Parallel Management for Traffic Signal Control

Y.-F. Zhao, Q.-J. Kong, H. Gao, F.-H. Zhu, and F.-Y. Wang, *Fellow, IEEE*

***Abstract*—With the rapid growth of the number of urban vehicles, it will be not advisable to alleviate traffic congestion by changing the traffic facilities only. And the traditional control strategies for single intersection or regional multiple intersections have been confirmed to have some effect in the past few decades, but still need to be improved. Based on ACP (Artificial societies, Computational experiments, Parallel execution) idea, we firstly proposed the concept of "event agent" in this paper, which refers to the ratings that traffic states give corresponding timing plans. Based on event agent, we used computational methods to establish a Parallel transportation Management Systems (PtMS), which was a self-completing system. In the system plenty of artificial events were generated, and some of them can not only simulate the actual traffic events, but also be substitutes for the actual events. Then through the parallel execution between actual and artificial events, the system recommends the most suitable timing plans to the current traffic state. Different from traditional control strategies, event agent based PtMS takes results as an orientation according to the idea of data-driven, which is more adaptive to the characteristics of transportation systems. For ensuring the validity and accuracy of experiments, our related data are all based on the famous traffic micro-simulation software Paramics. Furthermore, we compared our method with the classic Webster method, and experiments achieved good results.**

***Keywords*—ACP; data-driven; event agents; Intelligent Transportation Systems; parallel transportation management system**

I. Introduction

Due to the rapid development of social economic, vehicle number raced up and up to an unprecedented level. And various problems ensued, such as the deterioration in the quality of environment, the decrease of travel safety and so on. Researchers noted that it would be not advisable to alleviate traffic congestion by changing the traffic facilities only. Then the Intelligent Transportation Systems (ITS) [1-13] was initialized, developed and developed, and indeed it made a real difference in terms of improving traffic conditions in the past few decades.

In the reality, ITS was developed in a systematic direction with the emergence of several generations of traffic management systems, such as TRANSYT, SCOOT, SCATS, and so on. These traffic management systems not only contains the methods or models for single intersections, but also the information fusion and coordination strategies for the intersection, the traffic flow forecasting and inducement functions etc. [5]

In the theory, the initial study of ITS could date back to classic Webster method proposed in the mid-20th century [1], which alleviated traffic congestion from the perspective of signal control. Because of its effectiveness, it is often used as a benchmark by other traffic signal control methods. And in order to describe the characteristics of transportation systems, macroscopic and mesoscopic traffic models were born in the early research work. Among them, the Lighthill Whitham Richards (LWR) [2, 3] method and Lattice Boltzman Method (LBM) [4] method are typical macro- and meso-models respectively, which are based on hydromechanics or statistical physics. These two models may have a good performance on describing the overall properties of traffic flow, but both of them lack the flexibility to describe the details of the system. Therefore, a lot of microscopic traffic models and simulation software came into the world. And as the computing technologies are developed, the Multi-Agent Systems (MAS) [5-8] become more and more popular now. Based on MAS, an Artificial Transportation Systems (ATS) can grow up in a bottom up way [6], with drivers, vehicles, roads, traffic lights being modeled as autonomous, collaborative and reactive agents. An ideal ATS even could be a substitute for actual transportation system, which is better for overcoming the problem that the methods and models cannot be repetitive experimentalized in reality [6].

Additionally, because transportation systems refer to the complexity issues of both engineering and social dimensions, they have two essential characteristics: 1) inseparability. Inseparability means that, the global behaviors of transportation systems cannot be determined or explained by independent analysis of their component parts. Instead, the system as a whole determines how their parts behave. 2) unpredictability. Intrinsically, with limited resources, the global behaviors of transportation systems cannot be determined or explained in advance at a large scope. The complexity of transportation systems not only embodies in the vehicles and pedestrians, roads, signal lights, but also includes the weather environment, legal policy, the influence of social economy and ecological resources. These components cannot be ignored when analyzing, because there are mutual influence between them, and it is difficult to know how they influence each other in detail. Traditional methods are trying to include these components as many as possible when modeling. However, they are based on some assumptions, which cannot reflect the real accurate relationship of these components and their effect on the final global behaviors, let alone make accurate predictions in advance. So this leads to that ITS is not really "intelligent" in reality [7].

Inspired by ACP-based parallel control and management systems [6-13], we proposed the concept of "event agent"

This work was supported in part by the National Natural Science Foundation of China Program under Grant 71232006, 61233001, 61104160, 61203079.

The authors are with the Qingdao Academy of Intelligent Industries, Qingdao, Shandong, 266109, China (e-mail: {yifei.zhao, qingjie.kong, hang.gao, fenghua.zhu}@ia.ac.cn, feiyue@gmail.com).

from the perspective of traffic signal control, and the goal is trying to mine the most suitable timing plans for the current traffic condition through parallel execution between "artificial events" and "actual events". The event agent here is mainly represented by the "rating" that traffic condition gives timing plan. What's more, "rating", such as delay time, stop time, traffic flow and so on, can reflect the extent of match between the traffic condition and timing plan. We research and explore based on the following considerations: 1) data-driven. As mentioned in [7], parallel control and management is a data-driven approach, and it is the direction of traffic problem research. In future traffic area will not lack of data, for detecting and information fusion technology [14, 15] is more and more mature. 2) global perspective and result-oriented. Based on the idea of big data in traffic and model-free adaptive control, we block out the relationship between the components and their impacts on the global behavior which are difficult to be quantized through data mining. And the system could improve itself according to the feedback of "event" results. To our best knowledge, this is a new way for applying parallel control and management systems for ITS, and it is the most important contribution of this paper.

Section II will give the system architecture and operation process, and section III will model for the system. Experiments will be given in section IV, while section V is the conclusion and future work. Some algorithms used in this paper will be added in the appendix.

II. System Architectures and Operation Processes

"Event agents" refer to the "ratings" of traffic states for timing plans. They may come from historical data, or be generated by some classic traffic signal control algorithms and simulation software. Although they are "virtual" for the current traffic states, they could be used as an alternative version of the actual traffic events. What's more, their results will be more reliable and robust. Therefore, our purpose is to build a Parallel transportation Management System (PtMS) based on event agents.

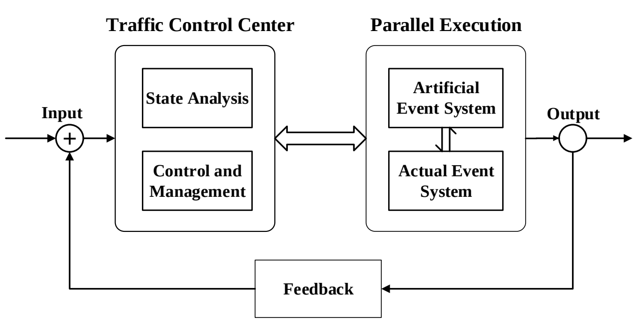

Figure 1. Manuscript of PtMS based on event agents

A. System Architectures

The system consists of two major parts: traffic control center and parallel execution. In the traffic control center, there are two modules: state analysis and control and management. And in the parallel execution, artificial event system and actual event system are included. Among them, some prediction and evaluation algorithms are stored in the control and management module, which are used for computational experiments of event agents. There are also some traditional control strategies in control and management module, in case that computational experiments on the artificial event system does not work well. At the same time, artificial event system holds many artificial events like (s,t,r), which means traffic state s gives a rating r to the timing plan t, while actual event system owns the dynamic actual events. The input of PtMS often refers to the detected traffic states, such as the density of traffic flow, delay time of vehicles, and so on. While the typical output of the system involves the timing plans which are thought to be suitable for the current traffic state.

B. Operation Processes

When receiving the input signal, namely the current traffic state, the state analysis module should carry on a preprocessing on the detected data for a further application. We need a standard format to describe the current traffic state, which is an incomplete event in the actual event system. Here "incomplete" means that the actual event only has s, without t and r. Then control and management module will analyze and mine the relation between the artificial and actual events, and computational experiments will be performed to predict the ratings of the current traffic state for the timing plans that never to be used before. In other words, new artificial events are built for the incomplete actual events. Here two points should be pointed out: 1) normally, some timing plans have been used by the current traffic state before, 2) more timing plans are never used by the current traffic state and the artificial traffic state. That is to say there are incomplete event agents in artificial event system. So some evaluations are required for us to choose the best data mining and machine learning algorithms, which are used to recommend timing plans to the artificial events. According to the prediction results and existing complete artificial events, the most suitable timing plans will be chosen for the current actual event, namely control and manage the actual and artificial systems through parallel execution. Here is a case, namely there is no appropriate timing plans in artificial event system that can improve the actual event significantly. If so, some traditional control strategies are required to help generate new artificial events. Then the chosen timing plans will be used in reality, and the actual result, like delay, flow, etc., will be recorded as a feedback. Afterwards state analysis module will receive the feedback, and transform it into standard form, like (s,t,r), with the help of computational experiments. Here are two cases. One is the timing plan that is entirely new. The new timing plan is generated by traditional control strategies. The other is the timing plan which is new for the current traffic state. In the first case, we directly enrich the artificial event system according to the feedback information; in the second case, we should compare the predicting value with the real value, then adjust the algorithms in the control and management module, and update the artificial event system according to the feedback information.

III. Modeling and Methodology

A. Artificial Events Generation

According to the system described above, we know that the artificial event system includes many pairs of traffic states and timing plans. Every traffic state will give different ratings to some timing plans while others' are missing. Our purpose is to predict the missing ratings according to the existing ones by adopting the idea of computational experiments. Due to the limits in reality, the data about existing ratings often comes from the simulations. Here we apply the famous traffic micro-simulation software Paramics as our experiment platform.

First of all, we should establish a road network. Here we perform an exploratory experiment on a single intersection shown in Figure 2.

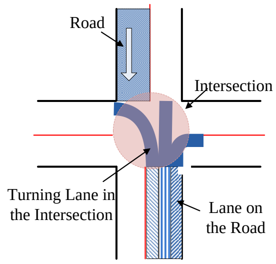

Figure 2. An intersection

For every road there are usually several lanes. The vehicles in a lane can be modeled as a queue with a fixed capacity. Once the capacity is exhausted no more vehicles can enter the queue. The right most and left most are dedicated lanes for turning. We assume that the vehicles follow given paths when turning.

The first element of the artificial event agent (s,t,r) is the traffic state. In Paramics, some ODs are set up to represent different traffic states based on the single intersection shown in Figure 2. For reflecting the various traffic states from unimpeded to congested, the number of vehicles included in OD should be set from few to excessive.

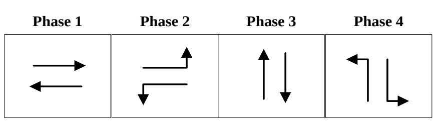

Figure 3. Phase sequence of the traffic lights

Phase sequence is the order of the different phases. For an intersection, there are several phases shown in Figure 3. We assume that there are $M = 4$ phases in the intersection shown in Figure 2, and there are green time and all-red time in every phase. Green time is the core of signal control that need us to calculate, while all-red time is set to have a fixed value AR = 3s. The settings of M phases and AR time constitute a timing plan, which is the second element of artificial event agent (s,t,r).

The performance indexes, such as delay, stop time, flow, etc., are regarded as ratings that traffic states give timing plans, because both indexes and ratings could reflect the suitable degree between traffic states and timing plans. For the ratings, the higher the better, while for some indexes like delay and stop timing, the lower the better. Therefore preprocess is required before further applications. Then we get the third element ratings.

Therefore, artificial events occur, and the agents of artificial events (s,t,r) appear. And the performance indexes mentioned above are used to describe the artificial event agents (s,t,r). We hope that the system we proposed not only simulates the actual traffic states, but also could be a substitute for the actual traffic system.

B. *Parallel Execution*

After artificial event system is built, we will get some event agents with the standard form (s_i, t_j, r_{ij}). Here, (s_i, t_j, r_{ij}) means that traffic state s_i gives a rating r_{ij} to the timing plan t_j. Among them some r_{ij} are unknown, because it is impossible for every traffic state to traverse all kinds of timing plans in reality. So our task is to predict the ratings of s_i for the timing plans when a traffic state s_i occurs, then recommend a most suitable timing plan t_j for s_i through parallel execution.

Through a simple mathematics manipulation, the standard form (s_i, t_j, r_{ij}) can be transformed into a matrix, in which the rows correspond to the timing plans, the columns represent the traffic states, and the matrix elements are filled with the corresponding ratings, as shown in TABLE I. But what we should note is that the matrix is sparse, so from the perspective of mathematics, the essence of computational experiments in the system is a matrix filling. Slope-one algorithm is a good choice to finish this task, as it is a rating-based collaborative filtering algorithm in recommendation systems. It was proposed by Daniel Lemire and Anna Maclachlan in 2005 [16], and was popular for its following characteristics:

1) esay to implement and maintain
2) updateable on the fly
3) efficient at query time
4) expect little from first visitors
5) accurate within reason

TABLE I. TRAFFIC STATE-TIMING PLAN RATING MATRIX

Rating	t_1	$\cdots$	t_i	$\cdots$	t_n
s_1	r_{11}	$\cdots$	r_{1j}	$\cdots$	r_{1n}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
s_i	r_{i1}	$\cdots$	r_{ij}	$\cdots$	r_{in}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
s_m	r_{m1}	$\cdots$	r_{mj}	$\cdots$	r_{mn}

These characteristics are applicable to solve the problems in the traffic systems. Various traffic states will happen every moment, and newest timing plans need to be immediate responded. So the artificial event system should be updated in time according to the dynamic changes of traffic conditions. Therefore, we select slope-one algorithm.

Based on the generated data and matrix transformation mentioned above, the system will work as the following processes:

Step 1: when receiving the detected data about the current traffic state, state analysis module should normalize the data and transform them into the standard format of event agents like (s_i, t_j, r_{ij}) (now t_j and r_{ij} are unknown);

Step 2: perform computational experiments for the current traffic state s_i to predict the ratings r_{ij} of s_i for the timing plans t_j in the artificial event system, where s_i has never been used before. Then many new event agents about s_i would be generated;

Step 3: control and management through parallel execution of artificial and actual event system, namely, recommend the most suitable timing plan t_j to the current traffic state s_i, by comparing the predicting ratings of s_i with the existing ratings of timing plans used by s_i before in the artificial event system;

Step 4: if the ratings of the timing plan recommended for s_i is lower than a given threshold, the traditional control strategies are required to generate more appropriate timing plan for s_i. If not, execute the result in step 3 in reality directly;

Step 5: Then the feedback information, such as delay time, stop time, flow, and so on, will be given to the state analysis module;

Step 6: the state analysis module will transmit the feedback information to the event agent with standard form, to update or further enrich the artificial event system.

IV. Computational Experiments

A. Dataset Description

The related experiment data is generated by microscopic traffic simulation software Paramics, and the experiment road network is shown in Figure 2. In Figure 2, 4 roads all have two links, and every link has three lanes. Other related Paramics simulation parameter settings are shown in TABLE II.

TABLE II. Paramics Parameters

Duration (hour)	1
Time step(s)	0.5
Demand factor (%)	100
Section length (m)	500
Orientation	Right Hand Drive
Units	Metric Units

When performing simulation, we collect the data from the 10th minute to the end of the simulation, because the first ten minutes are used as road initialization time. In the simulation, we set 7 ODs that we think can represent 7 different traffic states, therefore unimpeded and congested traffic states can be displayed. And 22 signal timing plans are included. Then we can obtain 154 groups of delay values. So 154 event agents are generated by some simple mathematical processing. In reality, it is impossible that every intersection condition has used every kind of timing plans, so we randomly choose 70 event agents from the 154 event agents above.

Furthermore, we get 7 optimal signal control plans according to Webster method. When using Webster method, the related parameter settings are shown in TABLE III.

TABLE III. Webster Parameters

AR (s)	L (s)	S (vehs/s)	minimum green time (s)	maximum green time (s)	optimum cycle (s)
3	5	2000	7	41	C_0

B. Algorithms Selected and Analysis

Based on the dataset generated above, we tried two different implementations for prediction: slope-one and weighted slope-one. When implementing, traffic states *s* are correspond to rows, timing plans *t* are correspond to columns, and the existing elements are the “ratings” which s have given to *t*. So the task is to fill the rating matrix through machine learning framework, namely predicting the matching degree between the traffic states and timing plans which have not been used by the traffic states. (See appendix for more detailed information about slope-one and weighted slope-one.) We tested them on 14 pairs of traffic states and timing plans, and compared their predicting ratings with the existing ratings in artificial event system, the result is shown in Figure 4.

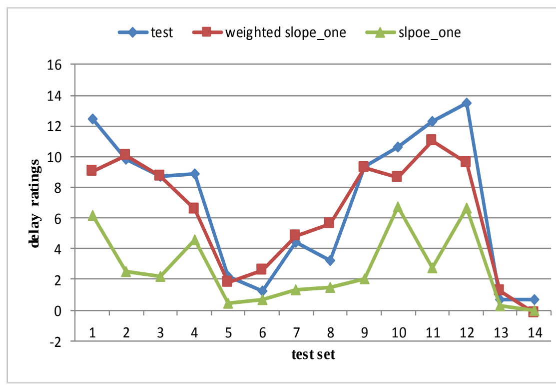

Figure 4. Delay Ratings of Test Set

Figure 4 shows that the ratings of weighted slope-one are more close to the test ratings. Especially in state 2, state 3, state 5, state 7 and state 9, the predictions are almost entirely correct. And more rigorously, from the mathematics perspective, the RMSE (root-mean-square error) between the slope-one and the test ratings is 5.1848, while that between the weighted slope-one and testing rating is 1.8076, which is significant beyond the slope-one. RMSE is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. Obviously we should select weighted slope-one in the experiment for its superiority.

C. Comparison with Webster

Through the computational experiments based on weighted slope-one algorithm, the best signal timing plans of seven traffic states in the artificial event system can be found. We also obtain 7 timing plans generated by Webster method, as shown in TABLE IV. In TABLE IV, the elements are the cycles of timing plans.

TABLE IV. TIMING PLAN GENERATION

	OD1	OD2	OD3	OD4	OD5	OD6	OD7
PtMS	48	88	148	120	72	40	120
Webster	72	104	176	140	88	64	176

Then we perform experiments on Paramics to get the delay time. The result is shown in Figure 5.

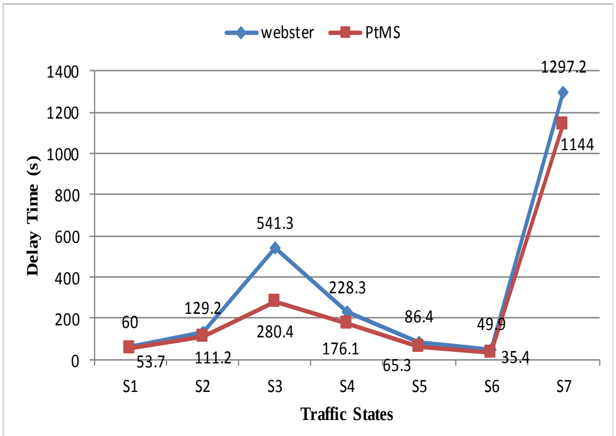

Figure 5. Delay time comparison

From Figure 5, we can find that the signal timing configurations recommended by PtMS, which are based on the event agents, outperforms that given by Webster method. It means that PtMS are more conducive to improve road conditions. Furthermore, the event agents in artificial systems have the ability of self-learning with the help of control and management module, i.e. the event agents could learn and revise the ratings through prediction and feedbacks; the system has the ability of self-completing by taking use of feedback information. Nevertheless, there are still much room for improvement in practice, especially in terms of artificial event agent generation. If the event agents are not enough, the actual computational experiments cannot be performed. And no matter how strong the learning and predicting algorithms are, there is no suitable event that can represent the reality, which could cause some unsatisfactory result, such as state 3 and state 7 in Figure 5. Now traditional control strategies are required to help build artificial event systems. So the abundant artificial events is the core idea of artificial transportation system. But fortunately, even the artificial event system is not abundant enough, the result will not be worse than historical results. So the experiments with the idea of computational experiments proved that, PtMS based on the event agents is an effective way to solve the traffic problems.

V. CONCLUSION & FUTURE WORK

To our best knowledge, this is a new way for application of parallel control and management for ITS. We proposed the concept of "event agent", and made a tentative experiment on applying it into parallel transportation management systems. And the experiments got good results, which proved the feasibility of "event agent".

Our future work should focus on the following aspects: 1) find a new way to generate more event agents which are more representative, 2) select better algorithms for control and management module.

APPENDIX

A. *Slope-one*

Slope-one is a family of algorithms used for collaborative filtering, introduced in a 2005 paper by Daniel Lemire and Anna Maclachlan [16]. Arguably, it is the simplest form of non-trivial item-based collaborative filtering based on ratings. Their simplicity makes it especially easy to implement them efficiently while their accuracy is often on par with more complicated and computationally expensive algorithms [16, 17]. They have also been used as building blocks to improve other algorithms [18, 19]. They are part of major open-source libraries such as Apache Mahout and Easyrec.

Next is an introduction of slope-one from mathematical perspective.

Essentially, instead of using linear regression from one item's ratings to another item's ratings ($f(\mathrm{x}) = a\,\mathrm{x} + \mathrm{b}$), it uses a simpler form of regression with a single free parameter ($f(\mathrm{x}) = \mathrm{x} + \mathrm{b}$). The free parameter is then simply the average difference between the two items' ratings. It was shown to be much more accurate than linear regression in some instances, and it takes half the storage or less.

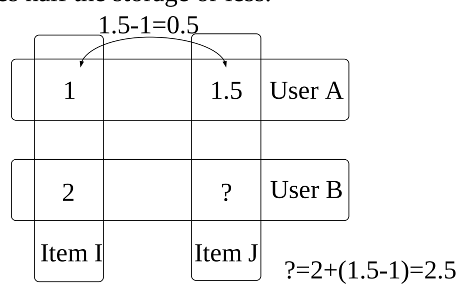

Figure 6. Basis of slope-one schemes: User *A*'s ratings of two items and User *B*'s rating of a common item is used to predict User *B*'s unknown rating.

Formally, given two evaluation arrays v_i and w_i with $i = 1, 2, \cdots, n$, we search for the best predictor of the form $f(x) = x + b$ to predict w from v by minimizing $\sum_i (v_i + b - w_i)^2$, then we can get $b = \frac{\sum_i w_i - v_i}{n}$. In other words, the constant b must be chosen to be the average difference between the two arrays. And, after getting *b*, for a new v_{new}, we can get its prediction value through $w_{new} = b + v_{new}$.

With the help of the pictorial diagram above, we define the average deviation of item *i* with respect to item *j* as:

$$dev_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{card\left(S_{j,i}(\chi)\right)}$$

Here $S_{j,i}()$ represents the set of users which rated both for item *i* and item *j*, and $card()$ represents the number of the set.

Then given that $dev_{j,i}+u_i$ is a prediction for u_j given u_i, a reasonable predictor might be the average of all such predictions

$$p(u)_j=\frac{1}{card(R_j)}\sum_{i\in R_j}(dev_{j,i}+u_i)$$

Where $R_j=\{i|i\in S(u), i\neq j, card(S_{j,i}(\chi))>0\}$ is the set of all relevant items.

If the data set is dense enough, we may use an approximation $\bar{u}=\sum_{i\in S(u)}\frac{u_i}{card(S(u))}\simeq\sum_{i\in R_j}\frac{u_i}{card(R_j)}$, then we can simplify the prediction formula for the slope-one scheme to

$$P^{S1}(u)_j=\bar{u}+\frac{1}{card(R_j)}\sum_{i\in R_j}dev_{j,i}$$

B. Weight Slope-one

One of the drawbacks of slope-one is that the number of ratings observed is not taken into consideration. Intuitively, to predict user *A*'s rating of item *L* given user *A*'s rating of items *J* and *K*, if 2000 users rated the pair of item *J* and *L* whereas only 20 users rated the pair of items *K* and *L*, then user *A*'s rating of item *J* is likely to be a far better predictor for item *L* than user *A*'s rating of item *K* is. Thus, we define the weighted slope-one prediction as the following weighted average

$$P^{wS1}(u)_j=\frac{\sum_{i\in S(u)-\{j\}}(dev_{j,i}+u_i)c_{j,i}}{\sum_{i\in S(u)-\{j\}}c_{j,i}}$$

Where $c_{j,i}=card(S_{j,i}(\chi))$.

C. Webster

Webster signal control is the most classical approach in ITS, and it is based on the vehicles' delay time when travelling through the intersection. Obviously, the target of Webster is to minimize the total delay time of the vehicles.

In Webster signal control, the best cycle time is given by

$$C_0=\frac{1.5L+5}{1-Y}$$

where C_0 is the best signal cycle with the unit seconds. The total loss time L can be described by $L=nl+AR$, among them, l is the loss time of every phase, n is the phase number, and AR stands for the all-red time in a cycle. The traffic flow rate Y comes from $Y=\sum_{i=1}^{n}y_i$.

Here we introduce the critical lane which refers to the lane having the largest traffic flow in each signal phase. It's usually assumed that traffic flow rate of critical lane is equal to the ratio of the traffic flow to the saturation flow.

References

[1] F. V. Webster, "Traffic signal settings," *Great Britain Department of Scientific A*, 1958.

[2] M. J. Lighthill, and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317-345, 1995.

[3] P. I. Richards, "Shock waves on the highway," *Operations research* vol. 4, no. 1, pp. 42-51, 1956.

[4] S. Chen, and G. D. Doolen, "Lattice Boltzmann method for fluid flows." *Annual review of fluid mechanics,* vol. 30, no. 1, pp. 329-364, 1998.

[5] C. Chen, F.-H. Zhu, and Y.-F. Ai, "A survey of urban traffic signal control for agent recommendation system," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, pp. 327-333, 2012.

[6] F.-Y. Wang, and S.-M. Tang, "Concepts and frameworks of artificial transportation systems," *Complex Systems and Complexity Science*, vol. 1, no. 2, pp. 52-59, 2004 (in Chinese).

[7] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630-638, Sept. 2010.

[8] F.-Y. Wang, "Agent-based control for networked traffic management systems," *IEEE Intelligent Systems*, vol. 20, no. 5, pp. 92-96, Sept.-Oct. 2005.

[9] F.-Y. Wang, "Artificial societies, computational experiments, and parallel systems: An investigation on computational theory of complex social economic systems," *Complex Syst. Complexity Sci.*, vol. 1, no. 4, pp. 25-35, 2004.

[10] F.-Y. Wang, "Parallel system methods for management and control of complex systems," *Control Decision,* vol. 19, no. 5, pp. 485-489, 2004.

[11] F.-Y. Wang and S. Tang, "Artificial societies for integrated and sustainable development of metropolitan systems," *IEEE Intell. Syst.*, vol. 19, no. 4, pp. 82–87, Jul./Aug. 2004.

[12] F.-Y. Wang, "Toward a revolution in transportation operations: AI for complex systems," *IEEE Intell. Syst.*, vol. 23, no. 6, pp. 8–13, Nov./Dec. 2008.

[13] F. Zhu, et al., "Computational traffic experiments based on artificial transportation systems: an application of ACP approach," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 189-198, 2013.

[14] Q.-J. Kong, et al., "An approach to urban traffic state estimation by fusing multisource information," *IEEE Intell. Trans. Syst.*, vol. 10, no. 3, pp. 499-511, 2009.

[15] Q.-J. Kong, et al., "A fusion-based system for road-network traffic state surveillance: A case study of shanghai," *IEEE Intell. Trans. Syst. Magazine,* vol. 1, no. 1, pp. 37-42, 2009.

[16] D. Lemire, and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," *SDM, SIAM,* pp. 1-5, 2005.

[17] F. Cacheda, et al., "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web (TWEB)*, vol. 5, no.1, pp. 2, 2011.

[18] P. Wang, and H. Ye, "A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering," *IIS. IEEE Int Conf. Industrial and Information Systems*, pp. 152-154, 2009.

[19] F. Cacheda, et al., "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web (TWEB)*, vol. 5, no.1, pp. 2, 2011.