

Cooperative Traffic Light Control Based on Semi-real-time Processing

Qin Zhu, Chao Peng, Jingmin Shi, Pengfei Duan, and Yu Bao

Shanghai Key Lab of Trustworthy Computing, Software Engineering Institute, East China Normal University, China
Email: {51121500045, cpeng}@ecnu.cn

Mengjun Xie

University of Arkansas at Little Rock, Little Rock, AR 72204, USA
mxxie@ualr.edu

Abstract—In this paper, we investigate cooperative traffic light control for multiple intersections based on semi-real-time processing. As urban traffic congestion problems is increasingly aggravating, existing traffic light controllers can no longer satisfy the rising demands for efficiently easing traffic pressure. In this paper, we propose an adaptive traffic light control algorithm based on semi-real-time processing and cooperation among traffic light controllers. We set fixed phase duration between traffic light phases in advance. For each intersection, the controller determines the next traffic light phase by prediction of traffic situation for the next period. To evaluate our proposed algorithm, we construct four traffic scenarios and run simulations with combination of NS3 and SUMO. The simulation results demonstrate that our proposed algorithm is effective and practical in different scenarios; it can reduce traffic load and average waiting time of vehicles, as well as enhance traffic throughput of intersections.

Index Terms—traffic light, traffic simulation, cooperative control, SUMO, NS3

I. INTRODUCTION

Recent years have witnessed worldwide rapid development in automobile industries and growing numbers of vehicles, along with the spread of population from countryside to cities. Consequently, the thorny issue of traffic congestion has bothered more and more people living in cities, particularly in those megalopolises with huge population. Traffic congestions will bring about substantial time losses for people stuck in jams. Meanwhile, low speed vehicles will lead to increased gasoline consumption [1], thus generate increased exhaust emission, aggravating environmental pollution. To reduce traffic congestions in cities, we should start with urban main roads which bear most traffic flow. Traffic situation will be significantly improved if we are

able to enhance traffic throughput of intersections and reduce traffic load and stop rate of vehicles on main roads.

It is with the above-mentioned awareness that a lot of related researches have been studied during recent decades. One of the primary directions of such researches is traffic light control [2]-[5].

According to USA National Electrical Manufacturers Association (NEMA), most present traffic light control can be classified into two types: fixed-time control and traffic-response control [6]. Unfortunately, both control methods have their own limitations (we will further discuss this in related works).

To avoid the deficiency of fixed-time control and traffic-response control, we combine the ideas of two control methods together. In this paper we present an adaptive traffic light control algorithm based on semi-real-time processing and cooperation among traffic light controllers. In reality, the traffic situation of urban main roads is extremely complicated. It is almost impossible to instantly calculate the traffic situation for the entire traffic network due to its highly dynamic environment. On the other hand, it is also not necessary to perform real-time processing since the traffic control cycle is usually not very short, so in our approach we set fixed phase duration between traffic light phases in advance. Each intersection has a traffic light controller and each controller decides the next traffic light phase by predicting the traffic situation of its own intersection in the next period. As the traffic situation for the next period will be effected by coming vehicles from neighboring intersections, we make every controller cooperate with its neighboring controllers.

The rest of this paper is organized as follows. In the next section, we briefly discuss related works on existing approaches of traffic light control. Section 3 models the traffic network. We then describe the details of our cooperative traffic light control algorithm in Section 4. In Section 5 we run simulations with NS3 and SUMO under four scenarios, and evaluate our algorithm. Finally, Section 6 concludes this paper.

Manuscript received September 20, 2014; revised February 11, 2015.

This research is supported by the Shanghai Municipal Natural Science Foundation (14ZR1412400), the Innovation Program of Shanghai Municipal Education Commission, the Natural Science Foundation of China under Grant No.91118008, the Shanghai Knowledge Service Platform Project (No.ZF1213). Corresponding author is Chao Peng.

II. RELATED WORKS

As we have mentioned in the previous section, there are two main categories of present traffic light controls: fixed-time control and traffic-response control.

Fixed-time control defines traffic light phase and the cycle time in advance. This method usually defines the phase and the cycle time based on historical traffic data. 24 hours of one day will be divided into several periods and different schemes of traffic light control will be operated during different periods. The main problem of such control is that the performance can be quite poor under the circumstance of unstable traffic flow. However, urban traffic flow is fast-changing in real world, and the instant traffic situation might be largely different from the historical data, with the consequent low flexibility and efficiency on real time basis.

Traffic-response control can change the traffic light phase duration. This control can vary the phase duration according to the numbers of waiting vehicles detected by sensors at each intersection. Recent researches are more focused on this kind of control. D. Helbing proposed a traffic-response controller based on fluid-dynamic model [7]. C. Gershenson presented self-organizing control systems based on the queue size of arriving vehicles [8]. D. Houli made researches on Reinforcement Learning method which enabled each traffic light agent to learn to control the traffic light via interacting with the environment [9]. Most of these traffic-response controllers gather traffic information with inductive loop detectors [10], [11], cameras [12]-[14] or radars. The deficiency of this control lies in the fact that it will suffer a lot from gathering real-time traffic data in large network, limiting it to relatively small scale traffic network.

In recent years, as VANET (Vehicular Ad-hoc NETWORK) has attracted more and more attention from academic community, some researchers have begun to study traffic light control using VANET. S. Kwatirayo presented a case study based on their adaptive traffic light control algorithm in VANET. They took into consideration both vehicles traffic density and vehicle's relative location to the intersection [15]. In Maythem K. Abbas's paper, VANET is being employed to help collect traffic data from the roads for the traffic light controllers [16]. However, VANET has not been popularized in real world yet and is still far from being an effective method to collect necessary information for decision making [17].

If we could combine the concepts of fixed-time control and traffic-response control together, not only can we enhance the flexibility and efficiency of traffic light control, but also can apply this combined control to a large traffic network, so as to improve traffic efficiency, and ultimately relieve the increasingly serious traffic pressure in an economic way.

III. NETWORK MODELING

In this section, we describe our traffic network model.

As illustrated in Fig. 1, a traffic network can be modeled into a directed graph $G(V,E)$. V denotes the set of intersections in the network while E denotes the set of

roads between intersections. For a pair of neighboring nodes v_i and v_k , $\langle v_i, v_k \rangle$ denotes the road from intersection v_i to v_k , while $\langle v_k, v_i \rangle$ denotes the road from intersection v_k to v_i . $n(v_i)$ denotes the set of all neighboring intersections for v_i , and $|n(v_i)|$ denotes the number of v_i 's neighboring intersections. For example, $n(v_7) = \{v_2, v_6, v_{12}, v_8\}$, and $|n(v_7)| = 4$.

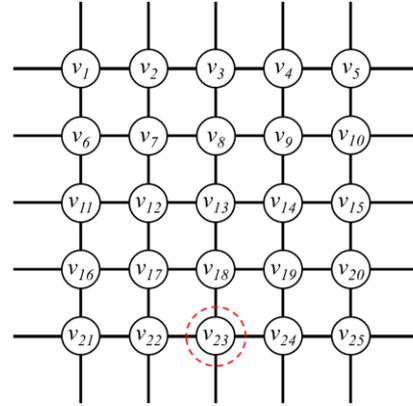


Figure 1. A 5x5 traffic network

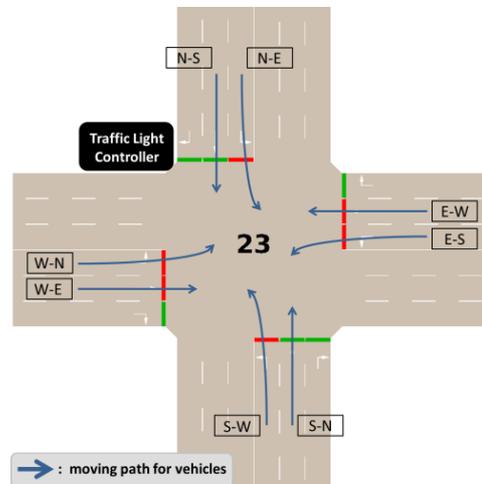


Figure 2. Details of an intersection

Take intersection v_{23} for example, Fig. 2 shows the details of an intersection in our network model. Every intersection consists of four directions, i.e., north, south, east and west. In this paper, we set three incoming lanes and three outgoing lanes for each direction. For an incoming direction, vehicles turning right drive on the rightmost lane, vehicles turning left are only allowed to use the leftmost lane and the middle lane is for vehicles running straight. There is a traffic light controller c_i at each intersection v_i to control the traffic lights for 12 incoming lanes. $l_i(j,k)$ denotes the incoming lane for intersection v_i which is from intersection v_j and is connected to v_k . Let us take v_7 for an example. A vehicle on $l_7(6,8)$ means that this vehicle is now at intersection v_7 , it came from intersection v_6 and is going to drive straightly toward intersection v_8 . A vehicle on $l_7(6,2)$ means that this vehicle also came from intersection v_6 but it is going to turn left at intersection v_7 and drive toward intersection v_2 . A vehicle on $l_7(6,12)$ means that it is

going to turn right and drive toward intersection v_{j2} . During period p , the traffic light state for lane $l_i(j,k)$ can be denoted by $s_i((j,k),p)$. $s_i((j,k),p) \in \{\text{green, red}\}$, which means the green light state and red light state respectively. We do not consider yellow state as we regard the yellow state as a special red state.

There are eight moving paths for vehicles at each intersection (Fig. 2). In this paper, we divide these eight moving paths into four pairs and each pair is composed of two compatible moving paths, be noted that all vehicles in the network drive on the right side:

- From north to south (N-S) or south to north (S-N);
- From east to west (E-S) or west to east (W-E);
- From north to east (N-E) or south to west (S-W);
- From east to south (E-S) or west to north (W-N).

Vehicles are allowed to turn right at any time. During any phase period, only a pair of lanes with compatible moving paths which have the largest traffic flow can be given green lights. Traffic light state for the other six lanes will be red. During period p , the traffic light phase for intersection v_i consists of all traffic light states for 12 incoming lanes (from four directions and each direction has three incoming lanes) and is denoted by $S(v_i,p)$. We define $S(v_i,p)$ with (1):

$$S(v_i,p) = \cup_{v_j \in n(v_i)} \cup_{v_k \in \{n(v_i)-v_j\}} s_i((j,k),p) \quad (1)$$

In (1), $|n(v_i)|=4$ and $|n(v_i)-v_j|=3$, thus $S(v_i,p)$ includes all traffic light states for $4 \times 3=12$ incoming lanes.

IV. COOPERATIVE CONTROL ALGORITHM

A. Factors

The premise of our algorithm is that the duration of traffic light phase period is fixed and needs to be determined in advance. During each period p , controller c_i needs to predict the traffic flow at intersection v_i of every incoming lane at the beginning of next period $p+1$. Take the lane $l_i(j,k)$ for example, the prediction will be affected by the following factors:

- 1) Vehicles which have arrived at intersection v_i on lane $l_i(j,k)$ at the beginning of period p . The number of such vehicles is denoted by $T_i((j,k),p)$. $T_i((j,k),p)$ can be obtained by sensors and cameras on roads.
- 2) Vehicles which are running on road $\langle v_j, v_i \rangle$ during period p . A portion of these vehicles will arrive at intersection v_i on lane $l_i(j,k)$ before next period. The number of this portion of vehicles is denoted by $Tr_i((j,k),p)$. $t((j,i),p)$ denotes the number of vehicles on road $\langle v_j, v_i \rangle$ during period p and it also can be obtained by sensors and cameras. We use $P_i(j,k)$ to denote the probability of vehicles on road $\langle v_j, v_i \rangle$ choosing to turn to intersection v_k after arriving at v_i . Such probability can be attained by statistical analysis based on historical data. Therefore, we define $Tr_i((j,k),p)$ with (2):

$$Tr_i((j,k),p) = P_i(j,k) \times t((j,i),p) \quad (2)$$

And the entire traffic situation for intersection v_i which is denoted by $T(v_i,p)$ can be defined with (3):

$$T(v_i,p) = \cup_{v_j \in n(v_i)} \cup_{v_k \in \{n(v_i)-v_j\}} (T_i((j,k),p) \cup Tr_i((j,k),p)) \quad (3)$$

- 3) Vehicles which will depart intersection v_i from lane $l_i(j,k)$ during period p . The number of such vehicles is denoted by $Td_i((j,k),p)$. We can get $Td_i((j,k),p)$ by analyzing historical data. If $s_i((j,k),p) = \text{red}$, vehicles on lane $l_i(j,k)$ cannot depart v_i , thus $Td_i((j,k),p) = 0$.
- 4) Vehicles which will depart v_i 's neighboring intersections and drive toward v_i during period p . A portion of these vehicles will come to intersection v_i on lane $l_i(j,k)$ before next period. The number of these newly coming vehicles is denoted by $Tc_i((j,k),p)$ and defined with (4):

$$Tc_i((j,k),p) = \sum_{v_h \in \{n(v_i)-v_j\}} P_i(j,k) \times Td_j((h,i),p) \quad (4)$$

With the above factors, we are able to predict the numbers of vehicles which will arrive at intersection v_i on lane $l_i(j,k)$ at the beginning of next period $p+1$, defined as $T_i((j,k),p+1)$ with equation (5), by adding up the number of vehicles which have arrived at intersection v_i on lane $l_i(j,k)$ at the beginning of period p , the number of vehicles which will arrive at v_i on lane $l_i(j,k)$ before next period both on road $\langle v_j, v_i \rangle$ and from neighboring intersections, and then subtracting the numbers of vehicles which will depart intersection v_i from lane $l_i(j,k)$.

$$T_i((j,k),p+1) = T_i((j,k),p) + Tr_i((j,k),p) + Tc_i((j,k),p) - Td_i((j,k),p) \quad (5)$$

For each vehicle on lane $l_i(j,k)$, c_i counts its waiting time since the vehicle stops. Once the vehicle departs lane $l_i(j,k)$, its waiting time will be reset to 0. $w_i((j,k),p)$ denotes the largest waiting time for vehicles on lane $l_i(j,k)$ at the beginning of period p . And we define the information of waiting time for all lanes of intersection v_i with (6):

$$W(v_i,p) = \cup_{v_j \in n(v_i)} \cup_{v_k \in \{n(v_i)-v_j\}} w_i((j,k),p) \quad (6)$$

B. Algorithm

Algorithm 1 describes the main idea of our proposed cooperative traffic light control. We set a fixed duration of traffic light phase period in advance. At the beginning of each period, algorithm 1 will be run. First, controller c_i sets the traffic light states for every incoming lane of intersection v_i according to the decision made in the previous period (line 3). Then c_i gets the information of waiting time for all incoming lanes of intersection v_i (line 4). If $w_i((j,k),p)$ is the largest one among all incoming lanes, and $w_i((j,k),p)$ is larger than $maxWT$ (a pre-defined parameter of the maximum waiting time for every vehicle) (line 5), c_i will give green light to lane $l_i(j,k)$ and its pairing lane with a compatible moving path with $l_i(j,k)$ (line 6). Otherwise, c_i gets current traffic flow data of each incoming lane (line 8) and share this information and current traffic light phase with its neighboring controllers (line 9). After that, c_i gets the information of current traffic phase and traffic flow of its neighboring intersections (line 10), and predicts the traffic flow of v_i

at the beginning of next period with all information it gets (line 11). Finally, c_i determines which pair of lanes with two compatible moving paths will have the largest traffic flow at the beginning of next period, and gives green lights to this pair of lanes while sets red lights to the other six lanes (line 12). So “compute $S(i,p+1)$ ” line 12 is a subprogram to compute the traffic light phase of intersection v_i for next period by analyzing the traffic data controller c_i gets. If two pairs of lanes have the same traffic flow, c_i will give green lights to the pair of lanes with larger vehicle waiting time. One thing should be noted is that traffic light states for right-hand lanes will always be green as vehicles are allowed to turn right as long as there is no obstacles or human beings ahead.

Algorithm 1. Cooperative traffic light control

```

1: at the beginning of period p
2: for all  $c_i$  in  $C$  do
3:   set traffic lights for  $v_i$ ;
4:   gets  $W(v_i,p)$ ;
5:   if  $\max W(v_i,p) > \max WT$ 
6:     go to 12;
7:   end if;
8:   get  $T(v_i,p)$ ;
9:   share  $T(v_i,p)$  and  $S(v_i,p)$  with  $n(v_i)$ ;
10:  get  $S(n(v_i),p)$  and  $T(n(v_i),p)$ ;
11:   $T(v_i,p+1) \leftarrow T(v_i,p) \cup S(v_i,p) \cup T(n(v_i),p) \cup S(n(v_i),p)$ ;
12:  compute  $S(i,p+1)$ ;
13: end for
    
```

V. SIMULATION EVALUATION

In this section, we present our simulation experiments and evaluate them. In our experiments, we use SUMO as the traffic flow simulator. SUMO (Simulation of Urban MObility) is an open source, microscopic and multi-modal traffic simulation package designed to handle large road networks [18]. SUMO enables us to build a simulative traffic network, generate traffic flows with different features, and control traffic lights. As SUMO is relatively weak in programming, we use NS3 (a discrete-event network simulator) in the meantime to make use of its strong programming ability. We establish a feedback loop between SUMO and NS3 using TraCI (Traffic Control Interface, an extension of SUMO to communicate with external applications). With the combination between SUMO and NS3, we implement simulations by our proposed algorithm. Then we evaluate our algorithm by analyzing simulation data exported from SUMO.

A. Experiment Setup

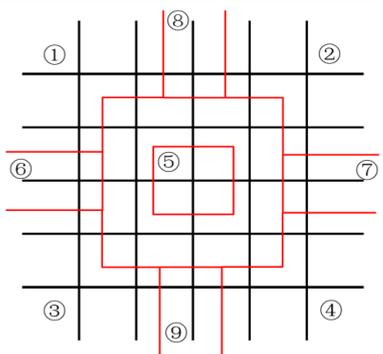


Figure 3. Nine blocks of the network

We use SUMO to construct a 5x5 network with 25 intersections and 60 roads. Each road is 200 meters long with six lanes. In our simulation experiments, we divide the whole network into nine blocks, and each block is defined by a unique identifier (Fig. 3).

In our simulation experiments, we use SUMO to generate traffic flow by OD-matrices (Origin-Destination-Matrices). Vehicles will be generated with random departure times and travel routes. Generated vehicles will appear in the scenario at the origins and then run to the destinations along the routes defined according to OD-matrices. After a vehicle arrives at its destination, it will be removed from the scenario automatically.

Table I shows the basic information of all vehicles generated with SUMO in our experiments.

TABLE I. BASIC INFORMATION OF VEHICLES

Acceleration	3.0 m/s
Brake Acceleration	6.0 m/s
Length	5 m
Minimum Vehicle Gap	2.5 m
Speed Limit	40 km/h

B. Scenarios for Simulation

To evaluate our algorithm under different traffic situations, we construct four scenarios, and each scenario corresponds to a specific OD-matrix. In all scenarios, the departure time of each vehicle is randomly arranged within one hour.

In our first scenario, we simulate a traffic situation with a low traffic density. We use SUMO to generate 650 vehicles to drive among the nine blocks.

In our second scenario, we simulate a traffic situation with a high density. We arrange 7200 vehicles to drive among the nine blocks.

In our third scenario, we simulate a traffic situation for the morning rush. Since during morning rush hour, a great number of people drive to work from suburbs to the downtown, we arrange 3000 vehicles to drive among the nine blocks, and most vehicles are from peripheral blocks to the central block.

In our fourth scenario, we simulate the evening rush. During evening rush hour, more vehicles drive from the downtown area to suburban areas, as people return home from work. Thus we arrange 3000 vehicles to drive among the nine blocks, and the majority of the vehicles are from the central block to peripheral blocks.

C. Evaluation

In our simulation experiments, we set the phase duration as 30 seconds for we take into consideration the following two aspects. On the one hand, intuitively, the shorter the phase duration is, the less total waiting time for vehicles in the whole traffic network will be. On the

other hand, we must not neglect the basic demand of pedestrians. Since usually a six-lane main road is 20 – 23 meters wide, and plus non-motor vehicle lanes and the central green belt, pedestrians need to walk about 30 meters [19]. The Manual of Uniform Traffic Control Devices recommended 1.05 m/sec (3.5 ft/sec) as the walking speed for calculating the pedestrian clearance time [20]. So we set the traffic light phase duration as 30 seconds to meet both needs of ensuring relatively short waiting time for vehicles and enabling pedestrians to walk across the roads within the phase duration.

For comparison, we have also performed simulations with independent intersection control. In this mechanism, each controller computes traffic light phase based on traffic information of its own intersection. Neighboring intersections do not cooperate with each other.

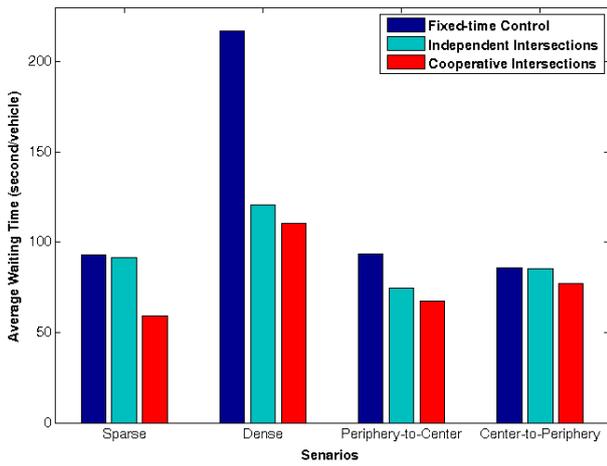
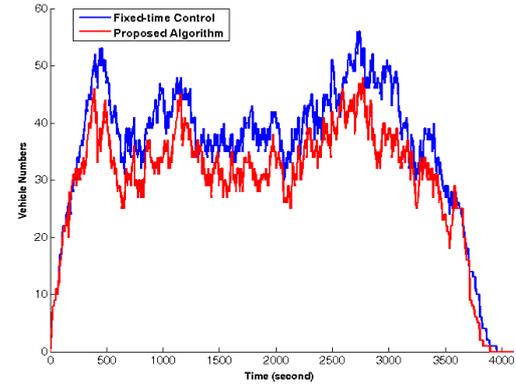
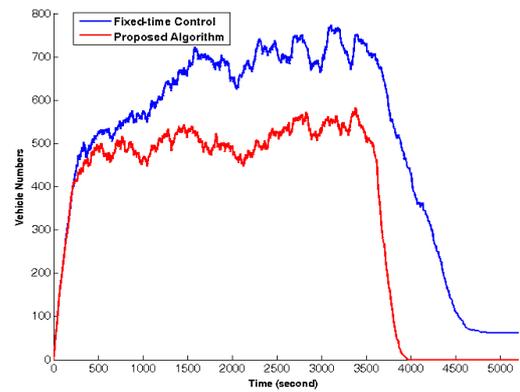


Figure 4. Average waiting time in four scenarios

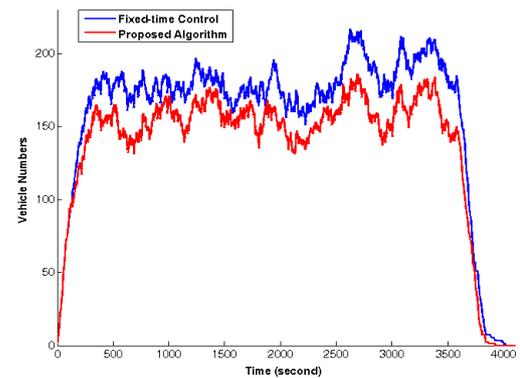
Fig. 4 displays the average waiting time for every vehicle running in the four scenarios by three methods: fixed-time control, independent intersection control without cooperation and our proposed algorithm with cooperation among intersections. Evidently, our algorithm achieves better performance than independent intersection control without cooperation. Our algorithm can reduce average waiting time in all four scenarios, particularly the dense scenario in which our algorithm lowers the average waiting time by 49.09%. The reduction rate of our algorithm is 36.62% and 28.18% in sparse scenario and periphery-to-center scenario respectively. In center-to-periphery scenario, independent intersection control can barely reduce the average waiting time while our algorithm achieves a decrease rate of 9.83%. However, the reduction rate is much lower than that in other scenarios. That is because the process of vehicles running from center to periphery is shunting traffic flow itself and the room for improvement is less than that of other scenarios.



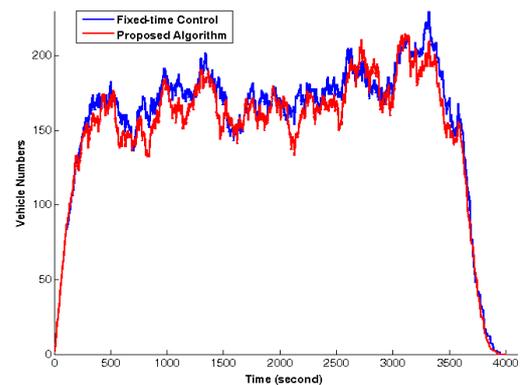
(a) Sparse scenario



(b) Dense scenario



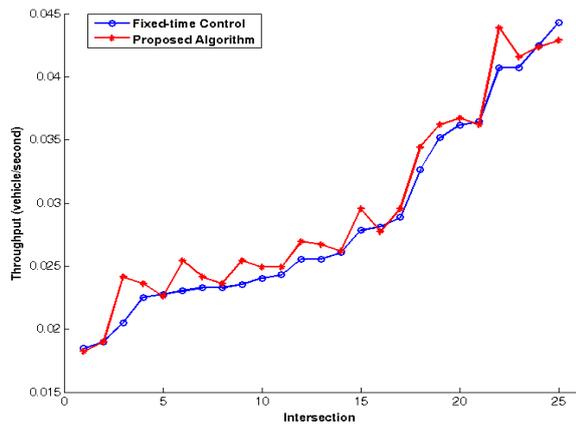
(c) Periphery-to-Center scenario



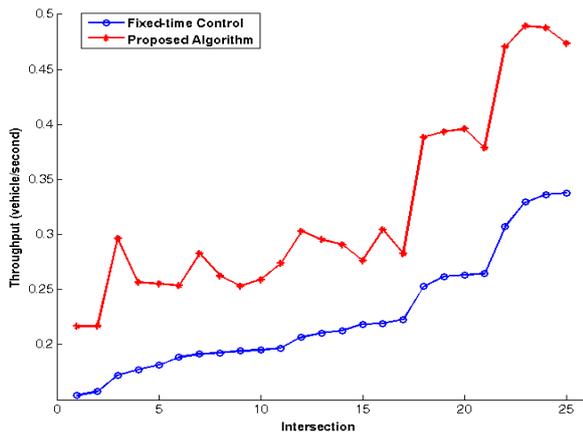
(d) Center-to- Periphery scenario

Figure 5. Vehicle number dynamics in the testbed

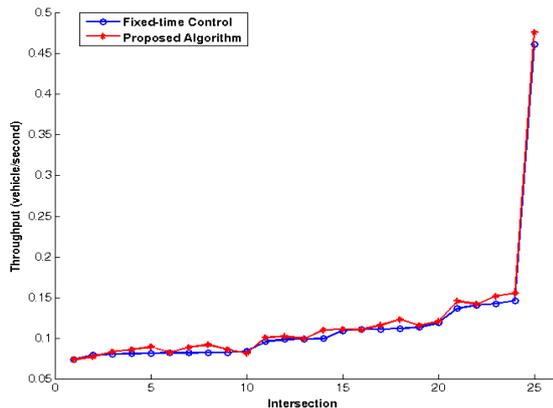
Fig. 5 illustrates the vehicle numbers for every second within each scenario. Apparently, after about 200 seconds our algorithm can obtain a steady reduction of vehicle numbers with time, especially in dense scenario (Fig. 5.b). It should be noted that the scale of vehicle numbers varies in different scenario. In sparse, periphery-to-center and center-to-periphery scenarios, room for improvement is relatively little since traffic loads in these three scenarios are not so heavy. But in dense scenario, our algorithm can significantly relieve the much heavier traffic load. The experiment results corroborate the efficiency of our algorithm in reducing traffic load in the whole network, particularly in a dense network with heavy traffic load.



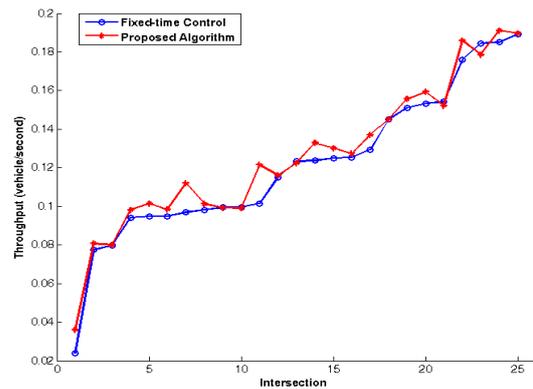
(a) Sparse scenario



(b) Dense scenario



(c) Periphery-to-Center scenario



(d) Center-to-Periphery scenario

Figure 6. Traffic throughput situation of each intersection

Fig. 6 shows the traffic throughput of each intersection. The traffic throughput value of an intersection is calculated by the total number of vehicles departing that intersection per second. For a more vivid display of the distinction between our algorithm and fixed-time control, we illustrate the traffic throughput in an ascending order. The scales of throughput are different in the four scenarios. In sparse, center-to-periphery and periphery-to-center scenarios, there are only slight improvements of traffic throughput. However, in dense scenario (Fig. 6b) which has a high probability of traffic congestions and is quite difficult to deal with, our algorithm achieves an obvious increase in traffic throughput.

VI. CONCLUSION

In this paper, we combine the concepts of fixed-time traffic light control and traffic-response control together and propose an adaptive traffic light control algorithm which can dynamically adjust the traffic light phases for multiple intersections of urban main roads by prediction of traffic situation and cooperation among traffic light controllers. We run simulations of four different scenarios in a traffic test-bed network with 25 intersections and 60 roads. Our algorithm performances better than fixed-time control in all four scenarios, particularly in the scenario with a high traffic density. The results demonstrate the efficiency and practicability of our algorithm in reducing unnecessary waiting time for vehicles and traffic load, thus enhancing traffic throughput of intersections. In future works, we will construct network models based on real urban main roads to evaluate the performance of our algorithm under more complicated scenarios. In addition, we will combine our algorithm with VANET, so as to obtain more accurate data of incoming vehicles at each intersection.

REFERENCES

- [1] M. U. Aslam, H. H. Masjuki, M. A. Kalam, H. Abdesselam, T. M. I. Mahlia, and M. A. Amalina, "An experimental investigation of CNG as an alternative fuel for a retrofitted gasoline vehicle," *Journal of Fuel: The Science and Technology of Fuel and Energy*, vol. 85, no. 5-6, pp. 717-724, March-April 2006.
- [2] Q. Wang, L. Wang, and G. Wei, "Research on traffic light adjustment based on compatibility graph of traffic flow," in

Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2011 International Conference, Zhejiang, China, Aug. 2011, vol. 1, pp. 88-91.

- [3] K. M. Yousef, J. N. Al-Karaki, and A. M. Shatnawi, "Intelligent traffic light flow control system using wireless sensors networks," *Journal of Information Science and Engineering*, vol. 26, no. 3, May 2010.
- [4] K. C. Yit, K. Y. Wei, K. L. Wei, and T. K. Min, "Q-Learning traffic signal optimization within multiple intersections traffic network on computer modeling and simulation (EMS)," in *Proc. 2012 Sixth UKSim/AMSS European Symposium*, Valetta, Malta, Nov. 2012, pp. 343-348.
- [5] M. A. Khamis and W. Gomaa, "Enhanced multiagent multi-objective reinforcement learning for urban traffic light control," in *Proc. 11th International Conference Machine Learning and Applications (ICMLA)*, pp. 586-591, Boca Raton, USA, Dec. 2012.
- [6] National Electrical Manufacturers Association, NEMA Standards Publication TS 2-2003 v202.06—Traffic Controller Assemblies with NTCIP requirements, 2003.
- [7] D. Helbing, S. Lämmer, and J. Lebacque, "Self-organized control of irregular or perturbed network traffic," *Optimal Control and Dynamic Games*, Springer, pp. 239-274, 2005.
- [8] C. Gershenson, "Design and control of self-organizing systems," PhD thesis, Vrije Universiteit Brussel, 2007.
- [9] D. Houli, L. Zhiheng, and Z. Yi, "Multiobjective reinforcement learning for traffic signal control using vehicular adhoc network," *Journal on Advances in Signal Processing (EURASIP)*, vol. 2010, pp. 7, 2010.
- [10] Y. Zhu, X. Liu, M. Li, and Q. Zhang, "POVA: Traffic light sensing with probe vehicles, on parallel and distributed systems," *IEEE Trans. on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1390-1400, July 2013.
- [11] B. Zhou, J. Cao, and H. Wu, "Adaptive traffic light control of multiple intersections in WSN-based ITS," in *Proc. 73rd IEEE Vehicular Technology Conference (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1-5.
- [12] P. Choudekar, S. Banerjee, and M.K. Muju, "Implementation of image processing in real time traffic light control," in *Proc. the 3rd International Conference on Electronics Computer Technology (ICECT)*, Kanyakumari, India, April 2011, vol. 2, pp. 94-98.
- [13] A. Kanungo, A. Sharma, and C. Singla, "Smart traffic lights swithcing and traffic density calculation using video processing," in *Engineering and Computational Sciences (RAECS)*, Chandigarh, India, March 2014, pp. 1-6.
- [14] S. K. Asare and R. A. Sowah, "Design and development of a microcontroller based traffic light control system using image processing techniques: A case study prototype for Legon-Okponglo Junction, University of Ghana," in *Proc. IEEE 4th International Conference on Adaptive Science & Technology (ICAST)*, Kumasi, Ghana, Oct. 2012, pp. 59-64.
- [15] S. Kwatirayo, J. Almhana, and Z. Liu, "Adaptive traffic light control using VANET: A case study," in *Proc. 9th International Conference on Wireless Communications and Mobile Computing Conference (IWCMC)*, Sardinia, July, 2013, pp. 752-757.
- [16] M. K. Abbas, M. N. Karsiti, M. Napiah, and B. B. Samir, "Traffic light control using VANET system architecture," in *Proc. National Postgraduate Conference (NPC)*, Kuala Lumpur, Malaysia, Sept. 2011, pp. 1-6.
- [17] H. Hartenstein and K. P. Laberteaux, *Vehicular Applications and Inter-Networking Technologies*, Wiley, 2010, ch. 1, pp. 1-4.
- [18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—simulation of urban mobility: An overview," in *Proc. the 3rd International Conference on Advances in System Simulation (SIMUL 2011)*, Barcelona, Spain, October, 2011, pp. 55-60.
- [19] I. B. Potts, D. W. Harwood, and K. R. Richard, "Relationship of lane width to safety for urban and suburban arterials," in *Proc.*

Transportation Research Board of the National Academies Annual Meeting (TRB 2007), vol. 2023, 2007, no. 1, pp. 63-82.

- [20] Manual of uniform traffic control devices, 2009.



interests include VANET.

Qin Zhu received the B.Eng. degree in Major of Computer Science and Technology from Department of Computer Science and Technology, East China Normal University, Shanghai, China in 2012. She is currently studying for the M.Eng. degree in Major of Software Engineering in Software Engineering Institute, East China Normal University, Shanghai, China. Her research



Chao Peng received his Ph.D. degree in Information Science from Japan Advanced Institute of Science and Technology in 2006. He is an Associate Professor of the Software Engineering Institute at East China Normal University, Shanghai, China. His research interests include internet of vehicles, cyber physical systems, algorithms and network systems.



Pengfei Duan received the B.Eng. degree in Major of Software Engineering in Software Engineering Institute, East China Normal University, Shanghai, China in 2014. He is currently studying for the M.Eng. degree in Major of Software Engineering in Software Engineering Institute, East China Normal University, Shanghai, China. Her research interests include VANET.



Jingmin Shi received the B.Eng. degree in Major of Software Engineering in Software Engineering Institute, East China Normal University, Shanghai, China in 2012. She is currently studying for the M.Eng. degree in Major of Software Engineering in Software Engineering Institute, East China Normal University, Shanghai, China. Her research interests include VANET.



Yu Bao received his Ph.D. degree in Computer Science from East China Normal University in 2005. He is an Assistant Professor of the Software Engineering Institute at East China Normal University, Shanghai, China. His research interests include software architecture and enterprise resource planning systems.



Mengjun Xie received the Ph.D. degree in Computer Science from College of William and Mary, Williamsburg, in 2009. He is an Assistant Professor of Computer Science at University of Arkansas at Little Rock, Little Rock. His research interests include network security, information security, network systems, and operating systems.