Random Ensemble Reinforcement Learning for Traffic Signal Control

Ruijie Qi\textsuperscript{1}, Jianbin Huang\textsuperscript{2*}, He Li\textsuperscript{3}, Qinglin Tan\textsuperscript{4}, Longji Huang\textsuperscript{5} and Jiangtao Cui\textsuperscript{6}

\textsuperscript{1,2,3,4,5,6} School of Computer Science and Technology, Xidian University, Xi’an 710126, China
qruijie@stu.xidian.edu.cn, jbhuang@xidian.edu.cn, hli@xidian.edu.cn, qltan@stu.xidian.edu.cn, longji.huang018@gmail.com, cuijt@xidian.edu.cn

Abstract

Traffic signal control is a significant part of the construction of intelligent transportation. An efficient traffic signal control strategy can reduce traffic congestion, improve urban road traffic efficiency and facilitate people’s lives. Existing reinforcement learning approaches for traffic signal control mainly focus on learning through a separate neural network. Such an independent neural network may fall into the local optimum of the training results. Worse, more, the collected data can only be sampled once, so the data utilization rate is low. Therefore, we propose the Random Ensemble Double DQN Light (RELight) model. It can dynamically learn traffic signal control strategies through reinforcement learning and combine random ensemble learning to avoid falling into the local optimum to reach the optimal strategy. Moreover, we introduce the Update-To-Data (UTD) ratio to control the number of data reuses to improve the problem of low data utilization. In addition, we have conducted sufficient experiments on synthetic data and real-world data to prove that our proposed method can achieve better traffic signal control effects than the existing optimal methods.

1 Introduction

Traffic congestion has always been a problem in the construction of smart cities. Congested vehicles not only affect the traffic efficiency of urban roads but also cause great environmental pollution and waste of resources. There needs to be a smart traffic signal control algorithm to handle the situation, increase the throughput of intersections, reduce the travel time of vehicles, and shorten the length of the vehicles queue waiting for the green light to pass.

Traditional traffic signal control methods are usually artificially set in advance. Such as pre-defined fixed-time method[Webster, 1958; Miller, 1963], actuated control method[Fellendorf, 1994; Mirchandani and Head, 2001], adaptive control method[Haddad \textit{et al.}, 2010; Silcock, 1997; Wong and Wong, 2003], optimization-based control method[Varaiya, 2013]. These methods require experts to design rules based on experience. However, the transportation system is dynamically changing. Fixed rules cannot effectively control traffic signal in response to real-time changes.

Recently, researchers start to investigate reinforcement learning (RL) techniques for traffic signal control. Reinforcement learning learns through the real-time interaction between the agent and the environment, so it has better learning ability for complex dynamic environments. Various work based on reinforcement learning technology has performed better than traditional control methods[Wei \textit{et al.}, 2018; El-Tantawy and Abdulhai, 2012; Nishi \textit{et al.}, 2018].

Existing reinforcement learning approaches for traffic signal control mainly focus on learning through a separate neural network. Such an independent neural network may cause non-convergence, instability, or even fall into the local optimum of the training results. In the process of reinforcement learning, research on improving data utilization has also received a lot of attention in recent years. REDQ[Chen \textit{et al.}, 2021] introduces the Update-To-Data (UTD) ratio to improve data utilization, but it is not applicable for traffic signal control problems with discrete action spaces.

To solve these problems, we first randomly initialize multiple DDQN networks to conduct the learning process together according to the idea of ensemble learning[Touretzky \textit{et al.}, 1997]. By mixing networks, it can effectively avoid falling into the local optimum and learn the optimal traffic signal control strategy. During the interaction, all networks vote together to participate in the choice of action. During the learning process, a subset of networks is randomly selected as the target network to update all networks. The strategies learned by multiple networks can effectively expand the strategy space, and only a subset of them used for updating can speed up the learning speed of the agent.

In this article, we propose a novel traffic signal control framework named Random Ensemble Double DQN Light(RELight), which uses a model-free algorithm for discrete action space but effectively improves data utilization. To improve data utilization, we introduce the Update-To-Data (UTD) ratio in the traffic signal control problem, which is the amount of data taken by the agent to update compared to the amount of data that the agent interacts with the environment. We use the Q network to score each action to select the action and solve the problem of discrete action space in traffic signal control.
control. We propose to use the variance of queue length as a part of the reward function. Because the minimization of the variance of the queue length can balance the queue length in different directions at the intersection. In this way, the learned strategy will not fall into the local optimum and avoid sacrificing waiting vehicles in one direction.

Through carefully designed experiments, we provide a specific detailed analysis on RELight. The experimental results prove that our method can learn a more optimal strategy to control traffic signals, shorten vehicle traffic time and reduce the length of queuing vehicles. In addition, the learning process is more stable, and the data utilization rate is significantly improved. We conducted an ablation study and the results showed that RELight is very robust in the choice of hyperparameters.

The rest of this paper is organized as follows. Section 2 discusses the literature. Section 3 formally defines the problem. The method is shown in Section 4 and the experiment results are shown in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

In this section, we firstly introduce traditional traffic signal control methods, then introduce reinforcement learning methods[Sutton and Barto, 1998].

2.1 Conventional Traffic Signal Control

Many traffic signal control methods were first proposed in the field of transportation research.

Pre-defined fixed-time control method[Urbanik et al., 2015; Roess et al., 2011] is a control method that is completely designed in advance by human experts based on experience. Actuated control is method also composed of pre-defined rules, but traffic signals can be adapted to real-time traffic data. Adaptive control method is not a set of rules but a set of traffic signal plans. Optimization-based control method[Robertson, 1969] can be adjusted according to real-time traffic conditions, unlike the previous method that relies on some hand-designed external rules. Although this method is based on traffic data for traffic signal control, it uses too many assumptions and is difficult to apply in practice.

2.2 Reinforcement Learning for Traffic Signal Control

Recently, many works have applied reinforcement learning technology to the research of traffic signal control. These results show that reinforcement learning has a better effect on the control of dynamic systems. Q-learning is the most classic algorithm in reinforcement learning algorithms. It is suitable for scenarios in discrete action space, and the action space in traffic signal control problems is conventionally defined as a set of discrete actions[Wei et al., 2018]. Some work has also achieved good results by redefining the traffic state and also using the DQN network model[Wei et al., 2019]. There is also work by redefining the phase of the traffic signal and constructing the corresponding network structure for effective learning and control[Zheng et al., 2019].

However, Q-learning may usually cause over-estimation of the Q value. Due to the instability of the dynamic environment and the random initialization of the deep reinforcement learning network, the final convergence process of the reinforcement learning algorithm is prone to be unstable, and large deviations occur during different training processes. In order to improve this situation, we naturally thought of the ensemble idea of machine learning.

Research on the characteristics of ensemble learning[Touretzky et al., 1997], uses some simple models designs to analyze different levels of the ensemble. Combining ensemble learning with reinforcement learning,[Chen et al., 2021] has given a good demonstration. In this paper, for the first time, random ensemble learning and deep reinforcement learning are applied to traffic light control problems and have achieved good results.

3 Preliminary

3.1 Environment

In this paper, we study traffic signal control in a single intersection scenario. We use a standard four-way intersection as an example for the following explanation. But these concepts can be easily transplanted to a three-way or a five-way intersection.

- **Entering approach:** The four directions of each intersection are named: North, South, East and West("N", "S", "E", "W" for short).
- **Signal phase:** In order to define the problem in a simple and formal way, we assume that the traffic signal has only two phases: a green light in the east-west direction and a red light in the other direction(WEG); a green light in the north-south direction and a red light in the other direction(NSG).

| Notation | Meaning |
|----------|---------|
| $S$      | state space |
| $A$      | action space |
| $R$      | reward function |
| $\gamma$ | discount factor |
| $L$      | queue length |
| $\bar{L}$ | average value of queue length on all lanes |
| $l_i$    | queue length on each lane i |
| $w_i$    | vehicle waitong time on each lane i |
| $n_i$    | number of vehicles in the queue on each lane i |
| $p_c$    | current phase |
| $p_n$    | next phase |
| $V$      | variance of queue length |
| $d_i$    | Delay of vehicles on each lane i |
| $w_t$    | waiting time of vehicles on each lane i |
| $C$      | whether to switch phase signal |
| $N$      | number of vehicles passing through the intersection in a time step |
| $T$      | travel time of vehicles N passing in a time step |

Table 1: Notation
3.2 Problem Definition

In our problem, an intersection is controlled by an independent agent. The agent observes the traffic environment, makes actions, and then learns according to the rewards of the environment’s feedback. The goal of the agent is to learn an optimal strategy for manipulating traffic phases in order to optimize the travel time of the vehicle through the intersection. This traffic signal control problem can be formalized as a Markov decision process $<S, A, P, R, \gamma>$ [Sutton and Barto, 1998].

**Problem 1.** Given the state space $S$, the action space $A$, and a reward function $R(s, a)$. Our work is based on the model-free method, so the state transition probability function $P$ is unknown. The goal of the agent is to learn an optimal strategy $\pi(a|s)$ to maximize the expectation of discounted rewards. The agent selects best actions for different states according to this strategy.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \quad (1)$$

4 METHOD

4.1 Framework

In order to reduce the variance in the Q function estimation, we adopt the ensemble learning framework REDQ as our algorithmic framework [Chen et al., 2021]. Similar with REDQ, standard reinforcement learning is divided into two parts: acting and learning. The RELight framework we proposed is shown in Figure 1. The dotted line in Figure 1 is acting, and the solid line is learning.

4.2 Agent Design

In this part, we introduce the specific design of state space, action space, and reward function.

- **State space $S$.** In this paper, we define the state of one intersection. For each lane $i$ in the intersection, the state consists of the following parts: queue length $l_i$, vehicle waiting time $w_i$, and the number of vehicles in the queue $n_i$, and traffic signal state information: current phase $p_c$, next phase $p_n$.
- **Set of actions $A$.** The action space is defined as whether to change the phase of the traffic signal $a$. If the agent decides to switch traffic signal phase, set $a = 1$; otherwise, $a = 0$.
- **Reward $R$.** In this paper, reward is defined as the weighted sum of the following parts:

$$Reward = w_1 \cdot V + w_2 \cdot \sum_{i \in l} d_i + w_3 \cdot \sum_{i \in l} w_i + w_4 \cdot C + w_5 \cdot N + w_6 \cdot T. \quad (2)$$

The detailed definition of each item in the reward function is as follows:

1. The variance of queue length on all approaching lanes. The queue length is defined as the number of waiting vehicles on each road, we count the number of waiting vehicles on each road, and then calculate their variance on all lanes.

$$V = \sum_{i \in l} (L_i - \bar{L})^2. \quad (3)$$

2. Delay of each lane $i$.
3. Waiting time of vehicles on each lane $i$.
4. Whether to switch the sign of the traffic light, it only contains 0 or 1.
5. The number of vehicles $N$ passing through the intersection in a time step.
6. The travel time of vehicles $N$ passing in a time step.

4.3 Network Structure

The specific network structure is shown in Figure 2. In the learning process, we adopted Double DQN as the core Q network.
Basicly, our network takes the vehicle state and the traffic
signal phase state at the intersection as input. The output is
the predicted score for each action according to the Bellman
equation:
\[ y_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta_i). \] (4)

We choose N Double DQN networks for ensemble to learn
the optimal strategy. In the double DQN algorithm [van
Hasselt et al., 2016], the calculation of the Q value is no
longer the calculation of the maximum Q value of each ac-
tion through a separate network as, but choose the action cor-
responding to the maximum Q value is first found in the pre-
diction network \( Q(s_t; \theta) \):
\[ a_{m}^{max} = \arg \max_a Q_m(s_t+1, a; \theta_i). \] (5)

Then use the selected action \( a_{m}^{max} \) to calculate the target Q
value in the target network \( Q' \), and choose the min Q value
among the M sub-Q networks:
\[ y_t = r_t + \gamma \min_{m \in M} Q'_m(s_{t+1}, a_{m}^{max}; \theta_i''). \] (6)

Then we use the expand \( Y_t \) to update all N networks. The tar-
get network revalues the parameter from the prediction net-
work after each update through soft update [Lillicrap et al.,
2016].

4.4 update-to-data ratio

The model-based method uses the UTD ratio, which is the
ratio of the amount of data the agent uses for learning to the
amount of data that the agent actually interacts with the en-
vironment, which greatly improves the sampling efficiency.
This is because the model-based method can perform data
simulation through the model, and mix the simulated data
with the real data to improve the sampling efficiency, and the
UTD ratio can reach up to 40.

However, the previous research on UTD ratio mainly fo-
cused on the model-based methods, and the UTD ratio based
on the model-free methods is less researched. In our model
learning process, we take multiple samples cyclically from
memory to enhance data utilization. Moreover, we have
proved that the UTD ratio can effectively improve data uti-
lization and enhance the learning effect through synthetic data
and real-world data in the experiments.

4.5 Algorithm

The process of online interaction between the agent and the
intersection, the update process of the traffic signal control
strategy is shown in Algorithm 1. \( G \) is the Update-To-Data
ratio. We use the soft update method to update the networks.

Algorithm 1 Random Ensemble Double DQN Light

1: Initializes N Q-function parameters \( \theta_i \), i=1,2,...,N, empty
   replay buffer \( D \). Set target parameters \( \theta_i' \), i=1,2,...,N
2: while Agent is interacting with the Environment do
3:     Choose \( a_t \in A \) based on \( Q_i(s_t; a_t; \theta_i) \)
4:     Vote for the performed action \( a_t \) in set \( \{ a_t \}_i \)
5:     Collect state \( s_{t+1} \) and reward \( r_t \) from the environment
6:     Store transitions \( (s_t, a_t, r_t, s_{t+1}) \) in replay buffer \( D \)
7:     for \( G \) updates do
8:         Sample a mini-batch \( B = \{(s_t, a_t, r_t, s_{t+1})\} \) from
         replay buffer \( D \)
9:         Randomly sample \( m \) different numbers from \( \{1, 2, ..., N\} \)
         as the index of chosen Q-function.
10:    Calculate target \( Y_t \) in the selected \( Q_m, m \in M : \)
       \[ y_t = r_t + \gamma \min_{m \in M} Q'_m(s_{t+1}, a_{m}^{max}; \theta_i''). \]
11:    for i = 1, 2,..., N do
12:        Update \( \theta_i \) with gradient descent using
       \[ \nabla_{\theta} \frac{1}{|B|} \sum_{(s_t, a_t, r_t, s_{t+1}) \in B} (Q_i(s, a; \theta_i) - y_t)^2 \]
13:    end for
14:    end for
15:    Update target networks with \( \theta'_i \leftarrow \rho \theta' + (1-\rho) \theta_i \)
16:    end while

Figure 2: RElight framework.

The traffic signal control process in the real world is very
complicated. We will use Example 1 to explain it.

Example 1. We assume there is an intersection with two
phases, West-East direction is green (WEG) and North-South
direction is green (NSG). We take the current intersection
state and traffic signal phase as input. Each DDQN learns
different strategies, and even if the observed state is the same,
its possible to score different results for each action. There-
fore, each DDQN chooses to switch the traffic signal state or
maintain the current signal phase according to its strategy.
Then we vote in these actions to decide whether to change the
signal phase.
5 EXPERIMENT

In this part, we conducted experiments using synthetic data and real-world data. After that, we showed the results of comparing our method with other methods and some case studies.

5.1 Settings

Following the tradition of traffic signal control research, the simulation platform we used in the experiment is SUMO. We obtain the state of the traffic environment according to the simulator as the input of the agent, and then pass the action of the agent to the simulator through the traffic light API to control the phase of the traffic light, and finally get feedback rewards from the simulator. According to general rules, there should be a three-second yellow light between the green light and the red light, so that vehicles that have entered the intersection can safely leave the intersection.

The parameter setting in the experiment and the weight coefficient of the reward function are listed in Table 2 and Table 3 respectively.

| Model Parameter                  | Value       |
|----------------------------------|-------------|
| Action time interval             | 5 seconds   |
| $\gamma$ for future reward       | 0.8         |
| $\epsilon$ for exploration       | 0.05        |
| batch size                       | 20          |
| memory length                    | 1000        |
| learning rate                    | 0.01        |
| number of Q-function N           | 10          |
| number of Q-function in-target M | 4           |
| UTD ratio                        | 40          |
| model soft upadate polyak        | 0.995       |

Table 2: Parameter Settings

| $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ |
|-------|-------|-------|-------|-------|-------|
| -0.25 | -0.25 | -0.25 | -5    | 1     | 1     |

Table 3: Weight Coefficient of the Reward Function

5.2 Datasets

Synthetic data

The experimental environment for synthetic data is a four-way intersection. The road length in the four directions of this intersection is 150 meters, and there are six lanes in each direction, three entry lanes and three exit lanes. There are two phases of traffic lights: WE-Green for Through and Left(WEG), NS-Green for Through and Left(NSG). The specific data characteristics are shown in Table 4.

Real-world data

The real-world dataset is collected by surveillance cameras in Hangzhou. The data collect the traffic situation of each direction in every second of a four-way intersection within an hour. By analyzing the data recorded by the camera, we obtained the trajectory data when the vehicle passed the intersection, and put it as traffic flow data into SUMO for experiment. The details of the data set are listed in the table 5.

| Config     | Begin (s) | End (s) | Number (cars) | Condition |
|------------|-----------|---------|---------------|-----------|
| switch     | 0         | 36000   | 14400         | WEG       |
|            | 36001     | 72000   | 14400         | NSG       |
| equal      | 0         | 72000   | 2400          | WEG       |
|            | 0         | 72000   | 2400          | NSG       |
| unequal    | 0         | 72000   | 14400         | WEG       |
|            | 0         | 72000   | 2400          | NSG       |
| synthetic  | 0         | 72000   | 14400         | switch    |
|            | 72001     | 14400   | 216000        | equal     |
|            | 144001    | 216000  |               | unequal   |

Table 4: Configurations for synthetic traffic data. It mainly includes four traffic flow conditions: (a) switch direction traffic flow, (b) equal traffic flow, (c) unequal traffic flow, and (d) complex traffic flow, which is a collection of the above situations.

| Dataset    | time(s) | Arrival rate(cars/s) |
|------------|---------|----------------------|
| Hangzhou   | 3600    | 0.514                |

Table 5: Details for real-world traffic data

5.3 Performance Comparison

Compared Methods

In order to measure the effect of our model, we used two types of methods to compare the experimental results: traditional transportation methods and RL methods. Traditional methods include Fixed-cycle control method[Miller, 1963] and Self-Organizing Traffic Light (SOTL) control method[Gershenson, 2005]. RL methods include DRL[Van der Pol and Oliehoek, 2016] method and IntelliLight[Wei et al., 2018] method.

Evaluation Metric

We follow the common metrics of existing research to compare the results of different methods: The average queue length at the intersection (Queue length). The average delay at the intersection (Delay). The average travel time of a vehicle through an intersection (Travel time).

Comparison with the performance of mentioned earlier methods on synthetic data.

We first use four baselines and our proposed method to conduct comparative experiments on four synthetic datasets. It can be seen from the following four tables that our proposed method performs better than the baseline in the four synthetic situations, especially on the metric of travel time. In configuration 1, the SOTL method is similar to our results in the first two metrics, and in config unequal, the IntelliLight method has a similar situation. But our method shows more obvious advantages in more situations, which shows that our method has the ability to deal with more complex and changeable situations and the generalization ability to different situations. RELight learned a better traffic signal control strategy through random ensemble and improvement of data utilization.
Comparison with the performance of mentioned earlier methods on real-world data. On the real-world dataset, we also compared our proposed method with the baseline method through experiments. According to Table 7, it can be clearly seen that our proposed method has obvious improvement in queue length and travel time. The ensemble learning and multiple updating of RELight effectively avoid overestimation of the Q value and effectively reduce excessive actions. It prevents the traffic flow in one direction at the intersection to make it wait for a long time. Therefore, it can avoid falling into the local optimum and learn a better global optimization strategy.

5.4 Study of RELight

UTD When the value of the UTD ratio is different, the learning effect is also different. We have carried out comparative experiments on different values of UTD parameters when the other parameters are optimal (N=M=10). Taking the synthetic data config unequal as the traffic condition, and comparing the UTD values with different values, the following figure is obtained. The introduction of UTD effectively improves the stability of the learning process. When UTD=40, the agent can learn the optimal strategy.

N Q-function and M choosed Q-function In this part, we conduct comparative experiments on different combinations of N and M, select the appropriate size of N and the optimal number of subsets M. By selecting the M parameter, we can effectively improve the convergence speed, accelerate the convergence process, and does not affect the learning effect. At the same time, we keep UTD set to 40. The data set used is also config unequal. According to the figure 3, it is obvious that when we ensemble 10 single DDQN (N=10), the effect of the strategy learned by the agent is much better than that of using only a single DDQN. We can see that when N=10, M takes 4 to get a shorter queue length. This shows that in the ensemble Q functions, we randomly select a subset of the Q function for optimization and learn a better strategy.

Table 6: Performance on synthetic data

| Data   | Model name | Queue length | Delay | Travel time |
|--------|------------|--------------|-------|-------------|
| switch | Fixed-cycle| 8.532        | 2.479 | 42.230      |
|        | SOTL       | 0.006        | 1.598 | 24.129      |
|        | DRL        | 91.412       | 4.483 | 277.430     |
|        | IntelliLight| 8.125       | 1.883 | 56.109      |
|        | RELight    | 0.076        | 1.619 | 3.841       |
| equal  | Fixed-cycle| 1.105        | 2.614 | 34.278      |
|        | SOTL       | 19.874       | 4.384 | 177.747     |
|        | DRL        | 3.405        | 3.431 | 52.075      |
|        | IntelliLight| 0.261       | 1.760 | 0.868       |
|        | RELight    | 0.223        | 1.733 | 0.835       |
| unequal| Fixed-cycle| 4.159        | 3.551 | 36.893      |
|        | SOTL       | 20.227       | 5.277 | 69.838      |
|        | DRL        | 16.968       | 4.704 | 66.485      |
|        | IntelliLight| 0.865       | 2.676 | 3.133       |
|        | RELight    | 0.896        | 3.111 | 3.016       |
| synthetic| Fixed-cycle| 4.601       | 2.883 | 39.707      |
|        | SOTL       | 13.372       | 3.753 | 54.014      |
|        | DRL        | 91.887       | 4.917 | 469.417     |
|        | IntelliLight| 3.062       | 2.177 | 18.782      |
|        | RELight    | 0.485        | 2.161 | 2.728       |

Table 7: Performance on real-world data

| Model name | Queue length | Delay | Travel time |
|------------|--------------|-------|-------------|
| Fixed-cycle| 19.542       | 3.377 | 84.513      |
| SOTL       | 16.603       | 4.070 | 64.833      |
| DRL        | 54.148       | 4.209 | 166.861     |
| IntelliLight| 8.425       | 4.44  | 20.920      |
| RELight    | 2.733        | 3.610 | 5.837       |

6 CONCLUSION

In this article, we use the ensemble reinforcement learning method to solve the traffic signal control problem. During the entire experiment and testing process, no prior knowledge and pre-training is required. This method can learn the optimal strategy only through complete online learning, and a large number of experiments have proved the superiority of our method. We conducted extensive experiments on a variety of synthetic traffic data and real traffic data and proved that the superiority of our proposed method exceeds the current best method. In addition, we optimized the RELight algorithm by comparing UTD, M, N, and other parameters to select the optimal parameters for traffic signal control.
References

[Chen et al., 2021] Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021.

[El-Tantawy and Abdulhai, 2012] Samah El-Tantawy and Baher Abdulhai. Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC). In 15th International IEEE Conference on Intelligent Transportation Systems, ITSC 2012, Anchorage, AK, USA, September 16-19, 2012, pages 319–326. IEEE, 2012.

[Fellendorf, 1994] Martin Fellendorf. Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority. In Ite Meeting, 1994.

[Gershenson, 2005] Carlos Gershenson. Self-organizing traffic lights. Complex Syst., 16(1), 2005.

[Haddad et al., 2010] Jack Haddad, Bart De Schutter, David Mahalel, Ilya Isoslovi, and Per Olof Gutman. Optimal steady-state control for isolated traffic intersections. IEEE Trans. Autom. Control., 55(11):2612–2617, 2010.

[Lillicrap et al., 2016] Timothy P. Lillicrap, Jonathan H. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[Miller, 1963] Alan J. Miller. Settings for fixed-cycle traffic signals. OR, 14(4):373–386, 1963.

[Mirchandani and Head, 2001] Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. Transportation Research Part C: Emerging Technologies, 9(6):415–432, 2001.

[Nishi et al., 2018] Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. Traffic signal control based on reinforcement learning with graph convolutional neural nets. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 877–883, 2018.

[Roess et al., 2011] R. P. Roess, E. S. Prassas, and W. R. McShane. Traffic engineering. Traffic engineering, 2011.

[Robertson, 1969] D. I. Robertson. Transyt: A traffic network study tool. Road Research Laboratory Report, 1969.

[Silcock, 1997] J.P. Silcock. Designing signal-controlled junctions for group-based operation. Transportation Research Part A: Policy and Practice, 31(2):157–173, 1997.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. IEEE Trans. Neural Networks, 9(5):1054–1054, 1998.

[Touretzky et al., 1997] Edited Touretzky, Peter Sollich, and Anders Krogh. Learning with ensembles: How over-fitting can be useful. Adv. Neural Inform. Processing Sys., 8, 03 1997.

[Van der Pol and Oliehoek, 2016] Elise Van der Pol and Frans A. Oliehoek. Coordinated deep reinforcement learners for traffic light control. In NIPS’16 Workshop on Learning, Inference and Control of Multi-Agent Systems, December 2016.