Technical Report of KDD CUP 2021 City Brain Challenge

Xibo Zhou
azxbnb9s@163.com
BOE Technology Group Co., Ltd.
Beijing, China

Jinxiao Wen
wenjinxiao_metis@163.com
BOE Technology Group Co., Ltd.
Beijing, China

Zhuoshi Yang
yangzhuoshi@boe.com.cn
BOE Technology Group Co., Ltd.
Beijing, China

ABSTRACT

We describe the ideas and methodologies that we developed in addressing the KDD Cup 2021 City Brain Challenge, targeting on city-level traffic light signal control problem. We introduce the background and the given task, and explain the design and mechanisms of our proposed algorithm in detail. Finally, we discuss some issues we tried to optimize but didn't work in this competition.

KEYWORDS

kdd cup, traffic signal control, reinforcement learning

ACM Reference Format:
Xibo Zhou, Jinxiao Wen, and Zhuoshi Yang. 2021. Technical Report of KDD CUP 2021 City Brain Challenge. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Traffic signal control is a key problem in a city-scale smart transportation system [7]. An intelligent traffic signal control algorithm is essential for improving the traffic conditions in terms of reducing the average travel cost and reducing the traffic congestion. In practice, most traffic signals are still controlled with fixed-time plan, which is unaware of real traffic conditions. Many approaches have been proposed to solve this problem by utilizing the traffic conditions on the road network [2][3], among which reinforcement learning techniques [5][8][4][6][1] have been more and more used in recent years.

The KDD Cup 2021 City Brain Challenge, organized by Yunqi Academy of Engineering, aims to target on this complicated problem. In this challenge, the competitors were provided with a city-scale road network and its traffic demand derived from the traffic flow data. The task is to coordinate the traffic signals to maximize the number of vehicles running on the road network while maintaining an acceptable travel delay. More specifically, for a four-leg intersection shown in Figure 1, there are 8 candidate signal phases to select, each giving pass to a pair of non-conflict lanes. The task is to develop an algorithm to select traffic signal phases for all the intersections of a road network during each period of time step, in order to optimize the traffic performance. When a traffic signal phase is selected, it lasts for the next 10 seconds. If a traffic signal is switched to another phase, there will be a 5 seconds period of "all red" at the beginning of the next step, in which all vehicles are not allowed to pass.

The evaluation metric of the traffic performance include the total number of vehicles running on the road network, and the average delay index of all the vehicles. In particular, the delay index of a vehicle is denoted as:

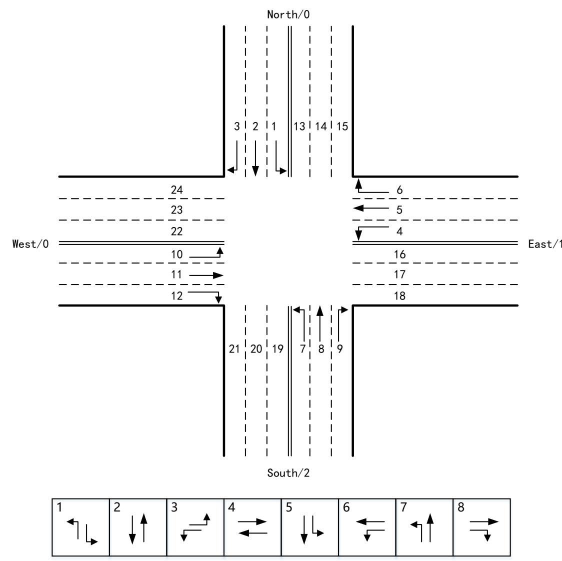

Figure 1: Traffic signal phases of a four-leg intersection

$$D = \frac{TT_i + TT_i^r}{TT_i^j} \quad (1)$$

where TT_i is the travel time of vehicle i, TT_i^r is the travel time of the rest trip estimated with free-flow speed, and TT_i^j is the travel time of the entire trip estimated with free-flow speed. The target is to maximize the number of vehicles running on the road network while keeping the average delay index under a predefined threshold (1.4 in the final phase of this competition).

In this challenge, our algorithm is run on a traffic simulation engine named CBEngine provided by the organizer. The input data includes 1) the road network and 2) the traffic flow. More specifically:

1) The road network consists of the intersection dataset, the road dataset, and the traffic signal dataset. The intersection dataset contains the coordinates and traffic signal installation information of each intersection. The road dataset contains the information of each road segment, including the upstream and downstream intersection ID, road length, the speed limit, and the number of lanes. All the road segments have two directions. The traffic signal dataset contains the information of each intersection followed with the road segments connected with it. Each intersection has up to

, ,

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

four connected road segments, listed in clockwise direction starting from the north. Some intersections only have three connected road segments. In this case, the road segment ID is replaced with -1 in the missing direction.

2) The traffic flow data consists of a series of traffic flows generated by a traffic generator. Each traffic flow is represented as a tuple: (t_s, t_e, t_i, r), which means there will be a new vehicle traveling along the route r every t_i seconds from t_s to t_e. The route r consists of a series of connected road segments. Each vehicle starts at the first intersection of r, and terminates at the last intersection of r.

When the simulation starts, the simulator first load and process the datasets, and initialize the environment. Next, the simulation runs continuously at each step of 10 second intervals of the real traffic. At each step, the simulator first calls our algorithm to determine the phase selections of all traffic signals, and then feed the decisions into CBEngine. Then the simulator will get an observation of the traffic on the road network, including the vehicle-level information and the lane-level information. For each vehicle, the observation includes its current road ID and road lane, the distance from this vehicle to the start intersection of the current road, current speed, and the timestamp when this vehicle enters the road network. The simulator computes the number of vehicles running on the road network and the average delay index of all vehicles every 20 seconds. When the average delay index reaches the predefined threshold, the simulation process terminates.

2 ALGORITHM DESCRIPTION

The overall process of our algorithm is divided into two parts. The first past is to compute the estimated cost of each traffic signal phase selection for each intersection, and the second part is to determine the traffic signal phase for each intersection.

2.1 Cost Estimation

The process of cost estimation of each traffic signal phase for each intersection is as follows:

1) Given the vehicle-level observations, we first computes the available distances of all the road lanes. The process is shown in Algorithm 1, where Obs_v is the vehicle-level observations, D_{stuck} is the set of available distances of all the road lanes, $r.length$ is the length of the road lane r, $v.speed$ is the current speed of v, $v.r$ is the current road lane, $v.dist$ is the distance from v to the start intersection of r.

2) For each vehicle v in the vehicle-level observations, we compute the remain time for v to reach the end intersection of current road. We assume the vehicle travels with an acceleration of 2.0 m/s if the current speed $v.speed$ is lower than the speed limit of the road $r.speed$, and then travels at a constant speed after reaching $r.speed$. Thus, the remain time t_r is calculated as follows:

$$t_a = \frac{r.speed - v.speed}{2.0} \tag{2}$$

$$d_a = \frac{(r.speed + v.speed) \times t_a}{2.0} \tag{3}$$

Algorithm 1 computing the available distance of all the road lanes.

```
Require: Obs_v
Ensure: D_stuck
for v in Obs_v do
    Obtain v.r
    if The r is empty then
        D_stuck(r) = r.length
    end if
    if v.speed = 0 & v.dist < D_stuck(r) then
        D_stuck(r) = v.dist
    end if
end for
return D_stuck
```

$$t_r = \begin{cases} \dfrac{-v.speed + \sqrt{v.speed^2 + 4 \times d_r}}{2.0} & if\ d_a > d_r \\ (r.length - v.dist - d_a)/r.speed + t_a & otherwise \end{cases} \tag{4}$$

where t_a is the expected time to accelerate to the speed limit of road r, d_a is the expected travel distance during t_a, and $d_r = r.length - v.dist$ is the remain distance for v to reach the end intersection.

3) Next, for each vehicle v in the vehicle-level observations, we compute the stuck time for v if it is not allowed to pass at the end intersection. Due to the "all red" period when the traffic signal phase switches, we split the stuck time into the first and last 5 seconds for the next step of 10 seconds. Normally, when the next road lane for v is not that congested, the stuck time of the first 5 seconds is

$$t_{v1} = 10 - t_r \tag{5}$$

if $t_r < 5$, and 0 otherwise; similarly, the stuck time of the last 5 seconds is

$$t_{v2} = 10 - t_r - t_{v1} \tag{6}$$

if $5 <= t_r < 10$, and 0 otherwise. However, if the next road lane for v is congested, the stuck time is the max available travel time on the next road lane for v. We use D_{stuck} to estimate whether the next road lane will be congested. Let r_n be the next road lane for v, then r_n is congested for the first 5 seconds if $\frac{D_{stuck}(r_n)}{r_n.speed} < t_{v1}$, and thus

$$t_{v1} = \frac{D_{stuck}(r_n)}{r_n.speed} \tag{7}$$

Similarly, r_n is congested for the last 5 seconds if $\frac{D_{stuck}(r_n)}{r_n.speed} < t_{v1} + t_{v2}$, and thus

$$t_{v2} = \frac{D_{stuck}(r_n)}{r_n.speed} - t_{v1} \tag{8}$$

Since the travel route of each vehicle is not visible in the signal phase selection algorithm, we assume that each vehicle stays on the same lane of the next road after passing the intersection. Note that in case the next road is too short, it is possible for a vehicle to finish traveling the entire road lane within 5 seconds. We temporarily ignore these roads and still use Equation 5 and 6 to compute t_{v1} and t_{v2}.

Moreover, there is a situation when no vehicle is able to reach an intersection from any direction in the next step. In order to save 5 seconds for signal phase switch, we compute the stuck time for v during the next 10 seconds after next step if it is able to reach the end intersection. The stuck time of the 10 seconds after next step is

$$t_{v3} = 20 - t_r - tv1 - tv2 \tag{9}$$

if $10 <= t_r < 20$, and 0 otherwise.

4) After computing the stuck time for all the vehicles, we aggregate the stuck time on all the road lanes. Thus, we can obtain the estimated time for each road lane if it is not given pass by the corresponding traffic light. Similar to vehicle stuck time calculation, we split the estimated time into the first and last 5 seconds for the next step of 10 seconds, and also compute the estimated time of the 10 seconds after next step.

2.2 Signal Phase Selection

The process of signal phase selection for for each intersection is as follows:

1) Given the estimated stuck time of each road lane for an intersection, we first compute the pass reward for each road lane. Since each traffic signal phase switch will cause a 5 seconds period of "all red" at the beginning of the next step, it is not encouraged to switch phases of a traffic signal too frequently. Therefore, we multiply the estimated stuck time by a phase maintenance coefficient γ_k for the road lanes that were allowed to pass in the previous step. In addition, we observe that the left lanes on the road network are far less likely to be released than the straight lanes. In order to lift the chance of being selected, we multiply the estimated stuck time by a left lane coefficient γ_l for the left lanes. After experimenting various mechanisms to set values for γ_k and γ_l, we finally fix the values to: $\gamma_k = 1.68$ and $\gamma_l = 1.47$.

2) Next, we compute the reward for each traffic signal phase. Given a traffic signal a. Note that only the phases selected in the previous step are able to give pass to their corresponding road segments for a full 10 seconds. The rest of the phases will only give pass for the last 5 seconds. Thus, the reward of a traffic signal phase p is calculated as follows:

$$c_p = \begin{cases} c1_{v1} + c2_{v1} + c1_{v2} + c2_{v2} & if\ p = p_{last} \\ c1_{v2} + c2_{v2} & otherwise \end{cases} \tag{10}$$

where $c1_{v1}$ and $c2_{v1}$ are the reward of the two lanes given pass by phase p for the first 5 seconds of the next step, $c1_{v2}$ and $c2_{v2}$ are the reward of the two lanes given pass by phase p for the last 5 seconds of the next step, and p_{last} is the phase of a selected in the previous step.

3) Finally, we decide the phase for each traffic signal. For a traffic signal a, if the reward of any phase is above 0, the algorithm returns the phase with the maximum reward. If the reward of all the phases for a is 0, which means no vehicle is able to reach the end intersection for a, the algorithm computes the reward of each phase in the 10 seconds after the next step, and returns the phase with the maximum reward. Moreover, if the reward of all the phases in the 10 seconds after the next step is still 0, the algorithm returns 0, which means no phase is selected.

2.3 Remain Time Correction

In the cost estimation of our algorithm, the remain time for each vehicle to reach the end intersection is calculated under a series of assumptions, which are often not true in the actual traffic situation. With the given observations from the simulator, it is not practical to develop a more detailed method to better estimate the remain time. Therefore, in each step, we develop a method to correct the estimated remain time for each vehicle by utilizing the observations of the previous step. The method works as follows:

1) In each step, we record the remain time of all the vehicles that are able to reach their end intersections (which means the remain time is less than 10 seconds).

2) In each step, we compare the observations of the current step and the last step, and find out the vehicles that should have moved to their next roads, but still stuck on the same roads of previous step. More specifically, a vehicle v is marked as "fail to pass" if and only if:

- v is recorded in the previous step
- v stays on the same road lane r from previous step to current step
- r is given pass by its traffic signal in the previous step, or r is a right lane
- $v.speed = 0$

3) After obtaining the vehicles marked as "fail to pass", we compute the average stuck time error of each road lane. Let V_{fl} be the set of vehicles marked as "fail to pass" on road lane l, the average stuck time error is computed as:

$$e_l = \frac{\sum_{v \in V_{fl}} (10 - t_r)}{|V_{fl}|} \tag{11}$$

where t_r is the remain time of v, and $|V_{fl}$ is the size of V_{fl}. Then, the average stuck time error of each traffic signal a is calculated as:

$$e_a = \frac{\sum_{l \in L_a} e_l}{|V_a|} \tag{12}$$

where L_a is all the road lanes controlled by a, and $|V_a|$ is the size of the vehicle set stuck at a marked as "fail to pass".

4) In each step, after computing the average stuck time error of all the traffic signals, we use the error to correct the travel time estimation for all the vehicles in Section 2.1:

$$t_r = t_r * e_a \tag{13}$$

where a is the traffic signal at the end intersection v is about to reach.

2.4 Dealing with Three-leg Intersection

A three-leg intersection is shown in Figure 2. Different from a four-leg intersection, there are six lanes missing from a three-leg intersection. As a result, for a three-leg intersection, most of the traffic signal phases are only able to give pass to one road segment, which is inefficient. Therefore, we reduce the phase choices to 3 for a three-leg intersection. For example, in Figure 2, the northern road is missing. Therefore we only keep phase 1, 4, and 6. Note that phase 1 still only give pass to one road segment, because the left lane of the southern road cannot be paired with any other road segment.

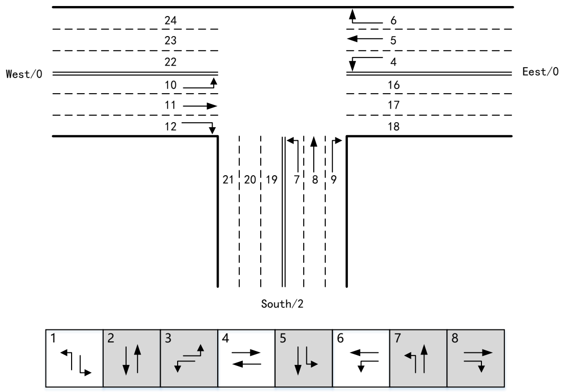

Figure 2: Traffic signal phases of a four-leg intersection

We also keep 3 phases for other types of three-leg intersections following the same mechanism.

Another issue for a three-leg intersection is that, some road lanes do not have a next road to travel. For example, in Figure 2, lane 6, 8, and 10 are expected to switch to the northern road that does not exist. Although based on our observations, no vehicle will travel on these lanes unless it is newly entering the road network or is finishing its trip, we still made some rearrangement for these lanes. When obtaining the next road for a vehicle, if it is traveling on the lane that cannot turn right or left, we assume it is going straight; if it is traveling on the lane that cannot go straight, we assume it is turning left. For example, in Figure 2, the vehicles traveling on lane 6 will be expected to move to lane 23, the vehicles traveling on lane 8 will be expected to move to lane 22, and the vehicles traveling on lane 10 will be expected to move to lane 17. Note that we only make these assumptions to compute the stuck time, we do not add the cost or reward of these vehicles to the rearranged road lanes.

3 INSIGHTS

The detailed mechanisms and methods introduced in Section 2 are tested to be effective during the competition. Besides that, there are several issues we tried to optimize but didn't result in a better score. We list some of the issues for future study:

- Lane switch. In the final phase, the complete route of vehicles are invisible to the phase selection algorithm. Without a precise knowledge of the lane switch, we can't estimate whether a vehicle will be stuck after it passes the traffic signal and moves to the next road. This sometimes leads to a consequence that the vehicles cannot move to the next road despite given pass by the traffic signal, because the next road is already congested.
- Short roads. In the road network, 13% of the road segments are too short for a vehicle to travel for 10 seconds at the speed limit. For these road segments, it is difficult to estimate the reward, because the vehicles that are able to reach the end intersections haven't been observed on these road segments yet. Since we also couldn't know which vehicles are going to move to these road segments due to the lane switch problem, there is a dead cycle. As a result, these short road segments usually have less advantage for the traffic signal to give pass to them. Moreover, since short road segments are easier to get stuck, they are more likely to be given up by traffic signals, which causes a more serious traffic congestion.
- Chains of congestion. When a road segment is continuously ignored by the traffic signal, it gets more and more congested. In some serious cases, an entire road segment can be congested, which causes the congestion on the following road segments.
- Frequent Signal Phase Switch. In some intersections, more than two road lanes are congested at similar degrees. In these cases, the signal phases sometimes switch frequently to give all the road lanes chances to pass. However, due to the "all red" period, this leads to a lower efficiency for any road lane to smoothly evacuate.

In conclusion, by the end of the competition, there's still a lot of issues that can be worked on. This is a chance for us to further study the traffic signal control problem, which is essential for improving the quality of transportation system in large cities.

REFERENCES

[1] 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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 3414–3421. https://aaai.org/ojs/index.php/AAAI/article/view/5744

[2] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. 2008. Self-Organizing Traffic Lights: A Realistic Simulation. In *Advances in Applied Self-organizing Systems*, Mikhail Prokopenko (Ed.). Springer, 41–50. https://doi.org/10.1007/978-1-84628-982-8_3

[3] Isaac Porche and Stéphane Lafortune. 1999. Adaptive Look-ahead Optimization of Traffic Signals. *J. Intell. Transp. Syst.* 4, 3-4 (1999), 209–254. https://doi.org/10.1080/10248079908903749

[4] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash V. 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, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1290–1298. https://doi.org/10.1145/3292500.3330949

[5] 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. *CoRR* abs/1905.05717 (2019). arXiv:1905.05717 http://arxiv.org/abs/1905.05717

[6] 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, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1913–1922. https://doi.org/10.1145/3357384.3357902

[7] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2496–2505. https://doi.org/10.1145/3219819.3220096

[8] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash V. Gayah, Kai Xu, and Zhenhui Li. 2019. Diagnosing Reinforcement Learning for Traffic Signal Control. *CoRR* abs/1905.04716 (2019). arXiv:1905.04716 http://arxiv.org/abs/1905.04716