IntelligentLight: Vote-Based Traffic Coordination Algorithm

Gang Song
1015120141@qq.com
Shanghai Jiao Tong University
Shanghai, China

Zheng Zhang
liszthorowitz@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Chufan Wang
wangchufan@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Yucheng Gu
enoch2090@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Zeyu Sun
ecori-01@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

In transportation and city planning fields, traffic control is an essential concept. Various methods have been proposed, but few of them are adapted to a city-scale roadnet. We propose a novel method based on analyses of real traffic flow. The method exploits the information from traffic flow, and dynamically adjusts its strategy of controlling signals. Through our tests on various generated traffic flows, we demonstrate that this method outperforms the previous methods on large-scale roadnets.

KEYWORDS

traffic signal control, multi-agent system

ACM Reference Format:
Gang Song, Zheng Zhang, Chufan Wang, Yucheng Gu, and Zeyu Sun. 2021. IntelligentLight: Vote-Based Traffic Coordination Algorithm. In ,. ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Traffic congestion has become one of the most predominant problems in the world. In some cities, this situation can be alleviated by rational road network planning, but one of the most effective and costless ways is coordinating the traffic signals dynamically according to the real-time traffic data.

In this challenge, we adjust the traffic lights to utilize the given road network at its maximum capacity, serving vehicles as many as possible while maintaining an acceptable delay. The evaluation time step is adjusted to twenty seconds in the final phase and the overall trip delay index should not exceed the threshold of 1.40.

We devise a vote-based traffic coordination algorithm, where each vehicle in the lane can "vote" for the signal phase in the next time step. Several factors will affect the weights of their votes, including the total preserved time, total travel time prediction, and the calculated pressure item of the vehicles. These components are carefully designed according to the characteristics of vehicle information. Our approach is proved to have high adaptability to real-time traffic data with randomness in the experiments.

In this paper, we will first give a precise definition of the problem. Section 3 then shows the framework and technical details of our algorithm, including the components that we designed. Section 4 then gives our choice of parameters. Experimental results and conclusion are given in Section 5 and Section 6.

2 PROBLEM DEFINITION

In this section, we revisit the problem under our notation. The city roadnet is defined over a directed multigraph $G = (I, L)$ where I is the set of intersections, and L a set of lanes. For each intersection i_k in I, there are 12 lanes $l_{i_k,1} \sim l_{i_k,12} \in L$ where i_k is their terminal vertex, and another 12 lanes $l_{i_k,13} \sim l_{i_k,24} \in L$ where i_k is their initial vertex. Table 1 gives the definition of the symbols we would use.

Table 1: Symbols

Symbol	Definition
i_k	Some intersection
v_j	Some vehicle
$v_{i_k,j}$	Some vehicle that belongs to the upstream lane of i_k
$l_{i_k,x}$	Some lane w.r.t. i_k
$p_{i_k,t}$	Available phases of i_k at time step t
$Q(p_{i_k,t})$	Action values of phases of i_k at time step t
$\rho_{l_x}^{down}$	Downstream vehicle density of lane l_x
$\rho_{l_x}^{up}$	Upstream vehicle density of lane l_x
$s_{v_j,i_k,t}$	Vote score of v_j at i_k, at time step t

The algorithm takes the information V_t of all vehicles over L, based on a 10-second-interval observation, and outputs its decision P_t, which is the set of next phases $p_{i_k,t}$ each intersection will take at timestep t. Table 1 summarizes the notations that we would use over the following sections.

3 METHOD

In this section, all mechanisms used by our algorithm will be described. Please note that it is not possible to reach every detail of the algorithm within this single report, so we will only cover the main ideas with which the algorithm can be reproduced. Implementation details can be reviewed by checking our code release.

3.1 Framework of Vote-Based Coordination Algorithm

The general framework of our algorithm is inspired by the state-of-the-art traffic signal control method *MaxPressure* [3]. Based on the information we gathered, a vote score $s_{v_j,i_k,t}$ is assigned to each vehicle. For each valid phase $p_{i_k,t}$, we define the action value $Q(p_{i_k,t}) = \sum_{V_t} s_{v_j,i_k,t}$. The phase with the maximum action value is then selected as the best action for this intersection at current timestep, i.e.

$$A_{i_k,t} = \operatorname{argmax}\left(Q(p_{i_k,t})\right) \tag{1}$$

The design of such a framework is based on the success of *Max-Pressure* method. To make the performance better, we further add various features based on observation of the traffic flow simulation.

3.2 Traffic Pattern Recognition

Before introducing the components we use to construct the vote score $s_{v_j,i_k,t}$, it is crucial to introduce a mechanism that we integrate within our algorithm, which is traffic pattern recognition.

Traffic patterning and grouping has long been a subject to be studied. Various researches have proved the effectiveness of such a method in relieving traffic pressure. Calafate Et Al. [1] proved in their work that modeling the traffic behavior by aggregating traffic flows with similar behavior would improve the congestion problem.

In our implementation, we use a simplified traffic pattern recognition mechanism. This mechanism works as follows: whenever a vehicle enters the roadnet, its route is recorded and cross-validated with existing records. We use the assumption that vehicles starting at the same intersection and moving in the same direction have a high probability to have similar routes. With this assumption, we use routes in the records to predict the future routes and total travel time of new vehicles in the roadnet.

The implementation of the above method requires another concept to be clarified, which is vehicle constraint. We classify vehicles into two kinds:

(1) **Unconstrained vehicles**, which refers to vehicles either reaching the end intersection of their journey or the remaining of their journey consists of no signalized intersections.
(2) **Constrained vehicles**, which refers to vehicles that are not unconstrained.

The route of a vehicle is recovered using our traffic pattern recognition method. If the route of a vehicle can not be deduced based on the roadnet history, it is treated as constrained.

For unconstrained vehicles, we modify their vote scores. Since they have no preference to downstream lanes, their downstream pressure (see section 3.3.2) is automatically set to 0.

3.3 Calculation of Vote Score

Recall that the vote score for a vehicle v_j at timestep t is $s_{v_j,i_k,t}$. We now define the score as follows:

$$s_{v_j,i_k,t} = T^f_{v_j,t} \cdot \mathcal{P}_{i_k,t} \cdot T^m_{v_j,l_x,t} \tag{2}$$

Table 2: Symbols in Calculation of Vehicle Score $s_{v_j,i_k,t}$

Symbol	Definition
$T^f_{v_j,t}$	Total travel time prediction of vehicle v_j at time step t
$\mathcal{P}_{i_k,t}$	Pressure item of intersection i_k at time step t
$T^m_{v_j,l_x,t}$	Total preserved time by allowing vehicle v_j to pass at time step t

The symbols used in (2) are defined in Table 2.

Each component of $s_{v_j,i_k,t}$ is ranged in $(0,1)$, so $s_{v_j,i_k,t} \in (0,1)$. The following parts will dive into the design of these components.

3.3.1 Total Travel Time Prediction.

Assuming a vehicle is driving on the route with no traffic signal and other vehicles' intervention, t_{ff} is used to define its travel time. With traffic pattern recognition defined in section 3.2, we can integrate the concept of t_{ff} into our score design.

The definition of delay index is:

$$d_i = \frac{TT_{v_j} + TT^r_{v_j}}{t_{ff,v_j}} \tag{3}$$

where TT_{v_j} is the travel time of vehicle v_j, $TT^r_{v_j}$ is the free-flow travel time speed and t_{ff,v_j} is t_{ff} of vehicle v_j. For those vehicles with smaller t_{ff}, their delay indexes will contribute more to the total delay index. Thus we increase the vote score of those vehicles and define the total travel time prediction item as

$$T^f_{v_j,t} = \begin{cases} \frac{k_{t_{ff}}}{t_{ff,v_j}}, & \text{if } v_j\text{'s route is known} \\ 1, & \text{if } v_j\text{'s route is unknown} \end{cases} \tag{4}$$

where $k_{t_{ff}}$ is a constant parameter determined by observation and experiments. Results show that adding this term to the equation does increase the performance.

3.3.2 Pressure Item Calculation.

For each vehicle that locates at some upstream lane l_x, the upstream pressure is apparently $\rho^{up}_{v_j,l_x}$; and since its destination is unknown, the downstream lane needs to be predicted. This is provided by the traffic pattern mechanism (3.2), in our implementation about 60% of all the vehicles' destinations can be predicted. Then there are 2 cases:

(1) **The destination can be predicted** according to the roadnet's history. The downstream pressure is then the pressure of the destination, denoted as $\rho^{down}_{v_j,l_y}$.
(2) **The destination can not be predicted**. The downstream pressure is then taken as $\max(\rho^{down}_{v_j,l_{left}}, \rho^{down}_{v_j,l_{mid}})$, where l_{left} is the inner lane of the vehicle's destination road, and l_{mid} the middle lane of that road. This is an approximation on the assumption that this vehicle won't change its lane in the upstream road.

With the upstream and downstream pressures, we define the pressure item as:

$$\mathcal{P}_{i_k,t} = \exp\left(k_1\left(\rho^{up}_{v_j,l_y} - \rho^{down}_{v_j,l_y}\right)\right) \cdot \left(1 - \left(\rho^{down,50}_{v_j,l_y}\right)^{k_2}\right) \tag{5}$$

In experiments, we notice that this design sometimes fails because the downstream lane has just accepted a batch of vehicles, and the upstream part of it is too crowded, while its overall pressure appears to be small. Fig.1 shows a specific case, where obviously $\rho_{l_2 2} < \rho_{l_1 6}$, but lane 16 is better at this moment to be selected.

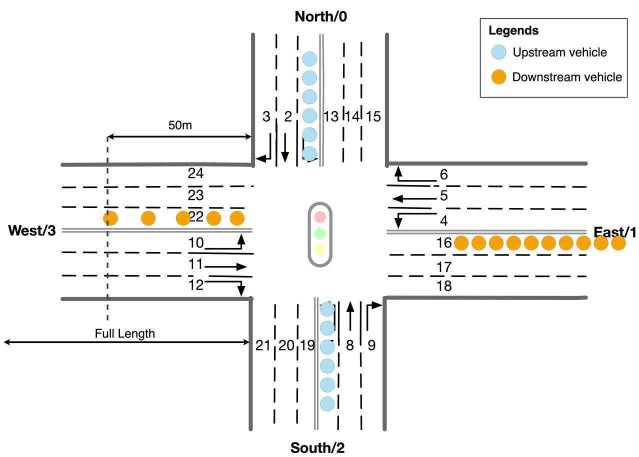


Figure 1: Example of Low ρ_{l_x} but not Suitable for Phase

To deal with such cases, we replace the $\rho_{l_x}^{down}$ in 5 with the pressure of the first 50 meters of the downstream lane, denoted as $\rho_{l_x}^{down,50}$. The number 50 is chosen as the product of the duration of a control cycle (10s) and the propagation speed of the green phase in a vehicle queue (approximately $5m/s$ according to our observation).

3.3.3 Total Preserved Time Item. Vehicles close to the intersection are more likely to reach the intersection sooner, and therefore should have larger weight in voting for the next phase. To make this weight more accurate, the vehicle speed is also introduced. Combining these two pieces of information, we devise the final component as

$$T^m_{v_j,l_x,t} = e^{-k_0 t_{v_j,l_x}} \tag{6}$$

To introduce the design of this component, we first define the following symbols.

For a vehicle reaching the intersection, it can be classified into one of the few situations. First, for a vehicle without constraint from previous vehicles, t_{v_j,l_x} and v_{v_j,l_x} can be calculated using a simple acceleration model. We denote this process as

$$v^{free}_{v_j,l_x}, t^{free}_{v_j,l_x} = \mathcal{H}(d_0, v_0, a) \tag{7}$$

The details of $H(d_0, v_0, a)$ are omitted since it is straight-forward.

For vehicles with constraints from previous vehicles, its speed is constrained by its previous vehicle, and the arrival time can be estimated using

$$t_{v_j,l_x} = \max(t^{free}_{v_j,l_x},\ t_{v_{j-1},l_x} + \Delta t_{follow}) \tag{8}$$

For the arrival speed v_{v_j,l_x}, when the vehicle has moved freely for some time, its speed should be the unconstrained speed $v^{free}_{v_j,l_x}$.

Table 3: Symbols for Calculating Total Preserved Time

Symbol	Definition
v_{v_j,l_x}	The actual speed of the j^{th} vehicle on lane l_x reaching the intersection
t_{v_j,l_x}	The actual time of the j^{th} vehicle on lane l_x reaching the intersection
$v^{free}_{v_j,l_x}$	The actual speed of the j^{th} vehicle on lane l_x reaching the intersection, without constraint of before vehicles
$t^{free}_{v_j,l_x}$	The actual time of the j^{th} vehicle on lane l_x reaching the intersection, without constraint of before vehicles
$v^{up}_{lv_j}$	Speed limit of the upstream lane for vehicle v_j
$v^{down}_{lv_j}$	Speed limit of the downstream lane for vehicle v_j
Δt_{follow}	Vehicle follow time interval by observation

Otherwise, we assume the speed difference of a vehicle with its previous vehicle is $1m/s^2$. Enforcing the speed limit of both upstream and downstream lanes, we have:

$$v_{v_j,l_x} = \begin{cases} v^{free}_{v_j,l_x}, & t_{v_j,l_x} = t^{free}_{v_j,l_x} \\ min(v^{up}_{lv_j}, v^{down}_{lv_j}, v_{v_{j-1},l_x} + 1), & t_{v_j,l_x} = t_{v_{j-1},l_x} + \Delta t_{follow} \end{cases} \tag{9}$$

3.4 Blacklist Mechanism

In some scenarios, the downstream lane to some intersection i_k may be full due to rush hours or accidents. The design of pressure item (3.3.2) may fail under such circumstances and lead to a phase where no vehicle can move. To deal with this problem, we devise a dynamic blacklist mechanism. An additional parameter $M_{v_j,t}$ is appended to (2), so it becomes:

$$s_{v_j,i_k,t} = T^f_{v_j,t} \cdot \mathcal{P}_{i_k,t} \cdot T^m_{v_j,i_k,t} \cdot M_{v_j,t} \tag{10}$$

This additional parameter depends on whether the lane vehicle v_j is located at has been blacklisted. In our implementation, if the first vehicle on any lane is detected not moving during a green phase, this lane is considered to have an anomaly.

At each time step lanes with anomalies will be checked for reasoning and then temporarily added to a blacklist. For the scenario in Fig.2, lane 5 will be considered to have an anomaly and will be temporarily blocked until the pressure at its downstream lane is released, and lane 8 will therefore be considered. There are two kinds of blacklist reasoning in the current version of the implementation. Table 4 lists their corresponding details.

At each time step for each intersection i_k, our agent first checks whether any upstream lane at i_k and on the blacklist satisfies its condition to leave. Then it rechecks if any lane at i_k should be added to the blacklist. Note that conditions to leave are usually designed stronger than the conditions to enter except for unknown reasons, i.e. $\neg C_l \Rightarrow C_e$. Therefore, $\mathcal{T}_l < \mathcal{T}_e$.

Table 4: Blacklist Implementations

Blacklist Reason	Conditions to enter C_e	Conditions to leave C_l	$m^R_{v_j,t}$
High Downstream Pressure	$max(\rho_{l_x}) > \mathcal{T}_e$	$max(\rho_{l_x}) \leq \mathcal{T}_l$	0
Unknown	Unreasonable anomaly	After preset time steps	$\mathcal{B}^{T_0-t}$

Table 5: Parameters of the Final Algorithm

Parameters	$\mathcal{T}_e$	$\mathcal{T}_l$	k_{tff}	k_1	k_2	k_0	Δt_{follow}
Values	0.64	0.30	700	0.30	0.85	0.27	1

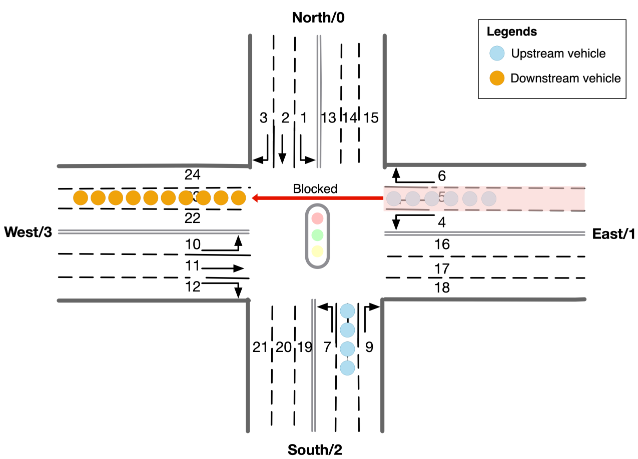


Figure 2: High Downstream Pressure Scenario

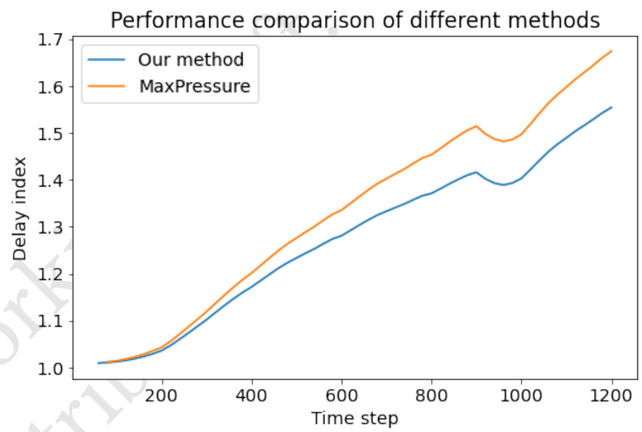


Figure 3: Performance comparison

When voting, each vehicle's vote score is reduced according to the measure factor $M_{v_k,t}$ of the lane it is located at, where $M_{v_k,t} = \prod m^R_{v_k,t}$ for all reasons. If the lane is not present in some blacklist, $m^R_{v_k,t} = 1$ for that reason R.

According to our experiment, for those lanes with unreasonable anomalies, it is best to set an exponentially decaying measure factor, so that the focus on these lanes will be temporarily transferred until they recover from the anomaly. Note that our design of blacklist is highly expandable, other kinds of blacklist reasons can be added to improve the adaptability.

4 CHOICE OF PARAMETERS

With the method introduced in Section 3, we further finetune the parameters of the algorithm to make it fit on the roadnet provided. The tuned parameters are listed in Table 5.

5 EXPERIMENTS

5.1 Comparison of Performance

To test the effectiveness of our approach, the algorithm is run on the default flow provided by the CityBrainChallenge Team [2]. We use the delay index as comparison, figure 3 shows that our approach outperforms MaxPressure at almost every time step. From Fig.3 we can see that from time step 200, our method performs significantly better than the MaxPressure method. Table 6 shows evaluating the two methods using 1.4 as delay index threshold and 20 seconds as evaluation interval.

Table 6: Performance Comparison with Delay Index Threshold=1.4

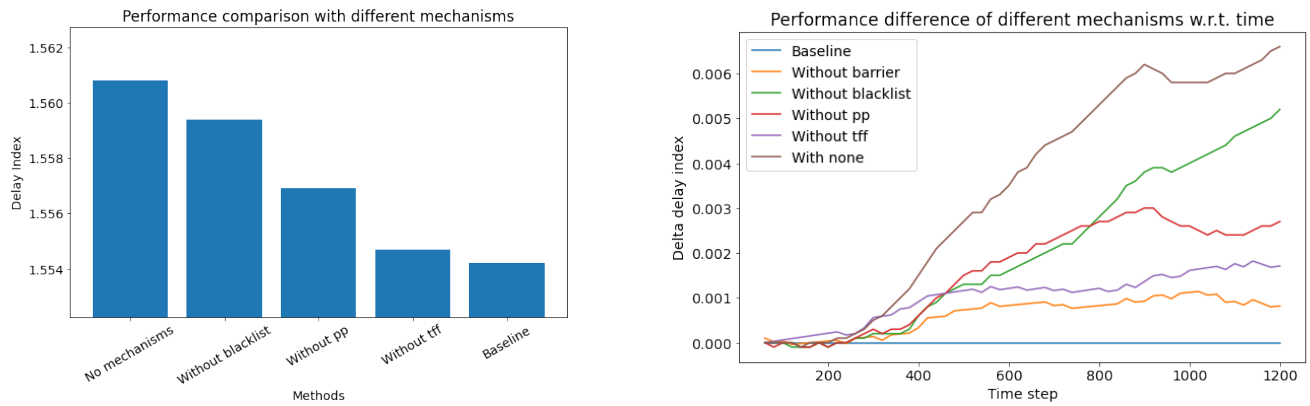
Method	Time step	Served vehicles
Our method	860	392384
MaxPressure	700	349504

5.2 Effectiveness of mechanisms

To prove the effectiveness of the above-listed mechanisms, we test our vote-based algorithm with each mechanism turned off, using the same settings in 5.1. The final delay index of time step 1200 is compared in Fig.4, where the baseline run is the final submission of our team.

6 CONCLUSION

In this paper, we devise a vote-based traffic coordination algorithm to relieve the traffic congestion problem. Our method combines the calculation of a few mechanisms, and by focusing on the minimum



(a) Performance of Mechanisms at 1200s **(b) Delay Index Increasing Rate of Mechanisms**

Figure 4: Performance Difference of Mechanisms

unit of a roadnet, which is vehicles, our method selects the next phase for each intersection in a given roadnet.

Experiments show that our algorithm has outperformed the *MaxPressure* method, and by a series of experiments we also show that each of our unique mechanisms has its effect. In KDD Cup 2021 City Brain Challenge, our method can serve 360637 vehicles in the final round, and achieved a delay index of 1.402714 with a runtime limit of 20 minutes, which outperforms other competitors.

We believe that by using a more deliberate design of traffic pattern recognition and the enrichment of our blacklist mechanism, this algorithm can be further improved. Dynamic parameter adjustment can also increase the adaptability of this algorithm to different kinds of vehicle flows. What's more, from 5.2 we may conclude that, the more information about each individual vehicle available to support our mechanisms, the better this algorithm can perform. With the data collection technique becoming more and more advanced in contemporary cities, our method has its unique advantage towards city-scale challenges.

ACKNOWLEDGMENTS

To KDD Cup Committee, for the wonderful traffic simulation engine and the computing resource.

REFERENCES

[1] Carlos T. Calafate, David Soler, Juan-Carlos Cano, and Pietro Manzoni. 2015. Traffic Management as a Service: The Traffic Flow Pattern Classification Problem. *Mathematical Problems in Engineering* 2015 (20 Oct 2015), 716598. https://doi.org/10.1155/2015/716598

[2] Z.Y Liu, Chang Liu, and Kan Wu. 2021. KDDCup2021-CityBrainChallenge-starter-kit. https://github.com/CityBrainChallenge/KDDCup2021-CityBrainChallenge-starter-kit.

[3] Pravin Varaiya. 2013. The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. *Advances in Dynamic Network Modeling in Complex Transportation Systems* (02 2013), 27–66. https://doi.org/10.1007/978-1-4614-6243-9_2