Intelligent Traffic Lights Schedule
—Based on Reinforment Learning

Zhuo Chen
ShanghaiTech University
chenzhuo@shanghaitech.edu.cn

Haoxin Liu
ShanghaiTech University
liuhx@shanghaitech.edu.cn

Zheng Shu
ShanghaiTech University
shuzheng@shanghaitech.edu.cn

Yuyang Gao
ShanghaiTech University
gaoyy@shanghaitech.edu.cn

Abstract

In modern society, traffic lights play an important role in scheduling traffic flow in different directions. However, existing traffic light scheduling schemes often cause traffic jams especially during rush hour, for the reason that it only obeys a simple scheme of periodic alternating and usually cannot fit traffic conditions in real world. In this work, we design a new intelligent traffic lights schedule based on reinforcement learning and take humanization into consideration. This work shows great performance on three different simulated traffic conditions which means we can beat the traditional traffic lights schedule.

1. Introduction

1.1. Motivation

Modern traffic lights have used almost the same mechanism since their birth, which is alternating periodically and each signal lasts for a fixed time. This basic scheme cannot adapt to different traffic conditions, so there exists many pain points about this immutable mechanism. For example, it may cause severe traffic jam during rush hour since the traffic flow in one direction is significantly larger than other directions and make drivers waiting for a lone time. What's more, it usually directs traffic with low efficiency and can not allocate resource sufficiently, and need policemen to provide help sometimes. Inspired by these pain points, we dedicate to solve this realistic problem by design an intelligent traffic lights schedule using reinforcement learning.

1.2. Problem statement

To show our work intuitively, we will split the whole project into the following aspects. First of all, the intelligence of the systems we design is embodied in that it can dynamically adjust for different traffic conditions, which tends to change signal light alternation mechanism according to the real traffic. Secondly, this system means that we design new traffic lights whose behavior will be changed from the fixed pattern to free combinations. The algorithm applied in this work is mainly based on approximate Q-learning and we will focus on reducing the average waiting time as the evaluation mechanism. Finally, we also add some humanization into consideration, such as maintaining fairness for all lanes to avoid waiting for a long time.

2. Model Setup

In order to reproduce the real-world traffic situation as much as possible, we went to the intersection of Rd.Jinke and Rd.Zhucongzhi to collect the traffic flow data on Wednesday morning rush hour. With information extracted from real world, we construct our intersection model as below:

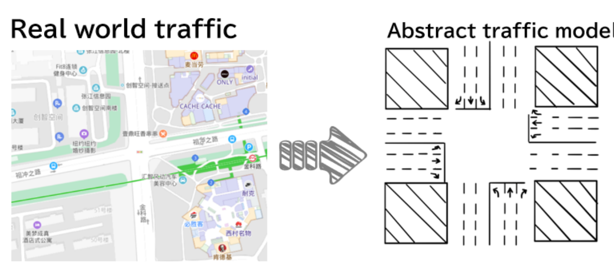

Figure 1. Abstract traffic model

In our traffic model, we assume that the basic unit time of green light is fixed and straight green light will last for 30s while left green light is 15s. Each trasition between states will choose to continue with the current green light or turn on another green light and each green light will last for the fixed time mentioned just now, and then perform

next transition. Besides, we also assume that the maximum number of passed cars in one time unit is fixed: the straight lane can pass 25 cars while the left lane can pass 10 cars at maximum. These data are all extracted from the real world traffic infomation. To simplify our model, we allow right-turn cars to pass directly by default so there will be no need to consider right-turn lights.
As the figure 1 shows, each road in the traffic model is composed of three lanes: left-turn lane, straight lane and right-turn lane, but only the first two lanes have traffic lights to control their traveling. So, the number of combinations of traffic lights are also limited. Here is a table to show possible lights combinations.

Light Location	1	2	3	4	5	6
South-North Straight Light	G	G	R	R	R	R
South-North Left-turn Light	R	G	G	R	R	R
East-West Straight Light	R	R	R	G	G	R
East-West Left-turn Light	R	R	R	R	G	G

Table 1. Lights combinations(R-Red; G-Green)

3. Algorithm

Taking the magnitude of states an intersection may have, which is exponential with respect to the amount of cars on a single lane, we decide to apply approximate Q-learning to our intelligent traffic light controller. For an approximate Q-learning method, the two most significant factors are features we extract from a given state and the reward function we use to evaluate a transition.

3.1. Features

The feature we extract from a state is a 5×1 vector. First we define the degree of crowdedness by dividing it into 4 levels-sparse, normal, crowded and traffic jam. These four entries of feature vector stand for the degree of crowdedness of four pairs of lanes in an intersection respectively. *A pair* here means the two lanes controlled by the same traffic light, for example, the lanes for cars go straight from the south to the north and cars go straight from the north to the south. The last entry in our feature vector is the time of the unfortunate car that has been waiting for the longest time among all cars in the intersection. This feature is going to be very useful when we are avoiding the cars on spars lanes starving from the cars of crowded lanes (cars that cause rush hour). Figure 2 shows an example of the feature vector and the illustration of crowdedness level. And we use a linear function to calculate Q-value like the equation below:

$$Q(s, a) = \omega_1 f_1(s, a) + ... + \omega_5 f_5(s, a)$$

$$\omega_i \leftarrow \omega_i + \alpha[difference]f_i(s, a)$$

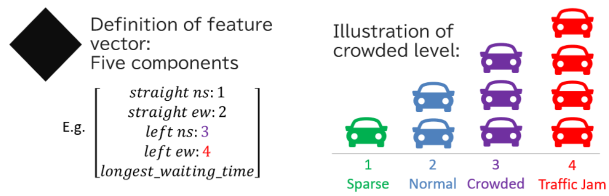

Figure 2. An example of the feature vector and the illustration of crowdedness level.

3.2. Reward function

Reward function of our design is divided into 3 parts: multiplication, one bonus and one penalty.
First, we take the product of number of cars just passed after a transition and the number of cars on that lane before this transition. By taking the product of these two numbers we basically hope these two transitions happen. One is that we want the intelligent traffic light to prefer choosing the green light that can let more cars pass, and the other is that we want it to attach more importance to the lanes that are more crowded.
Second, we have a penalty for the transition that caused excessive waiting time for some car(s) in the intersection. To be more specific, after a transition is done, which means an action (which green light to set) is chosen, the waiting time of cars that are blocked will all be updated. As long as the waiting time of some of them reach a certain threshold, we activate this penalty.
Third, this part of our reward is a bonus for a transition, which encourages our intelligent traffic light to choose consecutive green lights combinations rather than spilt them apart, for example, when the traffic flow is mostly evenly distributed, we prefer a light selection of two green lights for north-south straight lanes consecutively then two green lights for east-west straight lanes consecutively rather than that the selection of north-south, east-west, north-south and east-west. The reason for giving this bonus is that in real traffic situation there is always a fixed overhead for cars to launch and go cross the intersection. That is, if the transition decides to choose a green light which is the same as last one it will receive a bonus for this choice. Note that the weight of this part is not as large of as the first two.

$$Reward = Number\_of\_passed\_cars \times crowded\_level$$

$$+Penalty$$

$$+Bonus$$

4. Experiment

This section contains our detailed work about constructing data set, evaluate mechanism and result presentation.

4.1. Data set construction

When choosing or generating data-set, purpose takes matters. For our Reinforcement Learning Algorithm and expectation, a data set whose traffic density is very low is useless, since there exists high random and it is hard to predict even if the data we use to learn and the data we want to test is similar. In other words, the traffic may not show its statistical characteristics during a short time. Actually, even though our algorithm can work for above condition, it rarely performs better than traditional traffic lights. A good idea is only researching rush time and traffic jam, and trying to make our model best for them.

After the purpose is clarified, researchers observe and record traffic situation at the intersection of Rd.Jinke and Rd.Zuchongzhi during morning rush time. Uneven traffic density from each direction is a normal phenomenon. More specifically, The cars from east are much more than those from east, since workplaces and residences are located in different directions, meanwhile, the cars choosing to go straight is about twice more than those turning left(the cars choosing to turn right are ignored since they does not influence our model for single intersection).

We generated three kinds of datasets to simulate three different traffic conditions. Here is one table to show the property of these datasets.

Name	Corresponding situation	Property
Random	Normal traffic condition	The traffic flow in all directions is basically the same without congestion
Crowd	Rush hour	There is significantly more traffic in one direction than in the other
Extreme	Road construction or accidents	The road is blocked in one direction

Table 2. Three kinds of datasets

4.2. Evaluation mechanism

We mainly use average waiting time for all cars and longest waiting time among all cars for a certain period of time. The average waiting time measures the overall performance of our model with respect to reducing the waiting time for cars during rush hour or common case while the longest waiting time measures the degree of humanization of our model. Basically these two factors can not be optimized at the same time because taking longest waiting time into consideration is bound to extend the average waiting time of all cars and being more humanized means we have to sacrifice some time in letting cars pass in less crowded lanes.

4.3. Result

First, the figure showing below illustrates the performance of our model in a normal traffic situation which means that cars from all directions are randomly distributed. In this case, our model still does a decent job as in such case there is not so much space that can be optimized. When encountering the same set of traffic flow our model has a little better performance than a traditional periodic traffic light.

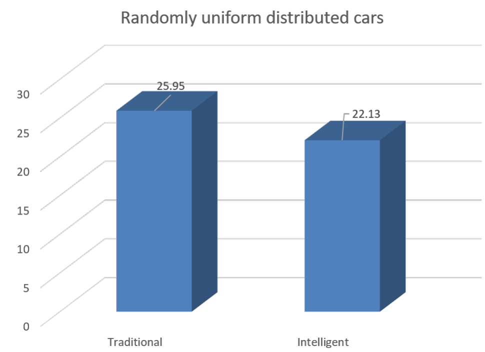

Figure 3. Random uniform distributed cars

Second, the achievement of our model on traffic flow during rush hour is the most essential part among all situations because this perfectly solve the problem we proposed at the beginning of this project, and the characteristic of traffic flow in this case is based on the real situation of rush hour at the intersection where we did a field work. Recall that when we are speaking of humanization, it means we do not want that cars from crowded lanes always are prior to cars from sparser lanes. The first graph shows the performance of our model when our intelligent traffic light is not humanized, and the second one illustrates the performance on the same traffic flow when we take humanization into consideration. It can be easily inferred that taking humanization will result in a slight decrease in average waiting time, but the longest waiting time is well limited, which is shown in Figure 4. During our simulation on more unbalanced traffic flow the difference of longest waiting time is even larger between these two models. Therefore, we think that the model taking humanization into consideration is more realistic one in real life.

Third, we also run our model in a relatively extreme traffic situation in which the cars are mostly from very few directions and are taking simplex action at the intersection. For example, cars are mostly from north and they are mostly choose to go straight. This case aims to simulate some situation when there is an unexpected car accident or road con-

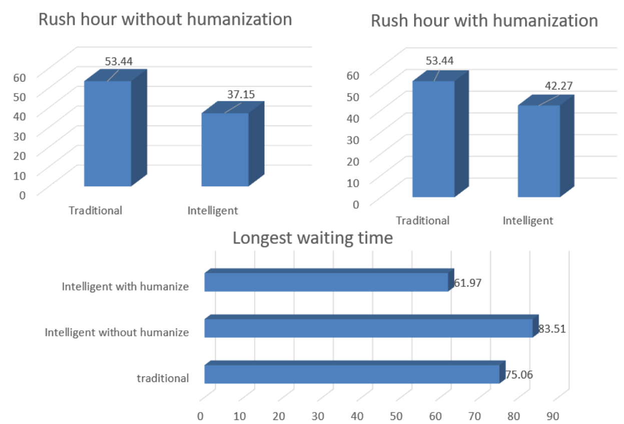

Figure 4. Rush hour results comparasion

struction in this intersection. When this happens it might cause that some action at the intersection is forbidden. We can imagine that a traditional periodic traffic light will do poor in this situation as it wastes a lot of time waiting for NO cars to pass. However, our model appears to be much cleverer because it learns that giving green light to lanes that contains NO cars is meaningless. Similar to the case of rush hour, an unfair traffic lights combination in this case is more possible to happen and the difference on longest waiting time is more obvious. Hence a humanized model is more suitable for real life.

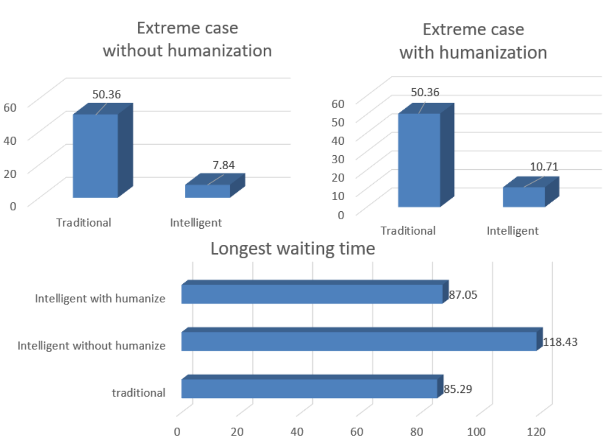

Figure 5. Extreme case results comparasion

5. Future Work

Further development for our model can be discussed in two main aspects. The first one is that we can make more efforts to adjust the parameter settings in our intelligent traffic light scheduler, for example, the difference between the number of cars can pass in two consecutive green light and that in two separate green lights. Now although we are encouraging the traffic light to take consecutive green lights the number of cars can pass the intersection is still the same as a normal transition from a red light to green light. The other aspect is that the model we design is flexible and thus it can be extended to multi-intersection system. Each intersection can be controlled by an independent intelligent agent and the agents can communicate with surrounding intersections to achieve a better performance in a larger view rather than a single intersection.

References

[1] Xie, Xiao-Feng, Stephen F. Smith, and Gregory J. Barlow. ”Smart and scalable urban signal networks: methods and systems for adaptive traffic signal control.” U.S. Patent No. 9,159,229. 13 Oct. 2015.

[2] Al-Khateeb, Khalid, and Jaiz AY Johari. ”Intelligent dynamic traffic light sequence using RFID.” 2008 International Conference on Computer and Communication Engineering. IEEE, 2008.

[3] Han, Taehee, and Chiho Lin. ”Design of an intelligence traffic light controller (ITLC) with VHDL.” 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM’02. Proceedings.. Vol. 3. IEEE, 2002.

[4] Taskin, Harun, and Remzi Gumustas. ”Simulation of traffic flow system and control using fuzzy logic.” Proceedings of 12th IEEE International Symposium on Intelligent Control. IEEE, 1997.

[5] Xie, Xiao-Feng, et al. ”Real-time traffic control for sustainable urban living.” 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014.

[6] Wiering, M. A., et al. ”Intelligent traffic light control.” (2004).

[7] Steingrover, Merlijn, et al. ”Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion.” BNAIC. 2005.