REINFORCEMENT LEARNING-BASED TRAFFIC SIGNAL CONTROL IN SPECIAL SCENARIO

Dingyi Zhuang
Department of Civil Engineering and Applied Mechanics
McGill University
Montreal, Quebec, H3A 0C3, Canada
Email: dingyi.zhuang@mail.mcgill.ca
ORCID: 0000-0003-3208-6016

Zhenyuan Ma
Department of Civil Engineering and Applied Mechanics
McGill University
Montreal, Quebec, H3A 0C3, Canada
Email: zhenyuan.ma@mail.mcgill.ca

Lijun Sun, Corresponding Author
Department of Civil Engineering and Applied Mechanics
McGill University
Montreal, Quebec, H3A 0C3, Canada
Email: lijun.sun@mcgill.ca
ORCID: 0000-0001-9488-0712

Word count: 5482 words text + 0 table(s) x 250 words (each) = 5482 words
Submission Date: August 1, 2020

ABSTRACT

Traffic signal control (TSC) is an important task to optimize the performance, both economic and environmental, of the transportation system. Recently reinforcement learning (RL) techniques have shown promising results in optimizing control policies for basic and standard intersections, which only receive traffic flow from connected arterials. However, the typical urban traffic networks are composed of freeways and arterial. The exchange flow between those two always causes congestion. This paper studies the effects of this type exchange flow on reinforcement learning-based TSC controllers. The first studied traffic network is built upon standard intersections. Then we build another network by adding an freeway off-ramp to the first network. Several up-to-date deep RL algorithms are used to learn optimal TSC signal for those two networks. Experimental results show the exchange flow between freeways and arterials greatly influence the learning rewards and safety. Positions and length settings of the ramp are also discussed, showing that RL results are sensitive to the infrastructure settings. We appeal for more attention on the practical traffic scenario discussion when conducting RL-based traffic control.

Keywords: Reinforcement learning, Traffic light control, Freeway off-ramp scenario

INTRODUCTION

The steadily increasing number of population and economic activities has led to the rise of urban congestion. Better solutions for TSC problem on the road will result in less traffic congestion and safer driving environment, which improves the efficiency and safety of the transportation system. Several advanced control methods have been widely used for reducing congestion. One promising strategy is the use of advanced RL-based TSC methods. In the RL context, the control problem will be modelled as an Markov Decision Process (MDP) with sequentially state-action-reward tuples. Observations change according to the selected action under a certain policy, with reward received to judge the policy. Back to transportation field, how to formulate the MDP framework is one of the most important tasks. Intuitively, we will consider the traffic lights as agents, and their phases as the states (1, 2). Whether the traffic lights change the phase and which phase they change to will be the action set. The objective is to find a policy that that maximize the expected return, e.g. the delay of vehicles.

However, current researches focus on improving the design of RL formulation and agent modeling, and they tend to apply models on larger road network to test the coordination performance. For instance, Wei et al. (1) consider a traffic road network with 196 intersections and design the cooperation algorithm among them. However, this large-scale road network only consists of arterial blocks and grids, where most of the intersections are simple ordinary 4-way intersections (i.e. crossroads) connected with typical arterial. This kind of road network is common in cities, but the real-world urban network is composed of freeways and arterial roads. The existing RL studies only consider the typical arterial roads, the coordination of traffic flow between the freeway and arterial are not fully studied. Lim et al. (3) have demonstrated that the existence of freeway off-ramp generates the spillover to the mainstream traffic flow, which decreases the freeway vehicle capacities and influence the TSC of the overall functions of urban freeways. Therefore, proper signal control in the neighboring intersection is more desired, which can save average delay between 15% and 40%, and average travel time up to 25% according to U.S. Federal Highway Administration (4).

Inspired by that, our goal of this study is to test RL models on a special traffic scenario in which the arterial network is connected with a freeway off-ramp. We will analyze the impact on RL model performance and road safety given changes in traffic flow or infrastructure influence (like the length of ramps) to discuss whether small changes in the environment setting will influence RL greatly. Our work is dedicated to examine whether RL models trained on simplified scenarios without discussion of infrastructures is reasonable. One important finding is that simply adding a freeway off-ramp will greatly influence the final rewards, which is much higher than increasing the same amount of inflow vehicles. We hope our work can appeal for more attention on the practical traffic scenario discussion rather than more fancy algorithm design on RL-based TSC.

The remainder of this paper is as follows. In section 2 we review some literature about classical signal control methods and reinforcement learning based ones. Section 3 describes our special traffic scenario design and how our RL model works. Section 4 shows all our experiments and results. Finally in section 5, we conclude our work.

LITERATURE REVIEW

Classical Methods

There exist a set of classical transportation methods for traffic signal control problems. They are based on the knowledge of signal planning and phase changing rules, which can be intuitively

understood and explained. Generally, these classical methods work as baselines comparing to RL-based methods. Most of the classical methods use equations related to groundtruth of physics, such as relationship between time, length and speed.

GreenWave (5) is one of the most classical methods focusing on undirectional traffic signal. It optimizes an offset for signals in each intersection by reducing the number of stops of vehicles when traveling in one certain direction. However, it requires a pre-defined signal plan applied to all intersections. A similar method named Maxband (6) has nearly the same setting with GreenWave. The difference occurs in the objective, as Maxband considers vehicles traveling along two opposite directions.

Actuated control is a more flexible way than static signal control, and is a more controllable way than RL-based signal control. There are rules for changing signal, by detecting whether there exists a sufficient time gap between current vehicle and its successive vehicle. A maximum and a minimum time limit is needed for phases, so that there will be balance for each direction in the intersection. This control method is commonly used in Germany.

Queue length of an intersection can be considered as "pressure" of the intersection. Max-pressure control is used to balance the pressure of each intersection and avoid over-saturation where the pressure is formally defined as the difference of queue length between incoming lanes and outgoing lanes (7).

Reinforcement Learning Methods

Because of more data resources and more computational power, RL-based methods have been applied widely in traffic signal control field. Since the traffic scenario is depicted via MDP, as discussed above, we will consider the traffic lights as agents, and their phases as the states (1, 2). Whether the traffic lights change the phase and which phase they change to will be the design of action. Usually, the reward of the algorithm is related to the total delay of vehicles.

However, there are much more possible settings of state, action and reward of RL problem. The length of waiting queue in front of traffic light (1), the level of congestion happened in a lane (8), waiting time of vehicles (9) could also be defined as the reward of agents. Although the goal of Reinforcement Learning problem is to maximize the expected return, according to different state setting, the reward function can be different. A reward function can also be a linear combination of several factors, so that we will know more about how the influence of factors to the traffic signals. From different perspectives, RL-based algorithms can be categorised differently. We can choose the most suitable algorithm by considering whether we need to learn the state transition funciton, whether we learn the policy parameter, and whether we learn by using a table for state-action pairs.

Game theory is a useful tool that can be applied into Reinforcement Learning methods, especially multi-agent RL model. By using the knowledge of game theory, signals or traffic lights are considered as players in a game, and they choose their best actions according to their payoffs (i.e. objectives in traffic signal control setting) (10). Finding an equilibrium in a game theory is similiar to the communication between agents in Reinforcement Learning and find the optimal solution. Moreover, different game theory models can be considered to fit different traffic environments and different settings of agents.

RL-based method can be also used together with a classical one. There is an example of doing so which is PressLight (11), using the idea of max-pressure (7). By this way, the knowledge in transportation field can be well used and make reinforcement learning methods more intuitive, as well as outperform both approaches.

Special Traffic Scenarios

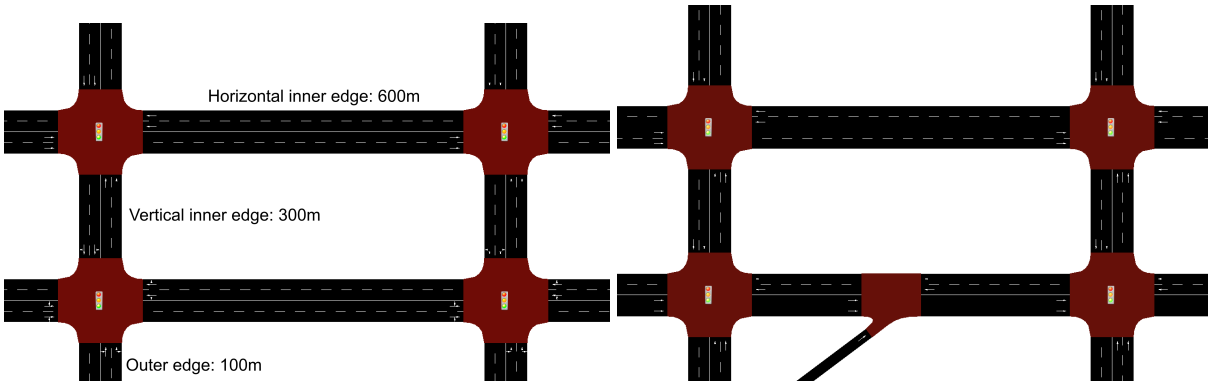
Comparing to the road network that only consists simple ordinary 4-way intersections (i.e. crossroads) (1), special traffic scenarios are less common appeared, both in our daily life and in research area. This means special traffic scenarios are worth to be researched into. T-junction is one of the simplest intersections, but it contains more possible phase design than common crossroad (12). A combination of two T-junctions with different directions is called LR-type staggered intersection (13). Congestions and even accidents tend to happen more in those non-regular traffic area.

Bottlenecks and intersections near freeways are another type of special traffic scenarios. Vehicles in both scenarios will face different width of roads before and after crossing an intersections. Moreover, due to different speed limit in different type of roads, traffic collisions appear more than other type of traffic scenarios. Existing research shows that if we apply traffic signal control, even basic control system, there will be a reduce of around 20% of collisions. (14)

Not only special traffic networks affect, some traffic events matter in traffic signal control. (15) and (16) mention there may be uncertain supply happened as traffic disruptions on links such as incidents, freeway crash and lane closure. These events can be treated as part of special traffic scenarios and will affect the control of signal and the traffic flow of the total traffic networks.

PROBLEM STATEMENT

Scenario Design



(a) A traffic grid with 4 regular intersections (b) A traffic grid with off-ramp from freeway

FIGURE 1 : Scenario design that (b) simply adds an off-ramp from freeway in the traffic grid of (a). If the traffic flow in the ramp is 0, these two environments are exactly the same for RL model.

We design a simple traffic grid that influenced by a freeway off-ramp shown in Figure 1(b). This grid is a generalization from regular traffic grid, which contains four regular crossroad intersections, as displayed in Figure 1(a). The scenario simulates how vehicles exiting from the freeway will merge the city freeway traffic flow. Original road setting is 300 meters with 4 lanes for every vertical inner edges, and 600 meters with 4 lanes for every horizontal inner edges. For outer edges we only show 100 meters for simplicity. For each regular intersection, we allow traffic inflow to come from two outer edges of the intersection with 25% probability to add new vehicle every timestep. Moreover, we can set random number of vehicles in the grid when initializing the scenario. If the vehicle flow in off-ramp is 0, meaning that no vehicles added in the ramp, then scenarios with or without the ramp are the same for RL models. In each regular intersection, we

set a traffic light with specified phases, without setting the time limit of phases. The traffic lights are what we treat as agents in our RL formulation. There is no traffic light in the area that ramp is connected with the grid in the very basic setting.

Some variables in this scenarios can be changed and then used to compare how the environment change will affect the traffic, which serve as a reference when government design their real traffic environment. Changeable variables include the inflow rate of vehicles in outer edges and ramp, the length of ramp, the position of ramp (i.e. how close it is to the right bottom intersection). Other variables such as the length of inner edges, the speed limit of vehicles or the number of lanes will also influence, but with smaller effect.

Environment Design

Working as a reinforcement learning problem, we need to specify the environment setting of this Markov Decision Process. Key elements in the environment includes agent, state, action and reward are designed as:

- **State**: Our state is defined as vehicle information for each intersection. The state component includes which intersection is selected, the speed of each vehicle, the distance between each vehicle and selected intersection and the edge where each vehicle is on. This state design is the same as the work of Wu et al. (17).

- **Action**: Because we design the action phase to make sure that vehicles can only pass the intersections horizontally or vertically, the action space **a** only consists of a list of float variables ranging from 0 to 1, specifying whether a traffic light is supposed to switch or not. The actions are sent to the traffic light in the grid from left to right and then top to bottom.

- **Reward**: Reward is calculated by the negative per vehicle delay minus a penalty for switching traffic lights.

- **Policy agent**: Since it is an environment with multiple agents (lights) and observations, we can consider both "single-agent" and "multi-agent" policy environment. Here we use the term "agent" again for policy trainer as it is defined in the RLlib (18). This agent definition is only used in algorithm discussion. If these is an super-agent controlling the observation of four traffic lights, we treat this environment as single-agent policy environment. Otherwise, in multi-agent environment, agents will model each light's own goal and action.

- **Termination**: An epoch is terminated if the expected timesteps are reached.

METHODOLOGY

Our research focuses more on the special transportation scenario and its comparison. Consequently, we will implement widely-applied methods from RLlib (18, 19). In this section, we would firstly introduce the vanilla policy gradient algorithm. Then we discuss the Asynchronous Actor-Critic Agents (A3C) and Proximal Policy Optimization (PPO) algorithms, which belongs to Policy Gradient method family but more efficient. These three algorithms are the basic models discussed in the experiment part.

Policy Gradient

The goal of reinforcement learning is to find an optimal strategy for agents to achieve the optimal rewards. As an important member, policy-gradient algorithms do not suffer from issues that value-based RL methods might have, such as complexity problem from the continuous states and actions (20). The basic idea of policy gradient is that, by sampling trajectories (sequences of state, action and reward tuples) from the environment from current policy function, we can gather the rewards with different trajectories where high-reward paths receive more weights.

Suppose we have a differentiable policy function $\pi_\theta(\mathbf{a}|\mathbf{s})$, where $\mathbf{a}$ is the action, $\mathbf{s}$ is the state, and θ represents the weight parameters to be learned. Suppose the reward function is $r(\mathbf{s},\mathbf{a})$. The optimal policy can be obtained by

$$\theta^* = \arg\max_\theta \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[r(\tau)] = \arg\max_\theta \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\sum_t r(\mathbf{s}_t,\mathbf{a}_t)]$$

where τ denotes the sampled trajectory of $(\mathbf{s}_t,\mathbf{a}_t)$ tuples from policy π_θ. The optimization process can be approximated by the gradient descent method. Since we are updating the weight parameters of policy π_θ, the method is referred as Policy Gradient. The updating rule can be approximated by (21):

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[(\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t))(\sum_t r(\mathbf{s}_t,\mathbf{a}_t)] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}[(\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}))(\sum_t r(\mathbf{s}_{i,t},\mathbf{a}_{i,t})]\end{aligned} \tag{1}$$

The vanilla Policy Gradient algorithm is summarized in Algorithm 1. Within each epoch, step 2 samples state-action-reward trajectories and step 3 and 4 calculate policy updating where $J(\theta)$ stands for the objective function we evaluate. The process can assign larger weights on high-reward samples to make them easier to be sampled. Thus, these well performed samples will further improve the policy updating.

Algorithm 1 Vanilla Policy Gradient

Initialize: A differentiable policy function$\pi_\theta(\mathbf{a}|\mathbf{s})$, learning rate α

1: **for** each epoch **do**
2: Sample trajectory $\mathbf{s}_{i,t},\mathbf{a}_{i,t},r(\mathbf{s}_{i,t},\mathbf{a}_{i,t}),\mathbf{s}_{i,t+1}$ from $\pi_\theta(\mathbf{s}|\mathbf{a})$ in the simulator
3: Calculate $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}[(\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}))(\sum_t r(\mathbf{s}_{i,t},\mathbf{a}_{i,t})]$
4: $\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$
5: **end for**

Actor-Critic Agents

Even though policy gradient methods enforce good sampling, the sampled rewards in practice usually have large variances, inducing large variance of the estimated gradients. Therefore, the vanilla policy gradient method is unstable and inefficient.

A3C add asynchronous mechanism where each agents interacts with its respective environments asynchronously, learning interaction with the others. While the core idea is still Actor-Critic,

which solves the variance problems by adding a value function, as "critic", to the vanilla policy gradient algorithm. It evaluates the value function of the current policy by observing the states and rewards feedback from the environment. The policy π_θ is the "actor" that generates the trajectory in Algorithm 1. Before updating the policy, the value function approximates the expected rewards using methods like neural network using the generated samples. The value function is used to assist in improving the policy's parameter θ updating by comparing the average value of the current state.

Algorithm 2 Actor-critic

Initialize: A differentiable policy function$\pi_\theta(\mathbf{a}|\mathbf{s})$, learning rate α

1: **for** each epoch **do**
2: Sample trajectory $\mathbf{s}_{i,t}, \mathbf{a}_{i,t}, r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}), \mathbf{s}_{i,t+1}$ from $\pi_\theta(\mathbf{s}|\mathbf{a})$ in the simulator
3: fit $\hat{V}_\phi^\pi(\mathbf{s})$ to sampled rewards
4: Evaluate the advantage as $\hat{A}^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) = r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1}) - \hat{V}_\phi^\pi(\mathbf{s}_{i,t})$
5: Calculate $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}[(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}))\hat{A}^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})]$
6: $\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$
7: **end for**

As shown in Algorithm 2, step 3 updates the value function using the generated states-actions-reward tuples and then used to calculate the advantage $\hat{A}^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$ in step 4. Advantage estimates the improvement of the corresponding action achieves compared to the average value function we have estimated. Thus, by multiplying the gradient of the policy function with the advantage in step 5, we can update the parameters so that high-reward actions will be selected more likely.

Proximal Policy Optimization

In practice, vanilla policy gradient is not scalable for large scale problems. To solve the problem of hard convergence in vanilla policy gradient, Schulman et. al propose PPO algorithm with adaptive Kullback–Leibler divergence (KL-divergence) (22). KL-divergence is defined as $\bar{D}_{KL}(P||Q) \approx -\sum_i P(i) \ln \frac{Q(i)}{P(i)}$, which penalizes the objective if the new policy P is different from the old one Q. Details can be found in Algorithm 3.

It can be found that the key is step 4 where KL-divergence constraints ensure that the new policy and the old one would not differ greatly. In this way, the convergence of the algorithm can be guaranteed.

EXPERIMENTAL RESULTS

For experiments, we customized scenario network and both single and multi-agent environments based on Berkeley Flow project (17). Simulation is conducted in SUMO (23) with 3000 timesteps in each epoch. Each case is run 5 times under different random seeds.

Reinforcement Learning Experiments

We have run three basic algorithms with RLlib package (19) for both single-agent and multi-agent policy environment including Asynchronous Actor-Critic Agents (A3C), Policy Gradient (PG), and Proximal Policy Optimization (PPO). As mentioned, here the term "single-agent" means there is a

Algorithm 3 PPO

Initialize: policy parameter θ_0, KL penalty β_0, target KL-divergence δ

for k=0,1,2,... **do**
  Sample trajectory $\mathbf{s}_{i,t}, \mathbf{a}_{i,t}, r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}), \mathbf{s}_{i,t+1}$ from $\pi_{\theta_k}(\mathbf{s}|\mathbf{a})$ in the simulator
  Evaluate the advantage as $\hat{A}^{\pi_{\theta_k}}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) = r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \hat{V}^{\pi}_{\phi}(\mathbf{s}_{i,t+1}) - \hat{V}^{\pi}_{\phi}(\mathbf{s}_{i,t})$
  Compute policy update $\theta_{k+1} = \arg\min_{\theta} J(\theta_k) - \beta_k \bar{D}_{KL}(\theta||\theta_k)$ by taking K steps of minibatch stochastic gradient descent.
  if $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \geq 1.5\delta$ **then**
    $\beta_{k+1} = 2\beta_k$
  else if $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \leq \delta/1.5$ **then**
    $\beta_{k+1} = \beta_k/2$
  end if
end for

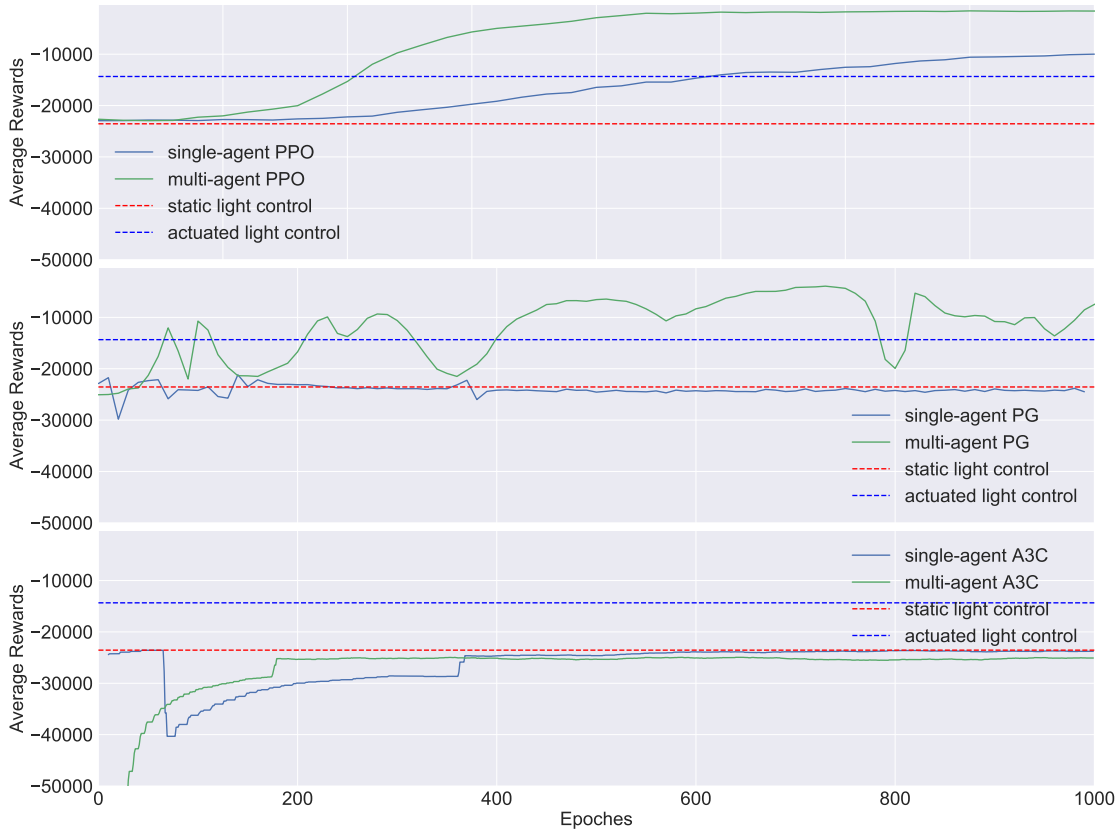


FIGURE 2 : Comparison of multi-agent and single-agent in three algorithms.

super agent controls the policy of 4 lights, instead of meaning we only only use one traffic light's observation. Contrarily, "multi-agent" means we train separately policies for different observations of the lights. Besides the RL control, There are two non-RL baseline methods comparing to three RL algorithms mentioned before. The first one is static light phase length. We set all green light phases with a length of 31 seconds and all yellow phases with 3 seconds. The second baseline

method is actuated traffic signals which is more commonly used in Germany. It will prolong traffic phases whenever a continuous stream of traffic is detected. and it switches to the next phase after detecting a sufficient time gap between successive vehicles.

Figure 2 show the mean rewards of PPO, PG and A3C algorithms respectively. From the result, we observe that A3C is not suitable for this traffic scenario. It even performs worse than static light control. And in A3C case, there is no much difference between single and multi-agent algorithms. Multi-agent PG works better than its single version, and it may be used in this traffic signal control problem, since it at least performs better than actuated signal control. But the problem is that it will not converge. Actually, PPO works quite well, especially in multi-agent environment. After convergence, multi-agent PPO performs 10 times better than the single one.

Because our task is to discuss the influence from infrastructural settings and the traffic flow (i.e. the independent variables), we only need to pick up the best performance algorithm as the base model. It can be found that multi-agent PPO suits this problem most, thus for the following infrastructure experiments, we keep on using this model setting as default, and evaluate on how infrastructure settings like ramp length and position affect.

Infrastructure Experiments

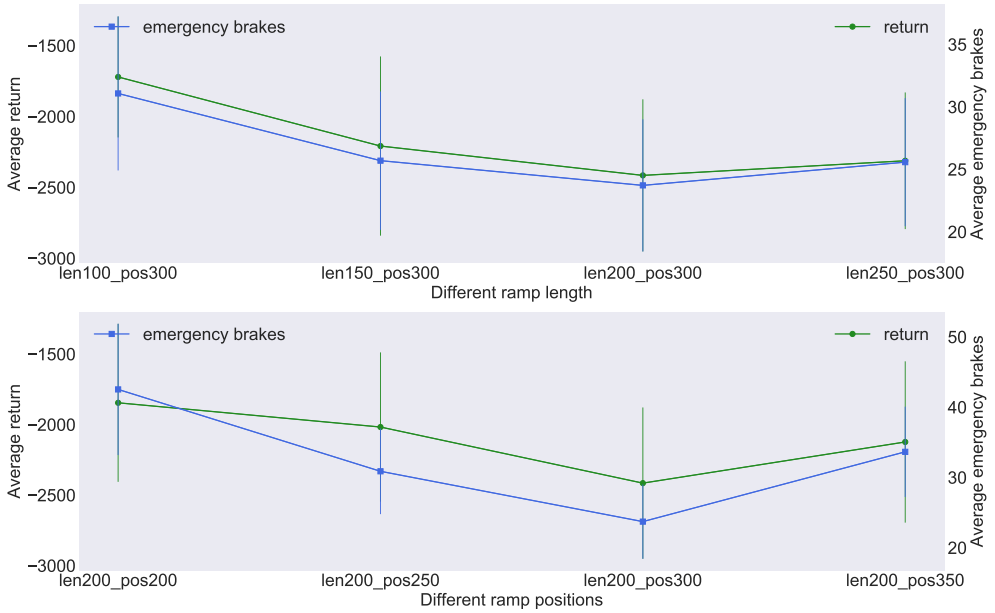
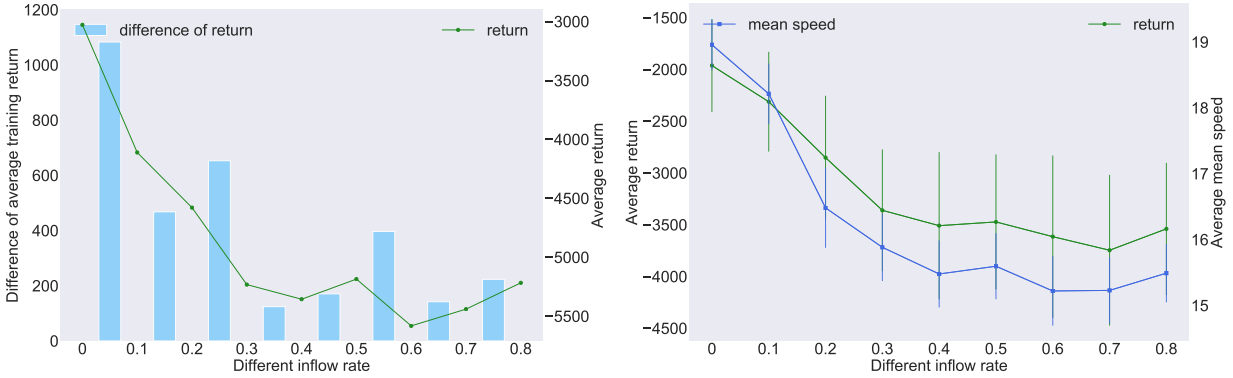


FIGURE 3 : Average return and number of emergency brakes under different ramp infrastructure settings

We then run experiments with different network settings to discuss how infrastructure of special scenarios will influence the RL control. The length of the ramp varies from 100 meters to 250 meters, and the position of the ramp to the right bottom intersection varies from 200 meters to 350 meters, both with 50 meters as intervals. In addition to the returns, we would like to apply reinforcement learning on traffic signal control to bring efficiency along with safety. Herein, we use the number of emergency brakes happened in each epoch as safety index. Figure 3 shows the results of average return and the number of emergency brakes under different infrastructure

settings after 100 epochs training with 5 different random seeds.

We observe that shortest ramp length gives best training reward, while there may not be large difference when ramp is too long (e.g. more than 200 meters). Slightly closer to the one of the intersections will have positive impact on traffic efficiency. Because the vehicles from ramp will turn right, the return of ramp closer to the left intersection is better than the return of the one closer to the right intersection. This result can be used as a simulation reference for infrastructure building. For safety issue, we find out that the average return and the number of emergency brakes have similar trend. However, lager the return is better, while less the emergency brakes is better. The setting that gives best training reward, which is ramp with length 100 and position 300, does not guarantee the safest control. The reason could be that balanced infrastructure setting, such as position 300 which denotes the ramp in the middle of the inner edge, will influence more on the flow by separating the road equally, but will make less emergency brakes also by separating the road equally.



(a) Average training return of difference inflow rates and their absolute value difference

(b) Average return and average mean speed of difference inflow rates

FIGURE 4 : Discussion of the impacts of inflow rate on training rewards.

We still need to answer why we use special scenarios as our research targets. The experiments on different inflow rates of ramp edge reflect our main motivation working on special traffic scenarios. From figure 4(a) and 4(b), we can clearly find that there is a large gap between inflow probability from 0 to 0.1. If we set the probability as 0, it means there is no inflow from ramp, thus this scenario becomes an ordinary traffic grid with only four crossroads. Once we add 0.1 probability of inflow to the ramp, the rewards will decrease a lot, comparing to other 0.1 probability change (e.g. from 0.1 to 0.2). When the probability of inflow is more than 0.4, there is no much difference between the return, as well as the average mean speed, also seen from Figure 4(b). We have run several simulations and then noticed that when the inflow probability rate is greater than 0.4, there will be congestion on the ramp. In this case, the traffic on freeway that generate the ramp will be affected more than the traffic grid that is connected to the ramp. Thus, the return we get that reflect the traffic situation for only grid area will not follow a decrease trend and change a lot when the inflow probability is greater than 0.4.

CONCLUSIONS

In this project, we manage to build a special traffic scenario to see how special scenario like adding an off freeway ramp to regular intersections will affect RL training performance from both training

returns and safety. By applying multi-agent PPO we have found that adding a freeway off-ramp will greatly influence the model performance, where difference is much higher than increasing the same amount of inflow rate. Also, we further explore the infrastructure influence, and run experiments on different ramp length and position combination. We find that shorter ramp length will promote the training reward while ramp position in the middle of the road will undermine the training performance. Therefore, the infrastructure designs have unignorable impacts in RL-based TSC problem. Thus more attention should be paid upon the infrastructure design methods for RL-based signal controllers.

Several interesting questions stem from our paper, which we plan to study in the future. For safety issue, we would like to add safety index directly into reinforcement learning reward, instead of just consider the number of emergency brakes. More special scenarios should be built and analyzed. We expect to find the most compatible algorithm and infrastructure for each special scenario. Autonomous vehicle is another direction. How to treat vehicles and traffic lights as agents together in one reinforcement learning control system is a challenge.

ACKNOWLEDGEMENT

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Fonds de recherche du Quebec - Nature et technologies (FRQNT), and the Canada Foundation for Innovation (CFI). The authors declare no competing financial interests with respect to this work.

AUTHOR CONTRIBUTIONS

D.Z. and Z.M. designed and performed the research; D.Z., Z.M. and L.S. wrote the paper.

REFERENCES

[1] Wei, H., N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, CoLight: Learning Network-Level Cooperation for Traffic Signal Control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Association for Computing Machinery, New York, NY, USA, 2019, CIKM '19, p. 1913–1922.

[2] Wei, H., G. Zheng, H. Yao, and Z. Li, Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2496–2505.

[3] Lim, K., J. H. Kim, E. Shin, and D.-G. Kim, A signal control model integrating arterial intersections and freeway off-ramps. *KSCE Journal of Civil Engineering*, Vol. 15, No. 2, 2011, pp. 385–394.

[4] Federal Highway Administration, U. D. o. T., *Ramp Management and Control Handbook*, 2006.

[5] Roess, R., E. Prassas, and W. McShane, *Traffic Engineering*. Traffic Engineering, Pearson/Prentice Hall, 2004.

[6] Little, J. D. C., M. D. Kelson, and N. H. Gartner, MAXBAND : a versatile program for setting signals on arteries and triangular networks, 1981.

[7] Varaiya, P., The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. *Advances in Dynamic Network Modeling in Complex Transportation Systems*, 2013, pp. 27–66.

[8] Bakker, B., S. Whiteson, L. Kester, and F. Groen, *Traffic Light Control by Multiagent Reinforcement Learning Systems*, Vol. 281, pp. 475–510, 2010.

[9] Wei, H., G. Zheng, H. Yao, and Z. Li, IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control, 2018, pp. 2496–2505.

[10] Elhenawy, M., A. A. Elbery, A. A. Hassan, and H. A. Rakha, An Intersection Game-Theory-Based Traffic Control Algorithm in a Connected Vehicle Environment. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 343–347.

[11] Wei, H., C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, 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*, Association for Computing Machinery, New York, NY, USA, 2019, KDD '19, p. 1290–1298.

[12] Borg, D. L. and K. Scerria, Constrained Dynamic Control of Traffic Junctions. *Procedia Computer Science*, Vol. 32, 2014, pp. 293 – 300, the 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014).

[13] Cai, Z., M. Xiong, D. Ma, and D. Wang, Traffic design and signal timing of staggered intersections based on a sorting strategy. *Advances in Mechanical Engineering*, Vol. 8, No. 4, 2016, p. 1687814016641292.

[14] Cambridge Systematics, I., *Twin Cities Ramp Meter Evaluation*, ????

[15] Zhu, F. and S. V. Ukkusuri, Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, Vol. 41, 2014, pp. 30 – 47.

[16] Aslani, M., M. S. Mesgari, and M. Wiering, Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, Vol. 85, 2017, pp. 732 – 752.

[17] Wu, C., A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, *Flow: A Modular Learning Framework for Autonomy in Traffic*, 2017.

[18] Liang, E., R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, Ray rllib: A composable and scalable reinforcement learning library. *arXiv preprint arXiv:1712.09381*, 2017.

[19] Liang, E., R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, Ray RLLib: A Composable and Scalable Reinforcement Learning Library. *CoRR*, Vol. abs/1712.09381, 2017.

[20] Mao, C., Y. Liu, and Z.-J. M. Shen, Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, Vol. 115, 2020, p. 102626.

[21] Sutton, R. S. and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

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

[23] Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, Microscopic Traffic Simulation using SUMO. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.