6
100 Nodes	497.1	476.3	448.0	412.5	379.8

network, the downward trends of all the four algorithms are very close. However, we observed that in the 50-node network and 100-node network, the control of the randomly selected nodes using DRQN has a smaller impact on the improvement of the traffic efficiency, FA performs slightly better than BA and CRRank still performs best. This finding demonstrates that prioritized control on the critical nodes with an optimized strategy can improve the traffic efficiency of road networks. Compared with the baseline algorithms, CRRank has an advantage in identifying the critical nodes on large-scale road networks.

Signal control policy learning

In this subsection, we evaluate the effectiveness of DRQN, and we use the following baselines.

- *Fixed Signal Timing plan* (*FST*) with each signal phase of 30 seconds
- *Self-Organizing Traffic Lights* algorithm (*SOTL*) (Cools, Gershenson, & D'Hooghe, 2013) turning the signal phase according to the elapsed time and the number of vehicles in queues
- *Q-Learning* (*QL*) representing the policy function as a table
- *Deep Q-network* (DQN) combining deep neural network and Q-learning

The configurations of QL and DQN are the same as DRQN, including the state space, action space, the reward and the discount factor γ. In each network, we select the top 10 critical nodes as the controlled nodes using CRRank. Figure 5 and Table 1 show the overall performance of the five methods. As shown in Figure 5, for the three algorithms based on Q-learning, QL, DQN and DRQN, the average travel time of all vehicles on all three networks decreases as the number of simulation episodes increases. After approximately 200 episodes, the models converge and learn the signal control policies that improve the traffic efficiency compared with the FST and SOTL. DQN outperforms QL slightly, since QL requires discretization of the traffic state space, which decreases the accuracy, while DQN can sense the fine-grained traffic situation with the advantage of deep learning. DRQN performs best, which indicates that the hidden state of LSTM capturing historical traffic information can improve the policy.

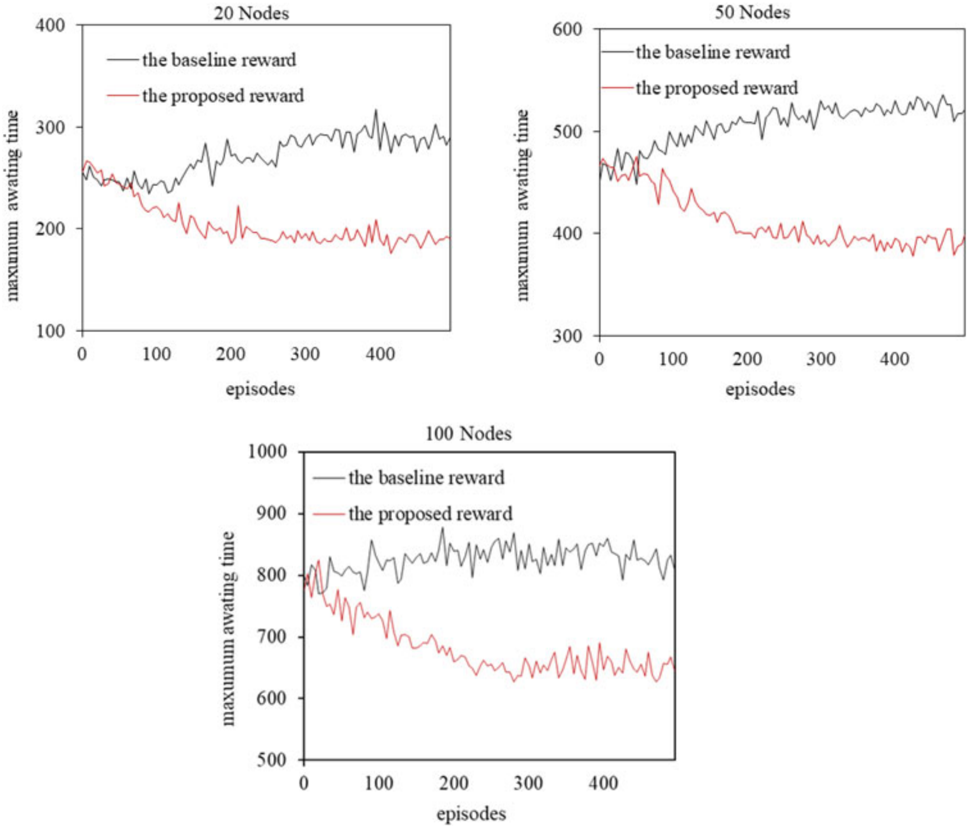

Figure 6. The relation between the maximum cumulative delay and the number of episodes using different reward in DRQN.

To understand the difference between the proposed reward function and the baseline reward defined in formula (10), we implement both of them in DRQN, and investigate the maximum delay defined as the maximum waiting time of all of the vehicles in a simulation episode. The waiting time of a vehicle is the cumulative duration of speed less than 10 km/h. Figure 6 shows the relationship between the maximum delay and the number of training episodes using different rewards in the three networks. From Figure 6, we observe that the proposed rewards can reduce the maximum delay to a certain extent as we expected. This approach prevents several drivers from blocking traffic for a quite long time and is of vital importance to improve the travel experience.

Conclusions

This paper proposed a signal control framework to improve the traffic efficiency of a large-scale road network. To make the problem tractable, a small number of critical nodes are preferred to control by optimal strategies. First, to discover the critical nodes from vehicle trajectories, we use CRRank, which ranks node importance simultaneously considering capacity, centrality, substitutability of a node, and spatial relationships among neighboring nodes. Second, for each critical node, we use an LSTM trained by RL to learn an optimal signal control policy. The experiments show that the proposed method outperforms all of the baseline methods.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China under Grant No. U1509205 and No. 51408237, and Toyota Research Center.

References

Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3), 278–285.

Aziz, H. A., Zhu, F., & Ukkusuri, S. V. (2018). Learning-based traffic signal control algorithms with neighborhood information sharing: An application for sustainable mobility. *Journal of Intelligent Transportation Systems*, 22(1), 40–52.

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

Batista, R. D. A., & Bazzan, A. L. C. (2015). Identification of central points in road networks using betweenness centrality combined with traffic demand. *Polibits*, (52), 85–91.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. In P. H. Enslow & A. Ellis (Eds.) *WWW 1998. Proceedings of the 7th International Conference on World Wide Web* (pp. 107–117). Brisbane, Australia: Elsevier Science Publishers B. V.

Casas, N. (2017). Deep deterministic policy gradient for urban traffic light control. arXiv preprint arXiv: 1703.09035.

Cools, S. B., Gershenson, C., & D'Hooghe, B. (2013). Self-organizing traffic lights: A realistic simulation. In M. Prokopenko (Ed.) *Advances in applied self-organizing systems* (pp. 45–55). London: Springer.

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 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 preprint arXiv:1705.02755.

Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142.

Grégoire, P. L., Desjardins, C., Laumônier, J., & Chaib-Draa, B. (2007, September). Urban traffic control based on learning agents. In D. J. Dailey (Ed.), ITSC 2007. *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems* (pp. 916–921). Seattle, WA: IEEE.

Hausknecht, M., & Stone, P. (2015). Deep recurrent q-learning for partially observable MDPS. CoRR, abs/1507.06527.

Jeon, H., Lee, J., & Sohn, K. (2017). Artificial intelligence for traffic signal control based solely on video images. *Journal of Intelligent Transportation Systems*, 22, 1–13.

Justesen, N., Bontrager, P., Togelius, J., & Risi, S. (2017). Deep learning for video game playing. arXiv preprint arXiv:1708.07902.

Khamis, M. A., & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, 29, 134–151.

Kirkley, A., Barbosa, H., Barthelemy, M., & Ghoshal, G. (2018). From the betweenness centrality in street networks to structural invariants in random planar graphs. *Nature Communications*, 9(1), 2501.

Kuyer, L., Whiteson, S., Bakker, B., & Vlassis, N. (2008). Multiagent reinforcement learning for urban traffic control using coordination graphs. In W. Daelemans & K. Morik (Eds.), *ECML PKDD 2008. Proceedings of the Machine Learning and Knowledge Discovery in Databases 2008* (pp. 656-671). Antwerp, Belgium: Springer.

Lämmer, S., Gehlsen, B., & Helbing, D. (2006). Scaling laws in the spatial structure of urban road networks. *Physica A: Statistical Mechanics and Its Applications*, 363(1), 89–95.

Lample, G., & Chaplot, D. S. (2017). Playing FPS games with deep reinforcement learning. In S. P. Singh, & S. Markovitch (Eds.), AAAI-17. *Proceedings of the 31th AAAI Conference on Artificial Intelligence* (pp. 2140–2146). San Francisco, CA: AAAI.

Lu, S., Liu, X., & Dai, S. (2008, June). Incremental multistep Q-learning for adaptive traffic signal control based on delay minimization strategy. In Y. Sun, X. Guo, J. He, et al. (Eds.), WCICA 2008. *Proceedings of the 7th World Congress on Intelligent Control and Automation* (pp. 687–691). Chongqing, China: IEEE.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Mousavi, S. S., Schukat, M., Corcoran, P., & Howley, E. (2017). Traffic light control using deep policy-gradient and value-function based reinforcement learning. arXiv preprint arXiv:1704.08883.

Niu, S. Y., Li, B., Niu, W. J., Zhang, J. S., & Liu, W. F. (2015). Evaluation of highway network node importance via node benefit function and weighted node betweenness. In *CICTP 2015* (pp. 2182–2191).

Ozan, C., Baskan, O., Haldenbilen, S., & Ceylan, H. (2015). A modified reinforcement learning algorithm for solving coordinated signalized networks. *Transportation Research Part C: Emerging Technologies*, 54, 40–55.

Park, K., & Yilmaz, A. (2010, April). *A social network analysis approach to analyze road networks*. Paper presented at ASPRS 2010 Annual Conference, San Diego, CA.

Pham, T. T., Brys, T., Taylor, M. E., Brys, T., Drugan, M. M., Bosman, P. A., & Steenhoff, D. (2013, May). *Learning coordinated traffic light control*. Paper presented at the 11th Adaptive and Learning Agents workshop (at AAMAS-13), St. Paul, MN, USA.

Qian, Y., Wang, B., Xue, Y., Zeng, J., & Wang, N. (2015). A simulation of the cascading failure of a complex network model by considering the characteristics of road traffic conditions. *Nonlinear Dynamics*, 80(1–2), 413–420.

Richter, S., Aberdeen, D., & Yu, J. (2007). Natural actor-critic for road traffic optimisation. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), NIPS 2007. *Proceedings of the Advances in neural information processing systems* (pp. 1169–1176). Vancouver, B.C., Canada: Curran Associates.

Scardoni, G., & Laudanna, C. (2013). Identifying critical road network areas with node centralities interference and robustness. In In R. Menezes, A. Evsukoff & M. C. González (Eds.), *Complex networks* (Vol. 424, pp. 245–255). Berlin: Springer.

Steingrover, M., Schouten, R., Peelen, S., Nijhuis, E., & Bakker, B. (2005, October). Reinforcement learning of traffic light controllers adapting to traffic congestion. In K. Verbeeck, K. Tuyls, A. Nowé, B. Manderick & B. Kuijpers (Eds.), BNAIC 2005, *Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence* (pp. 216–223). Brussels, Belgium: KVAB.

van der Pol, E., & Oliehoek, F. A. (2016). *Coordinated deep reinforcement learners for traffic light control*. Paper presented at the NIPS 2016 Workshop on Learning, Inference and Control of Multi-Agent Systems, Barcelona, Spain.

Walraven, E., Spaan, M. T., & Bakker, B. (2016). Traffic flow optimization: A reinforcement learning approach. *Engineering Applications of Artificial Intelligence*, 52, 203–212.

Li, D., Jiang, Y., Rui, K., & Havlin, S. (2014). Spatial correlation analysis of cascading failures: congestions and blackouts. *Scientific Reports*, 4(4), 5381.

Li, L., Lv, Y., & Wang, F. Y. (2016). Traffic signal timing via deep reinforcement learning. IEEE/CAA *Journal of Automatica Sinica*, 3(3), 247–254.

Wu, J. J., Gao, Z. Y., & Sun, H. J. (2007). Effects of the cascading failures on scale-free traffic networks. *Physica A: Statistical Mechanics and Its Applications*, 378(2), 505–511.

Wu, J. J., Gao, Z. Y., Sun, H. J., & Huang, H. J. (2006). Congestion in different topologies of traffic networks. *Europhysics Letters*, 74(3), 560.

Xu, M., Wu, J., Liu, M., Xiao, Y., Wang, H., & Hu, D. (2018). Discovery of critical nodes in road networks through mining from vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*. doi:10.1109/TITS.2018.2817282

Zou, Z., Wu, J., Gao, J., & Xu, X. (2014). Cascade defense in urban road network by inserting modular topologies. *Kybernetes*, 43(5), 750–763.