Finding a better traffic signal policy with RL

Sung Haebin (20160463)
Computer Science and Engineering

haebin307@postech.ac.kr

***Abstract*—We tend to come up with a good traffic signal policy for various traffic situations using RL. We propose a simple problem formulation that would be a good example for solving this problem.**

***Index Terms*—Reinforcement Learning, Traffic Signal Control**

I. Introduction

The emergence of RL(Reinforcement Learning) has made some serious advance on several control problems that were too complex for humans to explicitly solve, such as robotic control, autonomous driving, or games like poker and go. But there are so much decision-making problems left in the real world yet to solve, like politics, economy, traffic, or even the world itself. The world might be a better place if RL can give answers to those very hard decisions in our life.

This proposal suggests an application of RL in the field of traffic signals. Why traffic? For me, I always have thought that I was wasting so much time on waiting for traffic signals, and when taking taxis, the estimated time of arrival is so volatile due to traffic signals or traffic congestion. Usually real-life traffic signal policies are fixed over time, although some different versions of fixed policy are used place by place, which are mostly symmetric but differs on how they take turns giving green lights. If we take a closer look when waiting for traffic signals in crossroads in real life, we can frequently observe situations where green lights don't come so soon when so many people are waiting, or situations where green lights come when no one is waiting. Certainly, it's not hard to see that there is some space for improvement in the current traffic signal policy. Instead of explicitly coding an algorithm that covers every traffic situation and crossroad, it would be better if we could find an adequate traffic policy for various situations using RL.

II. Objective

The purpose of this research is formulating(modeling) a simple problem for finding a good traffic signal system, within the given environment. Some places might have more cars coming from a particular side, so a different signal strategy regarding the environment would be efficient. RL would be convenient in finding adequate policies for various environment settings. That is why we formulate an MDP problem, which is easily trainable by RL. We also want to consider humans in crosswalks as well as cars in crossroads, since a good traffic signal system must take account humans as well.

III. Rationale

Some previous works on using RL in traffic signal control happens to exist, both value-based, policy-based, model-based, model-free, and even using multi-agent learning settings. [1] [2] [3] Multi-agent learning is used on multi-intersection problems in traffic signal control, where the agents are defined as signal controllers of each intersection. There even exists a training environment called cityflow [4], which is a multi-agent reinforcement learning environment for large-scale city traffic scenario.

These previous approaches mainly consider cars and roads, but there seems to be no approach yet that considers humans walking in the road, especially ones waiting and crossing the crosswalk, which are heavily related to the signal system of cars. In this research proposal, I would try to come up with a problem formulation that not only considers about car traffic and car accidents, but also cares about human traffic and human accidents, or even human-car accidents.

IV. Significance

The reason I focused on traffic is because transportation has been always the key of society development. If we can make the physical world closer, there would be huge advantages on production and resource efficiency of the society. Traffic signal control could be rather a small step in transportation evolution compared to projects like HyperLoop or teleportation, but it also could be an easily applicable yet powerful advancement in the current status, where most of the population are wasting time everyday going to work because of traffic congestion and inefficient traffic signals. Also if traffic signals become better, there would be fewer people who violate the signals because they are in a hurry, which would obviously would reduce car accidents. Good traffic signal systems try to reduce delays, stops, fuel consumption, emission of pollutants, and accidents. And the application of RL would be convenient, without humans having to suffer from finding a good signal policy for every crossroad in the world.

V. Methods

We could come up with several problem formulations for the traffic signal problem. Remember that our focus is on developing a simple example that covers cars in crossroads(intersections), and also humans in crosswalks. Multi-agent frameworks are widely used in multi-intersection problems, but since we want a simple example, we will not directly formulate a multi-intersection case in this paper. Rather we will stick to the single-intersection case, although the extension to multi-intersection will be quite straightforward if we refer to previous works. [1] Basically, we will define the state as the situation of the road(intersection), define action as the signal change, and define reward as some objective we want to achieve in traffic management. We will discuss the details below.

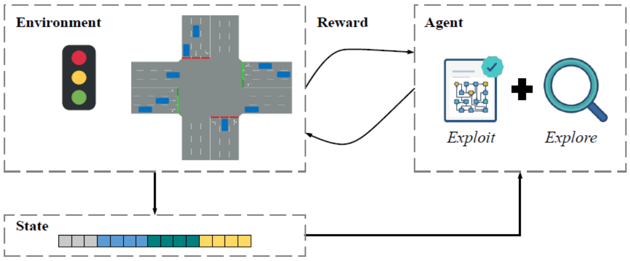

Fig. 1. Single Agent RL framework for single-intersection traffic signal control. Our approach adds 4 crosswalks in each side of the road, and also takes consider human traffic.

A. Environment

We should first consider the environment that we are going to handle. We want to handle a single intersection with 4 roads crossing, and we want 4 crosswalks on each side of the road too. Of course, a real world environment would be best for training, but obviously we can't risk people dying from faulty signals, so we somehow need a simulation environment for traffic. Our goal is a simple environment that represents a single intersection with cars and pedestrians. Figure 2.(a) is a typical environment of a single-intersection with only cars. We regard it as a discrete time step environment just for simplicity. You can see that going right has no signals like in the real world, and each straight, left directions are signaled with green/red. For only cars, we have 2^8 numbers of joint signals in a single intersection. Now we add crosswalks in each side of the road, and signals(green/red) in each crosswalk. This extension would increase the number of joint signals to 2^{12}. Then to finish the environment, we need to decide the dynamics of cars and pedestrians. Since this problem is about finding a good signal policy for each real-world road situation, the best dynamics would be a real-world sample of the target road, regarding how many cars emerged from each side, or how many cars didn't abide the signals in some situation, or how many accidents occurred and what types they were. For this simple example, we can explicitly simulate traffic with some probabilities (although then it would be solvable with only MDP planning, no need for RL). For each time step, cars could show up from each side by probability P_{car} , and cars proceed until they meet a signal. We decide the lane length L as a hyperparameter, which is the time needed to get to the signal, as well as the number of maximum cars in lane. If the signal is green, then in the next time step a car moves to its target lane and eventually go away as time passes, and if the signal is red, cars stop and wait. For pedestrians, they could show up from each side by probability P_{human}, and they cross when signals are green. Every car and human can ignore the signal and proceed with a small probability P_{troll}. And most importantly, when any car meets car, or car meets human during the time step transition, the next time step is marked as an *accident*, and the *degree of accident* is denoted by the weighted sum of cars and humans involved in the collision. Car-Car accidents have more impact and damage, so they might be more considerable than Car-Human accidents. But we definitely do not want to neglect Car-Human accidents, because it would lead to a signal policy that lets cars pass freely when humans are walking across the road, which nullifies the whole meaning of this proposal.

$$\begin{aligned} &\text{Degree of Accident} = \\ &\beta * [\text{cars involved in collision}] + \\ &[\text{humans involved in collision}], \\ &\text{where } \beta \text{ is a hyperparameter} \end{aligned} \tag{1}$$

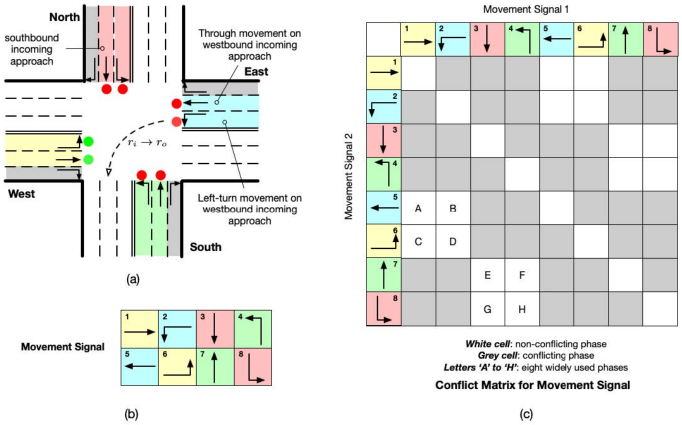
North
southbound incoming approach
Through movement on westbound incoming approach
East
$r_i \to r_o$
West
Left-turn movement on westbound incoming approach
South
(a)
Movement Signal
1
2
3
4
5
6
7
8
(b)
Movement Signal 1
Movement Signal 2
A
B
C
D
E
F
G
H
White cell: non-conflicting phase
Grey cell: conflicting phase
Letters 'A' to 'H': eight widely used phases
Conflict Matrix for Movement Signal
(c)

Fig. 2. Traffic environment for single-intersection, car-only. We extend this environment adding crosswalks and pedestrians.

B. MDP formulation

Now we define the state of our MDP. This is basically the task-relevant part of the environment. We use the same discrete state time step as the environment. There could be many states used in traffic signal control, such as queue status, waiting time, delay, speed, and the most popular one seems to be queue status. [1] For a near-minimal state, we track the number of cars in each lane, and the number of people in each side of the crosswalk, which would make the state space $24+8=32$.

Next we define action. Every 2^{12} signals could be an action, and leaving as it is may be sufficient because we have a concept of accident in the environment, so we could penalize it. But of course it would be better if we can boil down the number of actions, and this case it is definitely possible to remove some stupid signals. Figure 2.(c) shows a conflict matrix of two signals, which denotes if a collision occurs in each signal pair. Similarly for our case, we could make a 12 x 12 conflict matrix, and rule out actions that have more than one conflicting pairs.

Finally we define reward. What could be the objective of our problem? What is a good signal policy? If we concern time, we can minimize the sum of travel time or maximize throughput, or minimize the number of stops. Or if we concern road capacity, we can try to minimize queue length. The most popular objective seems to be queue length. [1] So for our proposal, we choose the reward as the negative of total queue length, with some penalization regarding the degree of accident. We choose a ”good” policy as a policy that minimizes the number of cars and humans waiting, and minimizes the degree of accident.

$$\begin{aligned} &\text{Reward} = \\ &- [\text{Total Queue length}] - \alpha * [\text{Degree of Accident}], \\ &\text{where } \alpha \text{ is a hyperparameter} \end{aligned} \tag{2}$$

As usual, we prefer instant rewards, since we cannot have people waiting. Thus we introduce a discount factor γ on the reward.

In conclusion, we have formulated an MDP environment and problem that could be a good example of the traffic signal control problem, even with pedestrians. It would be quite interesting to see what happens if we train a policy for this problem with a specific real-world intersection data.

References

[1] Hua Wei and Guanjie Zheng and Vikash Gayah and Zhenhui Li, ”A Survey on Traffic Signal Control Methods” IEEE ITSC 2020

[2] F. Rasheed, K. -L. A. Yau, R. M. Noor, C. Wu and Y. -C. Low, ”Deep Reinforcement Learning for Traffic Signal Control: A Review,” in IEEE Access, vol. 8, pp. 208016-208044, 2020, doi: 10.1109/ACCESS.2020.3034141.

[3] N. S. Jadhao and A. S. Jadhao, ”Traffic Signal Control Using Reinforcement Learning,” 2014 Fourth International Conference on Communication Systems and Network Technologies, 2014, pp. 1130-1135, doi: 10.1109/CSNT.2014.231.

[4] Zhang, Huichu et al. “CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario.” The World Wide Web Conference (2019): n. pag. Crossref. Web.