Ensemble Reinforcement Learning for Large-Scale Traffic Signal Control

Zhenzhe Ying
443407384@qq.com
HangZhou, ZheJiang

Zhuoer Xu
xuzhuoer.rex@gmail.com
HangZhou, ZheJiang

Hui Li
lihuiknight@gmail.com
HangZhou, ZheJiang

Haotian Wang
1031821435@qq.com
HangZhou, ZheJiang

Shiwen Cui
278044960@qq.com
HangZhou, ZheJiang

ABSTRACT

Traffic signal control is an important problem that affects people's daily life in commuting. However, it has been quite a challenge to design effective policy to control traffic signal for a large number of interactions in the road network. In this paper, we formulate the problem as a Markov Decision Process (MDP). Specifically, we model each interaction with a valid traffic light as an agent and all the interactions (agents) need to make a *joint policy* coordinately so that the traffic system can serve more vehicles with less *delay*. However, learning the joint policy for the large-scale traffic signal control is intractable because its action space is exponentially large over the number of intersections. To address this issue, we employ deep reinforcement learning to approximate the *joint policy* for the traffic signal control based on a carefully-designed MDP, where the reward is a self-defined *delay* index and the state is designed based on some heuristic methods. To make our policy generalize well on the unknown road networks, we learn multiple policies on some randomly generated road networks and archive the final policy via ensemble learning. We conduct our method on multiple partially observed traffic flows as well as a real-world road network: 1004 traffic lights in Nanchang, one of the largest cities in China. The experimental results show that our policy has a strong generalization and won the second place from 1156 teams [1].

ACM Reference Format:
Zhenzhe Ying, Zhuoer Xu, Hui Li, Haotian Wang, and Shiwen Cui. 2021. Ensemble Reinforcement Learning for Large-Scale Traffic Signal Control. In *City Brain Challenge (KDD2021), July, 2021, Online.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

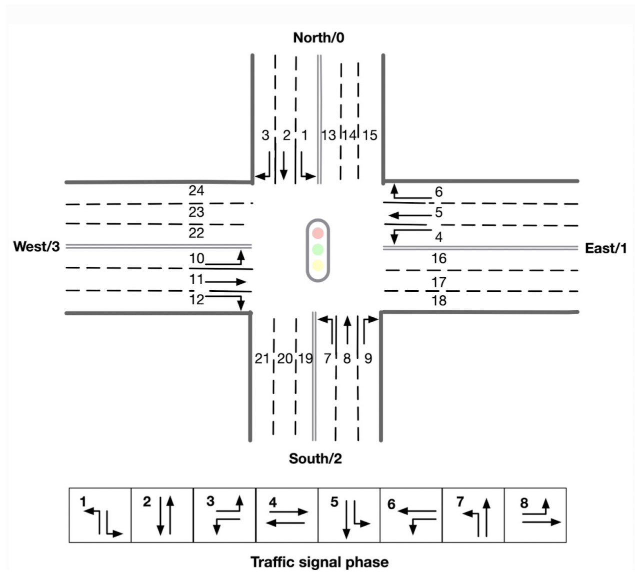

Figure 1: Illustration of a a four-leg intersection (agent) and its traffic signal phase.

1 INTRODUCTION

Traffic congestion is one of the biggest issues in the city traffic, however, it's unclear the substantive cause. Is it because that the number of vehicles has exceeded the capacity of the city or that we fail to utilize the road network at its maximum capacity? For example, Tokyo and New York City rank similarly by traffic congestion index. However, Tokyo has 43% more registered vehicles than New York City while Tokyo only has 15% more signalized intersections and 32% more road length than New York City [4]. Why is Tokyo able to serve more vehicles than New York City? Is New York City operating the traffic at its maximum capacity? To answer these questions, we are invited to design policy to coordinate the traffic signals and find the maximum number of vehicles that can be served with a city-scale road network in City Brain challenge in KDD CUP 2021.

We organize the paper as following. We introduce the problem of traffic signal control in section 2, then formulate the problem as a Markov Decision Process (MDP) and present the proposed method in section 3. We show the experimental results in section 4 and give the conclusion and future work in section 5.

2 PROBLEM AND CHALLENGES

Problems. Figure 1 presents a four-leg intersection. In each period of time step, only one of eight types of signal phases can be selected

[1]The details about the City Brain challenge (KDD CUP 2021) can be found from http://www.yunqiacademy.org/home/leaderboard.

(KDD2021), July, 2021, Online

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and a corresponding pair of non-conflict traffic movements will be served. For example, phase-1 gives right-of-way for left-turn traffic from northern and southern approaches. In the real-world road network, there are typically a large number of intersections. The intelligent traffic signal system needs to make a decision from a learned or specific joint policy to coordinate the movements of vehicles at each intersection. We are required to design a joint policy for the traffic system to coordinate the traffic signals and make the traffic system serve more vehicles while maintaining a lower acceptable delay.

Evaluation Metrics. Total number of vehicles served refers to total number of vehicles entering the network. The trip delay index is computed as actual travel time divided by travel time at free-flow speed. For an uncompleted trip, the free-flow speed is used to estimate the travel time of rest of the trip. The delay index is computed as average trip delay index over all vehicles served, *i.e.*, $D = \frac{1}{N}\sum_{i=1}^{M} D_i$, where M refers to the number of vehicles. D_i refers to the delay of i-th vehicle, which is defined by

$$D_i = \frac{TT_i + TT_i^r}{TT_i^f} \tag{1}$$

where TT_i refers to travel time of vehicle, TT_i^r refers to the rest of trip travel time which is estimated with free-flow speed, and TT_i^f refers to full trip travel time at free-flow speed.

Challenges. Many researchers have addressed this problem via optimization techniques, such as reinforcement learning (RL) [5, 8, 12], however, this problem is still far from solved because of the following challenges:

- There are lack of efficient reinforcement learning environments which can simulate the movements for thousands and thousands of vehicles in the large-scale road network [10]. For example, the environment provided by City Brain KDD CUP 2021 is not that efficient. Specifically, the provided environment costs about 3 minutes for one episode (about 300 steps) when running on a standard 8-cores machine. It's difficult for RL agent to play millions of episodes, which is typically necessary for a RL agent to learn its policy well from scratch [9].
- The action space is exponentially large over the large number of valid intersections (agents). For example, each agent has 8 actions. The road map with 1000 intersections will have 8^{1000} joint actions, which makes it intractable to learn the optimal policy within the limit time and computation resources.
- Reinforcement learning typically leads to an overfitting policy on the specific environment. However, our policy is evaluated on the unknown environment with the limited times of submissions. It requires us to learn a well-generalized policy.

In the next section, we provide a practical approach to handle above challenges.

3 METHOD

In this section, we formulate the traffic signal control as a Markov Decision Process (MDP) and employ an ensemble reinforcement learning to learn the approximate joint policy for large-scale traffic signals.

3.1 Environment

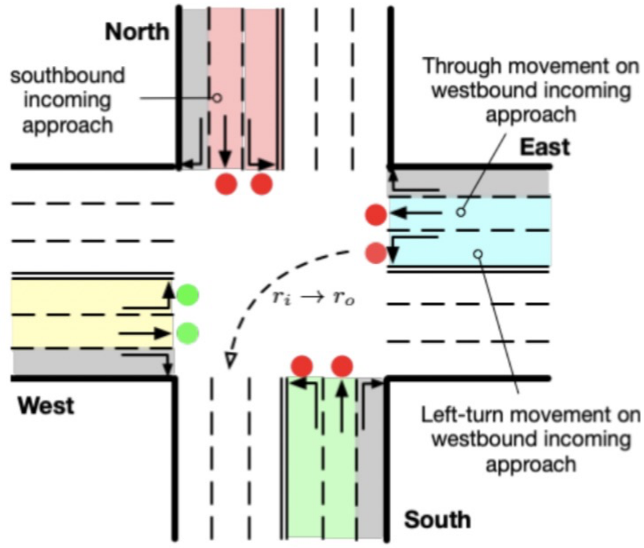

Figure 2: Traffic movement for the specific signal phase.

The environment simulates the vehicle movements and provides the observation for all vehicles, roads and N intersections (agents). A road consists of a set of lanes. There are two kinds of lanes: incoming lanes and outgoing lanes (also known as approaching/entering lane and receiving/exiting lane). There are some important parameters including speed limit, length, latitude, longitude and *etc.* to represent each lane. An intersection may have a traffic signal or not. Our target intersections are those which have signals. A movement signal is defined on the traffic movement, where *green* denotes the corresponding movement is allowed and *red* denotes the movement is prohibited. For a traffic signal, there are at most 8 phases ($1-8$). Each phase allows a pair of non-conflict traffic movements to pass this intersection. Figure 2 illustrates one of the traffic movements for the specific signal phase. The *action* is defined as the traffic signal phase for each intersection to be selected in the next 10 seconds. If an agent is switched to a different phase, there will be a 5 seconds period of *all-red* at the beginning of the next phase, which means all vehicles could not pass this intersection.

After analyzing the environment carefully, we found there are five challenges for this task:

- The traffic flow exceeds the load of road net. It is impossible for the traffic system to deliver all vehicles. So we have to design intelligent policy to deliver the high-value vehicles first.
- There is uncertainty in the environment. The same policy could generate two different delay values (about 0 to 0.001 difference).
- The environment is not efficient to play millions of episodes. We need to develop multi-processing tasks within the limited memory and computation resources.
- It's difficult to design a suitable observation (state) because each intersection may have 0 to 1000 vehicles under different

states. Also, the designed state should represent not only vehicles in the current intersection but also its neighbor intersections.

- Each intersection has 8 actions, N intersections in total 8^N actions. The action space of joint policy will be exponentially large over the number of intersections. It is intractable to develop hand-coded solutions and compute the optimal solutions [2].

3.2 Formulation

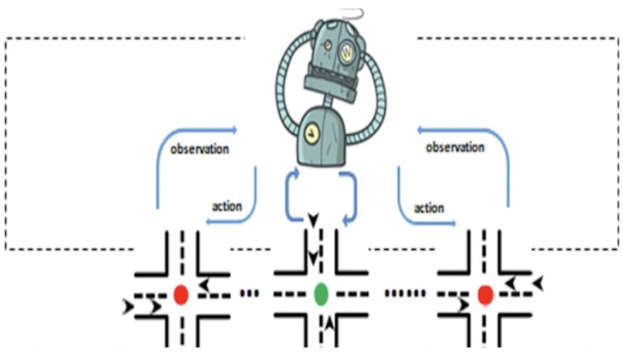

Figure 3: Learning reinforcement learning policy from the formulated MDP.

The environment simulates the vehicle movements and provides an initial state $\mathbf{s}^0 \in \{\mathcal{S}_i\}_{i=1}^N$ for all vehicles, roads and N intersections (agents) as illustrated by figure 3. After observing t-th state $\mathbf{s}^t$, all the agents take actions $\mathbf{a}^t \in \{\mathcal{A}_i\}_{i=1}^N$ according to the specific policy $\pi(\mathbf{a}^t|\mathbf{s}^t)$ and received their rewards $\mathbf{r}^t \in \{r_i\}_{i=1}^N$. The environment transforms its state to $\mathbf{s}^{t+1} \in \{\mathcal{S}_i\}_{i=1}^N$. Therefore, the whole process will be a standard Markov Decision Process (MDP), which is characterized by a tuple $(\{\mathcal{S}_i\}_{i=1}^N, \{\mathcal{A}_i\}_{i=1}^N, \{r_i\}_{i=1}^N, \gamma)$, where N is the number of agents, S_i is the local state of the agent i and $\mathcal{A}_i$ denotes the set of action available to agent i, γ denotes the decay factor. The goal is to learn a joint policy π that maximizes the following optimization problem

$$\eta(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \sum_{i=1}^{N} r_i^t] \tag{2}$$

To make the above optimization problem tractable, we need to define the two important factors in the MDP: the state $\{\mathcal{S}_i\}_{i=1}^N$ and the reward $\{r_i\}_{i=1}^N$, which are important in our scenario. In this paper, we employ Deep Q-Network (DQN) [6] to learn the policy π via an end-to-end fashion. We introduce the details of the two factors in the next sections.

3.3 Feature engineering

We derived more than 200 features in our model. Because of the limitation of space, we only talk about some important features as following.

Vehicle information. (1) The number of vehicles that are closer or further to the intersections. (2) According to our observation, it's interesting to notice that most of the roads are longer than 200 units, such a clue indicates that the vehicle positions are very important. We segment the road by the time to the intersection with free-flow speed, which makes it easily for the agent to determine when to switch the other phase.

Road information. The length of roads. Lane will be busy easily if it is too short. When a total traffic jam happens in a lane, the other lane in the same road is not available and cars can not enter into the intersection. So we put the length of roads into features, and specific a bigger weight on cars in the shorter road.

Route prediction. The route of vehicles matters a lot in the competition while the metric depends on TT_i^f as defined in equation (1). However, we can not observe the whole route in the environment. we predict the whole route for each vehicle according to its history route. This trick reduces the delay performance by about 0.01.

3.4 Training

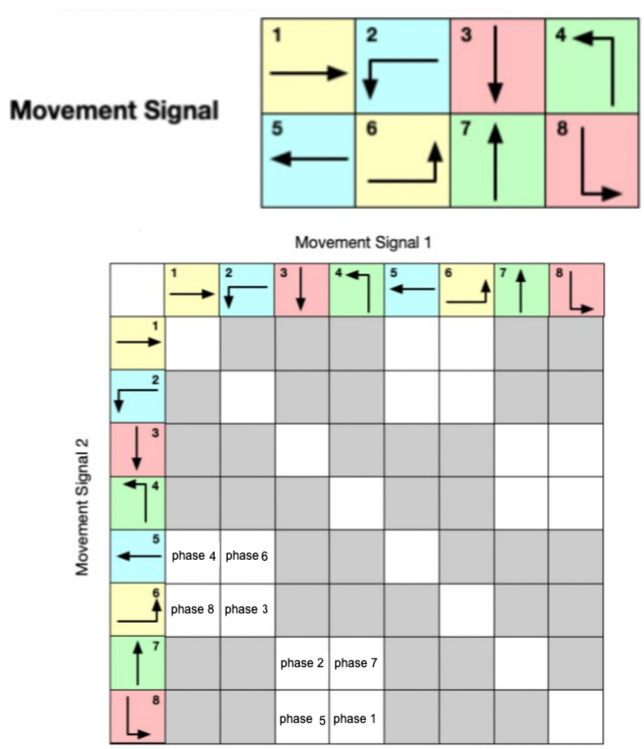

Figure 4: Signal and Phase.

Because the environment is not that efficient when the number of vehicles is larger than 10000, we develop a multi-processing learning task for DQN. Note that, DQN is an off-policy algorithm, which is suitable for the multi-processing task [7]. Specifically, in the competition, we train each model with 4 ∼ 8 process by 24 ∼ 48 hours.

Also, we design a new reward according to the evaluation metric in equation 1, which is defined by

$$D_i^t = \sum_{j=0}^{M} \left(\frac{TT_j + TT_j^r}{1000} \right) \tag{3}$$

$$r_i^t = \sum_{j=0}^{2} \left(\frac{D_i^{t+j}}{(j * 0.6 + 1)} \right) \tag{4}$$

where i refers to agent id, t refers to the step number in each episode.

Note that, in equation 4, we sum the rewards of sequential three steps to make the model benefit from long-term reward. Some other articles [3, 11] also provide the task-specific rewards, however, they doesn't meet the requirement in our scenario.

3.5 Ensemble

It's interesting to point out that the RL model trained on one specified traffic flow typically overfits the flow but performs poor on evaluated private flow in the leaderboard. To have diversity between the submissions, we use different features combination to train 4 models and 1 rule system to ensemble. Because the neural network has big uncertainty in regression task[1], we normalization the result of 4 neural networks and sum their predictions. And we develop a rule-based system by evaluating the value of action with different trials. The final action is ensemble by a ranking average of rule-based system score and neural networks' average score.

4 EXPERIMENT RESULTS

Since the leaderboard has changed a lot before the deadline, we do not have historical experimental data. Table 1 shows the top 10 team submissions.

Table 1: Experimental Results of different teams on Round 3. The larger of total served vehicles will be better. For the same total served vehicles, the lower delay will be better. GoodGoodStudy refers to our method.

Team Id	Total Served Vehicle	Delay
IntelligentLight	360637	1.4027135308052847
GoodGoodStudy	**358895**	**1.4052081109433103**
4PQC_team	358888	1.4025206513613906
SmartLight	357229	1.4027102867089065
BOE_IOT_AIBD	356463	1.4013649198707516
SUMO	335599	1.401792220782304
IF_BigData	335376	1.4020449312353032
alphabeta	318720	1.4034693515888321
bingo	317954	1.4010931499550239
D	316941	1.4021008708625378

Present the experimental results.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a practical reinforcement-learning method to handle the problem of multi-intersection traffic signal control, especially for large-scale networks. Our proposed method has shown its strong performance and generalization ability. In the future, one can use multi-agent reinforcement learning to capture the neighbors' information for each intersection to improve the performance.

REFERENCES

[1] Filipe Aires, Catherine Prigent, and William B Rossow. 2004. Neural network uncertainty assessment using Bayesian statistics: A remote sensing application. *Neural computation* 16, 11 (2004), 2415–2458.

[2] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans CA Groen. 2010. Traffic light control by multiagent reinforcement learning systems. In *Interactive Collaborative Information Systems*. Springer, 475–510.

[3] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3414–3421.

[4] KDD CUP. 2021. City Brain Challenge. http://www.yunqiacademy.org/poster

[5] Mengyu Guo, Pin Wang, Ching-Yao Chan, and Sid Askary. 2019. A reinforcement learning approach for intelligent traffic signal control at urban intersections. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 4242–4247.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[7] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G Bellemare. 2016. Safe and efficient off-policy reinforcement learning. *arXiv preprint arXiv:1606.02647* (2016).

[8] Chao Qu, Hui Li, Chang Liu, Junwu Xiong, James Zhang, Wei Chu, Weiqiang Wang, Yuan Qi, and Le Song. 2020. Intention propagation for multi-agent reinforcement learning. *arXiv preprint arXiv:2004.08883* (2020).

[9] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, and Julian Schrittwieser et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 7587 (2016).

[10] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. 2019. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8797–8806.

[11] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1290–1298.

[12] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1913–1922.