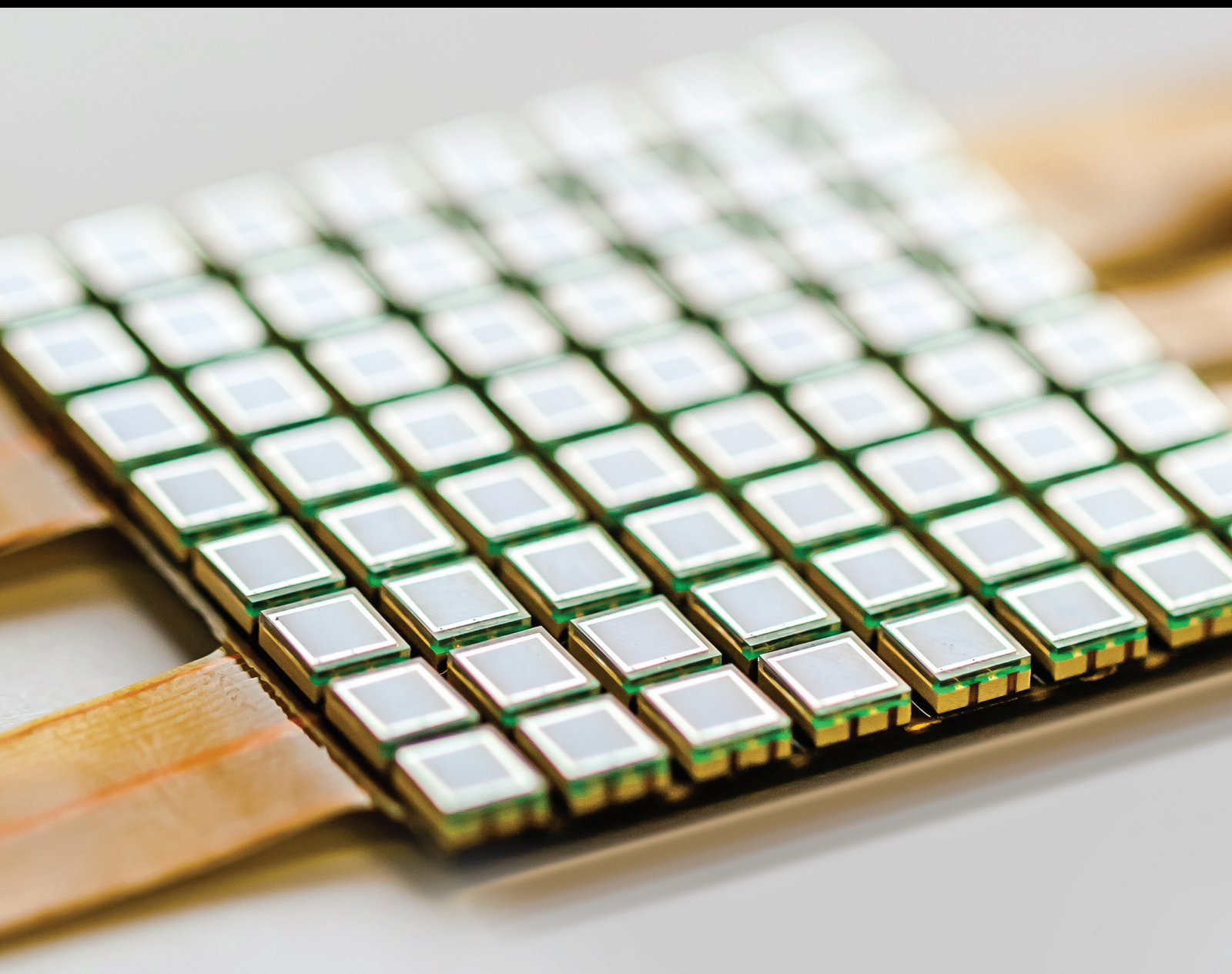
Journal of Sensors

Sensors in Connected Vehicle Technology: How Sensors Play a Critical Role

Lead Guest Editor: Chuan Ding
Guest Editors: Jia Hu and Xiaolei Ma



Sensors in Connected Vehicle Technology: How Sensors Play a Critical Role

Journal of Sensors

Sensors in Connected Vehicle Technology: How Sensors Play a Critical Role

Lead Guest Editor: Chuan Ding
Guest Editors: Jia Hu and Xiaolei Ma

Editorial Board

Contents

Hindawi
Journal of Sensors
Volume 2017, Article ID 8241932, 2 pages
https://doi.org/10.1155/2017/8241932

Editorial

Sensors in Connected Vehicle Technology: How Sensors Play a Critical Role

Chuan Ding,[1] Jia Hu,[2] and Xiaolei Ma[1]

[1]*School of Transportation Science and Engineering, Beihang University, Beijing 100191, China*
[2]*Federal Highway Administration, Washington, DC, USA*

Correspondence should be addressed to Chuan Ding; cding@buaa.edu.cn

Received 13 November 2017; Accepted 13 November 2017; Published 28 November 2017

A rapid development of connected vehicle (CV) technology has been observed around the globe in the past decade. The problems that the conventional traffic management systems bear are believed to be solved by introducing the technology. This CV technology puts diagnostic sensors onto vehicles/infrastructures and has the data collected transmitted wirelessly between vehicles and nearby infrastructures. It would no longer rely on conventional data collection equipment, like a loop detector or video detections, and it collects much more information than the conventional ways. Measurements that are previously unknown are available, which include but are not limited to vehicle speeds, positions, arrival rates, rates of acceleration and deceleration, queue lengths, and stopped time. With this extra information, many applications are made possible. However, on the other hand, the accuracy and the stability of sensors cause challenges for the development of CV application, for instance, the accuracy of GPS, communication delay, and LIDAR accuracy. These factors require special design embedded in those CV applications to accommodate the limitation of sensors. This special issue aims to serve as a major platform to facilitate the discussion and exchange of research ideas and technology development, encourage multidimensional knowledge sharing, and enhance research activities in investigating sensors in connected vehicle technology. In total, seven papers are included in this special issue and are summarized as follows.

There are several articles focusing on traffic control and analysis using the sensor-based data to improve traffic system. The autonomous vehicle is able to facilitate road safety and traffic efficiency and has become a promising trend of future development. W. Wu et al. proposed traffic control models based on cellular automata for intersections in the autonomous vehicle environment. Multiagent technology is used to simulate the proposed model. The simulation results show that the control strategy of the proposed model significantly reduces average delays and a number of stops as well as increasing traffic capacity. A classical control problem for an isolated oversaturated intersection is revisited with a focus on the optimal control policy to minimize total delay. J. Tan et al. used the gradient descent algorithm to convert the optimal control plan of the continuous-time model to the plan of the discrete-time model in many cases. Analytic proof and numerical tests for the algorithm are also presented.

From the perspective of sensor-based traffic operation and management, several articles analyzed the traffic condition and driver behavior. With the development of connected vehicle (CV) and vehicle to X (V2X) communication, more traffic data are being collected from the road network. In order to predict future traffic condition from connected vehicles' data in real time, P. Wang et al. proposed an online traffic condition evaluation model utilizing V2X communication. The contemporary vehicle data from the on-board diagnostic (OBD) is fused with the static road data in the roadside unit (RSU). Compared with traditional evaluation systems, the proposed model can handle more types of data but demands less data transfer. W. Zhao et al. did a sensor-based visual effect evaluation of chevron alignment signs' colors on drivers through the curves in snow and ice environment. The conclusions provide evident references for freeway warning products and the design of intelligent vehicles.

There are several articles focusing on driving and travel behavior using the GPS data. J. Wang and Y. Cao developed a method to identify whether there is deliberate speed-up or slow-down movement of a bus based on the recovery of GPS data. The effectiveness of the developed method was demonstrated using the data collected in Harbin, China. The results show that it can help bus enterprises to design reasonable time of day intervals and significantly improve their level of service. S. An et al. proposed a public transit riders' travel pattern measuring method based on divided cells and public transit vehicle's GPS data. A case study is carried out to evaluate the methods, which use the GPS data collected from taxis and buses in Harbin, China. The study is expected to provide a better understanding of public transit riders' travel patterns.

The remaining one article investigated spatial crowdsourcing problem caused by the increasingly widespread sensors. With the development of sensors, particularly vehicle sensors and mobile sensors, spatial crowdsourcing is gaining even more attention. B. Du et al. proposed a novel spatial crowdsourcing problem called software development team formation (SDTF). They proved that SDTF was NP-hard and designed three greedy algorithms and an index-based algorithm to solve the SDTF problem. Extensive experiments are conducted on synthetic and real datasets, and the results verify the effectiveness and efficiency of the algorithms.

Acknowledgments

The guest editors hope the information provided in this special issue is useful. Finally, we would like to thank the authors for an excellent contribution of their research works and very warmly acknowledged the reviewers for their valuable review comments.

Chuan Ding
Jia Hu
Xiaolei Ma

Hindawi
Journal of Sensors
Volume 2017, Article ID 9436054, 6 pages
https://doi.org/10.1155/2017/9436054

Research Article

Traffic Control Models Based on Cellular Automata for At-Grade Intersections in Autonomous Vehicle Environment

Wei Wu,[1] Yang Liu,[1] Yue Xu,[2] Quanlun Wei,[3] and Yi Zhang[4,5]

[1]*School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha 410076, China*
[2]*Polytech Nantes, Université de Nantes, 44600 Nantes, France*
[3]*College of Transport and Communications, Shanghai Maritime University, Shanghai 201306, China*
[4]*China Institute of Urban Governance, Shanghai Jiao Tong University, Shanghai 200240, China*
[5]*School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

Correspondence should be addressed to Yi Zhang; darrenzhy@sjtu.edu.cn

Received 29 July 2017; Revised 1 October 2017; Accepted 11 October 2017; Published 6 November 2017

Academic Editor: Xiaolei Ma

Autonomous vehicle is able to facilitate road safety and traffic efficiency and has become a promising trend of future development. With a focus on highways, existing literatures studied the feasibility of autonomous vehicle in continuous traffic flows and the controllability of cooperative driving. However, rare efforts have been made to investigate the traffic control strategies in autonomous vehicle environment on urban roads, especially in urban intersections. In autonomous vehicle environment, it is possible to achieve cooperative driving with V2V and V2I wireless communication. Without signal control, conflicted traffic flows could pass intersections through mutual cooperative, which is a remarkable improvement to existing traffic control methods. This paper established a cellular automata model with greedy algorithm for the traffic control of intersections in autonomous vehicle environment, with autonomous vehicle platoon as the optimization object. NetLogo multiagent simulation platform model was employed to simulate the proposed model. The simulation results are compared with the traffic control programs in conventional Synchro optimization. The findings suggest that, on the premises of ensuring traffic safety, the control strategy of the proposed model significantly reduces average delays and number of stops as well as increasing traffic capacity.

1. Introduction

The emergence and development of motorized transportation have not only promoted the progress of human civilization, but also brought about severe issues relating to road safety and traffic efficiency. According to statistics, more than 100,000 people died in road accidents per year worldwide [1–3]. In the United States traffic congestion has caused enormous economic losses. For example, in 2013, urban Americans experienced an extra 6.8 billion hours of travel and 3.1 billion gallons of fuel consumed because of traffic congestion [4–6]. With the development of motor vehicle, researchers have recognized that automata vehicle is probably one of the most effective ways to tackle those problems [7–10].

In the autonomous vehicle environment, real-time, mutual, and effective communication could be achieved not only between vehicles and vehicles, but also between vehicles and road [11–14]. With respect to road safety, the mutual cooperation helps to solve the traffic problems caused by the driving errors, insufficient visibility distance [15–17]. In terms of traffic efficiency, vehicles could form platoon through cooperation to reduce headways and increase road capacity [18–20].

A major part of previous studies on traffic flow in autonomous vehicle environment focused on continuous flow and the cooperative between vehicles on highway [21–23]. Few studies have proposed traffic control strategies in autonomous vehicle environment, with an emphasis on intermittent traffic flows. Considering the limitation of road space, even in autonomous vehicle environment, there will still be large number of plane intersections in urban areas. Therefore, it is imperative to investigate the traffic control strategies in autonomous vehicle environment in plane intersections. Although some related researches proposed traffic control

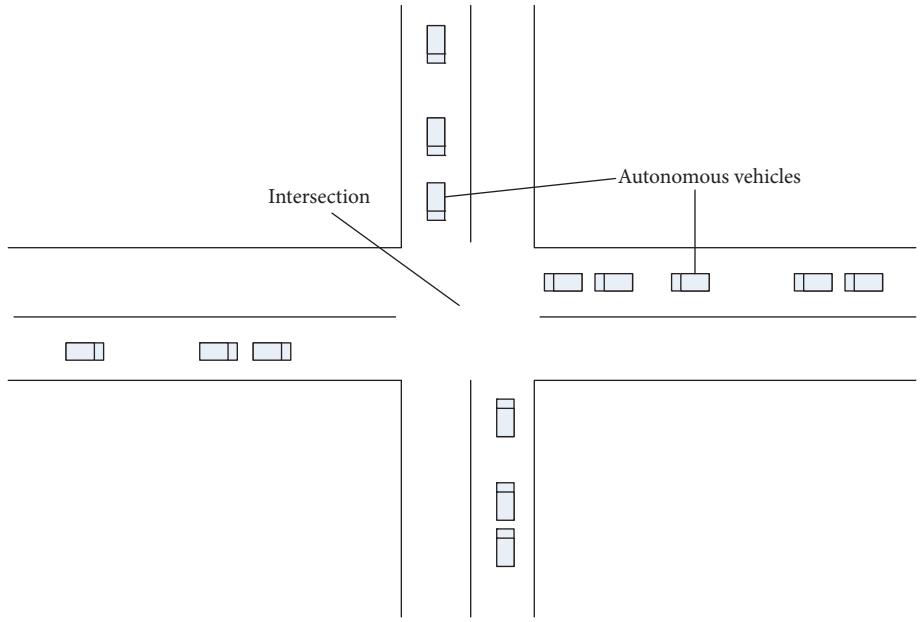


Figure 1: Basic idea of the research problem.

model in autonomous vehicle environment, the control algorithm is mainly on the basis of first-come-first-served (FCFS) algorithm [24], and the optimized object is focused on every single vehicle [25, 26] and thus has limited optimization effects.

Focusing on a single intersection, this paper proposed a traffic control model based on cellular automata in autonomous vehicle environment. The primary goals of this paper are to (1) explicitly capture the dynamic interaction between platoon and traffic control strategy; (2) establish cellular automata model to interpret the disciplines of traffic flow in autonomous vehicle environment; and (3) improve traffic efficiency based on minimize travel delays.

2. Development of Optimization and Simulation Approach

2.1. Problem Description. In autonomous vehicle environment, traffic lights are unnecessary. Traffics travel through plane intersections by communicating with each other. The research problem can be demonstrated in Figure 1. Autonomous vehicles arrive stochastically from four approaches. For the sake of calculation simplicity, in this paper, there is only one through lane considered at each approach. Pedestrians and cyclists are not considered. How to allocate passage time for each autonomous vehicle to maximize the system benefits of the intersection is studied in this paper. The common solution to this problem is first-come-first-served (FCFS) algorithm. However, FCFS algorithm has some disadvantages that can be demonstrated in Figure 2. There are two southbound autonomous vehicles in Figure 2. The estimated arrival times for these two vehicles are 2 and 5 seconds, respectively. There are a bunch of continuous westbound autonomous vehicles. The estimated arrival times are 3, 3.5, 4, 4.5, 5, 5.5, and 6 seconds, respectively. By employing FCFS algorithm, the southbound autonomous vehicle with 2 seconds' estimated arrival time will be served firstly. However, maybe better solution can be obtained if the continuous vehicle platoons can be able to proceed with priority. On the other hand, the optimization algorithm on account of each vehicle is very complex and inefficient. Optimization based on vehicle platoons may have applicability.

2.2. General Notation and Terminology. The notations used hereafter are summarized in Key Variables (Notations) Used in the Formulations.

2.3. The Traffic Flow Model Based on the Cellular Automata. Discrete time and discrete space are employed in cellular automata model. It is widely used in modeling traffic flows as it is able to simulate complicated traffic flow phenomena with a small number of simple rules. The basic traffic flow models based on cellular automata are as follows, from vehicle 1 to vehicle n.

The Acceleration Process. When an autonomous vehicle proceeds freely, it accelerates to the maximum speed, which is presented by the following formula.

$$\begin{aligned} &If\ \left(V_i^j(t) < \Delta X_{i-1}^j(t) + G_i^j(t)\right): \\ &\quad \{If\ \left(V_i^j(t) < V_{\max}\right), \\ &\qquad V_i^j(t+1) = V_i^j(t) + 1 \\ &else \\ &\qquad V_i^j(t+1) = V_{\max}\}. \end{aligned}$$

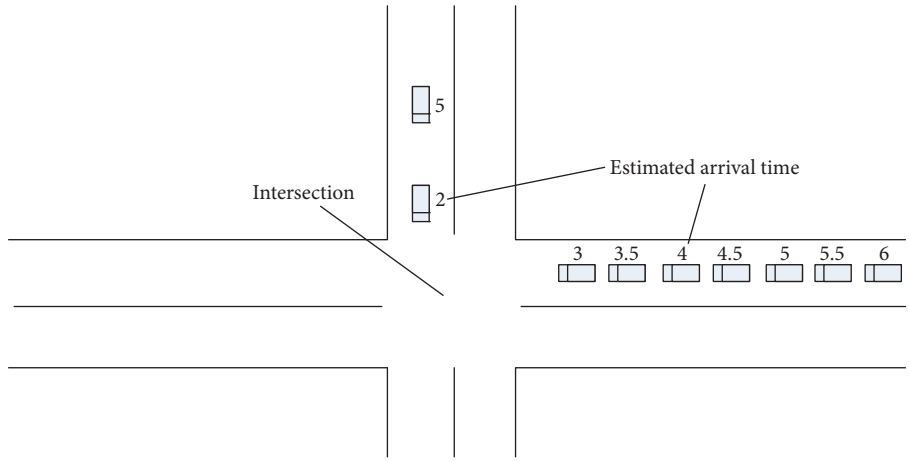


Figure 2: Disadvantages of first-come-first-served (FCFS) algorithm.

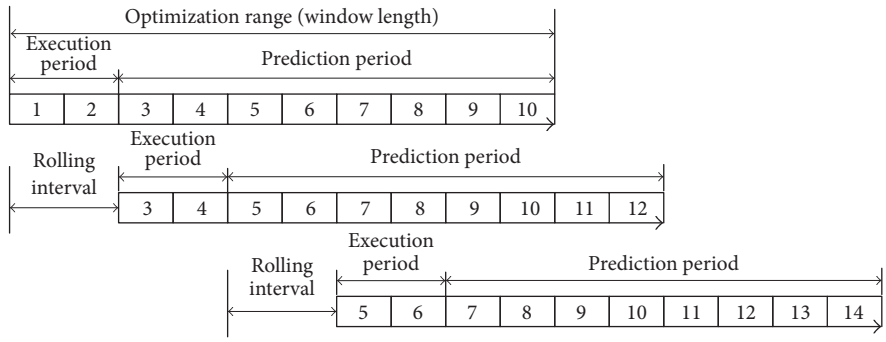


Figure 3: Optimization method of sliding time window.

The Deceleration Process. When an autonomous vehicle is blocked by a leading vehicle, it decelerates to avoid a crash and closely follows the leading vehicle.

$$If\ (V_i^j(t) \geq \Delta X_{i-1(t)}^j + G_i^j(t)):$$
$$\{V_i^j(t+1) = \min(\Delta X_{i-1}^j(t) + G_i^j(t), V_i^j(t))\}.$$

When an autonomous vehicle arrives at the intersection and does not get right-of-way, it decelerates as follows:

$$If\ (V_i^j(t) \geq D_i^j(t)):$$
$$\{If\ (\sigma_i^j(t) = 1),$$
$$V_i^j(t+1) = \min(\Delta X_{i-1}^j(t) + G_i^j(t), V_i^j(t))$$
$$If\ (\sigma_i^j(t) = 0),$$
$$V_i^j(t+1) = \min(D_i^j(t), \Delta X_{i-1}^j(t) + G_i^j(t), V_i^j(t))\}.$$

The Platoon Merging Process. If the time headway between two vehicles at the same approach is less than the critical time headway, then the two vehicles will be treated as platoons.

$$If\ ((X_n^{j-1}(t) - X_1^j(t))/V_1^j(t) < h_c):$$
$$\{\text{merging platoon } j \text{ into platoon } j-1\}.$$

The Platoon Split Process. When the waiting time of the traffics at the conflicting approach exceeds the maximum limit, the current vehicle platoon splits to two platoons, so as to give right-of-way to the conflicting traffic flows.

The Location Updating Process

$$X_i^j(t+1) = X_i^j(t) + V_i^j(t+1).$$

2.4. Rolling Window Time Control Strategy Based on Greedy Algorithm. In this paper, we employed the rolling time window approach as the optimization method. The basic process of the rolling time window approach is shown in Figure 3.

In the rolling step, we employed the greedy algorithm to optimize the control strategy of the platoon at the intersection. The optimization steps are as follows.

Step 1 (the beginning of optimization process). Identify the vehicle platoon on the four approaches. Optimization is started based on the platoon.

Step 2 (the determination of analysis scope). In the time span of each rolling step, search and determine the number of platoons that can pass through the four approaches.

Step 3 (the determination of control strategy). We number the four platoons that are closest to the intersections in

the four approaches of analysis scope, assuming that the length of each platoon is h, k, p, q. Using the greedy algorithm, when the 4 platoons request the conflicting right-of-way, we selected the longest platoon ($\max(h, k, p, q)$) to pass through the intersection. When a platoon has already passed, we reselect another four platoons that are closest to the intersections in the four approaches and return to the beginning of Step 3. The maximal waiting time is set for each platoon. When the actual waiting time exceeds the maximal waiting time, the platoon has the highest priority.

Step 4. Go to the next rolling time and begin from Step 1.

3. Simulation Modeling

In this paper, the simulation of traffic control model of plane intersection based on cellular automata in automatic driving environment is conducted in NetLogo platform. NetLogo is a multiagent programmable modeling environment for simulating natural and social phenomena. It was launched by UriWilensy in 1999 and was developed by the Link Learning and Computer Modeling Center (CCL), which is designed to provide a powerful and easy-to-use computer-aided tool for research and educational institutions.

At numerical simulation, each cellular length takes 3.5 m. $V_{\max} = 6$ (76 km/h); the simulation step size was set to 0.1 seconds, with continuous operation of 36000 steps of one-hour traffic conditions, taking into account the temporal and spatial distribution of traffic flow characteristics, using open boundary conditions. In the model, in order to eliminate the impact of random factors, using the last 900 simulation steps (15 minutes) for sample calculation, every sample represents 20 simulations, that is, each point of the chart is the average of 20 simulations.

4. Performance Analysis

To evaluate the performance of the proposed strategies, we compared the control strategy of proposed model and the conventional signal control strategy optimized by Synchro software. In Synchro model, the saturation flow was set to 1900 pcu/h; the amber time and all red time were set to 3 s and 2 s, respectively. Green splits and green phase time are optimized by Synchro software automatically. By adjusting the amount of approaching traffic volume, we changed the degree of saturation in the range of 0.1 to 1.3 (based on the saturation in Synchro). The length of the approach lane was set to 1000 m. The maximum waiting time was set to 30 s. The critical platoon headway was set to 2 s. The results are shown in Figures 4 and 5.

As we can see from Figures 4 and 5, in both proposed model and Synchro model, the delays and number of stops increase with the increase of saturation. In Synchro model, different cycle time shows similar effects. When the saturation is below 0.8, the delays and stops increase slowly. When the saturation is greater 0.8, the delays and stops increase significantly. Compared with Synchro model, in the proposed model, when the saturation is high, the delays and stops will be much lower in autonomous vehicle environment. When

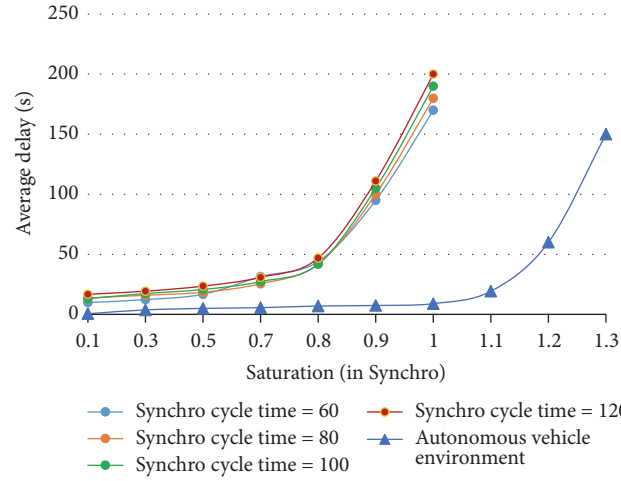


Figure 4: Comparison of average delay.

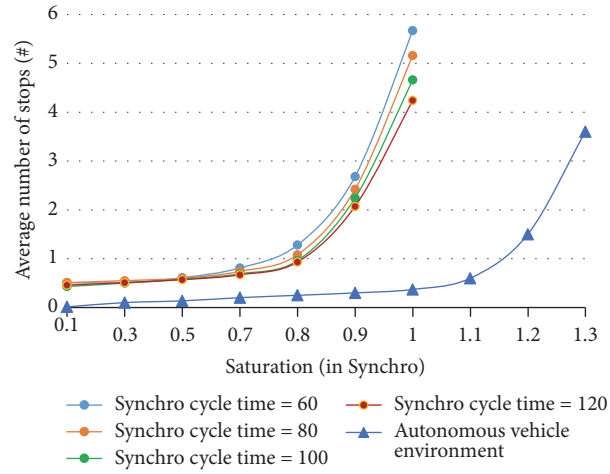


Figure 5: Comparison of average stops.

the saturation is as high as 1.1, the delays and stops will show significant rise in the proposed model. The findings represent that the proposed model is effective in reducing delays and stops.

As we can see from Figure 6, when the saturation is less than 0.5, the numbers of vehicles clearing the intersection in the proposed model and Synchro model are almost the same, indicating that both models meet the needs. However, as the saturation continues to increase, the numbers of vehicles clearing the intersection in the proposed model and Synchro model will show significant differences. When the saturation is close to 0.9, the number of vehicles clearing the intersection in Synchro model reaches maximum value. Meanwhile, when the saturation continues to rise to 1.1, the number of vehicles clearing the intersection in the proposed model is approaching stability. In the proposed model, the number of vehicles clearing the intersection when stable is 53.3% bigger than in Synchro model, suggesting that the proposed model is able to significantly improve traffic capacity.

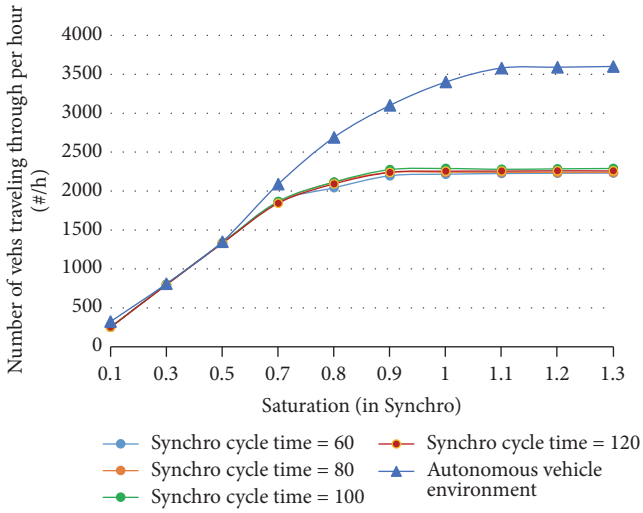


Figure 6: Comparison of number of vehicles passing per hour.

5. Conclusions

This paper presented a method of traffic control in the environment of automatic driving. Platoon is employed as the basic control unit. First, we establish the basic cellular automaton model. Then we optimized the control strategy based on rolling time window and greedy algorithm. Finally in this article, we employed the NetLogo multiagent simulation platform to simulate the proposed model. The simulation results show that, compared with Synchro model, the proposed model is able to reduce delays and the number of stops more effectively and to substantially increase the capacity of the intersection, which demonstrates the superiority of the proposed model.

This paper proposed preliminary simulation results and analysis for the proposed method at an isolated intersection. More extensive simulation experiments and field tests are required to be conducted in the future to assess the effectiveness of the proposed model under various traffic flow patterns. Corridor-wide evaluation of the proposed model's effectiveness should also be studied in future researches.

Key Variables (Notations) Used in the Formulations

i: Vehicle i
j: Platoon j
t: Time point
$V_i^j(t)$: The speed of vehicle i in platoon j at time t (m/s)
$G_i^j(t)$: The headway between vehicle i in platoon j and the vehicle in the front (m)
$D_i^j(t)$: Distance from the vehicle i in platoon j to the intersection (m)
$\sigma_i^j(t)$: Binary variable to represent if a vehicle gets right-of-way to travel through the intersection. If so, $\sigma_i^j(t) = 1$. If not $\sigma_i^j(t) = 0$
T_i^j: The maximum acceptable waiting time at the intersection (s)
$\Delta X_i^j(t)$: The moving distance of vehicle i in platoon j at time t (m)
$V_{\max}$: The maximum speed of the vehicle (m/s)
n: The maximum number of vehicles in the platoon (#veh)
h_c: Critical time headway (s)
$\rho(t)$: Average density at time t (pcu/km)
$N(t)$: Sum of the vehicles at time t (#veh).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research is supported by the National Natural Science Foundation of China under Grant no. 61773077, no. 51408065, and no. 51308336, the Fundamental Research Funds for the Central Universities of Shanghai Jiao Tong University (Projects 15JCZZ05 and 16JXYB08), the Open Fund of the State Engineering Laboratory of Highway Maintenance Technology (Changsha University of Science & Technology, Project KFJ160106), and the Urban China Initiative (UCI) Grant Project.

References

[1] H.-Y. Berg, J. Ifver, and M. Hasselberg, "Public health consequences of road traffic injuries - Estimation of seriously injured persons based on risk for permanent medical impairment," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 38, pp. 1–6, 2016.

[2] Y. C. Huang, R. Cheng-Shyuan, C. Jung-Fang et al., "Characteristics and outcomes of patients injured in road traffic crashes and transported by emergency medical services," *International Journal of Environmental Research & Public Health*, vol. 13, no. 2, article 236, 2016.

[3] C. Ding, D. Wang, C. Liu et al., "Exploring the influence of built environment on travel mode choice considering the mediating effects of car ownership and travel distance," *Transportation Research Part A Policy & Practice*, vol. 100, pp. 65–80, 2017.

[4] D. Schrank, B. Eisele, and T. Lomax, "2014 Urban mobility report: powered by Inrix Traffic Data," Tech. Rep., Southwest Region University Transportation Center, Texas, Tex, USA, 2015.

[5] W. Wu, W. Ma, K. Long, H. Zhou, and Y. Zhang, "Designing sustainable public transportation: Integrated optimization of bus speed and holding time in a connected vehicle environment," *Sustainability*, vol. 8, no. 11, article 1170, 2016.

[6] C. Ding, C. Liu, Y. Zhang, J. W. Yang, and Y. P. Wang, "Investigating the impacts of built environment on vehicle miles traveled and energy consumption: differences between commuting and non-commuting trips," *Cities*, vol. 68, pp. 25–36, 2017.

[7] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92–102, 2009.

[8] N. Kehtarnavaz, N. Griswold, K. Miller, and P. Lescoe, "A transportable neural-network approach to autonomous vehicle following," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 694–702, 1998.

[9] J. E. Naranjo, C. González, R. García, and T. De Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 438–450, 2008.

[10] F. Jiménez, M. Clavijo, J. E. Naranjo, and Ó. Gómez, "Improving the Lane Reference Detection for Autonomous Road Vehicle Control," *Journal of Sensors*, vol. 2016, Article ID 9497524, 2016.

[11] K. Tsukamoto, Y. Oie, H. Kremo, O. Altintas, H. Tanaka, and T. Fujii, "Implementation and performance evaluation of distributed autonomous multi-hop vehicle-to-vehicle communications over TV white space," *Mobile Networks and Applications*, vol. 20, no. 2, pp. 203–219, 2015.

[12] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Impact of communication erasure channels on the safety of highway vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1456–1468, 2015.

[13] W. Wu, P. K. Li, and Y. Zhang, "Modelling and simulation of vehicle speed guidance in connected vehicle environment," *International Journal of Simulation Modelling*, vol. 14, no. 1, pp. 145–157, 2015.

[14] H. Peng, d. li, Q. Ye et al., "Resource allocation for cellular-based inter-vehicle communications in autonomous multiplatoons," *IEEE Transactions on Vehicular Technology*, vol. 99, pp. 1–15, 2017.

[15] F. Jiménez, J. E. Naranjo, and Ó. Gómez, "Autonomous collision avoidance system based on accurate knowledge of the vehicle surroundings," *IET Intelligent Transport Systems*, vol. 9, no. 1, pp. 105–117, 2015.

[16] P. Koopman and M. Wagner, "Autonomous vehicle safety: an interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[17] C. Wuthishuwong, A. Traechtler, and T. Bruns, "Safe trajectory planning for autonomous intersection management by using vehicle to infrastructure communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1–12, 2015.

[18] S. Gong, J. Shen, and L. Du, "Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon," *Transportation Research Part B: Methodological*, vol. 94, pp. 314–334, 2016.

[19] V. A. C. van den Berg and E. T. Verhoef, "Autonomous cars and dynamic bottleneck congestion: the effects on capacity, value of time and preference heterogeneity," *Transportation Research Part B: Methodological*, vol. 94, pp. 43–60, 2016.

[20] Y. Xie, H. Zhang, N. H. Gartner, and T. Arsava, "Collaborative merging strategy for freeway ramp operations in a connected and autonomous vehicles environment," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 21, no. 2, pp. 136–147, 2017.

[21] S. E. Shladover, "Cooperative (rather than autonomous) vehicle-highway automation systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 1, pp. 10–19, 2009.

[22] J. Hu, L. Kong, W. Shu, and M.-Y. Wu, "Scheduling of connected autonomous vehicles on highway lanes," in *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM '12)*, pp. 5556–5561, Anaheim, Calif, USA, December 2012.

[23] J. Wei, J. M. Dolan, and B. Litkouhi, "Autonomous vehicle social behavior for highway entrance ramp management," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '13)*, pp. 201–207, Gold Coast, QLD, Australia, June 2013.

[24] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008.

[25] Z. Li, M. V. Chitturi, L. Yu, A. R. Bill, and D. A. Noyce, "Sustainability effects of next-generation intersection control for autonomous vehicles," *Transport*, vol. 30, no. 3, pp. 342–352, 2015.

[26] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.

Hindawi
Journal of Sensors
Volume 2017, Article ID 8568613, 10 pages
https://doi.org/10.1155/2017/8568613

Research Article

Finding Optimal Team for Multiskill Task Based on Vehicle Sensors Data

Bowen Du, Qian Tao, Feng Zhu, and Tianshu Song

State Key Laboratory of Software Development Environment, School of Computer Science and Engineering and IRI, Beihang University, Beijing, China

Correspondence should be addressed to Qian Tao; qiantao@buaa.edu.cn

Received 28 July 2017; Accepted 12 September 2017; Published 31 October 2017

Academic Editor: Jia Hu

These days, with the increasingly widespread employment of sensors, particularly those attached to vehicles, the collection of spatial data is becoming easier and more accurate. As a result, many relevant areas, such as spatial crowdsourcing, are gaining ever more attention. A typical spatial crowdsourcing scenario involves an employer publishing a task and some workers helping to accomplish it. However, most of previous studies have only considered the spatial information of workers and tasks, while ignoring individual variations among workers. In this paper, we consider the Software Development Team Formation (SDTF) problem, which aims to assemble a team of workers whose abilities satisfy the requirements of the task. After showing that the problem is NP-hard, we propose three greedy algorithms and a multiple-phase algorithm to approximately solve the problem. Extensive experiments are conducted on synthetic and real datasets, and the results verify the effectiveness and efficiency of our algorithms.

1. Introduction

These days, with the development of sensors (especially vehicle sensors and mobile sensors) [1–3], it is increasingly simple to acquire spatial and temporal information [4, 5].

Many studies based on vehicle sensors data have been conducted in recent years [6–8]. As a result, many applications now provide services based on users' real-time spatial information and these are becoming ever popular. Among these applications, some focus on crowdsourcing services that use spatial information. These applications usually require some workers to help an employer to accomplish a task. For example, Uber (https://www.uber.com) organizes drivers and provides users with a convenient taxi service, whereas +Meituan (http://www.meituan.com) provides a credible and fast food-delivery service. This area, called spatial crowdsourcing, is attracting significant attention.

The task assignment problem is one of the fundamental concerns in spatial crowdsourcing. For example, real-time taxi-calling platforms, such as Uber and Didi Chuxing [9], always need to assign each taxi-calling task to a suitable taxi (i.e., a crowd worker). An incorrect assignment may cause taxis to be dispatched to far-away places, which results in a slow response time and the loss of the platform. Many studies on the task assignment problem have been published in recent years [10–12]. However, most of them only consider the spatial information of tasks and workers, while ignoring the individual variations among workers. Namely, different people may excel or struggle with different tasks, and tasks also contain certain requirements for which some workers may be inadequate.

Take Figure 1 as an example. Suppose a website development task requires coders skilled in .NET, SQL, and HTML to assemble at the location of the origin, and there are three coders available (whose skills are presented in Figure 1). Although coder c_3 is located closer to the origin than c_1 and c_2, hiring c_3 will not help finish the task. In other words, it is necessary to further consider individual variations among different workers and special requirements of tasks.

As in [13, 14], each worker is associated with a set of skills representing their strengths. Tasks are also associated with a set of skills representing their special requirements.

R-trees [15] are a classical index structure for multidimensional data. Derived from B-tree, the data in an R-tree are stored in leaf nodes and all leaves are located in the same level of the tree. Every internal node contains between m and

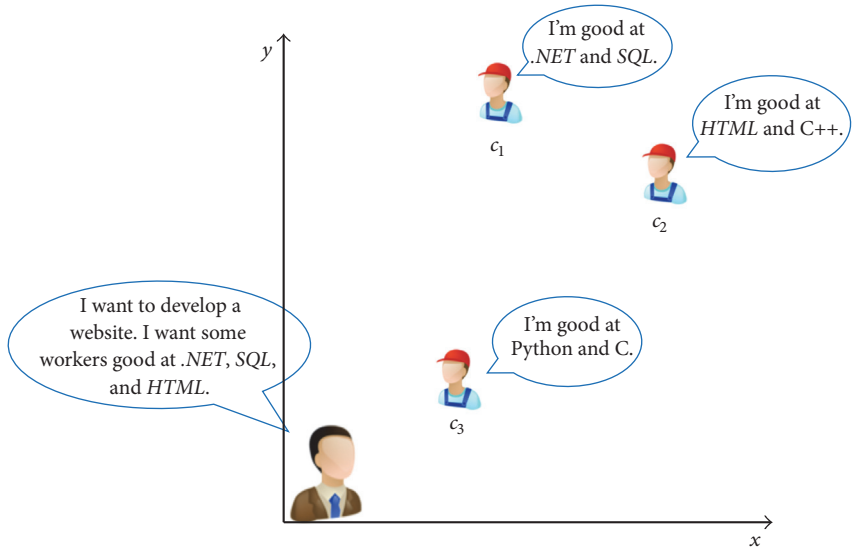


Figure 1: Example: variations among workers.

M child entries, and every leaf node contains between m and M data entries, where M is usually related to the size of disk pages, and m is predefined such that $m \leq M/2$. The tree is specially structured such that the children of a node overlap with few data from other nodes. Using an R-tree, we can dynamically insert/update/delete nodes, and rapidly search for all nodes located in a given rectangle.

The objective of our problem consists of two parts. First, workers need to move to the location of the task but receive no reward for this movement. In consideration of the workers, we attempt to reduce the gratuitous moving distance. Second, the employer wishes spend the minimum amount necessary to accomplish the task. In consideration of employers, we attempt to obtain a team at the lowest cost, on condition that the skill requirement is satisfied. As the problem definition in Section 2 shows, the objective of our work contains not only the distance between the task and workers, but also the total cost.

Contributions. In summary, our contributions are as follows:

(i) We propose a new Software Development Team Formation (SDTF) problem and prove that it is NP-hard.

(ii) Three greedy algorithms are provided to solve the SDTF problem.

(iii) We employ a multiphase algorithm based on R-trees.

(iv) We verify the effectiveness and efficiency of the proposed algorithms through extensive experiments on synthetic and real datasets.

Compared with our previous work [16], we propose a novel multiple-phase algorithm by using the index structure of R-trees. Additional experiments are also conducted on synthetic and real datasets.

The remainder of this paper is organized as follows. In Section 2, the problem is formally defined and proved to be NP-hard. In Section 3 three greedy algorithms are provided to solve the SDTF problem. In Section 4, we propose a multiple-phase algorithm based on R-trees. Extensive experiments on real datasets are described in Section 5. Previous work related to our problem is presented in Section 6, and the conclusions to this study are presented in Section 7.

2. Problem Statement

First, we introduce the two basic concepts of a task and a coder. We then formally define the Software Development Team Formation (SDTF) problem.

Definition 1 (task). A task t is defined as $\langle S, L\rangle$, where $t.S$ is a set of skills that are indispensable to complete the software development task t, and $t.L$ is the location specified to meet up and talk about task t, which, for example, can be described by longitude and latitude.

Similar to the definition of a task, a coder is formally defined as follows.

Definition 2 (coder). A coder c is defined as $\langle S, L, P\rangle$, where $c.S$ is a set of skills mastered by coder c, $c.L$ is the location of coder c, described similarly to that of a task t, and $c.P$ is the price of coder c.

Briefly, a team of coders is feasible for a task if the coders in the team can collaboratively accomplish the task.

Definition 3 (feasible team). A team T is defined as a set of coders $\{c_1, c_2, \ldots, c_{|T|}\}$. T is a feasible team for task t, if $\bigcup_{c_i \in T} c_i.S \supseteq t.S$.

Example 4. Suppose that we have a task t concerning website development, where $t.S$ = {LINUX, DATABASE, CSS, HTML} and a universal set of coders $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$.

Table 1: Coder profile.

Coder	*Skill*	*Price*	*Distance*
c_1	{PYTHON, CSS, C++}	70	1000
c_2	{CSS, LINUX, HTML}	60	5000
c_3	{LINUX, HTML}	50	7000
c_4	{DATABASE, JAVA}	55	2000
c_5	{PYTHON, DATABASE, LINUX}	65	6000
c_6	{HTML, C#}	60	3000

The skill set of every $c \in C$ is listed in Table 1. Team $T = \{c_2, c_4\}$ is a feasible team because $\bigcup_{c \in T} c.S = \{$CSS, LINUX, HTML, DATABASE, JAVA$\}$, which is a superset of $t.S = \{$LINUX, DATABASE, CSS, HTML$\}$.

We finally define our problem as follows.

Definition 5 (SDTF problem). Given a task t with a set of skills $t.S$ and a location $t.L$ and a universal set of coders $C = \{c_1, c_2, \ldots, c_{|C|}\}$, each with skill set $c_i.S$, location $c_i.L$, and price $c_i.P$, $1 \le i \le |C|$, we wish to find a team $T \subseteq C$ satisfying $\bigcup_{c \in T} c.S \supseteq t.S$, and we minimize Cost $= \alpha \cdot \max_{c \in T} |c.L, t.L| + (1 - \alpha) \cdot \sum_{c \in T} c.P$, where $|c.L, t.L|$ represents the distance between the location of coder c and task t and $\alpha \in (0, 1)$ is a parameter to adjust the weight of distance and price.

Theorem 6. *The SDTF problem is NP-hard.*

Proof. We prove the theorem by showing that a special case of the SDTF problem can be reduced from the weighted set cover problem. An instance of a weighted set cover problem consists of a set $U = \{1, 2, \ldots, n\}$ and a set $S = \{S_1, S_2, \ldots, S_{|U|}\}$, where $S_i \subseteq U$ for $1 \le i \le |U|$. Each S_i is associated with a positive value W_{S_i}, which can be viewed as the weight of S_i. The weighted set cover optimization problem aims to find a subset S^* of S satisfying $\bigcup_{S_i \in S^*} S_i \supseteq U$ minimizing $\sum_{S_i \in S^*} W_{S_i}$.

We consider a special case of the SDTF problem in which the task and coders are located in the same position, and the skill set of the task is the universal set of all skills. To reduce the weighted set cover problem to the special-case SDTF problem, we observe that each element in U corresponds to a skill in $t.S$, each element in S_i corresponds to a skill in $c_i.S$, and the weight of S_i corresponds to the price of c_i. As the task and all coders are at the same location, for every team T, $\max_{c \in T} |c.L, t.L| = 0$, and we need only minimize $\sum_{c \in T} c.P$. Obviously, there exists a solution to the weighted set cover problem if and only if there exists a solution to the special-case SDTF problem, and we can obtain an instance of the special-case SDTF problem from the instance of weighted set cover problem in polynomial time. Therefore, the general case of the SDTF problem is NP-hard. □

3. Greedy Solutions for SDTF

In this section, we present three greedy algorithms to solve the SDTF problem. The first two algorithms greedily choose the nearest/cheapest coder who can cover at least one uncovered skill. Because they only consider optimizing part of the objective function, the solution is sometimes not good enough. Thus we propose a third greedy algorithm that considers both price and distance when choosing a new coder.

input: Task t, Coders $C = \{c_1, c_2, \ldots, c_{|C|}\}$
output: A feasible team T
(1) $T \leftarrow \emptyset$;
(2) **while** T *is not feasible* **do**
(3) $\quad c \leftarrow \text{argmin}_{c \in C \&\& c.S \cap t.S \neq \emptyset} (c.P / |c.S \cap t.S|)$;
(4) $\quad T \leftarrow T \cup \{c\}$;
(5) $\quad t.S \leftarrow t.S - c.S$;
(6) **return** T

Algorithm 1: PF-SDTF.

3.1. Price First-SDTF Greedy Algorithm. The idea of the first greedy algorithm is to repeatedly add the cheapest coder to the team until the team is feasible. The whole procedure of this price first- (PF-) SDTF is illustrated in Algorithm 1. We assume that there exists at least one feasible team.

Considering that skills not in the skill set of the task contribute nothing to the accomplishment of the task, the term "cheapest coder" must be treated carefully. Here, we define the Average Price on Uncovered Intersecting Skills to describe how a coder contributes to the price part of the objective function:

$$\text{APUIS}(t, c) = \frac{c.P}{|c.S \cap t.S_u|}, \tag{1}$$

where S_u is the uncovered skill set of task t. We can see that APUIS describes how a coder influences the price part of the objective function if we add him/her to the final team. Choosing a coder with lower APUIS means we can satisfy the requirement of the skills with a lower total price. Note that when there is no intersection between the skill set of the worker and the uncovered skill set, APUIS will be infinity. Because we greedily choose the worker with the lowest APUIS, we omit this special case in (1).

In line (1) of Algorithm 1, we initialize an empty team T. In lines (2)–(5), when T is not feasible, we find a coder c who can cover at least one uncovered skill of task t and has the lowest $c.P/|c.S \cap t.S|$ value, add c to team T, and update $t.S$. Ties are broken by distance first, then arbitrarily. In line (6), we return the resulting feasible team T.

3.2. Distance First-SDTF Greedy Algorithm. The idea of distance first- (DF-) SDTF is to repeatedly add the nearest coder to the team until the team is feasible. The framework of DF-SDTF is similar to that of PF-SDTF. In each iteration, we find the nearest coder c_n who can cover at least one uncovered skill of task t; that is,

$$c_n = \underset{c \in C \&\& c.S \cap t.S_u \neq \emptyset}{\text{argmin}} |c.L, t.L|, \tag{2}$$

input: Task t, Coders $C = \{c_1, c_2, \ldots, c_{|C|}\}$
output: A feasible team T
(1) $T \leftarrow \emptyset$;
(2) **while** T *is not feasible* **do**
(3) $\quad c \leftarrow \text{argmin}_{c \in C \&\& c.S \cap t.S \neq \emptyset} |c.L, t.L|$;
(4) $\quad T \leftarrow T \cup \{c\}$;
(5) $\quad t.S \leftarrow t.S - c.S$;
(6) **return** T

Algorithm 2: DF-SDTF.

input: Task t, Coders $C = \{c_1, c_2, \ldots, c_{|C|}\}$
output: A feasible team T
(1) $T \leftarrow \emptyset$;
(2) **while** T *is not feasible* **do**
(3) $\quad c \leftarrow \text{argmax}_{c \in C \&\& c.S \cap t.S \neq \emptyset} \text{Utility}(c, t, T)$;
(4) $\quad T \leftarrow T \cup \{c\}$;
(5) $\quad t.S \leftarrow t.S - c.S$;
(6) **return** T

Algorithm 3: DP-SDTF.

where S_u is the uncovered skill set of task t. The whole procedure of DF-SDTF is illustrated in Algorithm 2. We assume that there exists at least one feasible team.

In line (1), we initialize an empty team T. In lines (2)–(5), when T is not feasible, we find the nearest coder c who can cover at least one uncovered skill of task t, add c to team T, and update $t.S$. Ties are broken by price first, then arbitrarily. In line (6), we return the resulting feasible team T.

3.3. Distance Price-SDTF Greedy Algorithm. The aforementioned two greedy algorithms are not effective, because they only try to optimize part of the objective function. To optimize both distance and price at every iteration, we design a utility function Utility. Given a task t, current team T, and coder c, the definition of Utility is

$$\text{Utility}(c, t, T) = \frac{|c.S \cap t.S|}{\alpha \cdot \Delta D(c, t, T) + (1 - \alpha) \cdot |c.P|}, \quad (3)$$

where $\Delta D(c, t, T)$ represents the increment in the maximum distance if c is added to team T, that is, $\Delta D(c, t, T) = |c.L, t.L| - \max_{c' \in T} |c'.L, t.L|$ if $|c.L, t.L| > \max_{c' \in T} |c'.L, t.L|$, and $\Delta D(c, t, T) = 0$ if $|c.L, t.L| \leqslant \max_{c' \in T} |c'.L, t.L|$. In fact, the value of $\alpha \cdot \Delta D(c, t, T) + (1 - \alpha) \cdot |c.P|$ in (3) is the increment in the objective function.

Using this utility function, we have a third greedy algorithm, Distance Price- (DP-) SDTF. The whole procedure of DP-SDTF is illustrated in Algorithm 3. We assume that there exists at least one feasible team.

In line (1), we initialize an empty team T. In lines (2)–(5), when T is not feasible, we find a coder c who gives the highest utility. Ties are broken by distance first, then arbitrarily. In line (6), we return the resulting feasible team T.

4. Multiple-Phase R-Tree Algorithm

In this section, we introduce an algorithm based on the R-tree data structure. Considering that some previous work has applied R-trees in Nearest Neighbor (NN) searching [17, 18], a naïve idea is to use an R-tree to accelerate the NN search in the DF-SDTF algorithm proposed in Section 3.2. However, this simple use of R-trees can only accelerate the search speed and does not help optimize the final cost. As our experiments will show, the DF-SDTF algorithm performs worse than the DP-SDTF algorithm proposed in Section 3.3. The above situation requires us to find an algorithm that is both efficient and effective in solving the SDTF problem.

Our original algorithm derives from an intuitive observation: if we query all nodes located in the square whose centroid is at the location of the task and whose side length is $2 \cdot l$, the distance between the task and the nodes in the result set will be at most $\sqrt{2} \cdot l$. This characteristic provides an applicable tool for the distance part of our objective function. By choosing a rectangle with suitable sides, we obtain a set of candidate coders who are close to the location of the task. The price part of the objective function can also be optimized if we employ a proper strategy to choose the next coder from the candidate coder set.

Based on the above observation, we propose the Multiple-Phase R-tree (MPR) algorithm. The main idea of our algorithm is as follows.

(1) Initialize a new R-tree and insert all coders into the tree.

(2) In each phase, obtain a candidate set of coders by querying all nodes located in the square whose centroid is at the location of the task.

(3) Sort all coders in the candidate set in descending order of APUIS. For each coder, add him/her to the final team T if his/her skills can cover at least one uncovered skill in the task.

(4) If team T is not feasible, return to step (2) and use a square with longer sides.

In detail, we generate the list of side lengths by uniformly dividing the maximum distance between the coders and the task. Given a parameter n_p denoting the number of phases, we first scan the whole set of coders and calculate the maximum distance between the coders and the task, maxDis. Then, we iteratively start a phase by using a square with side length $\text{maxDis}/n_p, 2 \cdot \text{maxDis}/n_p, \ldots, \text{maxDis}$, until we obtain a feasible team.

The pseudocode of our MPR algorithm is shown in Algorithm 4. First, we initialize the team T and find the maximum distance between the coders and the task in lines (1)-(2). Then, we calculate the step size of the sides between two phases in line (3). In each phase (iteration in lines (5)–(9)), we first query all nodes located in the square whose centroid is at the location of the task and whose side length is $2 \cdot i \cdot \text{step}$. We then alternately add coders with the minimum APUIS (lines (7)–(9)). Similarly, ties are broken by distance first, then arbitrarily. In line (10), we return the resulting feasible team T.

input: Task t, R-tree R, Coders
$C = \{c_1, c_2, \ldots, c_{|C|}\}$, number of phases n_p
output: A feasible team T
(1) $T \leftarrow \emptyset$;
(2) maxDis $\leftarrow$ maximum distance between coders and the task;
(3) step $\leftarrow$ maxDis$/n_p$;
(4) **for** $i \leftarrow 1$ *to* n_p **do**
(5) $\quad C_{\text{can}} \leftarrow R$.search$(t.L, 2 \cdot i \cdot \text{step})$;
(6) $\quad$ **while** T *is not feasible* **do**
(7) $\quad\quad c \leftarrow \text{argmin}_{c \in C_{\text{can}} \&\& c.S \cap t.S \neq \emptyset}(c.P/|c.S \cap t.S|)$;
(8) $\quad\quad T \leftarrow T \cup \{c\}$;
(9) $\quad\quad t.S \leftarrow t.S - c.S$;
(10) **return** T

Algorithm 4: MPR.

Table 2: Synthetic dataset.

Factor	Setting
α	0.1, 0.2, 0.3, 0.4, **0.5**, 0.6, 0.7, 0.8, 0.9
$\lvert t.\text{Skills}\rvert$	5, 6, 7, 8, 9, **10**, 11, 12, 13, 14, 15
$\lvert \bigcup_{c \in C} c.\text{Skills}\rvert$	100, 110, 120, 130, 140, **150**, 160, 170, 180, 190, 200
$\lvert C\rvert$	1 W, 2 W, 3 W, 4 W, **5 W**, 6 W, 7 W, 8 W, 9 W, 10 W

5. Evaluation

We applied our four algorithms to synthetic and real datasets. The algorithms were implemented in C++, and the experiments were performed on a machine with an Intel i7-4710mq 2.50 GHZ 4-core CPU and 8 GB memory.

5.1. Datasets. We use real and synthetic datasets to evaluate our algorithms. The real dataset is taken from CSTO (http://www.csto.com/) and includes 2033 active coders. In the CSTO dataset, each task is associated with a set of skills needed to complete a software development task, and each coder is associated with a set of skills and an average price that can be deduced from the history data. As few coders have associated price information (because many coders have not any completed tasks), we analyze the price distribution using coders associated with price information. Except for some expensive coders, the price of a coder is uniformly distributed in the range 0–5000 and is unrelated to the number of mastered skills. As the CSTO data are not associated with location information, we generate coordinates for each coder according to a uniform distribution.

For the synthetic data, based on our observations of the real dataset, we generate the price $c.P$ of coder c following a uniform distribution. We assume that each coder has 5–25 skills, which is common in practice. The distance from each coder to the task is generated according to a uniform distribution. The statistics and configuration of synthetic data are illustrated in Table 2, where the default settings are marked in bold font.

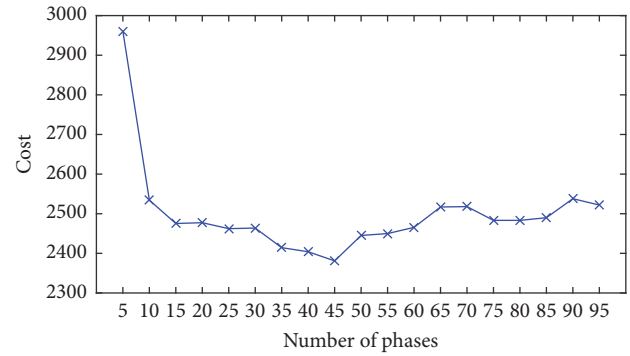


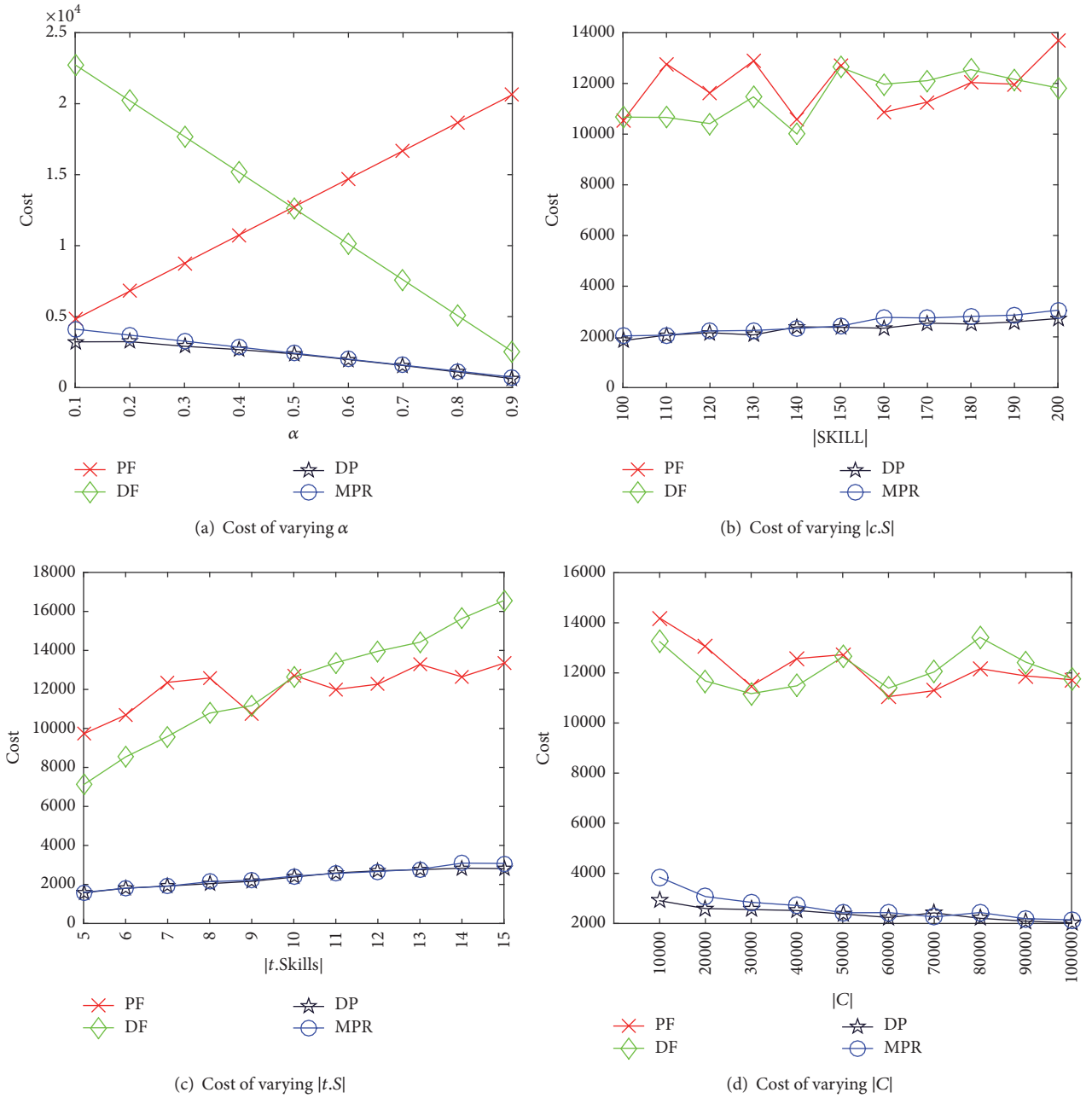
Figure 2: Cost of varying n_p.

5.2. Number of Phases in MPR Algorithm. In the MPR algorithm, we introduce a new parameter representing the total number of phases, n_p. Before conducting experiments on the synthetic and real data, we determined an appropriate value of n_p to ensure better performance of the MPR algorithm. We first generate a synthetic dataset with the default settings to preexamine how n_p affects the performance of the MPR algorithm. The results are shown in Figure 2 for n_p from 5 to 100. According to these results, we use $n_p = 45$ in all subsequent MPR experiments.

5.3. Experiments on Synthetic Datasets. The experimental results using the synthetic data are shown in Figures 3 and 4. In this section, we measure the effectiveness and efficiency of these four algorithms and analyze how various parameters affect the results given by each algorithm.

Effectiveness of Proposed Algorithms. Figure 3 shows the effectiveness of our four algorithms. The DP-SDTF and MPR algorithms offer similar performance and outperform both DF-SDTF and PF-SDTF.

Efficiency of Proposed Algorithms. Figure 4 shows the efficiency of our four algorithms. We can observe that although DP-SDTF and MPR have similar cost results, MPR is faster



(a) Cost of varying α

(b) Cost of varying $|c.S|$

(c) Cost of varying $|t.S|$

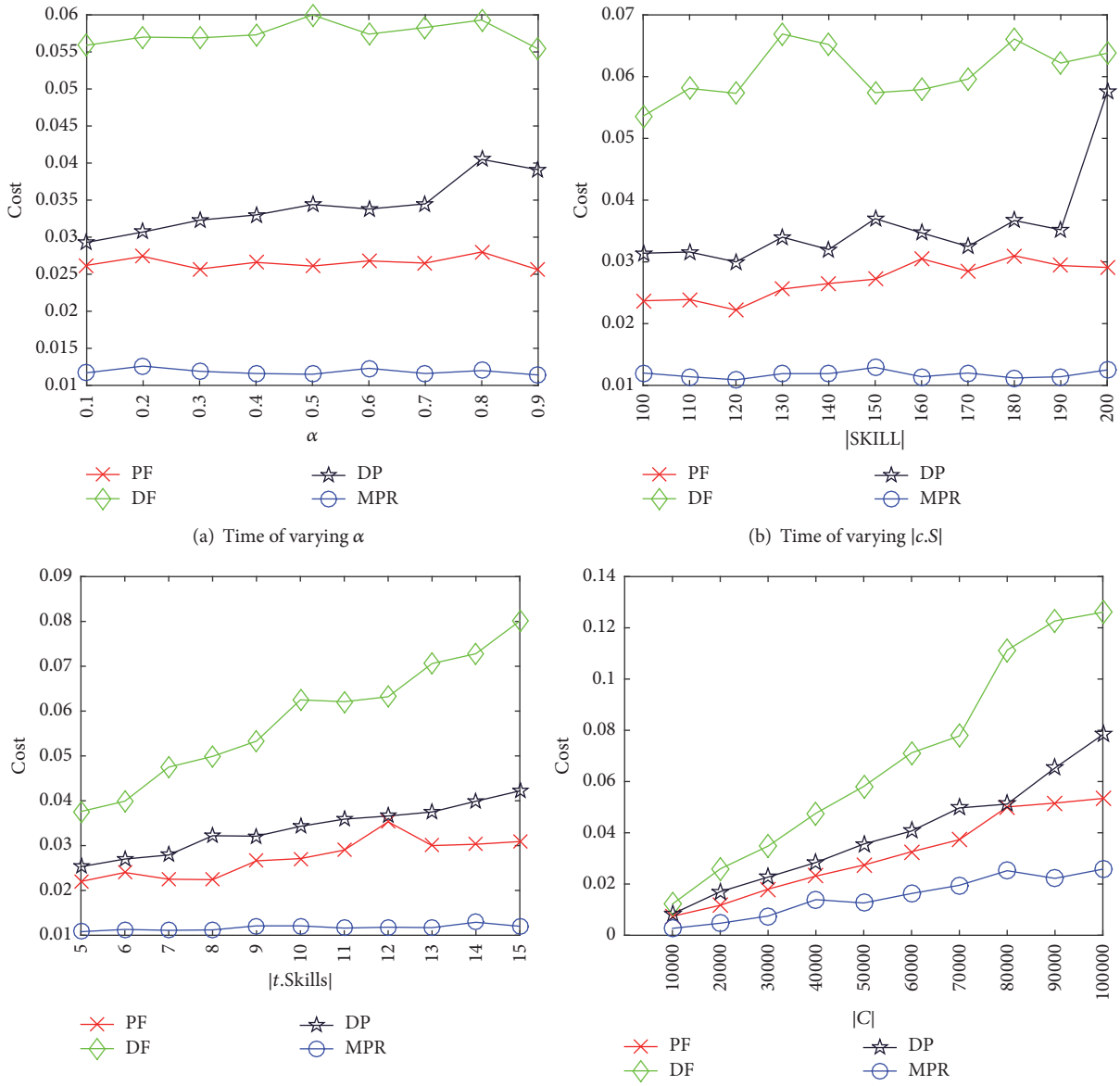
(d) Cost of varying $|C|$

Figure 3: Results on synthetic data.

than DP-SDTF. This is because we use the R-tree to prune some unvalued nodes and accelerate the process of the query. We can also observe how the restriction of skill satisfaction affects the running time of four algorithms. Although PF-SDTF, DF-SDTF, and DP-SDTF all use greedy strategy and their structures are similar, DF-SDTF algorithm consumes more time than that of PF-SDTF and DP-SDTF algorithms. This is because DF-SDTF algorithm only considers the effect of the distance. As a result DF-SDTF needs more coders to make the team feasible, resulting in more iterations than the PF-SDTF and DP-SDTF algorithms.

Effect of α. Figure 3(a) shows the effectiveness of varying α. As α varies from 0.1 to 0.9, the cost of DP-SDTF decreases smoothly, indicating that $\sum_{c \in T} c.P$ contributes more than $\max_{c \in T} |c.L, t.L|$. Because the DF-SDTF (PF-SDTF) algorithm only considers distance (price), when α is high (low), the performance is similar to that of the DP-SDTF and MPR algorithms. However, as α decreases (increases), the performance of DF-SDTF (PF-SDTF) becomes worse.

Effect of $|c.S|$, $|t.S|$, *and* $|C|$. The effect of varying $|c.S|$, $|t.S|$, and $|C|$ is illustrated in Figures 3(b), 3(c), and 3(d). Because



(a) Time of varying α

(b) Time of varying $|c.S|$

(c) Time of varying $|t.S|$

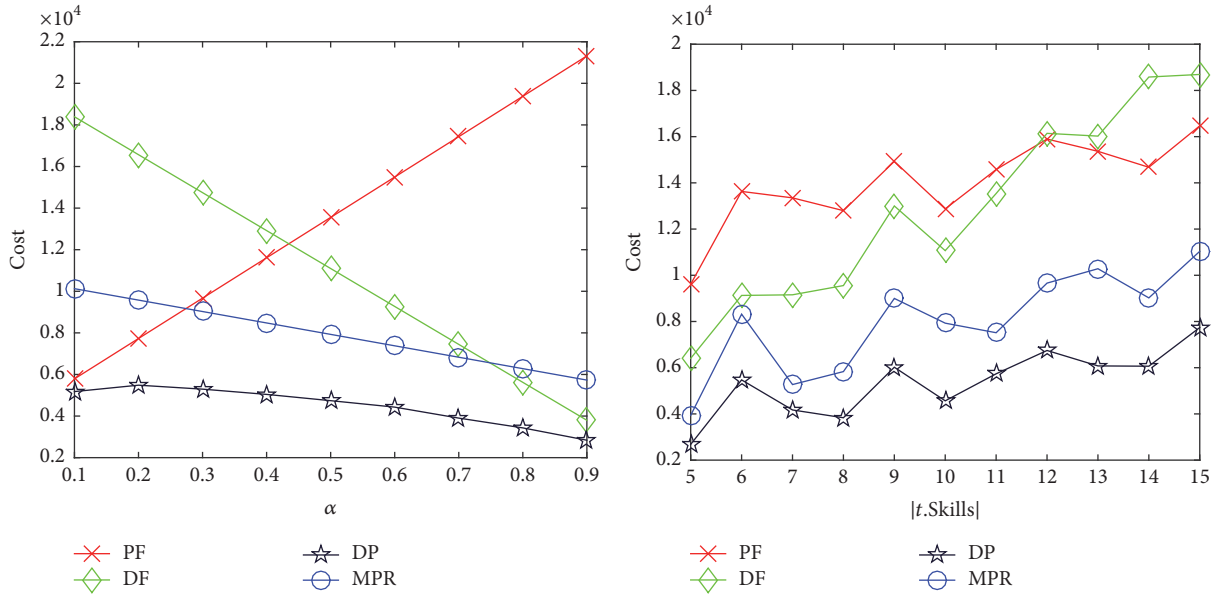
(d) Time of varying $|C|$

Figure 4: Running time on synthetic data.

the default setting of α is 0.5, finding a good team requires distance and price to be considered simultaneously. We can observe that the DP-SDTF and MPR algorithms perform better, with the DF-SDTF and PF-SDTF costing 3 to 4 times more.

5.4. Experiments on the Real Dataset. The experimental results using the real dataset are shown in Figure 5. Figure 5(a) shows the effects of varying α, and Figure 5(b) shows the effects of varying $|t.S|$. Varying α produces a similar effect as with the synthetic dataset. When varying $|t.S|$, the costs of the four algorithms oscillates, probably because of the structure of the CSTO dataset. Unlike the experiments on synthetic data, the MPR algorithm performs worse than DP-SDTF but still outperforms DF-SDTF and PF-SDTF. This is probably because, in real datasets, different skills may make different contributions, leading to a gap between results with synthetic data and real data.

Comparison with the Exact Result. Because the SDTF problem is NP-hard, we only conduct small-size experiments to compare the output of our DP-SDTF and MPR algorithms with the exact solution. The setting is $t.S = 5$ and $|C| = 300$, where coders are randomly chosen from the real dataset. The experimental results are shown in Figure 6. We can observe that the performance of DP-SDTF is similar to that of the exact algorithm, but the cost of the MPR algorithm is 1.25 to 1.5 times the exact minimum cost.



(a) Cost of varying α

(b) Cost of varying $|t.S|$

Figure 5: Results on real data.

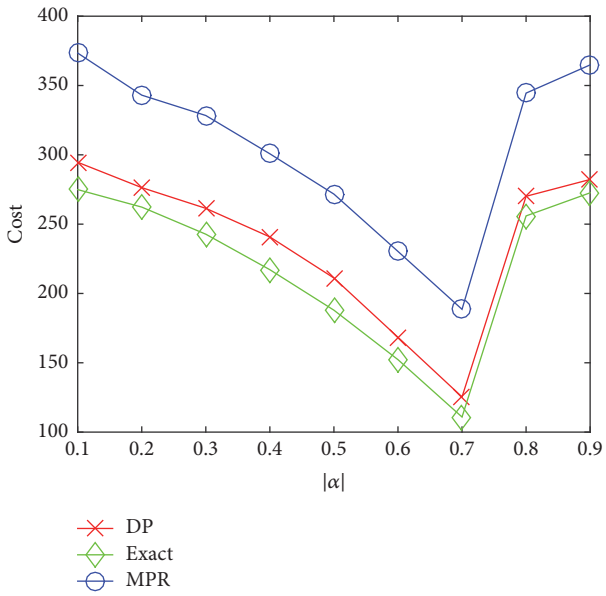


Figure 6: DP-SDTF versus MPR versus exact results.

Conclusion. From the extensive experiments conducted on both real and synthetic data to validate our four algorithms, we found that DF-SDTF (PF-SDTF) algorithm, which focuses on the distance (price) part of the objective function, performs better with larger (smaller) values of α. The DP-SDTF algorithm gives the best performance among the four algorithms discussed here because it considers both parts of the objective function. The fourth algorithm, MPR, accelerates the query process with little increase in the cost, which is more applicable in practice.

6. Related Work

The SDTF problem tackled in this paper covers the domains of *Team Formation* and *Spatial Crowdsourcing*. On the one hand, the SDTF problem can be simplified to the task assignment problem if we ignore the skill constraint. On the other hand, it is exactly the most distinctive requirement that the skills of a team must cover the skills of the task. Previous work related to these two domains is introduced in the following subsections.

6.1. Team Formation. The team formation problem was first proposed in [19]. The problem requires a team of workers that (1) its skills satisfy the requirement of the task; (2) the overall communication cost is minimum. In this paper, the NP-hard nature of this problem is also proved. The problem has been extended by associating each worker with a capacity [20], which is the maximum number of tasks assigned to the worker. To solve the capacitated team formation problem, two approximation algorithms with proved guarantees were proposed. Different from [19, 20], which only include a single task, the team formation problem has been considered with multiple tasks and workers in both offline and online scenarios [21]. While the above-mentioned studies attempt to optimize the overall communication cost, the workload can be balanced among workers by treating the communication cost as a restrictive constraint [22]. As the above shows, most studies on team formation focus on skills satisfaction in communicative graphs, while ignoring the influence of spatial information.

6.2. Spatial Crowdsourcing. The problem studied in this paper is an extension of the task assignment problem in spatial crowdsourcing, known as the server-assigned task assignment problem [10, 11], in which workers cannot reject

the assigned tasks. Recently, task assignment in real-time spatial crowdsourcing has also been studied by the online algorithmic model [12, 23]. Based on the original task assignment problem, both [24, 25] study the conflict-aware task assignment problem, in which tasks may conflict with each other and thus cannot be assigned to the same worker. In addition, the work [26] not only considers spatiotemporal conflicts of tasks but also schedules the plan that each worker complete tasks [26]. Furthermore, Kazemi et al. propose the quality-based task assignment problem [27], which utilizes majority voting techniques to guarantee the quality of task assignment results [28–30].

Although [13, 14] integrate the task assignment problem and team formation problem and propose a two-level-based framework to solve the problem, there are two main differences between [13, 14] and our work: (1) there is no capacity constraint in our work, which means that there are more candidates in the search space; (2) the objective of our work considers both the distance between the task and workers and the overall cost, whereas [13, 14] only attempt to minimize the overall cost.

7. Conclusion

With the development of sensors, particularly vehicle sensors and mobile sensors, spatial crowdsourcing is gaining ever more attention. In this paper, we propose a novel spatial crowdsourcing problem called Software Development Team Formation (SDTF). We prove that SDTF is NP-hard and design three greedy algorithms and an index-based algorithm to solve the SDTF problem. The first two greedy algorithms, DF-SDTF and PF-SDTF, only consider part of the optimization objective, and the performance is therefore below expectations. To overcome the shortcomings of these two algorithms, we design a third greedy algorithm, called DP-SDTF, which considers both parts of the optimization goal. In addition, we develop a multiple-phase algorithm based on R-trees called MPR. The MPR algorithm can accelerate the query process with little increase in cost. We conduct extensive experiments to evaluate the performance of our algorithms. The results show that our DP-SDTF algorithm achieves similar performance to the exact algorithm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported in part by National Grand Fundamental Research 973 Program of China under Grant 2015CB358700, NSFC Grant no. 71531001, and SKLSDE Open Program SKLSDE-2016ZX-13.

References

[1] G. Liang, J. Ca, X. Liu, and J. Liang, "Smart world: a better world," *Science China Information Sciences*, vol. 59, no. 4, Article ID 043401, 2016.

[2] B. Zhu, L. Xie, D. Han, X. Meng, and R. Teo, "A survey on recent progress in control of swarm systems," *Science China Information Sciences*, vol. 60, no. 7, 070201, 24 pages, 2017.

[3] J. Hu, B. B. Park, and Y.-J. Lee, "Transit signal priority accommodating conflicting requests under Connected Vehicles technology," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 173–192, 2016.

[4] J. Hu, M. D. Fontaine, and J. Ma, "Quality of private sector travel-time data on arterials," *Journal of Transportation Engineering*, vol. 142, no. 4, Article ID 04016010, 2016.

[5] C. Ding, D. Wang, X. Ma, and H. Li, "Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees," *Sustainability*, vol. 8, no. 11, article no. 1100, 2016.

[6] H. Jiang, J. Hu, S. An, M. Wang, and B. B. Park, "Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 290–307, 2017.

[7] C. Ding, X. Wu, G. Yu, and Y. Wang, "A gradient boosting logit model to investigate driver's stop-or-run behavior at signalized intersections using high-resolution traffic data," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 225–238, 2016.

[8] C. Ding, C. Liu, Y. Zhang, J. W. Yang, and Y. P. Wang, "Investigating the impacts of built environment on vehicle miles traveled and energy consumption: Differences between commuting and non-commuting trips," *Cities*, vol. 68, pp. 25–36, 2017.

[9] Y. Tong, Y. Chen, Z. Zhou et al., "The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1653–1662, 2017.

[10] L. Kazemi and C. Shahabi, "GeoCrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '12)*, pp. 189–198, ACM, November 2012.

[11] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, p. 2, 2015.

[12] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile Micro-Task Allocation in spatial crowdsourcing," in *Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE '16*, pp. 49–60, IEEE, Helsinki, Finland, 2016.

[13] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu, "Top-k team recommendation in spatial crowdsourcing," in *Proceedings of the 17th International Conference on Web-Age Information Management, WAIM '16*, vol. 9658 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 191–204, June 2016.

[14] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu, "Top-k Team Recommendation and Its Variants in Spatial Crowdsourcing," *Data Science and Engineering*, vol. 2, no. 2, pp. 136–150, 2017.

[15] A. Guttman, "R-trees:a dynamic index structure for spatial searching," in *Proceedings of the the 1984 ACM SIGMOD international conference*, pp. 47–57, June 1984.

[16] Q. Tao, T. Song, and K. Xu, "Finding optimal team for multi-skill task in spatial crowdsourcing," in *Proceedings of the 2th WORKSHOP on Data-Driven Crowdsourcing (DDC 2017) in conjunction with APWeb-WAIM 2017*, 2017.

[17] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries," *ACM SIGMOD Record*, vol. 24, no. 2, pp. 71–79, 1995.

[18] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," *ACM Transactions on Database Systems (TODS)*, vol. 24, no. 2, pp. 265–318, 1999.

[19] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pp. 467–475, July 2009.

[20] S. Datta, A. Majumder, and K. V. M. Naidu, "Capacitated team formation problem on social networks," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, pp. 1005–1013, August 2012.

[21] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Power in unity: Forming teams in large-scale community systems," in *Proceedings of the 19th International Conference on Information and Knowledge Management and Co-located Workshops, CIKM'10*, pp. 599–608, October 2010.

[22] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proceedings of the 21st Annual Conference on World Wide Web, WWW'12*, pp. 839–848, fra, April 2012.

[23] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in realtime spatial data: Experiments and analysis," in *Proceedings of the 42nd International Conference on Very Large Data Bases, VLDB 2016*, pp. 1053–1064, ind, September 2016.

[24] J. She, Y. Tong, L. Chen, and C. C. Cao, "Conflict-aware event-participant arrangement and its variant for online setting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2281–2295, 2016.

[25] Y. Tong, J. She, and R. Meng, "Bottleneck-aware arrangement over event-based social networks: the max-min approach," *World Wide Web*, vol. 19, no. 6, pp. 1151–1177, 2016.

[26] J. She, Y. Tong, and L. Chen, "Utility-aware social event-participant planning," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2015*, pp. 1629–1643, June 2015.

[27] L. Kazemi, C. Shahabi, and L. Chen, "GeoTruCrowd: trustworthy query answering with spatial crowdsourcing," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM SIGSPATIAL GIS 2013*, pp. 304–313, November 2013.

[28] C. C. Cao, Y. Tong, L. Chen, and H. V. Jagadish, "Wisemarket: a new paradigm for managing wisdom of online social users," in *Proceedings of the the 19th ACM SIGKDD international conference*, pp. 455–463, 2013.

[29] Y. Tong, L. Chen, and Y. Cheng, "Mining frequent itemsets over uncertain databases," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1650–1661, 2012.

[30] Y. Tong, L. Chen, and B. Ding, "Discovering threshold-based frequent closed itemsets over probabilistic data," in *Proceedings of the IEEE 28th International Conference on Data Engineering, ICDE 2012*, pp. 270–281, April 2012.

Hindawi
Journal of Sensors
Volume 2017, Article ID 9168525, 10 pages
https://doi.org/10.1155/2017/9168525

Research Article

A Sensor-Based Visual Effect Evaluation of Chevron Alignment Signs' Colors on Drivers through the Curves in Snow and Ice Environment

Wei Zhao,[1,2,3] Liangjie Xu,[1] Shaoxin Xi,[1] Jizhou Wang,[1] and Troy Runge[3]

[1]*School of Transportation, Wuhan University of Technology, Wuhan 430063, China*
[2]*School of Economics and Management, Inner Mongolia University of Science and Technology, No. 7 Aerding, Baotou, China*
[3]*College of Agricultural and Life Sciences, University of Wisconsin-Madison, 1552 University Ave, Madison, WI 53706, USA*

Correspondence should be addressed to Wei Zhao; zwei47@wisc.edu

Received 3 July 2017; Accepted 27 August 2017; Published 11 October 2017

Academic Editor: Chuan Ding

The ability to quantitatively evaluate the visual feedback of drivers has been considered as the primary research for reducing crashes in snow and ice environments. Different colored Chevron alignment signs cause diverse visual effect. However, the effect of Chevrons on visual feedback and on the driving reaction while navigating curves in SI environments has not been adequately evaluated. The objective of this study is twofold: (1) an effective and long-term experiment was designed and developed to test the effect of colored Chevrons on drivers' vision and vehicle speed; (2) a new quantitative effect evaluation model is employed to measure the effect of different colors of the Chevrons. Fixation duration and pupil size were used to describe the driver's visual response, and Cohen's **d** was used to evaluate the colors' psychological effect on drivers. The results showed the following: (1) after choosing the proper color for Chevrons, drivers reduced the speed of the vehicle while approaching the curves. (2) It was easier for drivers to identify the road alignment after setting the Chevrons. (3) Cohen's **d** related to different colors of Chevrons have different effect sizes. The conclusions provide evident references for freeway warning products and the design of intelligent vehicles.

1. Introduction

Freeway traffic crashes have become a key cause of death in the population in the current years. Crash occurrence and risk are significantly influenced by adverse weather conditions, especially for the northern freeway in snow and ice (SI) environment. Meanwhile, previous freeway traffic accident reports showed that the crash rate on curves was much higher than that on straight sections. Moreover, most crashes on curves were associated with lane departure and frontal collision [1]. The detailed result shows that the accident rate at the curve section is 0.41 per million vehicle-kilometers on no-snow days, while the number rises to 5.86 on snowy days, increasing by more than 13 times [2].

Since crash risk dramatically increases when passing curves in SI environment. Chevron alignment signs are utilized as an effective measure to improve traffic safety on curves. However, the colors of the signs were only roughly studied and tested. As a result, most of the existing researches focused on blue Chevrons, but the red and green Chevrons are often what has been used in China. To correct the improper application, three questions about colored Chevrons have to be urgently addressed:

(i) How much could the Chevrons positively contribute to the freeway traffic safety?

(ii) What is the process in which the drivers as well as the vehicles are influenced by the Chevrons?

(iii) Which color is the best for Chevrons on curves in snow and ice environment to decrease the traffic accidents?

This paper reports on tests that investigated the drivers' visual and motor feedback to colored Chevrons. This study quantitatively identified the effect of the Chevrons and, then, optimized the setting of Chevrons in order to decrease the occurrence of crashes in snow and ice environment.

Figure 1: Practical application of Chevrons in China.

To alert the drivers facing existing or potential hazards, it is important to place warning signs. The Chevrons are commonly applied to guide the driving direction and to indicate changes in road alignment. However, there are differences with the color settings of the Chevrons between China and America. Based on the National Traffic Sign and Device Standards of China [3], three main standards can be found. First, the Chevrons have blue background and white symbols on the freeway, while, in the expressway system, the Chevrons have green background and white symbols. Moreover, at the central islands, channelization islands, and bridges, along with other locations, the Chevrons have red background and white symbols. As shown in Figure 1, there are some practical applications of Chevrons in China. But, in America, almost all the warning signs are yellow and black, whereas white and blue signs are generally used to inform drivers of upcoming cities and roads. Although diverse colors are chosen, there is one common concept that information delivered by signs influences drivers' control of vehicles [4].

Snow and ice environment has a tremendous detraction on the driver's vision. The drivers would be influenced by the diffused reflection effect of snow and ice and the negative effect of monotonous color background in an SI environment. As a result, driving in an SI environment could take the edge off the driver's ability to recognize the changing environment. Xing et al. (2012) reached the conclusion that the visually perceived speed by drivers in a snow and ice-covered environment was 5%–14% lower than that in an environment without snow and ice [5]. Pasetto and Barbati (2012) found that diverse brightness and light conditions of road environments weighed heavily on the driver's visual perception ability [6]. Charlton (2007) concluded that drivers were likely to make operational mistakes while passing horizontal curves, as a result of failures in concentration, misperceptions of speed and curvature, and poor lane positioning [7].

Besides, the snow around roads has high reflectance. Light reflected by the snow during shiny days could strongly irritate drivers and is prone to cause eye fatigue. Chen et al. (2015) explored the effects of road environments on driving behaviors and cognitive performance of fatigued drivers, and the conclusion showed that the drivers were likely to overestimate the distance between vehicles [8].

Road curvature has a strong correlation with the traffic incidents and easily leads to the occurrence of crashes. In order to improve traffic safety on curved roads, Yotsutsuji et al. (2014) focused on the model research of drivers' cognitions to lead-vehicle speeds on curves [9]. Compared to the text symbols, colored ones bring stronger and faster impact on human vision. In addition, cool colors, warm colors, light colors, and dark colors would arouse different emotions [10]. Carson and Mannering (2001) found that the warning signs for icy roads could be optimized by repositioning, in order to reduce the frequency and severity of ice-related accidents [11]. De La Escalera et al. (1997) also found that the traffic signs provided drivers with useful information about the road conditions, the information would then influence drivers' feedback and finally make driving safer and easier [12]. Comte and Jamson (2000) explored four speed reducing methods when passing curves. They proved that all the information provided in any format could be effective in reducing speed on horizontal curves [13].

Research on the design and application of Chevrons has been updated continuously. Guan et al. (2014) explored the relationship between drivers' deceleration behavior and the setting method of traffic sign on curves. The results showed that the deceleration was highly related to how much information the drivers acquired and reacted to [14]. Choi et al. (2005) analyzed the sense of stability of Chevron alignment signs on the existing freeway through a practical test. The traffic delineator was found to regulate traffic speed on curves [15]. Furthermore, Charlton (2004) found that the Chevron alignment sign is an important traffic control device for warning drivers through delineating the alignment of the curved road [16]. The Chevron alignment sign not only could expand the vision when approaching a curve but also could provide warning and guidance while navigating a curve. Meanwhile, properly spaced Chevrons can be used to guide drivers to decelerate when approaching as well as passing a curve [17]. Rose and Carlson (2005) developed a spacing chart to find the proper spacing for Chevrons on horizontal curves. The former result showed that there was an obvious deceleration of about 3 mph at night with more than two Chevrons within vision around the curve. But less obvious deceleration was observed during daylight [18]. Wu et al. (2013, 2016) and Zhao et al. (2015) conducted an experiment with a driving simulator in order to measure drivers' eye movement, operations, and changes in psychological states. Detailed comparisons were made of drivers' recognition ability, operations, and changes in psychological reactions by driving through a simulated city expressway slope with and without Chevrons. The result proved that the Chevrons would let drivers pay more attention to hazardous traffic

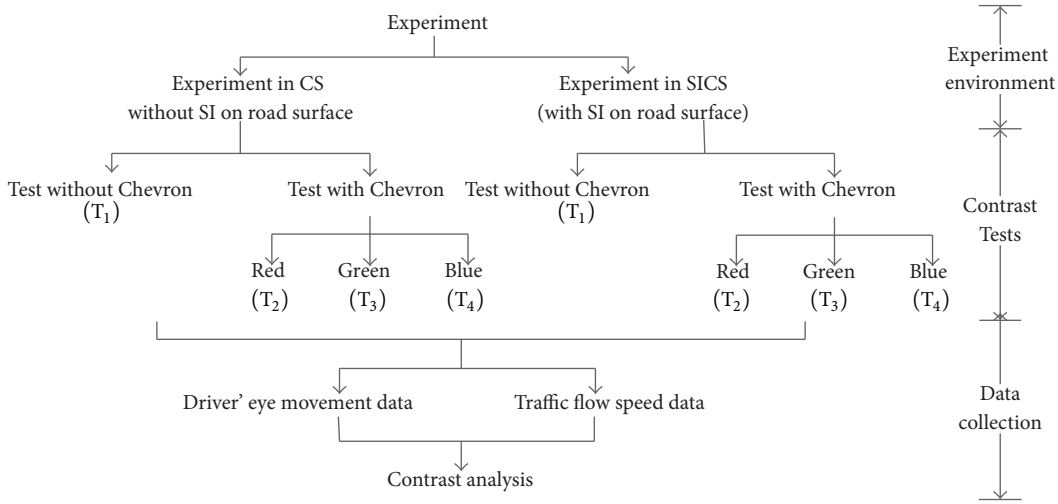


Figure 2: Flowchart of the experiment.

environments and decrease the crash risk. All these discussions lead to the common conclusion that Chevron signs can assist drivers in operating their vehicles while navigating curves [19–21].

Hardly enough attention has been given to drivers' vision feedback to colored traffic signs in existing research. Ritter et al. (1995) found that reasonable choice of color was the key to an efficient traffic sign. Moreover, the more distinctive the sign was, the rapider the drivers are able to trace information [22]. Liu et al. (2010) stated that, regardless of the traffic signs' background colors, the signs were perceived in a similar way by humans. Moreover, different contrasts between the background and foreground colors of traffic signs would make difference in cognizing [23]. Besides, the color scheme and the number of signs were significantly relevant to participants' response time to the information on the signs [24].

Thus far, although the Chevrons are able to decrease crash risk, little research has been done regarding the effect of Chevrons on visual feedback as well as driving reaction while navigating curves in a snow and ice environment, especially in the context of China. Summarizing the above discussion, only the setting method and effect of Chevrons on drivers have been analyzed, but the research results are not enough to guide the setting of Chevrons through curves in snow and ice environment. In addition, most studies on the psychological and physical effect of colored traffic signs including the Chevrons were completed by either mental questionnaire analyses or indoor simulation experiments. As a result of the color sensation and lighting condition, some results may contain a high distortion rate.

In this study, on-road experiments were conducted to collect vehicle and driver parameters, including speeds of the test vehicle and traffic flow, pupil size, and fixation duration, using the SMI eye-tracker system and NC200 portable traffic analyzer. The practical test vehicle was studied while passing through a common and representative curve of a freeway in China in an environment with and without snow and ice. An eye movement tracking technique was adopted to observe the driver's fixation distribution and pupil size. The NC200 portable traffic analyzer was utilized to observe the speed of the test vehicle as well as the distribution of traffic flow. A new quantitative method was then developed to evaluate the effect of colored Chevrons. In addition, different feedback of three different colors of the Chevrons in SI environment was calculated. This study provides theoretical guidance for optimizing traffic signs enabling the drivers to easily recognize road information including warning and guidance in SI environment and keep safety operation.

This paper is organized as follows: the next section provides experimental details including participants, experimental setup, data collection method, and experimental scenarios. Then, the model and statistical analysis are described, followed by the results and discussions. Conclusions and recommendations of this study are provided in the last section.

2. Experiment

The flowchart of the experiment is shown in Figure 2. A similar section of freeway was separately tested. In each test, there are two datasets with and without the colored Chevrons. Parameters, including vehicle speed, traffic flow, pupil size, and fixation duration, were measured and compared.

2.1. Experimental Scenarios. Two representatively horizontal curves located on National Rd. 302 in Jilin Province, China, were selected as the experimental road sections and are shown in Figure 3. It is worth mentioning that these two common sections of road are very similar both in shape and in visibility. These two curved sections were separately named as CS and SICS.

(i) Section 1: CS is located near Dachapeng Village. There was snow all along the roadside, but there was no snow or ice on the road surface. And the road markings could be seen clearly.

(ii) Section 2: SICS is located near West Sanjiazi Village. There was thick snow covering the road surface as well as the roadside. The road markings were not visible.

(a) (b)

Figure 3: Photos of the experimental CS (a) and SICS (b).

(a) (b)

Figure 4: SMI (a) and NC200 (b) sensors.

Both sections are two-lane freeways with one lane in each direction. Each lane is 4.5-meter-wide with a speed limit of 60 km/h. The length of the CS is 420 m, with a radius of 180 meters. The length of the SICS is 405 m, with a radius of 160 m. The surroundings of both experimental curves were covered by snow. Both curves were freeways without intersection or roadside buildings, so the experience was not impacted by the roadside traffic.

2.2. Experimental Objectives and Sensors. The objective of this experiment was twofold as followed. The experiment time was 10:00 a.m.–13:00 p.m., from Jan. 29, 2015, to Feb. 3, 2015.

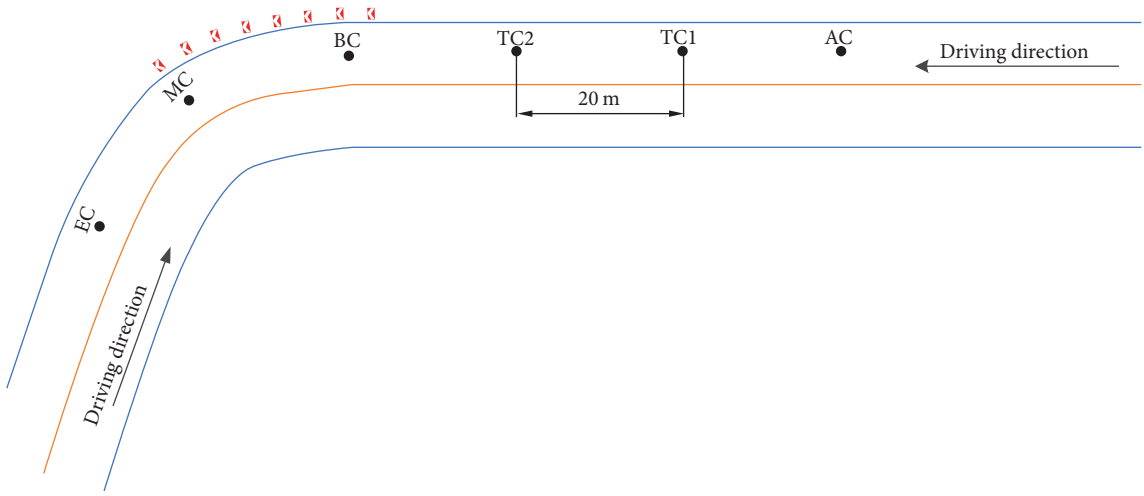
2.2.1. Objective 1: Collecting Drivers' Eye Movement. The experiment was conducted with thirteen participants between ages 24 and 35, who were asked to drive through the CS and SICS. According to the local DOT data, most drivers in this area are male and experienced. So the participants are most male drivers with over five years' driving experience. And the SMI eye-tracker (shown in Figure 4) was used to collect information of drivers' fixation distribution and changing pupil size as the vehicle passed through the two curves.

2.2.2. Objective 2: Observing Traffic Flow. The observation of traffic flow was conducted in a normal traffic environment, and all the vehicles were in free-flow running state. The traffic was mainly composed of light vehicles, especially sedans, so the influence of vehicle types could be neglected.

A traffic flow observation experiment was performed with NC200 portable traffic analyzer (shown in Figure 4). The real-time speed of all the traffic was supposed to be observed. The devices were placed in the middle of the lane. According to the length of the curves, 5 observation points were set in SICS and 6 observation points were set in CS. Each point, as is shown in Figure 5, was spaced 20 meters from the adjacent points, recording the speeds of vehicles passing through.

2.3. Experimental Procedure

2.3.1. Experiment Design. In order to study the different impacts on traffic flow as well as drivers' vision caused by Chevrons and colors in Chevrons, a series of contrast tests were designed. In this experiment, Chevrons of three commonly applied colors (red, green, and blue) were involved. The experiment was divided into two parts as was shown in Figure 2: experiments in SICS and experiments in CS. Each part contained 4 respective tests: (T1) test without Chevrons, (T2) test with red Chevrons, (T3), test with green Chevrons, and (T4) test with blue Chevrons. According to the standard of road sign regulation in China [3], 8 Chevrons with a 2-meter height and a size of 220 mm × 400 mm board were set in the middle of the curves, separated 6 meters from the nearest ones. A brief shed was built as waiting room in order to block the drivers' vision when different Chevrons were set



AC = approach of curve
BC = beginning of curve
MC = middle of curve
EC = ending point of curve
TC1 = transition point 1 of the curve
TC2 = transition point 1 of the curve (there is no this point in C-SICS)

Figure 5: The speed observation point of the curve.

up along the curves. As a result, after a long waiting time, the road condition would not be remembered in each test.

2.3.2. Data Collection. For objective 1, all drivers, wearing a SMI eye-tracker, drove the test car to pass through the curves successively. Then, they would do it again after another colored set of Chevrons was set up. The drivers' eye movement, the data of the driver's eye fixation duration, and pupil size would be collected by the eye-tracker. Meanwhile, the drivers were requested to give physiological feedback.

For objective 2, when the drivers saw the Chevrons, they would adjust the speed of the vehicle. The NC200 was used to collect speed of all vehicles that passed the curves.

3. Effect Evaluation Method

3.1. Assumptions

(i) Assumption 1: the experimental environment is the same except for the traffic signs.

(ii) Assumption 2: the speed of traffic flow passing the curves in SI environment obeyed normal distribution in general.

(iii) Assumption 3: the drivers with different effect sizes would make different driving operation. This changing operation mainly depended on whether there were traffic signs on the roadside.

3.2. Effect Evaluation Model. In the single factor experiments, the mean difference is expressed as the contrast of two groups of experiments. The linear contrast function is as follows:

$$\psi = c_1\mu_1 + c_2\mu_2 + \cdots + c_j\mu_i + \cdots + c_m\mu_l, \quad (1)$$

where ψ stands for speed difference; i and j represent the indices of the experimental groups; μ_i represents the mean speed of the ith group; c_j represents a group of constants which satisfy $c_1+c_2+\cdots+c_m = 0; i = 1, 2, \ldots, l; j = 1, 2, \ldots, m$.

To measure the effect of Chevrons' color on drivers, variance in road condition was controlled in the experiment. The different colored Chevrons along the curves were used as the experimental variables. The experiments were divided into tests T1 to T4, T1 is expressed by μ_w (without Chevrons group), T2–T4 are expressed by μ_c (colored Chevrons group including T2: μ_r-red Chevrons, T3: μ_g-green Chevrons, and T4: μ_b-blue Chevrons) The mean difference of different Chevrons' colors is expressed as

$$\psi = \mu_c - \mu_w. \quad (2)$$

The effect size of the linear contrast, **d**, is defined as

$$\mathbf{d} = \frac{\psi}{\sigma}, \quad (3)$$

where σ is the pooled standard deviation. In calculation, ψ can be estimated by using the mean value of tests T1 to T4 in order to replace population mean. The formula to calculate σ is as follows:

$$\sigma = \sqrt{\frac{(n_w - 1)S_w^2 + (n_c - 1)S_c^2}{n_w + n_c - 2}}, \quad (4)$$

where n_w represents the number of samples without Chevrons; n_c represents the number of samples with Chevrons; S_w is the variance of samples without Chevrons; S_c is the variance of samples with Chevrons.

Therefore, the estimated value (Cohen's **d**) [25] of effect size, which represents the Chevrons' psychological effect on drivers in SI environment, can be obtained by the following expression:

$$\mathbf{d} = \frac{\overline{x}_w - \overline{x}_c}{\sigma}, \quad (5)$$

Table 1: Grades of Cohen's **d** to evaluate effect.

Cohen's **d**	Grades of effect	The significance of effect grade
d < 0.2	Light effect	Light influence on the driver's visual psychology.
0.2 < **d** < 0.8	Medium effect	Medium influence on the driver's visual psychology.
d > 0.8	Strong effect	Strong influence on the driver's visual psychology.

Table 2: Fixation duration and pupil size in each scenario.

Curve environment	Chevrons tests (T1–T4)	Fixation duration (second)		Pupil size (pixel)	
		AVG	SD	AVG	SD
CS	Without Chevrons	0.543	0.457	31.196	1.001
	Red Chevrons	0.558	0.390	31.303	1.547
	Green Chevrons	0.277	0.241	33.452	2.382
	Blue Chevrons	0.334	0.256	30.285	1.398
SICS	Without Chevrons	0.352	0.199	36.736	2.196
	Red Chevrons	0.427	0.246	34.055	1.347
	Green Chevrons	0.348	0.273	34.355	2.500
	Blue Chevrons	0.345	0.215	37.179	1.948

where $\overline{x}_w$ is the mean speed of samples without Chevrons and $\overline{x}_c$ is the mean speed of samples with Chevrons.

3.3. Large Sample Evaluation Method. Effect evaluation model can indicate the correlative degree of variables. However, there is little relation between effect evaluation model and sample size. Combined with pretest and posttest design, the model used Cohen's **d** to measure the difference between the different tests (T1 to T4). Cohen's **d** is defined as the difference between the two means divided by the standard deviation of a sample. Also, Cohen's **d** represents the correlation degree of variables. Psychological Cohen's **d** is often used to evaluate the colored signs' psychological effect. Cohen's **d** uses the standard deviation unit to express effect size. Thus, **d** = 1.0 means the difference between the mean values of the tests of samples is 1 SD (standard deviation). Cohen and Jacob studied the size of effect [25], and Cohen's **d** values are divided into three grades (see Table 1).

4. Results and Discussion

4.1. Eye Movements Analysis

4.1.1. Descriptive Statistics. After the experiment for objective 1, the data from the SMI eye-tracker is used in the following analysis. Table 2 presents a summary of the average fixation duration (FD_AVG), standard deviation of fixation duration (FD_SD), average pupil size (PS_AVG), and standard deviation of pupil size (PS_SD) in different scenarios.

4.1.2. Fixation Duration. The first analysis compares the drivers' fixation duration approaching the curve in tests T1–T4. As is shown in Table 2, both in CS and in SICS, the FD_AVG value of sections with green (0.277 and 0.348) and blue (0.334 and 0.345) Chevrons is lower than that without Chevrons (0.543 and 0.352), while the FD_AVG with red Chevrons (0.558 and 0.427) is higher. The drivers' FD_AVG values in CS are all bigger than that in SICS. Drivers find it easier to identify the road alignment with setting of Chevrons; as a result, the fixation duration is reduced. Moreover, different colored signs produce different visual effects, and red is commonly considered as prohibition or warning, and the drivers easily pay more attention to it.

4.1.3. Pupil Size. In this study, drivers' pupil size is creatively utilized as a corresponding variable with fixation duration. These two variables are supplemental to each other and are enrolled in one function. The percentage of fixation duration is shown as follows:

$$f_a = \left(\frac{\sum_a^{a-1} t_x}{\sum_B^A t_x} \right) \times 100\%, \tag{6}$$

where f_a is the percentage of fixation duration when $a - 1 < x \leq a$; x is pupil size; t_x is the corresponding fixation duration of x; A and B are the boundary of pupil size distribution. Here $A = 24$ and $B = 44$.

Fixation duration characteristics of pupil size in different environments are shown in Figure 6. Combining the values in Table 2, the drivers' average pupil size in SICS was larger than that in CS. Because the very similar environments and road condition were controlled, it is found that drivers are more likely affected by the SI environment than by the CS environment. Also, drivers show a greater mental load staring at the target in SI environment than in CS environment.

The PS_AVG values in CS with the setting of blue Chevrons (30.285) are smaller than that without Chevrons (31.196), while the PS_AVG values with red (31.303) and green (33.452) Chevrons are larger than that without Chevrons. But the opposite results are found when passing through SICS, and the PS_AVG values with blue Chevrons (37.179) are larger than that without Chevrons (36.736), while the PS_AVG

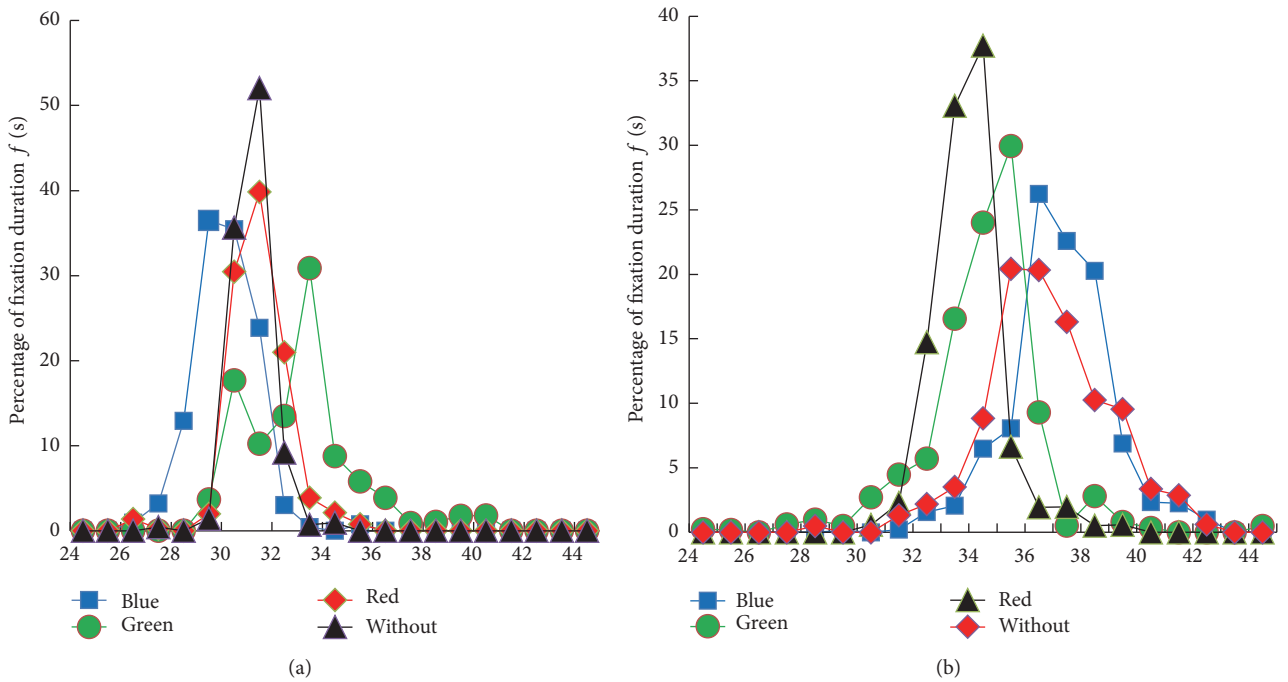


Figure 6: Fixation duration characteristics of pupil size in different environments.

Table 3: Evaluation of effect and speed distribution.

Curve environment	Chevrons tests (T1–T4) μ	Mean speed $\overline{x}$ (km/h)	Standard deviation S	Sample size n	Mean speed difference ψ	Pooled standard deviation σ	Effect size **d**	Effect evaluation
CS	Without	65.31	12.815	120	—	—	—	—
	Red	60.50	10.049	120	4.81	11.515	0.418	Medium
	Green	60.06	10.975	120	5.25	11.930	0.440	Medium
	Blue	62.39	10.020	120	2.92	11.502	0.254	Medium
SICS	Without	38.50	10.301	100	—	—	—	—
	Red	33.32	5.389	100	5.18	8.220	0.630	Medium
	Green	35.86	8.702	100	2.64	9.535	0.277	Medium
	Blue	37.81	10.155	100	0.69	10.228	0.067	Small

values with red (34.055) and green Chevrons (34.355) are smaller than that without Chevrons (36.736). The ability of visual recognition is different in different environments. In this study, this ability shows a visibly better result in CS environment. Furthermore, as a result of contrast tests of T2 to T4, the different colors of Chevrons, combined with ambient color, also have an effect on the driver's identification abilities.

4.2. Feedback of Driving Operation. In order to confirm the effect of the Chevrons, the feedback of driving operation is analyzed by the change of vehicle speed in different scenarios. The detailed methodology is explained in Section 2.1. The results will be discussed in three aspects as follows.

4.2.1. General Results in SI Environment. The analysis of the Chevrons' influence on the traffic with CS and SICS was analyzed. Based on the speed data recorded by NC200, the effects shown in different tests were presented in Table 3.

From Table 3, the results revealed a significant interaction between the different Chevrons and the speeds of passage through curves. The driving speed was significantly lower with Chevrons along the curves. Meanwhile, different colors showed significantly different feedback on drivers' operation. There are some common results of speed distribution for both CS and SICS: the red Chevrons and green Chevrons produced the most significant effect, and blue Chevrons produced the lightest effect. Under the condition of CS, Cohen's **d** values of all the colors indicated medium effect; meanwhile, the values of the green Chevrons and red Chevrons (>0.4) are greater than that of blue Chevrons (0.254). Under the condition of SICS, blue Chevrons had the smallest effect size (0.067).

After comparison in Table 3, the speed distribution of traffic flow under the condition of CS is higher than that in SICS. And the Chevrons were proven to be highly effective on feedback of speed. However, under the condition of SICS, the average traffic flow speed is smaller. And the red Chevrons,

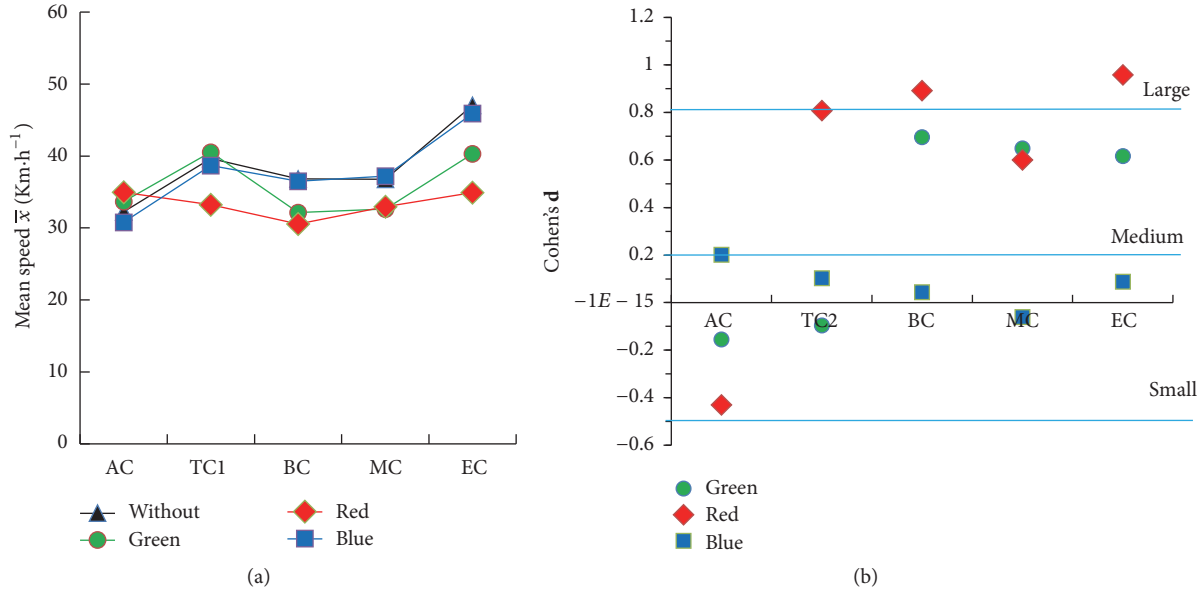


Figure 7: Mean speed (a) and effect sizes (b) obtained at observation points on SICS.

as the connotation of hazard warning, show the biggest effect size as 0.630.

4.2.2. Driver Performance in SICS. In the process of driving along the curve in SI environment, the Chevrons' effect on the drivers was time-varying and distance-varying. According to what they saw, the drivers made different feedback in their driving operation. The mean speed of traffic flow and effect sizes obtained from the observations points in SICS are shown in Figure 7.

Figure 7 shows that the Chevrons with different colors resulted in different mean speeds in SICS. Red Chevrons, green Chevrons, and blue Chevrons had different effects on improving the safety driving operation. The red Chevrons led the distinct mean speed changes. The green Chevrons resulted in a significant reduction compared to normal traffic flow speed. The blue Chevrons led the comparatively light mean speed changes. Moreover, by analysis of effect sizes at all the five observation points as shown in Figure 7, red Chevrons produced relatively large effect sizes at distances of 20 m, 40 m, and 80 m from the curve entrance. However, the effect sizes produced by the blue Chevrons were smaller. In general, the red Chevrons and the green Chevrons produced larger effect sizes (around 0.8), while blue Chevrons produced unobvious and small effect sizes (less than 0.2).

Under the condition of SICS, the drivers drove conservatively at a low speed; as a result, they could recognize the Chevrons earlier than the drivers keeping fast speed. Therefore, they would pay more attention to the Chevrons for a longer time. Meanwhile, the Chevrons had an obvious effect on the drivers. Due to the weak visibility and warning effect, blue Chevrons were unlikely to have a noticeable effect. In conclusion, the effect of the red and green Chevrons in the SICS will be better.

4.2.3. Driver Performance in CS. In the process of passing through the curve in CS, the Chevrons produced different effect sizes at the observation points. The mean speed of traffic flow and effect sizes are shown in Figure 8. Although there was no snow or ice on the road surface, there was snow all along the roadside.

As is shown in Figure 8, with the road surface in dry condition, the trend of speed measured at all the observation points was almost the same. There are a common rise at TC1, TC2, and EC and fall at BC and MC. Almost all the values with the Chevrons in whichever color are lower than that without the Chevrons. This means the mean speed of traffic flow decreases visibly after setting the Chevrons. The mean speeds with the red Chevrons, the green Chevrons, and the blue Chevrons are slightly different from each other, and such difference can be ignored. Figure 8 shows that the Chevrons' effect sizes at the observation points were generally within 0.2–0.8, presenting a stable distribution. The effect sizes show that all the Chevrons with the three colors had a basically medium effect under the condition of CS, while, after a detailed comparison, blue Chevrons have a slightly weaker effect than the red Chevrons and green Chevrons. However, the effect sizes caused by red Chevrons and green Chevrons at the same observation points differ from each other.

Because of the good condition of freeway in CS, the traffic flow speed was high. However, the snow roadside still had influence on drivers, reflecting vehicle speed and effect size. According to Figure 8, the speed influence could be explained as follows. When approaching the curve, the drivers were confident enough to accelerate a little to see the curve information clearly. Then they slowed down actively once being stimulated by the Chevrons as soon as they saw the information. Finally, they speeded up again to pass through the curve.

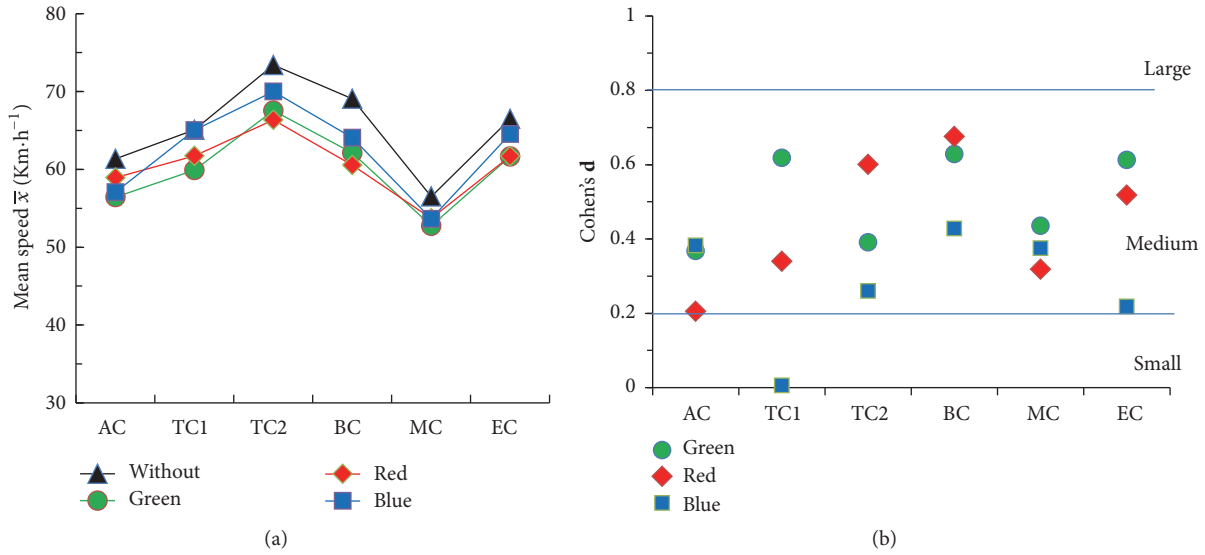


Figure 8: Mean speed (a) and effect sizes (b) obtained at observation points on CS.

By considering the actual application of the Chevrons in CS environment, the green Chevrons should be used in freeway ramp due to the higher speed. The blue Chevrons should be used on the general road due to the low speed. The red Chevrons would better be used as a warning sign in some special sections. The recommended setting method of the Chevron alignment signs in this paper is the same as China Standards [3].

5. Conclusion

The existent studies investigated the effect of Chinese Chevron alignment signs along the curves on two-lane rural freeways. Moreover, most studies were based on the driving simulation experiments. This paper contributed to developing and applying a new basic method on the multisensors. The contrast test environments without and with Chevrons were set on the actual road in dry and SI road conditions. The vehicle test was carried out in all the scenarios to research the influence of the Chevrons on drivers with snow covering the roadside. Moreover, this study did further research on effects of the colors of traffic signs on drivers, and the conclusions were helpful in providing a theoretical foundation for safety design of colors in traffic environment.

In an environment with snow covering the roadside, the significant effect of Chevrons on drivers' feedback is found through the contrast analysis of Chevrons' guiding effect and colors' psychological effect. Chevrons along the side of freeway curves could enhance the signs' effect on drivers as well as reduce the speed of traffic flow. As a result, setting of proper Chevrons improves driving safety passing through the curves in dry condition or SI condition.

There was a lower traffic flow speed with Chevrons along the road than without Chevrons. Meanwhile, the function of Chevrons on speed reduction was significantly affected by the Chevrons' color. All three colors can attract drivers' attention on roadside information along the curves and effectively guide them into making a correct response to the roads' alignment. Besides, the Chevrons play a significant role in encouraging drivers to reduce their speed, and this effect was sensitive to Chevrons' color. It was caused according to the drivers' special visual field. Since the Chevrons prevent excessive speeds and encourage drivers to pay more attention to curves, Chevrons appear to bring great benefit in reducing traffic crashes on curves.

With different road surface conditions, the driver's visual recognition ability is not the same. Generally, they have a better version in CS. Meanwhile, the contrast between Chevrons' color and environmental color also affects the driver's identification. Drivers find it easier to identify the road alignment when Chevrons were set as a result of the decrease of fixation duration. Because red is commonly used as warning, the driver will pay more attention to identify signs in red. The Chevrons' psychological effect was different with different kinds of road surface in SI environment. Cohen's **d** values for CS were within 0.2–0.5, which indicated medium effect. Significant difference was observed between the effect of different colors on drivers for SICS. The red Chevrons produced the greatest effect, followed by the green Chevrons. The blue Chevrons had the weakest effect. Therefore, proper color is both supplementary and complementary in setting the Chevrons on curves. Trade-off colored Chevrons are recommended in SICS regarding future Chevron sign implementation on roadways in harsh winter areas to improve driver visibility and traffic safety.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by Research Projects of Social Science and Humanity on Young Fund of the Ministry of Education

of China (16YJCZH157), Fund for Less Developed Regions of the National Natural Science Foundation of China (no. 71764020), and Natural Science Foundation of Inner Mongolia (2015BS0707).

References

[1] National Highway Traffic Safety Administration, Fatality Analysis Reporting System, 2008.

[2] A. J. Khattak, K. K. Knapp, K. L. Giese et al., "Safety implications of snowstorms on interstate highways," in *Proceedings of the 79th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, January, 2000.

[3] National Standards of the People's Republic of China. Road Traffic Signs and Markings–Part 2: Road Traffic Signs, 2009.

[4] C. Pankok Jr., D. B. Kaber, W. J. Rasdorf, and J. E. Hummer, "Driver attention and performance effects of guide and logo signs under freeway driving," in *Proceedings of the 94th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, January 2015.

[5] E.-H. Xing, R. Wang, and P. HAN, "Influence of snowy and icy road conditions on driver visual perception characteristics," *China Safety Science Journal*, vol. 22, no. 3, pp. 86–91, 2012.

[6] M. Pasetto and S. D. Barbati, "The impact of simulated road-space perception on driver's behavior," *Procedia-Social and Behavioral Sciences*, vol. 53, pp. 721–730, 2012.

[7] S. G. Charlton, "The role of attention in horizontal curves: a comparison of advance warning, delineation, and road marking treatments," *Accident Analysis and Prevention*, vol. 39, no. 5, pp. 873–885, 2007.

[8] C. Chen, G. Zhang, R. Tarefder, J. Ma, H. Wei, and H. Guan, "A multinomial logit model-Bayesian network hybrid approach for driver injury severity analyses in rear-end crashes," *Accident Analysis and Prevention*, vol. 80, pp. 76–88, 2015.

[9] H. Yotsutsuji, H. Kita, and K. Kitamura, "Accident-preventive measure selection method based on the speed cognition of lead-vehicle driver in curved roadway," *Procedia-Social and Behavioral Sciences*, vol. 138, pp. 592–601, 2014.

[10] Z. Lan, "Application of color in the design of urban traffic signs," *Integrated Transport*, no. 1, pp. 76–80, 2013.

[11] J. Carson and F. Mannering, "The effect of ice warning signs on ice-accident frequencies and severities," *Accident Analysis and Prevention*, vol. 33, no. 1, pp. 99–109, 2001.

[12] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[13] S. L. Comte and A. H. Jamson, "Traditional and innovative speed-reducing measures for curves: an investigation of driver behaviour using a driving simulator," *Safety Science*, vol. 36, no. 3, pp. 137–150, 2000.

[14] W. Guan, X. Zhao, Y. Qin, and J. Rong, "An explanation of how the placement of traffic signs affects drivers' deceleration on curves," *Safety Science*, vol. 68, pp. 243–249, 2014.

[15] H. Choi, H. K. Park, and I. J. Kang, "A study on the optimal intervals for chevron signs," *Journal of The Korean Society of Civil Engineers*, vol. 25, no. 2, pp. 331–339, 2005.

[16] S. G. Charlton, "Perceptual and attentional effects on drivers' speed selection at curves," *Accident Analysis and Prevention*, vol. 36, no. 5, pp. 877–884, 2004.

[17] R. Srinivasan, B. Persaud, K. A. Eccles, D. L. Carter, and J. Baek, "Safety evaluation of improved curve delineation with signing enhancements," in *Proceedings of the 89th Annual Meeting on the Transportation Research Board*, Washington, DC, USA, January 2010.

[18] J. Liu and A. J. Khattak, "Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 83–100, 2016.

[19] Y. Wu, X. Zhao, J. Rong, and J. Ma, "Effects of chevron alignment signs on driver eye movements, driving performance, and stress," *Transportation Research Record*, vol. 2365, pp. 10–16, 2013.

[20] T. Wang, R. R. Souleyrette, and K. Gkritza, "Incorporating safety into transportation planning at smaller agencies," in *Proceedings of the 92nd Annual Meeting on the Transportation Research Board*, Washington, DC, USA, January 2013.

[21] X. Zhao, Y. Wu, J. Rong, and J. Ma, "The effect of chevron alignment signs on driver performance on horizontal curves with different roadway geometries," *Accident Analysis and Prevention*, vol. 75, pp. 226–235, 2015.

[22] W. Ritter, F. Stein, and R. Janssen, "Traffic sign recognition using color information," *Mathematical and Computer Modelling*, vol. 22, no. 4, pp. 149–161, 1995.

[23] B. Liu, Z. Wang, G. Song, and G. Wu, "Cognitive processing of traffic signs in immersive virtual reality environment: an ERP study," *Neuroscience Letters*, vol. 485, no. 1, pp. 43–48, 2010.

[24] C.-J. Lai, "Effects of color scheme and message lines of variable message signs on driver performance," *Accident Analysis and Prevention*, vol. 42, no. 4, pp. 1003–1008, 2010.

[25] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Academic Press, Cambridge, Mass, USA, 2013.

Hindawi
Journal of Sensors
Volume 2017, Article ID 6290248, 11 pages
https://doi.org/10.1155/2017/6290248

Research Article

Continuous and Discrete-Time Optimal Controls for an Isolated Signalized Intersection

Jiyuan Tan,[1] Xiangyun Shi,[1] Zhiheng Li,[2] Kaidi Yang,[3] Na Xie,[4] Haiyang Yu,[5] Li Wang,[1] and Zhengxi Li[1]

[1]*Beijing Key Lab of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing 100144, China*
[2]*Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China*
[3]*The Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich, Switzerland*
[4]*School of Management Science and Engineering, Central University of Finance and Economics, Beijing 100081, China*
[5]*Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Haiyang Yu; hyyu@buaa.edu.cn

Received 13 March 2017; Revised 29 June 2017; Accepted 19 July 2017; Published 7 September 2017

Academic Editor: Jia Hu

A classical control problem for an isolated oversaturated intersection is revisited with a focus on the optimal control policy to minimize total delay. The difference and connection between existing continuous-time planning models and recently proposed discrete-time planning models are studied. A gradient descent algorithm is proposed to convert the optimal control plan of the continuous-time model to the plan of the discrete-time model in many cases. Analytic proof and numerical tests for the algorithm are also presented. The findings shed light on the links between two kinds of models.

1. Introduction

Along with the fast increase of auto population, urban streets are becoming more crowded nowadays. To relieve congestions and reduce accidents, various traffic control methods have been proposed since the late 1950s [1].

As a typical traffic scenario, oversaturated intersections attracted consistent interest during the last six decades [2, 3]. The term "oversaturated" means the following: the vehicles that remained since the last cycle plus the vehicles that newly arrived exceed the capacity of the intersection. This leads to the carryover of vehicle queues (at least in one leg of the intersection) to the next cycle.

Discussions on the road networks that consist of many oversaturated intersections can be found in researches done by Chang and Sun [4], Di Febbraro and Sacco [5], Dotoli and Fanti [6], Ma [7, 8], and Sun et al. [9, 10]. In the research done by Varaiya [11] and Le et al. [12], the study of pressure-based signal control developed stability properties of a decentralized signal timing policy for networks with stochastic arrivals. But for a real-time signal timing optimization problem, the data that could be used is the arriving information of vehicles in the recent several signal cycles (data could be gotten by connecting vehicle technology, etc.). The optimization objects and scenarios are different between the model in this paper and pressure-based policies. This paper will focus on the isolated oversaturated intersection.

Usually, researchers aim to find an optimal signal timing plan that minimizes the total delay of vehicles passing this intersection. The total delay is often defined as the time integral of the sum of all queue lengths for all legs of the intersection over a given time horizon. However, the total delay is a nonlinear and nonconvex function of control variables (e.g., green phases), which makes it difficult to optimize. One promising approach is to apply heuristic algorithms to solve the formulated optimization problem. For example, the genetic algorithm was applied by Park et al. [13] to optimize the total delay. An alternative approach is to first approximate the nonconvex total delay with some convex functions and then solve the newly formulated optimization problem. The

rest of this paper will focus on the second approach within a typical traffic scenario: an isolated intersection with only two movements.

There are mainly two kinds of convexified models for this scenario. The first kind of models originated from Gazis [3] who used continuous-time differential equations to describe the traffic dynamics. The cycle length, departing flow rates, and arriving flow rates in Gazis [3] were all assumed to be constant. Michalopoulos and Stephanopoulos [14, 15] extended the continuous-time model by including the maximum queue lengths constraints and time-variant arrival flow rates. Such formulations led to a classical control problem that can be solved via the Pontryagin Maximum Principle (PMP) [16]. However, the obtained continuous-time signal timing plan should be discretized into the corresponding discrete-time signal timing plan that can be executed in practice.

The second kind of models uses discrete-time difference equations to describe the traffic dynamics [7, 8, 17–19]. The corresponding design problem can then be formulated as a linear programming (LP) problem [20–25]. One interesting question that naturally arises is how to depict the difference and connection between the discrete-time model and the continuous-time model.

Recently, Ioslovich et al. [26] studied the formulated LP problem by considering the corresponding continuous-time approximation model and gave an elegant approximate solution in continuous-time forms. However, it was not verified how this approximate solution differs from the accurate solution (the solution obtained by the continuous-time model). Whether a discretized version of this continuous-time approximate solution is still optimal to the LP problem also needs further discussions.

Zou et al. [27] have given a preliminary result on the relationship between the continuous-time model and the discrete-time model. Zou et al. applied a graphical method to adjust the continuous-time approximate solution to a discrete-time accurate solution. They assumed that both streams are dispatched simultaneously and the adjustment method does not influence the clearance cycle. It is shown that the approximate solution can be adjusted to the optimal solution by changing the green ratio in the switching cycle. However, the discrete-time LP problem in Ioslovich et al. [26] is more complex than the one considered in [27], since the two streams are allowed to be cleared at different times. Merely changing the green ratios at the switching cycle may not obtain the accurate solution.

In this paper, the LP problems proposed by Ioslovich et al. [26] are directly attacked using the strong duality theorem [28]. It is first shown that the approximate solution and the accurate solution do not coincide in many situations. Then, the relationship between the approximate solution and the accurate solution will be discussed. The errors introduced by discretization are carefully studied. It is shown that, in many cases, the discretized approximate solution can be converted to the accurate solution within a few gradient descent adjustments. Finally, an algorithm is proposed to implement this conversion. These findings shed light on the connection between the continuous-time and discrete-time signal timing models.

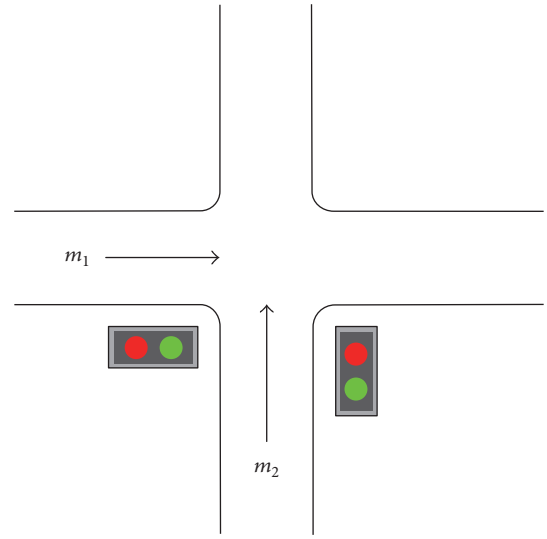


Figure 1: An isolated intersection with two one-way movements m_1 and m_2.

To give a detailed analysis, the rest of this paper is arranged as follows. Section 2 introduces the LP problem proposed by Ioslovich et al. [26] and lists the nomenclature used in this paper. Section 3 proposes two counterexamples to show that the optimality of the discretized approximate solution may not hold. Section 4 discusses the difference and connection between the discrete-time model and the continuous-time model. Finally, Section 5 concludes the paper.

2. Problem Presentation

2.1. Nomenclature and Assumptions. For presentation simplicity, the nomenclature in this paper follows Ioslovich et al. [26] as shown in Nomenclature List. Similar to Ioslovich et al. [26], consider an isolated intersection with two one-way streams m_1 and m_2 governed by signal lights, as shown in Figure 1. Note that the two-stream isolated intersection is chosen as an initial building block to understand the connection between the continuous-time model and the discrete-time model. This is a widely applied treatment even in recent works (e.g., [23]). Nevertheless, this assumption can be relaxed to consider general intersections. Both the continuous model and the discrete model are able to handle cases with more generalized intersections. Although it is hard to directly compare the two types of models in generalized intersections, we expect that the findings in this paper can shed light on more generalized intersections.

Likewise, the following assumptions are imposed as in Ioslovich et al. [26].

Assumption 1. (a) Each signal cycle starts with a green light for m_1.

(b) There is no lost time for each cycle. Suppose that the green ratio for stream m_1 in cycle k is denoted as $u(k)$; the green ratio for stream m_2 is then equal to $1 - u(k)$.

(c) The arrival flow rates and the saturation departure flow rates of both streams are constant over time. Denote a_i as the arrival flow rates for m_i and d_i as the departure flow rates.

(d) Without loss of generality, m_1 is assumed to be the main stream (i.e., $d_1 > d_2$).

(e) For some reasons (e.g., physical constraints), both streams are constrained by minimum green ratios. Define $u_{\min}$ and $1 - u_{\max}$ as the minimum green ratios for m_1 and m_2, respectively; then, the constraint on $u(k)$ is represented as $u_{\min} \le u(k) \le u_{\max}$.

(f) There exists a signal timing strategy such that queues in both streams can be cleared finally but may not be in the same cycle. Define $u_L = a_1/d_1$ and $u_H = 1 - a_2/d_2$. This assumption is written as $u_L < u_H$, $u_L < u_{\max}$, and $u_{\min} < u_H$.

(g) Cycle lengths are known to be constant T.

In Assumption 1, (a)–(f) are inherited from Ioslovich et al. [26] directly and (g) is added to simplify the discussion. In fact, if cycle lengths are not fixed, a similar conclusion can still be drawn but the detailed timing plan is much more complicated.

2.2. The Discrete-Time LP Model and Its PMP Solution. Similar to Ioslovich et al. [26], the discrete-time model is formulated as the following problem to minimize the total delay of vehicles:

$$\min \quad J_D = \sum_{k=0}^{N} \left[q_1(k) + q_2(k)\right] + \frac{a_1 + a_2}{2} T \sum_{k=0}^{N-1} u(k) \tag{1}$$

$$\text{s.t.} \quad q_1(k+1) \ge \max\left\{q_1(k) + d_1 T\left(u(k) - u_L\right), a_1 T\left(1 - u(k)\right)\right\} \tag{2}$$

$$q_2(k+1) \ge \max\left\{q_2(k) + d_2 T\left(u_H - u(k)\right), 0\right\} \tag{3}$$

$$u_{\min} \le u(k) \le u_{\max} \tag{4}$$

$$q_i(0) = q_{i,\text{int}}, \quad i = 1, 2, \tag{5}$$

for $k = 0, 1, \ldots, N-1$.

In (1)–(5), the objective function is a linear approximation for total delay. The decision variables are $u(k), k = 0, 1, \ldots, N$. For presentation convenience, it is assumed that the number of total cycles N is large enough such that both streams are cleared before cycle N. Equations (2) and (3) represent the evolution of both queues in time. Equation (4) gives the upper and lower bound of the green ratio in each cycle. Equation (5) is the initial queue lengths. It is easy to verify that this LP problem is equivalent to the Relaxed Discrete-Event Max-Plus Problem proposed by Haddad et al. [22].

In Ioslovich et al. [26], the above LP (see (1)–(5)) was transformed into an equivalent continuous-time optimal control problem (see (6)) with free terminal time t_f. The decision variable is $v(t)$.

$$\begin{aligned}
&\min \quad J_C = \int_0^{t_f} \left(q_1(t) + q_2(t)\right) dt \\
&\text{s.t.} \quad \frac{dq_1(t)}{dt} = d_1\left[u_L - v(t)\right] + w_1(t) \\
&\frac{dq_2(t)}{dt} = d_2\left[v(t) - u_H\right] + w_2(t) \\
&q_{i(T)} \ge 0, \quad i = 1, 2 \\
&w_i(t) \ge 0, \quad i = 1, 2 \\
&u_{\min} \le v(t) \le u_{\max} \\
&q_i(0) = q_{i,\text{int}}, \\
&q_i\left(t_f\right) = 0, \\
&i = 1, 2.
\end{aligned} \tag{6}$$

Ioslovich et al. [26] applied Pontryagin Maximum Principle (PMP) to derive the optimal solution to the continuous-time model (see (6)).

Particularly, the following four cases were discussed with respect to the order of $u_L, u_H, u_{\min}$, and $u_{\max}$: (I) $u_L < u_{\min} < u_H < u_{\max}$; (II) $u_L < u_{\min} < u_{\max} < u_H$; (III) $u_{\min} < u_L < u_{\max} < u_H$; (IV) $u_{\min} < u_L < u_H < u_{\max}$.

Table 1 lists the solution. The optimal continuous-time solution is denoted as $v^*(t)$. t_s is the switching time where the optimal control switches. M and R are represented as

$$\begin{aligned}
M &= \frac{d_1\left(u_{\max} - u_L\right)}{d_2\left(u_H - u_{\max}\right)}, \\
R &= \frac{u_{\min} - u_L}{u_H - u_{\min}}.
\end{aligned} \tag{7}$$

For example, Figure 2 illustrates the evolution of queue lengths under the optimal solution for Case I(a). It is observed that the optimal solution is a two-stage strategy, namely, bang-bang control. The green ratio remains $u_{\max}$ at first, causing the length of stream m_1 to decrease sharply and stream m_2 to increase. After the switching time t_s, the green ratio changes into $u_{\min}$ and both streams start to decrease. The optimal switch-over point where the solution switches from maximum to minimum green split is given in Table 1.

Table 1: The optimal solution to the continuous model.

Case	Condition	Solution	Switching time, t_s
I(a)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} > R$	$v^*(t) = \begin{cases} u_{\max}, & t \le t_s \\ u_{\min}, & t > t_s \end{cases}$	$\frac{q_{1,\text{int}} - Rq_{2,\text{int}}}{d_2(u_{\max} - u_H)(R - M)}$
I(b)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} < R$	$v^*(t) = u_{\min}$	
II(a)	$R < \frac{q_{1,\text{int}}}{q_{2,\text{int}}} < M$	$v^*(t) = \begin{cases} u_{\max}, & t \le t_s \\ u_{\min}, & t > t_s \end{cases}$	$\frac{q_{1,\text{int}} - Rq_{2,\text{int}}}{d_2(u_{\max} - u_H)(R - M)}$
II(b)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} < R$	$v^*(t) = u_{\min}$	
II(c)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} > M$	$v^*(t) = u_{\max}$	
III(a)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} > M$	$v^*(t) = u_{\max}$	
III(b)	$\frac{q_{1,\text{int}}}{q_{2,\text{int}}} < M$	$v^*(t) = \begin{cases} u_{\max}, & t < t_s \\ u_L, & t > t_s \end{cases}$	$\frac{q_{1,\text{int}}}{d_1(u_{\max} - u_L)}$
IV		$v^*(t) = \begin{cases} u_{\max}, & t < t_s \\ u_L, & t > t_s \end{cases}$	$\frac{q_{1,\text{int}}}{d_1(u_{\max} - u_L)}$

The discretized solution of $v^*(t)$ is obtained as follows. Suppose the cycle length is T; the discretization is done as

$$u_D(k) = v^*(kT), \quad k = 0, 1, \ldots, N-1. \tag{8}$$

For presentation simplicity, the solution $u_D(k)$, $k = 0, 1, \ldots, N-1$, is called the discretized approximate solution in the rest of this paper.

2.3. The Steady State and the Steady-State Solution. According to Haddad et al. [22], the steady state is defined as the state where the queue lengths are the same in the beginning and end of each cycle. This means, given the green ratio for a certain stream, the queue lengths of this stream keep the same in a certain period. Denote the queue lengths in the steady state as $q_{i,\text{ss}}$, $i = 1, 2$, and the green ratio in the steady state as u_{ss}. So, considering a single-cycle version of discrete-time model (see (1)–(5)), the steady state can be formulated into the following model:

$$\min \quad q_{1,\text{ss}} + q_{2,\text{ss}} + \frac{a_1 + a_2}{2} T u_{\text{ss}} \tag{9}$$

$$\text{s.t.} \quad q_{1,\text{ss}} \ge \max\left\{q_{1,\text{ss}} - d_1 T\left(u_{\text{ss}} - u_L\right), a_1\left(1 - u_{\text{ss}}\right)T\right\} \tag{10}$$

$$q_{2,\text{ss}} \ge \max\left\{q_{2,\text{ss}} - d_2 T\left(u_H - u_{\text{ss}}\right), 0\right\} \tag{11}$$

$$u_{\min} \le u_{\text{ss}} \le u_{\max}. \tag{12}$$

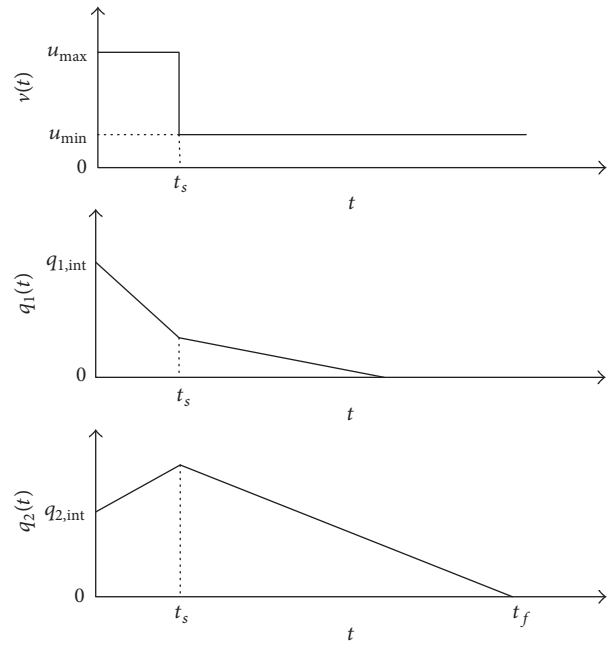


Figure 2: An illustration of the queue lengths and optimal timing plan in Case I(a).

The objective equation (9) is a linear approximation of delay in a single cycle. Equations (10) and (11) depict the queue evolution. Equation (12) gives the bound of the green ratio.

The optimal green ratio u_{ss} is called the steady-state solution. Haddad et al. [22] gave a neat analytical form of u_{ss} as follows:

$$u_{ss} = \begin{cases} \max\{u_L, u_{\min}\}, & a_1 < a_2, \\ \min\{u_H, u_{\max}\}, & a_1 > a_2. \end{cases} \tag{13}$$

According to Haddad et al. [22], the two streams that follow the discrete-time model (see (1)–(5)) will eventually enter the steady state under Assumptions (a)–(g). Therefore, in the accurate solution, the green ratios in the steady state are the steady-state solution u_{ss}.

3. The Nonoptimality of the Discretized Approximate Solution

In this section, it is shown that the discretized approximate solution does not always coincide with the accurate solution. The term "coincide" here means that the discretized approximate solution is not always optimal for the discrete-time model. In particular, two counterexamples are constructed. The first counterexample is used to deal with Case I and Case II where the minimum green ratio $u_{\min}$ plays an important role in the approximate solution. In contrast, the second counterexample is used to handle Case III and Case IV where the minimum green ratio $u_{\min}$ does not influence the approximate solution. The two counterexamples are as follows.

Counter Example 1 (for Case I and Case II). The parameters are set as $q_{1,\text{int}} = 60$, $q_{2,\text{int}} = 20$, $d_1 = 0.55$, $d_2 = 0.30$, $a_1 = 0.15$, $a_1 = 0.10$, $u_{\min} = 0.40$, $u_{\max} = 0.80$, $T = 160$, and $N = 6$. Then, the accurate solution and the discretized approximate solution are shown in Figure 3(a).

Counter Example 2 (for Case III and Case IV). The parameters are set as $q_{1,\text{int}} = 40$, $q_{2,\text{int}} = 40$, $d_1 = 0.60$, $d_2 = 0.35$, $a_1 = 0.15$, $a_1 = 0.10$, $u_{\min} = 0.15$, $u_{\max} = 0.80$, $T = 100$, and $N = 25$. Then, the accurate solution and the discretized approximate solution are shown in Figure 3(b).

As is shown in Figure 3, the accurate solution deviates from the discretized approximate solution in two ways. In Counterexample 1, the accurate solution roughly remains as a bang-bang form before the system enters steady state. The difference concentrates around the switching cycle and steady state. However, in Counterexample 2, the accurate solution exhibits oscillation and differs from the discretized approximation solution in almost every cycle.

Such deviations are caused by discretization and can be classified into two categories.

The first category of deviations is caused by the discretization of the model constraints. For the continuous-time model, the queue length for stream m_i is lower-bounded by the constraint $q_1(k) \geq 0$, while for the discrete-time model, the corresponding constraint is $q_1(k) \geq a_1 T(1 - u(k))$. When cycle length T is large, the discretization of model constraints may introduce huge errors. In Counterexample 1, the switching time locates in the middle of a cycle. As shown in Figure 4, the discretization leads to waste of green ratio around the switching cycle due to the restriction of the minimum green ratio. When T is small enough, the solution of the continuous-time model and the solution of the discrete-time model are coherent. A proper T should be selected to ensure consistency.

The second category deviation is caused by the discretization of the approximate solution. In Counterexample 2, the clearance time of stream m_1 locates in the middle of a cycle. So, the discretization causes oscillation in the accurate solution.

4. The Relationship between the Discretized Approximate Solution and the Accurate Solution

In this section, the connection between the discrete approximate solution and the accurate solution is further discussed.

For Case I and Case II, the errors are relatively concentrated. The discretized approximate solution does not deviate largely from the accurate solution in many cases. In fact, the discretized approximate solution can be modified to the accurate solution by some minor adjustments; see Algorithm 1. The details of adjustments will be discussed in Sections 4.1–4.3.

For Case III and Case IV, however, the dispersed errors make it hard to adjust the solution in a few steps. However, for some carefully chosen cycle length, the accurate solution coincides with the discretized approximate solution.

For example, in Case III, if the cycle length is chosen as $T = q_{1,\text{int}}/[nd_1(u_{\max} - u_L) + (d_1 - a_1)u_L]$, where n is any positive integer, the accurate solution can be written as

$$u(k) = \begin{cases} u_{\max}, & k \leq n, \\ u_L, & k > n. \end{cases} \tag{14}$$

It is easy to show that this accurate solution coincides with the corresponding discretized approximate solution.

In the rest of this section, the discussion will focus on the adjustments for Case I, and the adjustments for Case II are similar.

For presentation simplicity, define several denotations before starting the discussion. Define $\widehat{k}_1$ and $\widehat{k}_2$, respectively, as the clearance cycle indices in which streams m_1 and m_2 are cleared exactly:

$$\widehat{k}_1 := \max\{k : q_1(k) > a_1 T(1 - u(k))\}, \tag{15}$$

$$\widehat{k}_2 := \max\{k : q_2(k) > 0\}. \tag{16}$$

Apparently, in Case I(a), it holds that $\widehat{k}_1 \leq \widehat{k}_2$. So, both streams enter steady state after cycle $\widehat{k}_2$. Note that there is also a switch in the discretized approximate solution. Define $\widehat{k}_s$ as the switching cycle index where switch occurs; that is,

$$\widehat{k}_s : \min\{k : u(k) < u_{\max}\}. \tag{17}$$

The adjustments contain two parts. Section 4.1 deals with the adjustment in the steady state. Section 4.2 includes

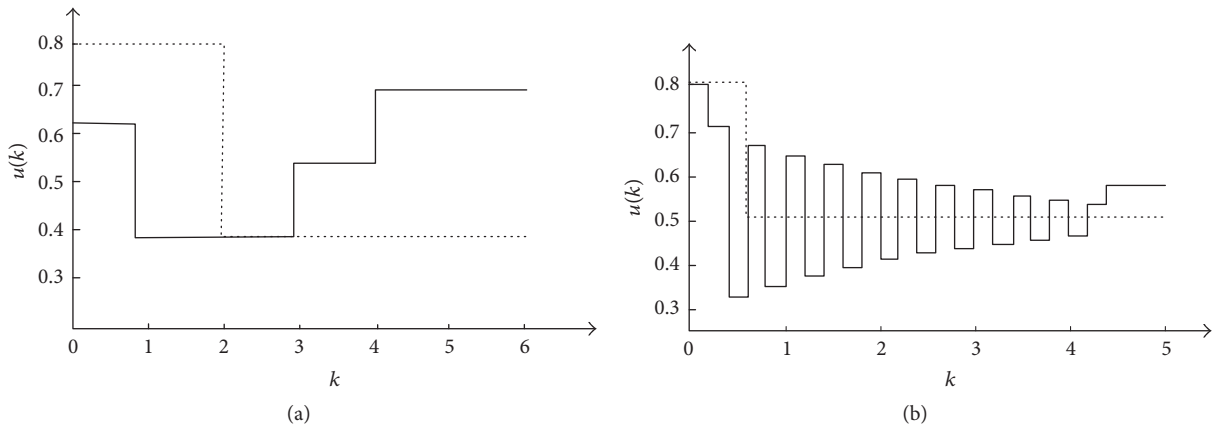


Figure 3: The two counterexamples. The dotted line represents the discretized approximate solution. The solid line represents the accurate solution.

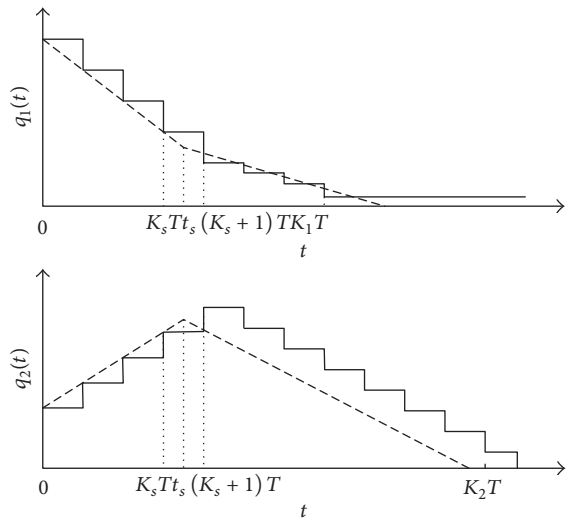


Figure 4: An illustration of the error caused by discretization. The dotted line represents the queue lengths in the continuous-time model. The solid line represents the queue lengths in the discrete-time model.

the adjustment around the switching cycle. Both parts are summarized as Algorithm 1. The optimality of Algorithm 1 will be proven in the Appendix. Finally, Counterexample 1 in Section 3 is revisited to better illustrate the adjustments.

4.1. The Steady-State Adjustment. The adjustment for the steady state is shown in lines (34)–(38) in Algorithm 1. Since the steady-state solution is optimal for the discrete-time model after the two streams enter steady state, $u(k)$ can be set as

$$u(k) = u_{ss}, \quad k > \widehat{k}_2. \tag{18}$$

However, in the clearance cycle $\widehat{k}_2$, the optimal green ratio is not u_{ss} due to discretization. So, $u(\widehat{k}_2)$ can be set as follows:

$$u\left(\widehat{k}_2\right) = \begin{cases} u_H - \dfrac{q_2\left(\widehat{k}_2\right)}{d_2 T}, & a_1 > a_2, \\ u_{\min}, & a_1 < a_2. \end{cases} \tag{19}$$

4.2. The Adjustment around the Switch. The adjustment around the switching cycle is shown in lines (2)–(33) in Algorithm 1. There are two cases of adjustment: increasing the green ratios after the switch and decreasing the green ratios before the switch. For presentation simplicity, this subsection only handles the case of increasing the green ratios after the switch. The adjustment to the other case is similar except for some details; see Algorithm 1, lines (9)–(20).

This adjustment can be viewed as a gradient descent approach to gradually reduce the total delay. It operates in three steps: (1) determine the direction of adjustment; (2) update the green ratio; (3) check feasibility of the new green ratio. It will be proven in the Appendix that such adjustment operations can be finished in a few steps.

Due to constraint equations (3) and (4), the discrete-time model is a nonsmooth model. Thus, a subdifferential will be used instead of a gradient. Without loss of generality, assume that both streams are cleared after the switching cycle (i.e., $\widehat{k}_1 \geq \widehat{k}_s$). The case of $\widehat{k}_1 < \widehat{k}_s$ can be transformed easily to the case of $\widehat{k}_1 > \widehat{k}_s$ (Algorithm 1, lines (2)–(8)).

In the first step, the direction of adjustment is determined by calculating the subdifferential. Denote the subdifferential of the total delay J_D with respect to $u(\widehat{k}_s)$ as ∂J_D. Define the function $Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s)$ as

$$Q\left(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s\right) = \begin{cases} \left[d_2\left(\widehat{k}_2 - \widehat{k}_s\right) - d_1\left(\widehat{k}_1 - \widehat{k}_s\right) + \dfrac{a_1 + a_2}{2}\right] T, & a_1 < a_2, \\ \left[d_2\left(\widehat{k}_2 - \widehat{k}_s\right) - d_1\left(\widehat{k}_1 - \widehat{k}_s\right) + a_1\right] T, & a_1 > a_2, \end{cases} \tag{20}$$

so as to represent the subdifferential in a more convenient way.

Clearly, the case of increasing the green ratios after the switch corresponds to the condition of $Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s) < 0$.

(1) Set $u(k)$ as the discretized approximate solution. Calculate $\hat{k}_1, \hat{k}_2, \hat{k}_s$ by Eq. (15)-(16);
(2) **if** $\hat{k}_1 < \hat{k}_2$ **then**
(3) Set $u(\hat{k}_s - 1) \leftarrow \min\{(q_1(\hat{k}_s - 1)/T)/(d_1 - a_1), u_{\max}\}$ and set $u(k) \leftarrow u_{\min}, k \geq \hat{k}_s$;
(4) Update $\hat{k}_2$ by Eq. (16) and set $\hat{k}_1 \leftarrow \hat{k}_s$;
(5) **if** $u(\hat{k}_s - 1) \leq u_{\min}$ **then**
(6) $u(\hat{k}_s - 1) \leftarrow u_{\min}$, $\hat{k}_s \leftarrow \hat{k}_1 - 1$ and update $\hat{k}_1$ and $\hat{k}_2$ by Eq. (15) and (16);
(7) **end**
(8) **end**
(9) **while** $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s - 1) > 0$ **do**;
(10) $\hat{k}_2 \leftarrow \hat{k}_2 - 1$. Update $u(\hat{k}_s - 1)$ such that stream m_2 is cleared exactly in cycle $\hat{k}_2$;
(11) Update $\hat{k}_1$ by Eq. (15);
(12) **if** $Q(\hat{k}_1, \hat{k}_2 + 1, \hat{k}_s - 1) < 0$ **then**
(13) $\hat{k}_2 \leftarrow \hat{k}_2$+1. Find the minimum $\hat{k}_1$ such that $Q(\hat{k}_1 + 1, \hat{k}_2, \hat{k}_s - 1) < 0$;
(14) Update $u(\hat{k}_s - 1)$ such that stream m_1 is cleared exactly in cycle $\hat{k}_1$;
(15) Set $\hat{k}_1 \leftarrow \hat{k}_1 + 1$;
(16) **end**
(17) **if** $u(\hat{k}_s - 1) < u_{\min}$ **then**
(18) $u(\hat{k}_s - 1) \leftarrow u_{\min}$, $\hat{k}_s \leftarrow \hat{k}_s - 1$ and update $\hat{k}_1$ and $\hat{k}_2$ by Eq. (15) and (16);
(19) **end**
(20) **end**
(21) **while** $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s) < 0$ **do**
(22) $\hat{k}_2 \leftarrow \hat{k}_2 + 1$. Update $u(\hat{k}_s)$ such that stream m_2 is cleared exactly in cycle $\hat{k}_2$;
(23) Update $\hat{k}_1$ by Eq. (15);
(24) **if** $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s) > 0$ **then**
(25) $\hat{k}_2 \leftarrow \hat{k}_2 - 1$. Find the maximum $\hat{k}_1$ such that $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s) \geq 0$;
(26) Update $u(\hat{k}_s)$ such that stream m_1 is cleared exactly in cycle $\hat{k}_1$;
(27) **else if** $Q(\hat{k}_1, \hat{k}_2 + 1, \hat{k}_s) > 0$ and $u(\hat{k}_s) \leq u_{\max}$ **then**
(28) break;
(29) **end**
(30) **if** $u(\hat{k}_s) > u_{\max}$ **then**
(31) $u(\hat{k}_s) \leftarrow u_{\max}$, $\hat{k}_s \leftarrow \hat{k}_s + 1$ and update $\hat{k}_1$ and $\hat{k}_2$ by Eq. (15)-(16);
(32) **end**
(33) end
(34) **if** $a_1 > a_2$ and $u_{\max} > u_H$ **then**
(35) Calculate $q_1(k)$, $q_2(k)$, by Eq. (2)-Eq. (3);
(36) Set $u(\hat{k}_2) \leftarrow u_H - q_2(\hat{k}_2)/(d_2 T)$;
(37) **for** $k \in (\hat{k}_2, N)$ **do** $u(k) = u_H$;
(38) **end**

Algorithm 1: The adjusting algorithm for the discretized approximate solution in Case I(a).

There are three situations based on the queue evolution in clearance cycles $\hat{k}_1$ and $\hat{k}_2$.

(1) Stream m_1 is cleared exactly in cycle $\hat{k}_1$; that is, $q(\hat{k}_1 + 1) = a_1 T(1 - u(\hat{k}_1)) = q_1(\hat{k}_1) - d_1 T(u(\hat{k}_1) - u_L)$. The subdifferential in this situation is calculated as

$$\partial J_D = \left[Q\left(\hat{k}_1 + 1, \hat{k}_2, \hat{k}_s\right), Q\left(\hat{k}_1, \hat{k}_2, \hat{k}_s\right) \right], \tag{21}$$

where "$[a, b]$" means the closed interval between real numbers a and b.

The physical meaning of the subdifferential J_D is shown as follows. The total delay changes if a slight disturbance is imposed on $u(\hat{k}_s)$. Denote $\Delta u(\hat{k}_s)$ as the increment in $u(\hat{k}_s)$ and ΔJ_D as the corresponding change in total delay. If $\Delta u(\hat{k}_s) > 0$, then ΔJ_D can be calculated as

$$\begin{aligned} &\Delta J_D \\ &= \left(d_2\left(\hat{k}_2 - \hat{k}_s\right) - d_1\left(\hat{k}_1 - \hat{k}_s\right) + \frac{a_1 + a_2}{2} \right) T \Delta u\left(\hat{k}_s\right) \\ &= Q\left(\hat{k}_1, \hat{k}_2, \hat{k}_s\right) \Delta u\left(\hat{k}_s\right). \end{aligned} \tag{22}$$

Similarly, if $\Delta u(\hat{k}_s) < 0$, the corresponding changes in total delay are written as $\Delta J_D = Q(\hat{k}_1 + 1, \hat{k}_2, \hat{k}_s)\Delta u(\hat{k}_s)$.

Therefore, if $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s) < 0$, the total delay can be reduced by increasing $u(\hat{k}_s)$. If $Q(\hat{k}_1, \hat{k}_2, \hat{k}_s) \geq 0$ and $Q(\hat{k}_1 + 1, \hat{k}_2, \hat{k}_s) < 0$, neither increasing nor decreasing $u(\hat{k}_s)$ will

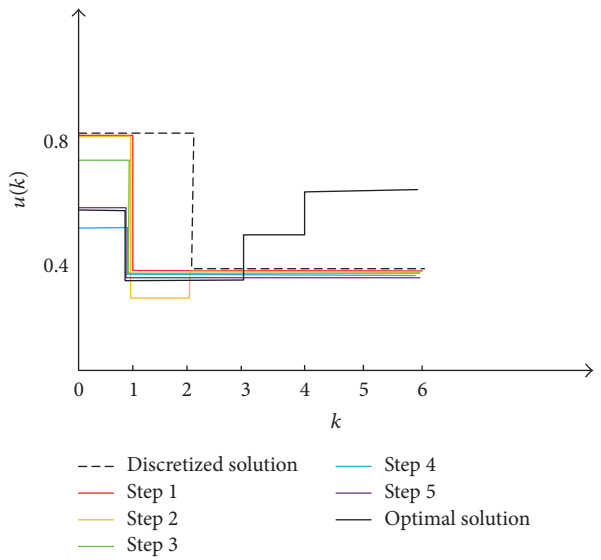


FIGURE 5: The adjustment of the discretized approximate solution.

render a better total delay and thus the optimal $u(\widehat{k}_s)$ is achieved.

(2) Stream m_2 is cleared exactly in cycle $\widehat{k}_2$; that is, $q_2(\widehat{k}_2+1)=0=q_2(\widehat{k}_2)-d_2T(u_H-u(\widehat{k}_2))$. The subdifferential can be calculated similarly as $\partial J_D=[Q(\widehat{k}_1,\widehat{k}_2,\widehat{k}_s),Q(\widehat{k}_1,\widehat{k}_2+1,\widehat{k}_s)]$.

Then, if $Q(\widehat{k}_1,\widehat{k}_2+1,\widehat{k}_s)<0$, the total delay can be reduced by increasing $u(\widehat{k}_s)$. If $Q(\widehat{k}_1,\widehat{k}_2+1,\widehat{k}_s)\geq 0$ and $Q(\widehat{k}_1,\widehat{k}_2,\widehat{k}_s)<0$, the optimal choice of $u(\widehat{k}_s)$ is obtained.

(3) Both streams are not cleared exactly in their clearance cycles. Then, the subdifferential is $\partial J_D=\{Q(\widehat{k}_1,\widehat{k}_2,\widehat{k}_s)\}$. If $Q(\widehat{k}_1,\widehat{k}_2,\widehat{k}_s)<0$, the total delay can be reduced by increasing $u(\widehat{k}_s)$.

In the second step, the green ratio $u(\widehat{k}_s)$ is updated by reducing $\widehat{k}_1$ and increasing $\widehat{k}_2$. The green ratio $u(\widehat{k}_s)$ is calculated such that either stream m_1 or stream m_2 is cleared exactly in its clearance cycle. The detailed calculation is shown in Algorithm 1 (lines (24)–(29)).

The third step is the feasibility check of whether the new green ratio $u(\widehat{k}_s)$ satisfies the constraint equation (4). In the case of increasing $u(\widehat{k}_s)$, the only possible violation is $u(\widehat{k}_s)>u_{\max}$. Therefore, if the feasibility check fails, $u(\widehat{k}_s)$ is set as $u_{\max}$ and a new bang-bang solution is obtained. Then, the algorithm goes back to the first step and determines the direction of adjustment.

4.3. The Numerical Examples. In this subsection, Counterexample 1 in Section 3 is revisited to illustrate the steps of Algorithm 1. Recall that the parameters are $d_1=0.55$, $d_2=0.3$, $a_1=0.15$, $a_2=0.10$, $u_{\min}=0.4$, $u_{\max}=0.8$, $q_{1,\text{int}}=60$, $q_{1,\text{int}}=20$, $T=160$, and $N=6$. Algorithm 1 will be applied to adjust the discretized approximate solution. The detailed operation is shown in Table 2.

The adjustment of the discretized approximate solution is also shown in Figure 5.

5. Conclusion

This paper studies the continuous-time and discrete-time signal timing models for the classical isolated signalized intersection with only two one-way vehicle flows. Against intuitions, the nonequivalence between the solutions of the two models is first shown by two counterexamples. Then, the differences between the solutions of the two models are explained. Finally, an algorithm is proposed to transform the discretized continuous-time solution to the discrete-time solution for many cases. Via this gradient descent based algorithm, a more concrete link is set up between the continuous-time and discrete-time signal timing models.

Appendix

Proof of the Optimality of Algorithm 1

The optimality is stated in the following theorem.

Theorem A.1. *The solution derived by Algorithm 1 is optimal for Case I(a).*

Theorem A.1 will be proven by the strong duality theorem (shown as Lemma A.2).

Lemma A.2 (strong duality theorem for linear programming). *A primal feasible solution and a dual feasible solution to a linear programming problem are optimal if and only if the corresponding primal and dual objectives are the same.*

Proof of Theorem A.1. For presentation simplicity, only the case of increasing the green ratios after the switch will be considered in this proof. The other case can be similarly proven. Since the adjustment for the switching cycle (lines (21)–(33)) is a gradient descent approach and the discrete LP model is apparently feasible, the algorithm will eventually terminate.

The dual problem to the discrete LP model (see (1)–(5)) is written as

$$\begin{aligned}\widetilde{J}_D=\max\quad & \sum_{i=1}^{2}q_{i,\text{int}}\alpha(0)\\ & +\sum_{K=0}^{N-1}\min\{S(k)u_{\max},(k)u_{\min}\} \qquad (A.1)\\ & +\sum_{K=0}^{N-1}\left[d_1u_L\alpha_1(k)-d_2u_H\alpha_2(k)\right]\end{aligned}$$

$$\text{s.t.}\quad -\alpha_i(k)+\alpha_i(k+1)-\beta_1(k)+1=0 \qquad (A.2)$$

$$-\alpha_i(N-1)-\beta_i(N-1)=0 \qquad (A.3)$$

$$\begin{aligned}S(k)=&-d_1\alpha_1(k)+d_2\alpha_2(k)+a_2\\ &+a_1(1-\beta_1(k)).\end{aligned} \qquad (A.4)$$

Table 2: The operation of the algorithm.

Cycle k	0	1	2	3	4	5
Discretized solution	0.8000	0.8000	0.4000	0.4000	0.4000	0.4000
Step 1	0.8000	0.4000	0.4000	0.4000	0.4000	0.4000
Step 2	0.8000	0.3833 (0.400)	0.4000	0.4000	0.4000	0.4000
Step 3	0.7833	0.4000	0.4000	0.4000	0.4000	0.4000
Step 4	0.5167	0.4000	0.4000	0.4000	0.4000	0.4000
Step 5	0.6636	0.4000	0.4000	0.4000	0.4000	0.4000
Optimal solution	0.6636	0.4000	0.5197	0.6667	0.6667	0.6667

There are three situations when the iteration of lines (21)–(33) terminates.

(1) The iteration terminates at line (28). Then, $u(\widehat{k}_s)$ is given such that stream m_2 is cleared exactly. The corresponding $\widehat{k}_1$, $\widehat{k}_2$, and $\widehat{k}_s$ satisfy $Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s) \le 0$ and $Q(\widehat{k}_1, \widehat{k}_2 + 1, \widehat{k}_s) > 0$.

(2) The iteration terminates at line (21) and $u(\widehat{k}_s)$ is set as $u_{\max}$. The corresponding $\widehat{k}_1$, $\widehat{k}_2$, and $\widehat{k}_s$ satisfy $Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s) \ge 0$.

(3) The iteration terminates at line (21) and $u(\widehat{k}_s)$ is given such that stream m_1 is cleared exactly. The corresponding $\widehat{k}_1$, $\widehat{k}_2$, and $\widehat{k}_s$ satisfy $Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s) \ge 0$ and $Q(\widehat{k}_1 + 1, \widehat{k}_2, \widehat{k}_s) < 0$.

The proofs of the three situations is similar. Only the first situation is handled here. Moreover, assume $a_1 < a_2$. The proof of $a_1 > a_2$ is the opposite.

Denote ϵ_1 and ϵ_2, respectively, as $\epsilon_1 = 0$ and $\epsilon_2 = -Q(\widehat{k}_1, \widehat{k}_2, \widehat{k}_s)/d_2$. Then the dual solution is rewritten as

$$\alpha_i(k) = \max\left\{\widehat{k}_i - k + \epsilon_i, 0\right\}, \quad i = 1, 2, \tag{A.5}$$

$$\beta_i(k) = -\alpha_i(k) + 1 - \alpha_i(k+1), \quad i = 1, 2. \tag{A.6}$$

Substituting (A.5) and (A.6) into (A.4), $S(k)$ can be represented as

$$S(k) \begin{cases} -d_1\left(\widehat{k}_1 - k\right) + d_2\left(\widehat{k}_2 - k + \epsilon_2\right) + \dfrac{a_1 + a_2}{2}, & k < \widehat{k}_1, \\ d_2\left(\widehat{k}_2 - k + \epsilon_2\right) + \dfrac{-a_1 + a_2}{2}, & \widehat{k}_1 \le k \le \widehat{k}_2, \\ \dfrac{-a_1 + a_2}{2}, & k > \widehat{k}_2. \end{cases} \tag{A.7}$$

With simple calculation, the sign of $S(k)$ is determined as

$$S(k) \begin{cases} < 0, & k < \widehat{k}_s, \\ = 0, & k = \widehat{k}_s, \\ > 0, & k > \widehat{k}_s. \end{cases} \tag{A.8}$$

Finally, it is obtained by comparing the primal and dual objectives that

$$\begin{aligned} \widetilde{J}_D - J_D = d_2 T \epsilon_2 \Bigg[& \frac{q_{2,\text{int}}}{d_2 T} - \widehat{k}_s\left(u_H - u_{\max}\right) \\ & - \left(u_H - u\left(\widehat{k}_s\right)\right) - \left(\widehat{k}_2 - \widehat{k}_s + 1\right)\left(u_H - u_{\min}\right) \Bigg] \\ = 0. & \end{aligned} \tag{A.9}$$

This proves that the primal and dual objective values are equal. By Lemma A.2, both primal and dual solutions are optimal. Therefore, Algorithm 1 gives the optimal solution for Case I(a). □

Nomenclature List

Known Constants

N: The total number of cycles in the observation period
T: The cycle length of the isolated intersection
m_i: The two streams, $i = 1, 2$
a_i: The arrival rate of stream m_i, $i = 1, 2$
d_i: The saturation departure flow rate of stream m_i, $i = 1, 2$

$q_{i,\text{int}}$: The initial queue length of stream m_i, $i = 1, 2$
u_L: The green ratio that keeps stream m_1 unchanged (i.e., $u_L \coloneqq a_1/d_1$
u_H: The green ratio that keeps stream m_2 unchanged (i.e., $u_H \coloneqq (d_2 - a_2)/d_2$
$u_{\min}, u_{\max}$: The minimum and maximum bounds of green ratio
u_{ss}: The steady-state solution of the discrete-time model.

Variables and Functions regarding the Continuous-Time Model

t_s: The switching time of signal timing plan in the continuous-time optimal solution
t_f: The final time in the continuous-time model
J_C: The objective of the continuous-time model, representing the approximate total delay
$v(t)$: The green time ratio of stream m_i at time t, the decision variable for continuous-time model
$v^*(t)$: The approximate solution, that is, the optimal solution to the continuous-time model.

Variables and Functions regarding the Discrete-Time Model

$u(k)$: The green time ratio of stream m_i in cycle k, the decision variable for the discrete-time model
$q_i(k)$: The queue lengths of stream m_i in the beginning of cycle k, $k = 0, \ldots, N-1$
$\widehat{k}_1$: The index of the cycle in which the stream m_1 is cleared
$\widehat{k}_2$: The index of the cycle in which the stream m_2 is cleared
$\widehat{k}_s$: The index of the cycle in which $u(k)$ switches from $u_{\max}$ to $u_{\min}$
J_D: The objective of the discrete-time model, representing approximate total delay
$u^*(k)$: The accurate solution, that is, the optimal solution to the discrete-time model
$u_D(k)$: The discretized solution of $v^*(t)$.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Support by Beijing Natural Science Foundation (4164083), National Natural Science Foundation of China (61603005, 71303269), Fundamental Research Funds for the Central Universities, and Central University of Finance and Economics Major Research Task of Fostering Project (14ZZD006) is gratefully acknowledged.

References

[1] F. Webster, Traffic signal settings. Road Research Technical Paper 39. Great Britain Road Research Laboratory, London, UK, 1958.
[2] D. Green, "Control of oversaturated intersections," *Operational Research Quarterly*, vol. 18, no. 2, pp. 161–173, 1967.
[3] D. C. Gazis, "Optimum control of a system of oversaturated intersections," *Operations Research*, vol. 12, no. 6, pp. 815–831, 1964.
[4] T.-H. Chang and G.-Y. Sun, "Modeling and optimization of an oversaturated signalized network," *Transportation Research Part B: Methodological*, vol. 38, no. 8, pp. 687–707, 2004.
[5] A. Di Febbraro and N. Sacco, "On modelling urban transportation networks via hybrid Petri nets," *Control Engineering Practice*, vol. 12, no. 10, pp. 1225–1239, 2004.
[6] M. Dotoli and M. P. Fanti, "An urban traffic network model via coloured timed Petri nets," *Control Engineering Practice*, vol. 14, no. 10, pp. 1213–1229, 2006.
[7] X. Ma, H. Yu, Y. Wang, Y. Wang, and J. Gomez-Gardenes, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, Article ID e0119044, 2015.
[8] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
[9] W. Sun, Y. Wang, G. Yu, and H. X. Liu, "Quasi-optimal feedback control for a system of oversaturated intersections," *Transportation Research Part C: Emerging Technologies*, vol. 57, pp. 224–240, 2015.
[10] W. Sun, Y. Wang, G. Yu, and H. X. Liu, "Quasi-optimal feedback control for an isolated intersection under oversaturation," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 109–130, 2016.
[11] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
[12] T. Le, P. Kovács, N. Walton, H. L. Vu, L. L. H. Andrew, and S. S. P. Hoogendoorn, "Decentralized signal control for urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 431–450, 2015.
[13] B. Park, C. J. Messer, and T. Urbanik II, "Traffic signal optimization program for oversaturated conditions: genetic algorithm approach," *Transportation Research Record*, no. 1683, pp. 133–142, 1999.
[14] P. G. Michalopoulos and G. Stephanopoulos, "Oversaturated signal systems with queue length constraints—I: single intersection," *Transportation Research*, vol. 11, no. 6, pp. 413–421, 1977.
[15] P. G. Michalopoulos and G. Stephanopoulos, "Oversaturated signal systems with queue length constraints-II. Systems of intersections," *Transportation Research*, vol. 11, no. 6, pp. 423–428, 1977.
[16] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko, *The mathematical theory of Optimal processes*, vol. 43, Wiley-Interscience, 1962.
[17] R. Allsop, "SIGSET: A computer program for calculating traffic signal settings," *Transportation Engineering Control*, vol. 13, no. 2, pp. 58–60, 1971.

[18] R. E. Allsop, "Estimating the traffic capacity of a signalized road junction," *Transportation Research*, vol. 6, no. 3, pp. 245–255, 1972.

[19] R. E. Allsop, "SIGCAP: A computer program for assessing the traffic capacity of signal-controlled road junctions," *Transportation Engineering Control*, vol. 17, no. 819, pp. 338–341, 1976.

[20] J. Haddad, B. De Schutter, D. Mahalel, and P.-O. Gutman, "Steady-state and N-stages control for isolated controlled intersections," in *Proceedings of the 2009 American Control Conference, ACC 2009*, pp. 2843–2848, June 2009.

[21] J. Haddad, B. De Schutter, D. Mahalel, I. Ioslovich, and P.-O. Gutman, "Optimal steady-state traffic control for isolated intersections," in *Proceedings of the 6th IFAC symposium on robust control design*, 2009, pp. 96–101.

[22] J. Haddad, B. De Schutter, D. Mahalel, I. Ioslovich, and P.-O. Gutman, "Optimal steady-state control for isolated traffic intersections," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 55, no. 11, pp. 2612–2617, 2010.

[23] L. Li, K. Yang, Z. Li, and Z. Zhang, "The optimality condition of the multiple-cycle smoothed curve signal timing model," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 46–57, 2013.

[24] K. Han, V. V. Gayah, B. Piccoli, T. L. Friesz, and T. Yao, "On the continuum approximation of the on-and-off signal control on dynamic traffic networks," *Transportation Research Part B: Methodological*, vol. 61, pp. 73–97, 2014.

[25] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.

[26] I. Ioslovich, J. Haddad, P.-O. Gutman, and D. Mahalel, "Optimal traffic control synthesis for an isolated intersection," *Control Engineering Practice*, vol. 19, no. 8, pp. 900–911, 2011.

[27] B. Zou, J. Hu, and Y. Zhang, "The optimal discretized timing plan for individual oversaturated intersections," in *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, ITSC 2012*, pp. 1656–1660, September 2012.

[28] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, vol. 116 of *International Series in Operations Research & Management Science*, Springer, New York, NY, USA, 3rd edition, 2008.

Hindawi
Journal of Sensors
Volume 2017, Article ID 1321237, 8 pages
https://doi.org/10.1155/2017/1321237

Research Article

Operating Time Division for a Bus Route Based on the Recovery of GPS Data

Jian Wang and Yang Cao

School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin 150090, China

Correspondence should be addressed to Yang Cao; caoyang_202@163.com

Received 24 April 2017; Revised 22 June 2017; Accepted 11 July 2017; Published 14 August 2017

Academic Editor: Xiaolei Ma

Bus travel time is an important source of data for time of day partition of the bus route. However, in practice, a bus driver may deliberately speed up or slow down on route so as to follow the predetermined timetable. The raw GPS data collected by the GPS device equipped on the bus, as a result, cannot reflect its real operating conditions. To address this concern, this study first develops a method to identify whether there is deliberate speed-up or slow-down movement of a bus. Building upon the relationships between the intersection delay, link travel time, and traffic flow, a recovery method is established for calculating the real bus travel time. Using the dwell time at each stop and the recovered travel time between each of them as the division indexes, a sequential clustering-based time of day partition method is proposed. The effectiveness of the developed method is demonstrated using the data of bus route 63 in Harbin, China. Results show that the partition method can help bus enterprises to design reasonable time of day intervals and significantly improve their level of service.

1. Introduction

A well-designed bus schedule scheme is important for increasing bus transit ridership [1]. Bus passenger demand differs greatly at different time intervals during the everyday operation. Before the overall design of a bus schedule scheme, the operating time of a bus route should be divided into multiple time intervals for which different schedule schemes should be made. This greatly helps formulate precise operating and dispatching schemes for buses and reduce the operational costs of a bus transit enterprise.

In recent years, buses in a number of large cities in China have been equipped with GPS devices [2–5]. Bus enterprises can now directly retrieve the bus travel time between any two stops from this database. However, given the predetermined timetables, the travel time may not reflect the actual performance of a bus. When faced with traffic jams, a bus driver may deliberately accelerate if the bus is to arrive at a downstream stop no later than the scheduled time. Although it may manage to arrive on time, the bus typically undergoes frequent acceleration and deceleration enroute which not only reduces the comfort of passengers but also increases the probability of traffic accidents. In contrast, the bus driver may deliberately slow down in smooth traffic so as to avoid early arrival at the downstream stop. Consequently, the lowered travel speed may leave the passengers with the impression that the bus service is inefficient. These two kinds of drivers' behavior are common in China [6, 7]. The root cause is that the initial timetables are usually nonoptimal considering the real-time traffic conditions. Therefore, the retrieved GPS data cannot be used directly. To obtain the actual travel speed and to further divide the operating time, the effect of drivers' behavior should be considered.

Scholars have conducted much research on the optimization of bus schedule schemes but have rarely investigated the division of the operating time [8–11]. To evaluate the effectiveness of a bus schedule scheme, Patnaik et al. [12] selected as indexes the numbers of passengers boarding and alighting the bus and the number of midway stops. The buses from the starting stop were then divided into several classes. The data used to develop the models were collected by the Automatic Passenger Counters (APC) on buses operated

by a transit agency in the northeast region of the United States. Guihaire and Hao [13] presented a global review of the crucial strategic and tactical steps of transit planning: the design and schedule of the network. They pointed out that the bus operating period mainly depended on the passengers' requirements which were different at different times. However, no analytic method has been developed for time of day partition. Using ridership data from a bus smart card system, Yue [14] obtained an ordered sampling of the passengers' arrival ratio curve and divided the operating time into multiple intervals using the Fisher optimal segmentation method. In his model, only the passenger volume was considered and the bus travel speed was neglected. As a result, the bus operating conditions were not fully considered during the partition. Shen et al. [15] proposed an improved K-means clustering algorithm for the division of the bus operating period based on GPS data. However, only the bus travel speed was used and the passenger demand was not considered. Given that, in different time intervals, a transit agency tends to arrange different bus dispatching frequencies because of the different passenger demand, this study becomes less practically promising. Bie et al. [16] selected the dwell time at each stop and the travel time between each pair of them as indexes and developed a rapid division algorithm. This is the first study that considers both the bus travel speed and the passenger demand in time of day partition. However, the GPS data were used directly without considering the deliberate speed-up or slow-down movement.

The existing methods for operating time division exhibit two shortcomings: (i) only the passenger flow volume is taken into account and (ii) data are obtained typically through manual work which consumes much manpower and many other resources. The method proposed in this study builds the relationship between time division and bus schedule scheme and successfully addresses these shortcomings.

The contributions of this study are twofold. Firstly, we develop a method to identify whether there is deliberate speed-up or slow-down movement of a bus. A recovery method is then established for calculating the real bus travel time based on raw GPS data. To the best of our knowledge, no research so far has investigated this kind of problem. Secondly, a sequential clustering algorithm is developed to partition the operating period into multiple intervals based on the recovered bus travel time and dwell time at stops.

The structure of this paper is organized as follows. In Section 2, a recognition method for bus operating state is first developed followed by a recovery method for the bus travel time. A discussion is provided as to why the recovery travel time and dwell time are selected as division indexes. In Section 3, a sequential sample clustering algorithm is proposed to divide the operating time into multiple time intervals using the recovered travel time and dwell time. Section 4 presents a real case study and Section 5 concludes the paper.

2. Development of the Operating Time Division Method

2.1. Recognition of the Bus Operating State. In this paper, unless stated otherwise, all time is measured in units of seconds. Let us assume that a bus i passes m stops in total during an operating period n. According to its timetable, the planned travel time of bus i from the mth stop to the $(m+1)$th stop is denoted as $T_i^n(m, m+1)$. The planned operating time T_i^n can be written as follows:

$$T_i^n = \sum_{m=1}^{M-1} T_i^n (m, m+1). \tag{1}$$

When traveling along a route, a bus usually passes through three different kinds of regions, namely, stops, road sections, and intersections. Therefore, the planned travel time of bus i from the mth stop to the $(m+1)$th stop can be further divided as follows:

$$\begin{aligned} T_i^n (m, m+1) = a_i^n (m, m+1) + b_i^n (m, m+1) \\ + c_i^n (m+1), \end{aligned} \tag{2}$$

where $a_i^n(m, m+1)$ denotes the travel time spent at road sections, $b_i^n(m, m+1)$ denotes the travel time spent at intersections, and $c_i^n(m+1)$ denotes the travel time spent at bus stops.

$a_i^n(m, m+1)$, $b_i^n(m, m+1)$, and $c_i^n(m+1)$ can be extracted from GPS data in combination with a geographic information system (GIS) map. The actual travel time of the bus i from the mth stop to the $(m+1)$th stop, denoted as $\widehat{T}_i^n(m, m+1)$, can be rewritten as follows:

$$\begin{aligned} \widehat{T}_i^n (m, m+1) = \widehat{a}_i^n (m, m+1) + \widehat{b}_i^n (m, m+1) \\ + \widehat{c}_i^n (m+1). \end{aligned} \tag{3}$$

(1) Recognition of a Driver's Deliberate Acceleration. Theoretically, the bus travel time at road sections and intersections increases under traffic jams.

$$\begin{aligned} \widehat{a}_i^n (m, m+1) &> a_i^n (m, m+1), \\ \widehat{b}_i^n (m, m+1) &> b_i^n (m, m+1), \\ \widehat{T}_i^n (m, m+1) &> T_i^n (m, m+1). \end{aligned} \tag{4}$$

At intersections, bus drivers tend to reduce speed because of the queuing vehicles and the restriction of changing lanes. However, a driver can frequently accelerate and decelerate at road sections to reduce the travel time and to ensure punctual arrivals at the downstream stops.

Case 1.

$$\widehat{T}_i^n (m, m+1) = T_i^n (m, m+1). \tag{5}$$

In Case 1, although a bus may be delayed at intersections, it still arrives at the downstream stops on time due to deliberate acceleration at road sections.

Case 2.

$$\begin{aligned}&\widehat{T}_i^n(m,m+1) > T_i^n(m,m+1),\\&\left[\widehat{T}_i^n(m,m+1) - T_i^n(m,m+1)\right]\\&\quad < \left[\widehat{b}_i^n(m,m+1) - b_i^n(m,m+1)\right]\\&\qquad + \left[\widehat{c}_i^n(m,m+1) - c_i^n(m,m+1)\right].\end{aligned} \tag{6}$$

In Case 2, although the driver may deliberately speed up the bus, it does not arrive on time at the downstream stops.

Case 3.

$$\begin{aligned}&\widehat{T}_i^n(m,m+1) > T_i^n(m,m+1),\\&\left[\widehat{T}_i^n(m,m+1) - T_i^n(m,m+1)\right]\\&\quad \geq \left[\widehat{b}_i^n(m,m+1) - b_i^n(m,m+1)\right]\\&\qquad + \left[\widehat{c}_i^n(m,m+1) - c_i^n(m,m+1)\right].\end{aligned} \tag{7}$$

In Case 3, the increase in the bus travel time at road sections exceeds or equals the total increase in the travel time spent at intersections and in the dwell time at stops. The bus may run normally or undergo deliberate acceleration.

(2) Recognition of a Driver's Deliberate Deceleration. When the traffic volume is low, the bus travel times at road sections and intersections may decline.

$$\begin{aligned}&\widehat{a}_i^n(m,m+1) < a_i^n(m,m+1),\\&\widehat{b}_i^n(m,m+1) < b_i^n(m,m+1),\\&\widehat{T}_i^n(m,m+1) < T_i^n(m,m+1).\end{aligned} \tag{8}$$

Theoretically, if the timetable is not optimized in real time, the driver may deliberately slow down the bus to enable punctual arrivals according to the schedule.

Case 1.

$$\widehat{T}_i^n(m,m+1) = T_i^n(m,m+1). \tag{9}$$

In Case 1, although the bus is slightly delayed at intersections, it still arrives at the downstream stops on time, since the driver deliberately slows down the bus.

Case 2.

$$\begin{aligned}&\widehat{T}_i^n(m,m+1) < T_i^n(m,m+1),\\&\left[T_i^n(m,m+1) - \widehat{T}_i^n(m,m+1)\right]\\&\quad < \left[b_i^n(m,m+1) - \widehat{b}_i^n(m,m+1)\right]\\&\qquad + \left[c_i^n(m,m+1) - \widehat{c}_i^n(m,m+1)\right].\end{aligned} \tag{10}$$

In Case 2, although the drive deliberately slows down the bus at road sections, the bus still arrives at the downstream stops ahead of the scheduled time.

Case 3.

$$\begin{aligned}&\widehat{T}_i^n(m,m+1) < T_i^n(m,m+1),\\&\left[T_i^n(m,m+1) - \widehat{T}_i^n(m,m+1)\right]\\&\quad \geq \left[b_i^n(m,m+1) - \widehat{b}_i^n(m,m+1)\right]\\&\qquad + \left[c_i^n(m,m+1) - \widehat{c}_i^n(m,m+1)\right].\end{aligned} \tag{11}$$

In Case 3, the decrease in the bus travel time at road sections exceeds or equals the total decrease in the travel time spent at intersections and in the dwell time at stops. The bus may run normally or undergo deliberate deceleration.

2.2. Recovery of the Optimal Travel Time on the Road. When a driver's deliberate acceleration or deceleration is recognized as discussed in Section 2.1, the retrieved GPS data cannot be directly used for the optimization of the schedule scheme. This effect should be considered for recovering the optimal bus travel time on the road.

The delay time of a bus at an intersection can be calculated by subtracting the travel time at a preset speed from the travel time spent at an intersection. During the operating period, n, a number of buses pass through the intersection and their average delay can be directly calculated. Assuming that $\overline{d}_i^1$ denotes the average delay at the timetable's initial operation stage, the traffic conditions will change after a certain period of time, and the average delay will become $\overline{d}_i^2$.

Generally speaking, the traffic flow on a road increases/decreases as a result of an increase/decrease in traffic flow at the adjacent intersection. According to the theory of traffic engineering, the travel time spent at a road section or at an intersection is directly proportional to the traffic flow. At a signalized intersection, the average delay $\overline{d}$ can be calculated by the following [17].

$$\begin{aligned}\overline{d} = {}& \frac{0.5C\left(1-\lambda_i\right)}{1-\left[\min\left(1,x_i\right)\cdot\lambda_i\right]}\\&+ 900T\left[\left(x_i-1\right)^2 + \sqrt{\left(x_i-1\right)^2 + \frac{4x_i}{\mathrm{Cap}_i\cdot T}}\right],\end{aligned} \tag{12}$$

where λ_i, x_i, and Cap_i denote the green ratio, degree of saturation, and traffic capacity, respectively, of the phase for bus i. T denotes the length of the analysis period and is generally set at 0.25 h.

$$x_i = \frac{q_i/S_i}{\lambda_i}, \tag{13}$$

where q_i and S_i denote the ratios of the arrival and saturation flows of the entrance lane for bus i, respectively.

For bus i, when the average delay changes from $\overline{d}_i^1$ to $\overline{d}_i^2$ while the other variables remain unchanged, the variation ratio of the flow at the entrance lane can be derived according to (12)-(13). Since r_i denotes the ratio of the flow after a certain period of time to the original one, r_i can also denote the

variation ratio of traffic flow which will be used for recovering the optimal travel time of the bus on the road.

Through field observations, a relationship is shown to exist between the average speed of traffic on urban roads and the flow. At low traffic flow, speed is insensitive to the increase in flow and only decreases slightly. When the flow increases and is close to the capacity of the road, the speed decreases significantly. When the flow is lower than the capacity of the road, the average speed varies with the flow in an approximate linear fashion:

$$q_i' = \alpha + \beta v_i, \tag{14}$$

where q_i' denotes the flow in pcu/h of the road section, v_i denotes the average speed of the traffic flow in km/h, and α and β are constants to be determined.

According to the characteristics of traffic flow, when free-flow speed u_f occurs, the traffic flow equals 0 ($q_i' = 0$). When the speed equals the optimal value u_m, the traffic flow q_i' reaches the maximum and the saturation flow ratio S_i is achieved. Therefore, the following equations hold:

$$\begin{aligned} \alpha + \beta \times u_f &= 0, \\ \alpha + \beta \times u_m &= S_i. \end{aligned} \tag{15}$$

By calculation, we can get $a = Su_f/(u_f - u_m)$, $b = S/(u_m - u_f)$.

$$v_i = \left(q_i' - \frac{S_i u_f}{u_f - u_m} \right) \frac{u_m - u_f}{S_i}. \tag{16}$$

Assuming that the flow changes to $r_i q_i'$ after the bus dispatching scheme is executed for a certain period of time, the average travel speed v_i' of the bus can be calculated by

$$v_i' = \left(r_i q_i' - \frac{S_i u_f}{u_f - u_m} \right) \frac{u_m - u_f}{S_i}. \tag{17}$$

Defining $r_i' = v_i'/v_i$, the following expressions can be obtained:

$$\begin{aligned} r_i' &= \frac{\left(u_f - u_m\right) r_i q_i' - S_i u_f}{\left(u_f - u_m\right) q_i' - S_i u_f}, \\ \widehat{a}^n(m, m+1) &= \frac{1}{r_i'} a^n(m, m+1). \end{aligned} \tag{18}$$

Let $a^n(m, m+1)$ denote the average travel time of the bus from the mth to the $(m+1)$th stop within the operating time period n at the timetable's initial operation stage. The optimal travel speed after a certain period of time becomes $\widehat{a}^n(m, m+1)$ which denotes the recovered average speed from the mth to the $(m+1)$th stop.

r_i' is the most important parameter which plays a decisive role in the travel time recovery process. Figure 1 illustrates the overall process for calculating r_i'.

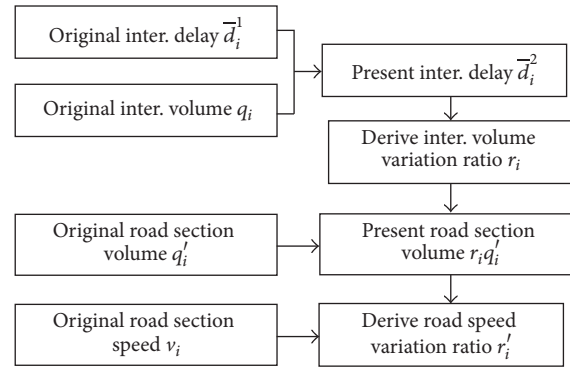


Figure 1: Flow chart of the bus travel time recovery.

2.3. Determination of the Operating Time Division Indexes. To divide the operating period, we first analyze all the historical data and then group into the same class buses with similar operating states and starting times into the same class. The corresponding operating time is referred to as a time interval. In this study, the dwell time at each stop and the interstop travel time (the recovered value as described in Section 2.2) are selected as the division indexes.

(1) Bus Dwell Time at Stops. The dispatching frequency affects the passenger volume of a bus route when the vehicle capacity of a bus is fixed. In each interval, the frequency is kept constant. Only when the passenger volume is also constant or slightly fluctuates will the passenger load factor of each dispatched bus be similar to one another. As a result, the uneven bus occupancy rate in an interval can be avoided.

With GPS data only, the number of alighting/boarding passengers at each stop is not available. However, empirical results show that the bus dwell time is positively proportional to the number of alighting/boarding passengers. Namely, a larger number of alighting/boarding passengers will result in longer dwell time. Though the bus dwell time is also affected by some other secondary factors such as the fare structure and the bus vehicle type (whether all doors can be used by the alighting passengers), they are all predetermined and remain unchanged for a given bus line. Hence the fluctuation of the bus dwell time at stops is mainly dependent on the number of alighting/boarding passengers. Therefore, the total dwell time at all stops is used to measure the passenger demand. Buses with similar total dwell time will be classified into the same time interval.

Let $\Delta D_{\max}$ denote the maximum permissible difference in the total dwell time at all stops for a bus in the same period. It can be calculated by $\Delta D_{\max} = \varepsilon \cdot \delta \cdot T_{ob}$ where δ denotes the maximum number of passengers, ε denotes the passenger carrying factor which is used mainly for adjusting the expected degree of crowdedness in the bus, and T_{ob} denotes the average boarding time of each passenger at each stop.

(2) Bus Interstop Travel Time. The bus operating status is affected not only by the arrival passenger volume at each stop, but also by the traffic conditions in real time. The

road traffic conditions influence the bus interstop travel time and hence the punctuality of the bus at each stop. For two buses dispatched consecutively from the same depot, if they have identical numbers of alighting/boarding passengers at each stop but different interstop travel time, they will experience different total travel time as well as different levels of punctuality at stops. Hence these two buses should not be classified into the same time of day interval.

Let $\Delta E_{\max}$ denote the maximum permissible difference in the total travel time among all stops for a bus in the same period. It can be calculated by $\Delta E_{\max} = \max\{0, H - \varepsilon \cdot \delta \cdot T_{ob}\}$, where H denotes the departure time interval for the buses as stated in the timetable.

Study [18] has analyzed the division indexes in different time of day for a bus route. However, the bus travel time obtained from GPS data was used directly. The deliberate acceleration or deceleration was not considered which renders the division results nonoptimal.

3. Operating Time Division Algorithm

Some classical clustering algorithms (such as K-means clustering) have achieved favorable results in index-based classification but are not suitable for this study. These classical algorithms do not take the order of data into account but quantify the correlation among data by using one of the distance metrics (such as the Euclidean and Mahalanobis distances). If the sequence of the buses is not taken into consideration, the buses with nonadjacent departure time intervals may be included in the same class. For example, when the first, second, third, fourth, tenth, and twentieth buses are included in the same operating period, this period can be divided into three subplots: subplot 1 includes the first, second, third, and fourth buses; subplot 2 includes the tenth bus; and subplot 3 includes the twentieth bus. Subplots 2 and 3 are quite short leading to frequent transitions between different bus dispatching schemes which reduces the management efficiency of the bus enterprise [19].

Given that the sequential sample clustering requires that the data sequence not be disturbed, a Fisher sequential sample clustering method (also referred to as optimal segmentation) is the most effective method [18]. There are 2^{n-1} division methods for n sequential samples. Each division method corresponds to segmentation. Among these segmentations, there exists an optimal segmentation that minimizes the difference within a segment and maximizes the difference among segments. To help achieve the optimal segmentation, the diameter of a class should be defined. After that the loss function is defined according to the constraint that the neighboring samples should be included in the same class. The optimal classification is found through a step-by-step recursive calculation with the objective of minimizing the loss function. The details of the procedure are described below.

(1) Calculation of the Diameter of a Class. In this study, the ordered variables are denoted as $x_1, x_2, \ldots, x_n$ (each variable x_i denotes an m-dimensional column vector, $i = 1, \ldots, n$). $m = 2$ given that the dwell time and the travel time are selected as two division indexes. Assuming that $\{x_i, x_{i+1}, \ldots, x_n\}$ denotes a segment ($1 \le i \le j \le n$), the diameter of a class (also referred to as the sum of the squares of deviation) $A(i, j)$ can be written as follows:

$$A(i, j) = \sum_{l=i}^{j} \left(x_l - \overline{x}_{i,j}\right)' \left(x_l - \overline{x}_{i,j}\right). \tag{19}$$

(2) Calculation of the Loss Function. For simplicity, the variable x_i ($i = 1, \ldots, n$) is denoted by its subscript i. Assuming that i_k denotes the first sample (vector) in the kth segment, the following method can be used for dividing the n ordered variables into K classes:

$$P(n, K) : \{i_1 = 1, i_1 + 1, \ldots, i_2 - 1\}, \\ \{i_2, i_2 + 1, \ldots, i_3 - 1\}, \ldots, \{i_K, i_K + 1, \ldots, n\}. \tag{20}$$

To use Fisher clustering, we need to define a loss function $e(P(n, K))$ to evaluate the quality of clustering. For a certain division method, the loss function $e(P(n, K))$ is defined as the sum of the squares of the deviations of all classes. Given n and K (the Fisher algorithm is applicable to cases with a known class number, K), the total sum of the squares of the deviations of all classes is fixed. Hence a smaller intraclass sum of squares and a larger interclass sum of squares give better classification results. In other words, clustering or segmentation aims to find a method which minimizes the loss function $e(P(n, K))$:

$$\text{Obj: } \min e(P(n, K)) = \min \sum_{k=1}^{K} A\left(i_k, i_{k+1} - 1\right). \tag{21}$$

To solve the above-described objective function, we use the following recursion:

$$\min e(P(n, K)) \\ = \min_{K \le i \le n} \{\min e(P(n - 1, K - 1)) + A(i, n)\}. \tag{22}$$

For example, when $K = 2$, $P^*(n, 2)$ is the optimal method among all possible division schemes that minimizes the loss function.

$$e\left(P^*(n, 2)\right) = \min e(P(n, 2)) \\ = \min_{2 \le i \le n} \{A(1, i - 1) + A(i, n)\}. \tag{23}$$

Using the method of induction, the recursion described in (23) can be derived which represents the optimal classification method of dividing n samples into K classes. It can be regarded as a combination of the optimal classification method of dividing $i - 1$ samples into $K - 1$ classes and the Kth segment which includes the remaining $n - i + 1$ samples.

There are two unique features of this algorithm. Firstly, it does not disturb the order of the dispatched buses. Hence the numbers of all buses that are classified into the same interval are adjacent. Secondly, the algorithm is not complex which takes less time to get the partition results and which can improve the computational efficiency.

(3) Final Division Based on Threshold Values of Two Indexes. By means of the above two steps, the dispatched buses are

Table 1: Time interval partition scheme and bus headway during the investigation period.

Interval number	Starting and ending times	Headway (min)	Bus quantity	Departure number
1	05:50–07:00	15	5	1–5
2	07:00–09:00	8	15	6–20
3	09:00–12:00	10	18	21–38
4	12:00–16:00	11	22	39–60
5	16:00–19:00	8	22	61–82
6	19:00–21:00	12	11	83–93

Table 2: Minimal lost function and starting codes of the last cluster in different partition methods.

	$K = 2$	$K = 3$	$K = 4$	$\cdots$	$K = 89$	$K = 90$	$K = 91$	$K = 92$
$n = 3$	0.0007 [2]			$\cdots$				
$n = 4$	0.05134 [3]	0.0437 [3]		$\cdots$				
$n = 5$	0.0094 [3]	0.0084 [3]	0.0008 [4]	$\cdots$				
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$n = 91$	9.121 [52]	8.068 [83]	7.482 [65]	$\cdots$	0.0026 [91]	0.0016 [91]		
$n = 92$	10.065 [52]	8.448 [83]	7.926 [65]	$\cdots$	0.0029 [92]	0.0026 [92]	0.0016 [92]	
$n = 93$	9.909 [48]	8.213 [82]	7.683 [72]	$\cdots$	0.0039 [93]	0.0029 [93]	0.0026 [93]	0.0016 [93]

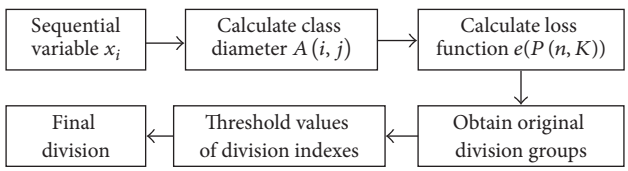


Figure 2: Flow chart of the division algorithm.

divided into K groups. However, it has not been determined whether the differences in the total dwell time and in the interstop travel time between two adjacent buses are smaller than the threshold values, which is thus evaluated in this step. In each group, if the differences in the two division indexes of two adjacent buses are larger than the threshold values, the two buses should be classified into different groups.

Figure 2 illustrates the overall process of the division algorithm.

4. Case Study

In this section, we apply the proposed time of day division method based on GPS data on the number 63 bus route in Harbin, China, as a case study.

4.1. Data Acquisition. Bus route 63 in Harbin has 21 stops in total. The line starts from Jiangong Community and goes all the way to Dajiang Community along the westbound direction. The operating distance of one direction is approximately 9.5 km. The bus enterprise has set the sampling interval of the GPS data at 30 seconds, which, however, cannot satisfy the requirement of this study. As a result, we carried out our own investigation of the bus line for two weeks (from Monday to Friday per week) during September 2013. In each bus, a GPS device was placed and connected to a laptop for real-time storage of the GPS data, which were later matched with a GIS map. Afterwards, the required travel time spent on road sections and intersections and the delay and dwell time at stops were extracted. During the investigation, the bus operating time was from 05:50 to 21:00. The operating period can be divided into 6 time intervals for each day. The specific starting and ending time points as well as the departure headways are listed in Table 1. In December 2014, we performed a second investigation for one week (Monday to Friday) and obtained the latest bus operating data. Compared with the first investigation, the total number of vehicles in Harbin had increased significantly. In addition, due to the winter snow on the road, vehicles moved more slowly and road congestion became even more serious. Acceleration and deceleration of the bus vehicles happened to be more frequent. As a result, the original schedule scheme was no longer suitable for the second investigation.

4.2. Division Results. Before the division of the operating time, the values of various parameters should be determined. δ and T_{ob} are constants which are set at 60 people per bus and 2.2 seconds per person, respectively. Given that $H = 8$ minutes, $H^a_{\max} = 2H$ and $H^a_{\min} = H$ and $\Delta D_{\max} = 112.2$ and $\Delta E_{\max} = 367.8$.

All the division indexes are normalized before used. Table 2 lists the minimal loss function based on the sequential clustering and the beginning label at the last time slot. The minimal loss function is calculated from the second column; that is, $K = 2$. The minimal loss functions of all the schemes of dividing the first i buses ($3 \le i \le 97$) into K classes are derived

to determine the optimal segmentation. Using $\min e(P(3,2))$ as an example, there are two division schemes which divide the first two buses into two classes, namely, ({1}, {2, 3}) and ({1, 2}, {3}).

$$\begin{aligned}\min e(P(3,2)) &= \min_{2\le j\le 3}\left(A_{1,j-1} + A_{j,3}\right)\\ &= \min\left[\left(A_{1,1} + A_{2,3}\right), \left(A_{1,2} + A_{3,3}\right)\right]\\ &= \min(0.0007, 0.0076) = 0.0007.\end{aligned} \quad (24)$$

The optimal segmentation is ({1}, {2, 3}), and the beginning label of the last class (i.e., 2) is recorded. As shown in the second row and the second column in Table 2, [2] on the right of 0.0007 represents the division of the first three buses into 2 classes where the beginning label of the second class is 2 and the corresponding minimal loss function is 0.0007. Moreover, the division indexes (the average dwell time at stops and the average travel time among stops) are different for different buses in a class which should be taken into account in the classification. The buses whose division indexes are smaller than the thresholds are grouped into the same class. For example, there are 92 division schemes when dividing 93 samples into 2 classes. Before the calculation of the loss function, we should first evaluate whether the thresholds $\Delta D_{\max}$ and $\Delta E_{\max}$ are satisfied and delete those division schemes that do not satisfy the requirement. Only after that can the loss functions of the remaining division schemes be calculated so as to determine the optimal division.

As shown in Table 2, the sequential clustering algorithm cannot determine the class number K but can only determine the optimal class number according to the variation in the minimal error function. It can be observed that, in the last row of Table 2 ($n = 93$), the minimal error function of 93 sample data decreases gradually with an increasing K. A greater K suggests a finer division and, accordingly, fewer buses are included in a class in which the difference is smaller. However, a bus enterprise does not necessarily want to increase the number of the operating time slots, since doing so will not only increase the frequency to update the dispatching schemes but also require more transition schemes between different dispatching schemes. Frequent transitions may reduce the operating efficiency of bus transit [7]. In studies [7, 8], the value of K was determined by the manager. For this study, with reference to the previous research, we consulted the administration department of the bus enterprise and finally set the value of K at 8; that is, the operating time of the number 63 bus is divided into 8 time intervals as shown in Table 3.

Table 3: Final partition results of operation time intervals for bus route 63.

Interval number	Starting and ending times	Interval number	Starting and ending times
1	5:50–7:05	5	12:49–15:01
2	7:05–8:09	6	15:01–16:39
3	8:09–11:05	7	16:39–19:15
4	11:05–12:49	8	19:15–21:00

5. Conclusion

This study first recovers the bus travel time on the road based on the historical GPS data and then divides the bus operating time using a sequential clustering algorithm. The main conclusions are as follows:

(1) The bus travel time data collected from the bus-mounted GPS cannot truly reflect the real operating state of the bus vehicle. Drivers' behavior should be taken into account for data correction.

(2) For the division of the operating time, the division algorithm is more sensitive to the threshold value of the dwell time at stops. A smaller threshold value may easily make the division finer.

(3) A sequential clustering method can ensure that the order of the adjacent buses is not disrupted in order to achieve a favorable division of the bus operating time.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Project no. 51578199). This work was performed at the Key Laboratory of Advanced Materials & Intelligent. Control Technology on Transportation Safety, Ministry of Communications, China.

References

[1] Z. Liu, Y. Yan, X. Qu, and Y. Zhang, "Bus stop-skipping scheme with random travel time," *Transportation Research C: Emerging Technologies*, vol. 35, pp. 46–56, 2013.

[2] X. Ma and Y. Wang, "Development of a data-driven platform for transit performance measures using smart card and GPS data," *Journal of Transportation Engineering*, vol. 140, no. 12, Article ID 04014063, 2014.

[3] X. Ma, Y. J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders' travel patterns," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 1–12, 2013.

[4] X. Ma, C. Liu, H. Wen, Y. Wang, and Y. Wu, "Understanding commuting patterns using transit smart card data," *Journal of Transport Geography*, vol. 58, pp. 135–145, 2017.

[5] Y. Li, X. Wang, S. Sun, X. Ma, and G. Lu, "Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 306–328, 2017.

[6] X. Gong, X. Guo, X. Dou, and L. Lu, "Bus travel time deviation analysis using automatic vehicle location data and structural equation modeling," *Mathematical Problems in Engineering*, vol. 2015, Article ID 410234, 9 pages, 2015.

[7] M. Chen, X. Liu, and J. Xia, "Dynamic prediction method with schedule recovery impact for bus arrival time," *Transportation Research Record*, no. 1923, pp. 208–217, 2005.

[8] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[9] C. Ding, D. Wang, C. Liu, Y. Zhang, and J. Yang, "Exploring the influence of built environment on travel mode choice considering the mediating effects of car ownership and travel distance," *Transportation Research Part A: Policy and Practice*, vol. 100, pp. 65–80, 2017.

[10] C. Ding, X. Wu, G. Yu, and Y. Wang, "A gradient boosting logit model to investigate driver's stop-or-run behavior at signalized intersections using high-resolution traffic data," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 225–238, 2016.

[11] C. Ding, S. Mishra, G. Lu, J. Yang, and C. Liu, "Influences of built environment characteristics and individual factors on commuting distance: a multilevel mixture hazard modeling approach," *Transportation Research Part D: Transport and Environment*, vol. 51, pp. 314–325, 2017.

[12] J. Patnaik, S. Chien, and A. Bladikas, "Using data mining techniques on apc data to develop effective bus scheduling plans," *Journal of Systemics, Cybernetics and Informatics*, vol. 4, no. 1, pp. 86–90, 2006.

[13] V. Guihaire and J. K. Hao, "Transit network design and scheduling: a global review," *Transportation Research A: Policy and Practice*, vol. 42, no. 10, pp. 1251–1273, 2008.

[14] D. Z. Yue, *Optimal timetable research based on passenger arrival rate*, School of Control Science and Engineering, Shandong University, Jinan, China, 2014 (Chinese).

[15] Y. D Shen, T. H Zhang, and J. Xu, "Homogeneous bus running time bands analysis based on K-means algorithms," *Journal of Transportation Systems Engineering and Information Technology*, vol. 14, no. 2, pp. 87–93, 2014 (Chinese).

[16] Y. Bie, X. Gong, and Z. Liu, "Time of day intervals partition for bus schedule using GPS data," *Transportation Research Part C*, vol. 60, pp. 443–456, 2015.

[17] Transportation Research Board, "Highway Capacity Manual," 2000.

[18] R. Guo and Y. Zhang, "Identifying time-of-day breakpoints based on nonintrusive data collection platforms," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 18, no. 2, pp. 164–174, 2014.

[19] P. Yu, H. Chi, and X. C Tan, "A study on flight altitude discrepancy base on the fisher ordinal samples cluster method," *Chinese Journal of Management Science*, vol. 18, no. 5, pp. 130–136, 2010.

Hindawi
Journal of Sensors
Volume 2017, Article ID 7248189, 11 pages
https://doi.org/10.1155/2017/7248189

Research Article

Online Traffic Condition Evaluation Method for Connected Vehicles Based on Multisource Data Fusion

Pang-wei Wang,[1] Hong-bin Yu,[1] Lin Xiao,[2] and Li Wang[1]

[1]*Beijing Key Lab of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing 100144, China*
[2]*National Research Council of the National Academies of Sciences, Engineering, and Medicine, 500 Fifth Street NW, Washington, DC 20001, USA*

Correspondence should be addressed to Pang-wei Wang; wpw@ncut.edu.cn

Received 11 April 2017; Revised 4 June 2017; Accepted 2 July 2017; Published 3 August 2017

Academic Editor: Hai-Feng Ji

With the development of connected vehicle (CV) and Vehicle to X (V2X) communication, more traffic data is being collected from the road network. In order to predict future traffic condition from connected vehicles' data in real-time, we present an online traffic condition evaluation model utilizing V2X communication. This model employs the Analytic Hierarchy Process (AHP) and the multilevel fuzzy set theory to fuse multiple sources of information for prediction. First, the contemporary vehicle data from the On Board Diagnostic (OBD) is fused with the static road data in the Road Side Unit (RSU). Then, the real-time traffic evaluation scores are calculated using the variable membership model. The real data collected by OBU in field test demonstrates the feasibility of the evaluation model. Compared with traditional evaluation systems, the proposed model can handle more types of data but demands less data transfer.

1. Introduction

Nowadays, traffic congestion is a serious issue due to the growing number of vehicles moving on the urban road networks. Connected Vehicle (CV) technology enhances the ability of traffic information collection and management through Vehicle to X communication (including Vehicle-To-Infrastructure (V2I) and Vehicle-To-Vehicle (V2V) communication), which presents one of the best ways to mitigate urban traffic congestion, improve traffic safety, and reduce fuel consumption [1].

With the development of connected vehicles, multiple sensors and communication modules tend to become standard equipment in vehicles. Through these sensors and communication modules, the required traffic information can be collected and distributed efficiently. Meanwhile, diversified traffic data sources and effective analysis methods provide a more reliable decision-making basis for traffic managers [2]. Furthermore, advanced data fusion technology can be used to deal with massive multisource traffic data to provide more accurate estimation of urban road conditions and improve the evaluation and prediction methods for urban traffic system [3].

In the urban traffic system, there are various traffic data acquisition methods, such as detector, video, and radar. With the application of V2X communication, more traffic data can be collected from connected vehicles, infrastructure, and other traffic sensors. The data can then be fed to the traffic condition evaluation, prediction, and decision-making system. If the traffic management department can leverage real-time traffic information from V2X communication to induce traffic flow and reduce unnecessary travel time, the operational efficiency of the transport network can be improved.

Huang proposed a data fusion method to optimize urban traffic flow based on neural network and fuzzy reasoning, which collected the traffic data from varied detectors on the urban road [4]. Quek et al. introduced a special class of fuzzy neural network known as the pseudo outer-product fuzzy neural network using the truth-value-restriction method (POPFNN-TVR) for short-term traffic flow prediction. The method combined the complementary capabilities of both neural networks and fuzzy logic; thus it constituted a more promising technique for modeling traffic flow [5]. Zhao et al. analyzed the characteristics of multisource data fusion and support vector machine (SVM). Following to the principle

of SVM, they collected multisource traffic flow data from Hanshin Highway [6]. Castillo et al. reviewed the roles of mathematical tools and methods in traffic flow observability, estimation, and prediction problems. The high number of possible combinations of these elements justifies the existence of a wide collection of methods for analyzing static and dynamic situations [7]. Thomas and Dia presented a neural network algorithm based on traffic data fusion and tested it with simulated data. It analyzed various influence factors on data collection, such as positions of detectors, numbers of floating cars, length of the urban road, and severities of traffic accidents. Several classical algorithms were applied in traffic information fusion, including Kalman filter, artificial neural network, exponential smoothing, and recursive estimation algorithm [8]. Yang et al. proposed a novel fusion model which can be used to identify traffic status and analyze traffic conditions, accidents, scope of coverage, and forecast of future traffic flow [9]. Ren et al. processed observation traffic data for the traffic volume of urban road using fuzzy fusion algorithm. The test results showed that this method can acquire more complete and reliable traffic data. To forecast long periods of traffic flow conditions [10] Stutz and Runkler used fuzzy clustering to classify and analyze the traffic jams on a German freeway [11]. Jiang et al. used fuzzy clustering to identify road traffic conditions; the feasibility of this model was proved by the simulation results [12]. Rizzi et al. proposed an application of a highly efficient classification system based on low complexity real-time Internet traffic flows, by considering traffic data sets collected in different epochs and places [13]. Guo et al. proved the urban road traffic conditions can be analyzed with traffic data of coil detector by improved fuzzy clustering method [14]. He et al. improved a fusion method with new data collected from mobile phone and microwave sensors, providing enough data for traffic analysis [15].

There are several achievements while applying V2X communication in ITS. Backfrieder et al. predicted future congestion based on the Bottleneck prediction method and V2X communication. It demonstrated promised performance through dynamic microscopic traffic simulations both in a real-world scenario and in an artificial road network scenario [16]. Schünemann proposed a flexible simulation tool which simulated real-time traffic flow by V2X. This tool can also be used to simulate various scenarios of future intelligent transportation systems [17]. Otsuki and Miwa designed an efficient content-delivery control algorithm using real-time traffic data generated from traffic situation. The algorithm utilized the route prediction information in order to share traffic data during the vehicles by V2X communication efficiently [18]. Wedel et al. introduced a novel algorithm that can be used for connected vehicle with navigation system to calculate routes circumnavigating congested roads [19].

We can draw a conclusion from the literatures above that current research on traffic evaluation mainly focuses on how to process traditional data from detectors. Researchers have conducted a full study of how to improve the accuracy and reliability of information fusion. However, due to the limitation of data types and inevitable errors from traffic detectors, the advantages of novel traffic evaluation method are not prominent. To fill this research gap, a real-time traffic evaluation method based on data fusion in V2X scenario is presented in this paper. The vehicle OBD data is collected and processed by the RSU installed at the intersection. OBD data will be later fused with the static road data in the method. Details are discussed in the following section.

2. Description of the Traffic Data Fusion Scenario

Unlike traditional floating car system which sends real-time data directly to the server via mobile network and provides few data for traffic control, this paper uses the OBD data as the source of vehicle dynamic data and keeps the data within RSU at intersection. There are two types of data stored in the RSU: static road parameters such as road grade, number of lanes, road length, and real-time dynamic data generated by the connected vehicles.

The scenario of connected vehicles with V2X communication in this paper is described as shown in Figure 1. With the RSU installed at the intersection, all kinds of vehicle data generated by the vehicle passing through the road section (Point A → Point D) are collected. By fusing the parameters of the road section, the system evaluates the traffic conditions at each collection interval.

The structure of CV system is shown as Figure 2. The On Board Unit (OBU) installed on the vehicle is an embedded acquisition system that receives the vehicle data through the Controller Area Network (CAN) protocol from the Electronic Control Unit (ECU). The RSU actively sends the handshake information to establish communication with the OBU that supports V2X communication and determines whether the vehicle enters or leaves the intersection by comparing the location information of the intersection and the GPS data of the vehicle.

The flow chart of traffic data fusion is shown in Figure 3. When the vehicle leaves the last intersection (Point A), it starts to record the running data of the current road section and sends the data generated on the road section to the RSU when leaving the next intersection (Point D). When the vehicle enters the communication range (Point B) of the RSU, the two parties will establish a stable V2X communication. The OBU sends its own vehicle basic information and continuously sends the positioning information before the RSU requests the OBD data. When the vehicle enters the intersection 2 (Point C), since the communication history of the RSU1 is recorded in the OBU, the current vehicle data belonging to entrance lane (intersection 1 → intersection 2) can be determined in the RSU2. According to this principle, the data of vehicles on multiple entrance lanes can be processed simultaneously in the RSU. After the data sent by the OBU is verified as valid, data fusion and evaluation will be done in the RSU according to the model in Section 3.

The OBD interface is chosen since it provides not only the vehicle sensor information, but also the vehicle internal control information and fault information. This distinguishes the proposed model from most traditional floating car data collection systems which obtain the vehicle data mainly from GPS module. Since the OBD interface integrates external

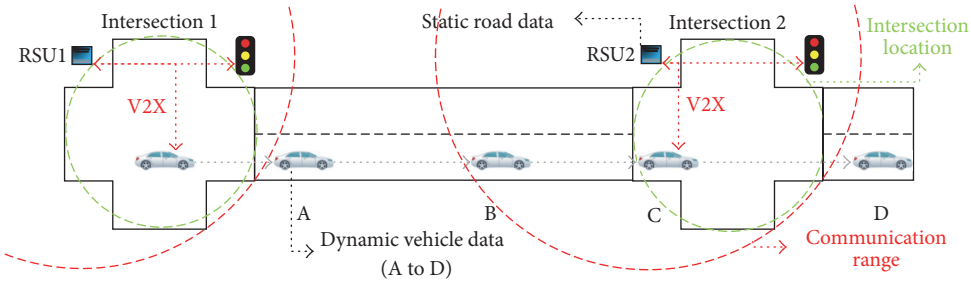


Figure 1: The scenario of CV system on intersections.

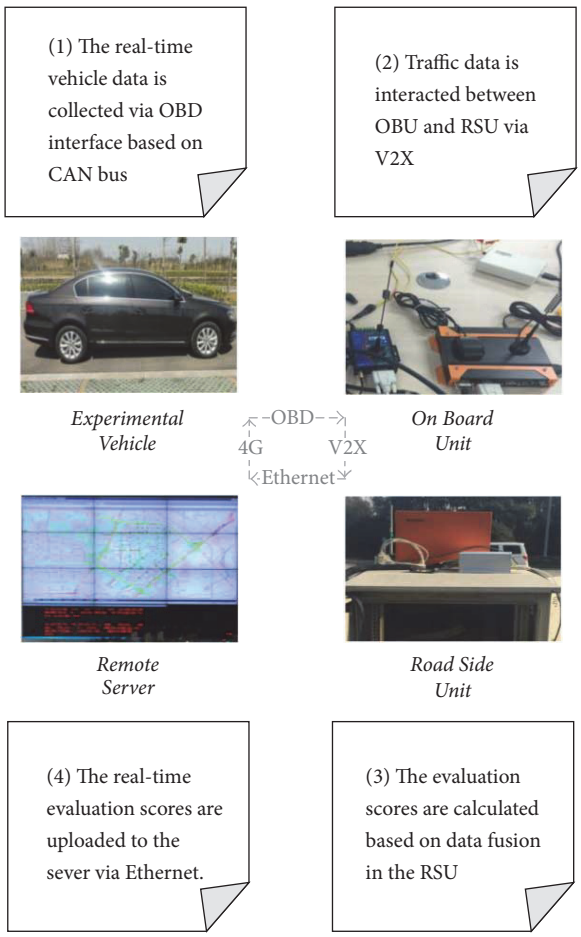


Figure 2: The evaluation process for CV system.

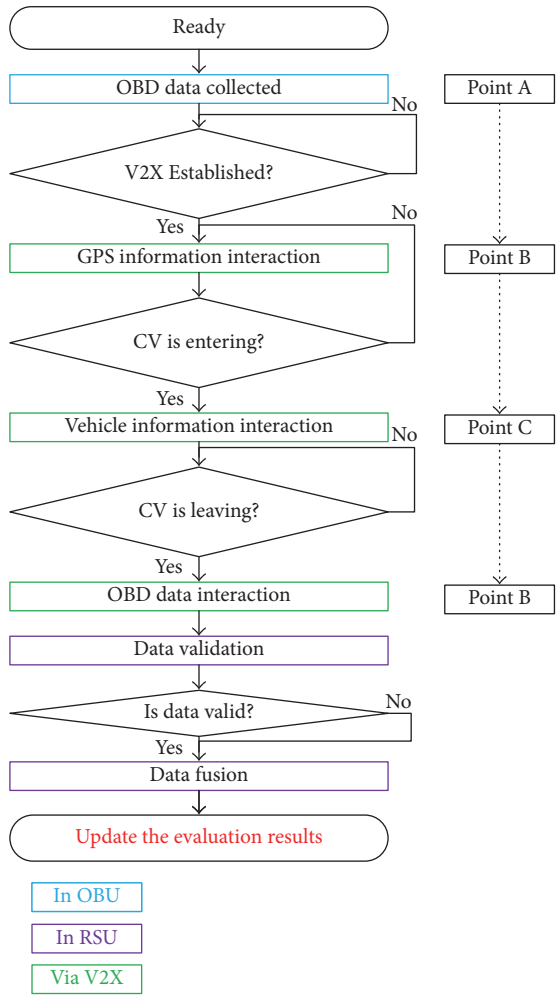


Figure 3: The flow chart of online traffic evaluation based on data fusion with V2X.

detectors, it greatly enhances the evaluation system's versatility.

With the V2X communication and vehicle OBD data, the scenarios described in this paper have the following advantages over traditional floating car systems:

(1) More traffic information is shared by V2X. The traffic flow data on the macroscale is fused and calculated at the RSU, whereas the running information of each vehicle is collected in V2X communication on the microscale.

Meanwhile, the information for driving guidance or alarm can be sent to vehicles as well. In a word, if the multisource data fusion with V2X communication is applied in urban traffic system, it can greatly improve the driving safety and traffic capacity. The accuracy error caused by the interference of the sensor can also be avoided.

(2) Traffic data from OBD interface is more accurate and computationally friendly. The method reduces the load and the computation amount of the data on the network and can effectively avoid errors caused by the collection system. Since the method collects corresponding dynamic data judged

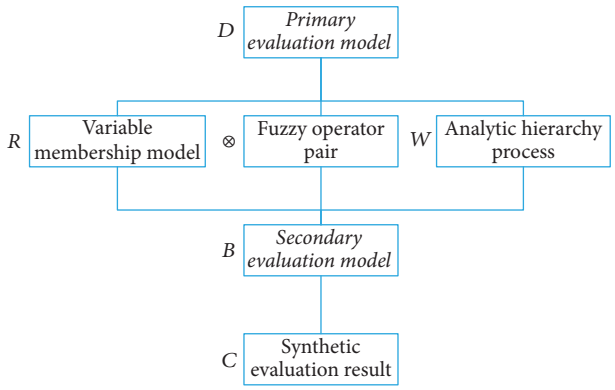


Figure 4: The structure of evaluation model.

based on the OBD data directly, it is not necessary to always track and calculate the vehicle GPS data. The process of data calculation and transmission is simplified by the method, and errors caused by sensors can also be avoided.

3. Evaluation Model Based on Real-Time Traffic Information Fusion

The evaluation model established in this paper is based on the multilevel fuzzy synthetic evaluation model. The basic idea is to establish the fuzzy judgment matrix by using the transformation principle to describe the data boundary of the factors in fuzzy set. Through the multilayer numerical calculation based on the evaluation criteria and weights, we will determine the results of the evaluation object [20, 21]. The structure of synthetic evaluation model is shown as Figure 4.

3.1. The Primary Evaluation Model. Define Q as a finite set, and Q: $Q = \{q_1, q_2, \ldots, q_n\}$, where q_i $(i = 1, 2, \ldots, n)$ is an object to be evaluated.

The objects to be evaluated (elements of "Q") in this paper are the road states at different times (e.g., 10 a.m. or 5 p.m.).

Define a finite set, that is, P: $P = \{p_1, p_2, \ldots, p_m\}$, and the element p_i $(i = 1, 2, \ldots, m)$ in P represents a different evaluation indices.

While evaluating a road's real-time status, there are many evaluation indexes that can be used. Based on the principle of measurability, the average travel time (ATT), average number of stops (ANS), and average stopped time (AST) are selected as the final evaluation indices in this model, as in (1). All these three indices can be calculated from data collected from OBD and V2X communication.

$$P = \{p_1, p_2, p_3\} = \{\text{ATT}, \text{ANS}, \text{AST}\}. \tag{1}$$

Travel time is defined as the difference between the data exchange time at the current intersection and that at the adjacent downstream intersection. While traveling to the next intersection, the OBD can record the number of stops and total stopped time. In addition, all detected data will be verified by communication integrity and data validation to ensure the validity of the data. Then, the above selected three real-time evaluation indices can be calculated by averaging the valid data.

Define u_j as the membership function of the j^{th} evaluation index $p_j \in P$, that is, $u_j = u(p_j)$, $u_j \in [0, 1]$; then U is a finite fuzzy subset; that is, $U = \{u_1, u_2, \ldots, u_m\}$.

Since the membership function is applicable to all evaluation objects, an evaluation matrix can be obtained as a fuzzy relation, that is, R: $Q \times U \to [0, 1]$, which is defined as

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{21} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}_{n \times m}, \tag{2}$$

where $r_{ij} = R(q_i, u_j) \in [0, 1]$ is the membership degree of the i^{th} object to be evaluated on the j^{th} evaluation index.

Define $S = (Q, U, R)$ as the primary evaluation space and give a fuzzy vector W:

$$W' = \left(w_1', w_2', \ldots, w_m'\right)^T. \tag{3}$$

In (3), element w_j of W represents the weight of each evaluation index with respect to the primary evaluation model:

$$D = R \otimes W. \tag{4}$$

3.2. The Variable Membership Model. Assuming that the evaluation result is a finite set $V = \{v_1, v_2, v_3, v_4\} = \{\text{excellent}, \text{good}, \text{medium}, \text{bad}\}$, each element in the set corresponds to a distribution interval of the membership function, which is shown in

$$\begin{aligned} u_j &\in [0.25(i-1), 0.25i], \\ v &= v_i \quad (i = 1, 2, 3, 4). \end{aligned} \tag{5}$$

Considering the negative correlation between the evaluation indices and evaluation results, this paper selects the membership function of the Cauchy type, as shown in

$$u_i = \mu\left(p_j\right) = \begin{cases} 1, & p_j \le c_j \\ \dfrac{1}{1 + \left[a_j\left(p_j - c_j\right)\right]^{b_j}}, & p_j > c_j \end{cases} \tag{6}$$

$$\forall a_j, b_j, c_j > 0.$$

It can be seen that, in the coordinate system $p_j - u_j$, each membership function distribution interval $0.25i$ must have a corresponding critical value p_{ij}. According to the set of values under the same membership function, the coefficients of the membership function, including a_j, b_j, and c_j, can be solved by regression analysis. Taking into account the actual traffic scenario, p_{ij} is a time-varying value changing with static traffic parameters. The dynamic adjustment strategy for p_{ij} is shown in

$$p_{ij} = \frac{(1 - \xi)\,\widehat{p}_{ij} L\omega_{mj}}{\widehat{L}\alpha_{nj}} + \xi\widehat{p}_{ij}, \quad (i = 1, 2, 3, 4), \tag{7}$$

Table 1: Parameters of the membership function.

	$\widehat{p}_{ij}$	ξ	ω_{mj}	α_{nj}	p_{ij}	a_j	b_j	c_j
u_1	(184, 138, 97, 71)	0.3	1	1.05	(219, 165, 117, 86)	0.0007	1.66	79.8
u_2	(3.8, 2.6, 1.6, 0.9)	0	0.95	1.10	(5.5, 3.8, 2.3, 1.2)	0.164	1.82	1.2
u_3	(75, 50, 35, 26)	0.85	0.95	1.03	(80, 54, 37, 28)	0.165	1.29	27.3

where $\widehat{p}_{ij}$ is a typical critical value for the length of the unidirectional road at the specified road grade and is calibrated by a large number of tests, ξ is a coefficient that represents the part of evaluation indices which is generated by the control of the signal, which depends mainly on the green signal ratio and the number of phases, $\widehat{L}$ is the standard length (500 m) for the urban road, α_{nj} is the influence coefficient of n lanes on the j^{th} evaluation index, and ω_{mj} is the influence coefficient of m branch road on the j^{th} evaluation index.

To obtain the critical value p_{ij}, we take the relevant static data (in Table 4) and typical values into (7). By the nonlinear regression analysis function "nlinfit" in MATLAB, we obtain the parameters of the Cauchy-type function in (6). Then the coefficients are fitted as a curve, including a_j, b_j, and c_j. The results are shown in Table 1 and Figure 5.

For the value of the membership function u_j, it is necessary to convert it into the membership degree with the corresponding indices: based on the trapezoidal membership, we define the interval as $0.25i$ of the evaluation result set V, which is the intermediate membership degree ($r_{ij} = 0.5$) of the two evaluation indices. The floating range is from $0.25i - 0.1$ to $0.25i + 0.1$. The final membership relationship is shown in Figure 6.

3.3. The Analytic Hierarchy Process. The Analytic Hierarchy Process (AHP) is used for organizing and analyzing complex decisions based on mathematics and psychology. Rather than prescribing a "correct" decision, the AHP helps decision makers find one solution that best suits their goal and their understanding of the problem. It provides a comprehensive and rational framework for structuring a decision problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions.

In the proposed model, there are many evaluation indexes that can be considered. However, the weights of each index are not predefined. For example, some researchers may regard that the average speed is the most important index to evaluate a road state, but others may regard the stops as more important than the average speed. Both of the two viewpoints are subjective assumptions. So the AHP method is to determine the weight of these evaluation indexes scientifically. The steps of AHP are shown as follows.

Step 1. According to the relevant research and practical experience, comparing the importance of the three evaluation indices, the judgment matrix table can be acquired as shown in Table 2.

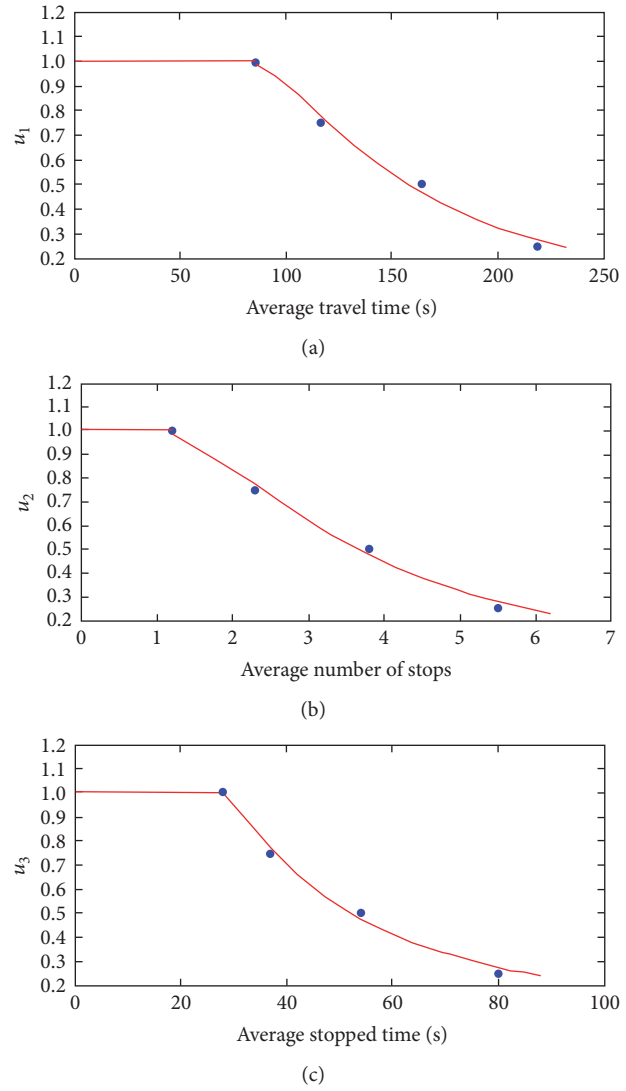


Figure 5: Curve fitting of the membership function.

Step 2. The data in Table 2 is brought into the following equation to obtain the judgment matrix A:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{21} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}_{n \times n}. \quad (8)$$

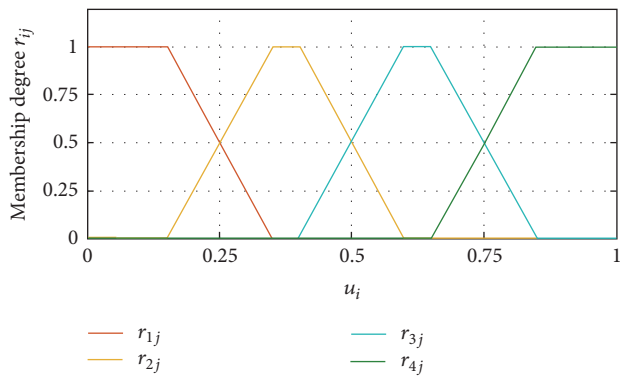


Figure 6: Membership relationship of the evaluation model.

Table 2: Parameters of the judgment matrix.

	a_1	a_2	a_3
a_1	1	3	2
a_2	$\frac{1}{3}$	1	$\frac{2}{3}$
a_3	$\frac{1}{2}$	$\frac{3}{2}$	1

Step 3. The columns of judgment matrix A are normalized as

$$\overline{a}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} \quad (i, j = 1, 2, \ldots, n). \tag{9}$$

To apply the AHP method, each element in (9) is recalculated as a proportion to the sum of its own column. By normalizing, the value of each element can be transformed as a percentage, which is the value that needs to be calculated in (10)

Step 4. The sum of the rows of the judgment matrix A is calculated as

$$\overline{w}_i = \sum_{j=1}^{n} \overline{a}_{ij} \quad (i = 1, 2, \ldots, n). \tag{10}$$

Step 5. Normalize $\overline{w}_i$ to get w_i; we can find the largest eigenvalue $\lambda_{\max}$ and eigenvector, according to $Aw = \lambda_{\max} w$:

$$w_i = \frac{\overline{w}_i}{\sum_{i=1}^{n} \overline{w}_i} \quad (i = 1, 2, \ldots, n). \tag{11}$$

Finally, the weight matrix can be calculated: $W = (0.52, 0.22, 0.26)^T$.

When the weight set is calculated by AHP method, we will make a consistency check to ensure that the results are reasonable. For example, to avoid logical errors, if the result is index A being more important than index B, and index B is more important than index C, but index C is more important than index A, then the results are unreasonable.

Step 6. Calculate the consistence index C.I; we can find the corresponding mean random consistency index R.I, where n represents the order of the judgment matrix A:

$$\text{C.I} = \frac{\lambda_{\max} - n}{n - 1}. \tag{12}$$

Step 7. The consistency ratio C.R is calculated as (13), where R.I represents a constant value, determined by n (e.g., $n = 3$, R.I = 0.52):

$$\text{C.R} = \frac{\text{C.I}}{\text{R.I}}. \tag{13}$$

Through the calculation in (13), the results of consistency check (C.R < 0.1) are accepted.

3.4. The Fuzzy Operator Pair and Secondary Evaluation Model. The symbol $\otimes$ in (4) represents a fuzzy operator pair. If more operator pairs are introduced at the same time, a new fuzzy subset can be obtained for each evaluation object:

$$U' = \{D_1, D_2, \ldots, D_p\}, \quad U' \in [0, 1]. \tag{14}$$

In (14), p represents the number of fuzzy operator pairs.

The fuzzy operator pair will determine the meaning of the fuzzy vector to a larger extent. Besides, the secondary evaluation space composed of multiple operator pairs will help to measure the influence of the evaluation indices (P) on the object to be evaluated (Q) from various aspects. In this paper, we select three operator pairs: $(\wedge, \vee)$, $(\bullet, \vee)$, and $(\wedge, \oplus)$, where "$\vee$" represents Max, "$\wedge$" represents Min, "$\bullet$" represents multiplication, and "$\oplus$" represents addition. These three operator pairs focus on the contribution of individual or multiple evaluation indexes, and $\sum_{j=1}^{p} w_j \neq 1$.

A new fuzzy relation can be obtained Combining Q and U', that is, R': $Q \times U' \to [0, 1]$:

$$R' = \begin{bmatrix} R'_1 \\ R'_2 \\ \vdots \\ R'_n \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1p} \\ d_{21} & d_{21} & \cdots & d_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{np} \end{bmatrix}_{n \times p}. \tag{15}$$

In (15), d_{ij} represents the primary evaluation value of the i^{th} object calculated from (4) when the j^{th} operator pair is used.

Thus, the secondary evaluation space $S' = (Q, U', R')$ is obtained. Each element $w_j \in W$ is given a fuzzy vector W' in the secondary evaluation space:

$$W' = \left(w'_1, w'_2, \ldots, w'_p\right). \tag{16}$$

In (16), the element w'_j of W' represents the weight of the j^{th} fuzzy operator pair for the secondary evaluation space, and $\sum_{j=1}^{p} w'_j \neq 1$, $w'_j \in [0, 1]$.

The secondary fuzzy vector matrix is obtained according to the AHP, that is, $W' = (0.17, 0.28, 0.55)$. Then the secondary evaluation model is obtained:

$$B = W' R'^T = \{b_1, b_2, \ldots, b_m\}. \tag{17}$$

Table 3: Data transmission performance with respect to communication distance.

Number	Communication distance (m)	Packet loss rate (%)	Time delay (ms)	Transmission rate (kbps)
1	20	1.6	<200	102.2
2	50	1.9	<200	99.5
3	100	3.3	<400	96.5
4	150	5.6	<400	91.9
5	200	7.2	<800	87.3
6	300	9.9	<800	80.5
7	400	21.7	>1200	71.5
8	500	46.6	>1200	50.4

Note. The test data consists of three packets, totaling 51 KB.

In (17), b_i represents the evaluation index of the j^{th} evaluation object, where $b_i = \sum_{j=1}^{p} w_j' d_{ij}$.

We can find that this mathematical model is a two-level fuzzy evaluation model. The final result is a set of evaluation results.

3.5. The Synthetic Evaluation Results. Finally, the evaluation results for connected vehicles are obtained based on the synthetic evaluation method. To ensure the feasibility, the original evaluation results B are integrated according to the weighted average principle. Firstly, the elements of B are normalized to obtain $\widetilde{b}_i$; then the final evaluation score C is calculated in

$$C = 100 - \sum_{i=1}^{m} \frac{100}{3} (i-1)\, \widetilde{b}_i. \tag{18}$$

We can find in (18), where $C \in [0, 100]$, that the road condition is proportional to the value of C.

4. Experiment and Analysis

4.1. Experimental Method. A data acquisition system with reference to a real V2X communication scenario is established in this article. The system has an embedded data acquisition device based on the chip of *Freescale i.MX6 Q*, which is installed in vehicles (OBU) and intersections (RSU). The device has a rich interface as shown in Figure 7, and the data of the Experimental vehicle can be passed into the device via the CAN interface. The OBU acquires OBD and GPS data according to the implementation procedure and communicates with RSU through the high power Zig-Bee module. The actual environment proved that the communication established by the Zig-Bee module was stable enough to simulate the real V2X communication. The test results are shown in Table 3.

A segment of the Pingguoyuan South Road in Shijingshan District of Beijing is picked as experimental section in this paper. The static data of this road is shown in Table 4. In the experiment, two experimental vehicles equipped with OBU device are traveling continuously on both sides of the road. The calculated evaluation indices are transmitted via wireless communication to the RSU device installed at the intersection. The experiment is carried out from 6:00 am to 8:00 pm on March 3, 2017. In order to improve the accuracy of the evaluation as much as possible, the experiment ensured that there are at least 10/16 sets of data in each section of the road during the peak/valley hours of traffic flow. At the same time, the flow data of the day is obtained through artificial observation and converted to Passenger Car Unit (PCU). Restricted by the experimental condition, the final evaluation result is calculated by deriving the vehicle data stored in the RSU into the personal computer. However, this does not affect the evaluation result and preserves more details of the original data to ensure the reliability of the results.

Table 4: Parameters of the road section.

Parameter name	Data
Road grade	Secondary trunk road
Road type	Two-way two-lane
Road length	698 m
The number of branch road	1/1
Traffic signal control	Yes

4.2. Experimental Data Analysis. A total of 407 data packets were collected by the experimental device, of which 377 were valid. That is to say, 92.6% of all data packets received by the device is valid. All valid data are processed at intervals of one hour as shown in Figures 8, 9, and 10 where (a) represents a section from west to east, (b) represents a section from east to west, the curve represents the three evaluation indices, and the gray shaded area represents the distribution of the data. As seen from the figures, the three evaluation indices share similar trend with respect to time. This proves the reliability of the data from one aspect.

The data sheets of the three evaluation indices are brought into a MATLAB program where the final synthetic evaluation result C is calculated; see the lower part of Figure 11. The two colors of the data represent the two directions of the road. The upper part of Figure 10 is the flow data for the day. Comparing the upper and lower parts of the figure, we can find that the evaluation score has a negative correlation with the original traffic flow data, which is consistent with the actual traffic situation. As shown in the figure, the peak flow in the day also

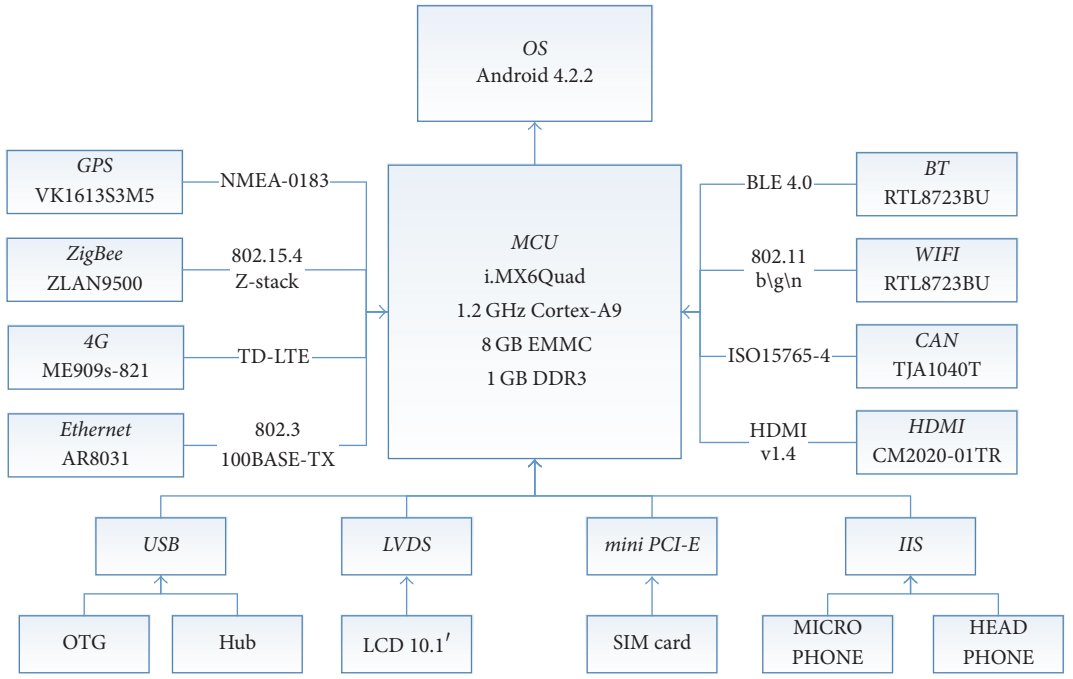


Figure 7: The hardware structure of the terminal for CV system.

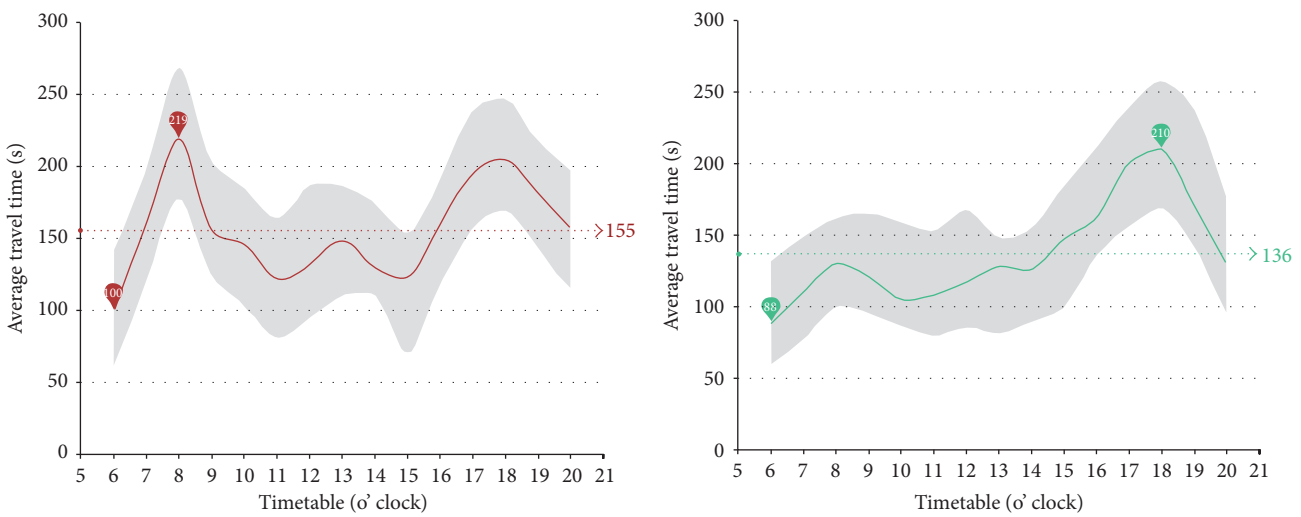
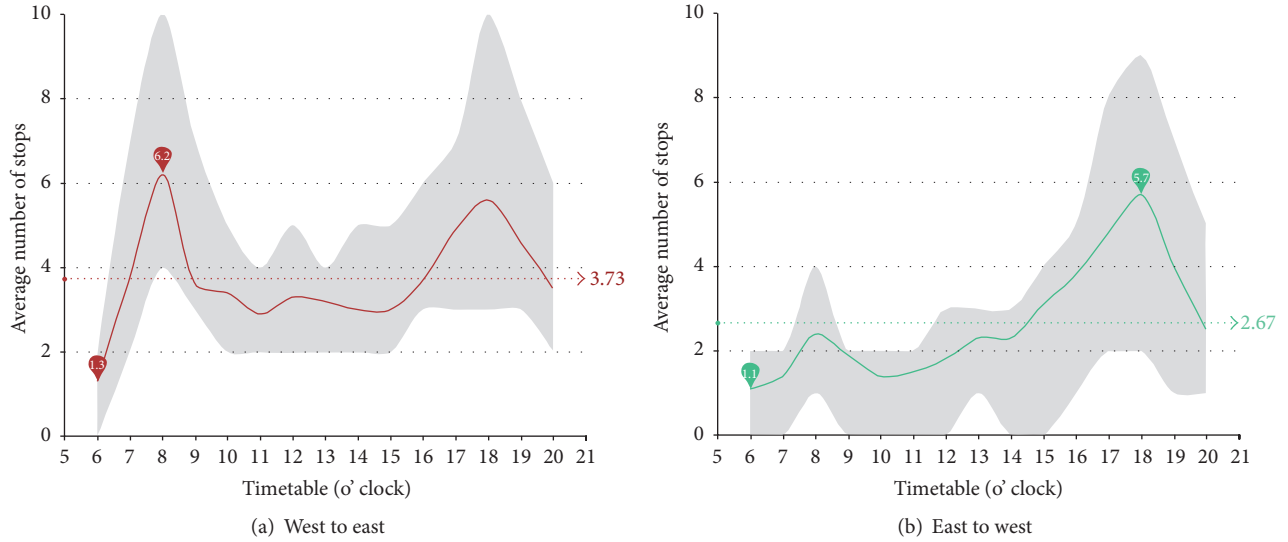


Figure 8: Results of average travel time.

corresponds to the lowest evaluation result. The difference between the peak hours and the valley hours is evident in the evaluation results.

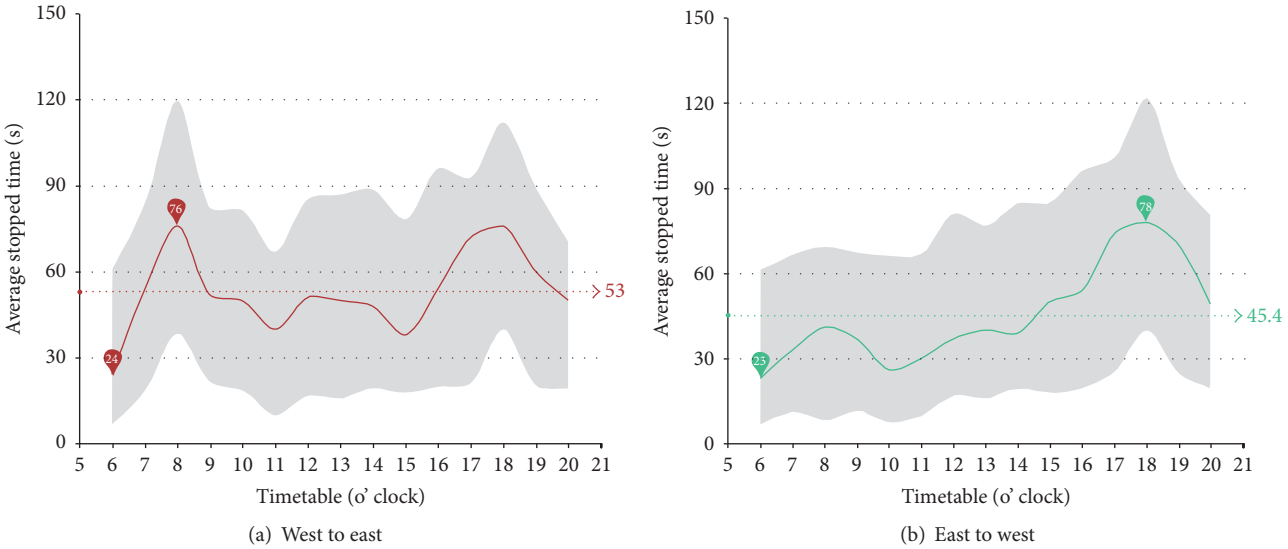
Since the two directions of the road have similar static road parameters, in order to describe the relationship correctly between the evaluation score and the traffic flow, all the evaluation data are arranged in ascending order of the traffic flow as shown in Figure 12. The red line in the figure represents the actual calculated evaluation score, and the green line represents the reference delay value of the grade road at different traffic flows.

It can be seen from the above analysis that the evaluation results of the multilevel fuzzy synthetic method used in this paper are in good agreement with the actual situation. If the device's communication coverage is further improved, the time interval of the traffic evaluation can be reduced to 5 to 15 minutes, which is sufficient to meet the requirements of the road evaluation system under the complex road network. It



(a) West to east (b) East to west

Figure 9: Results of average stops.



(a) West to east (b) East to west

Figure 10: Results of average stopped time.

provides a feasible solution for the traffic evaluation method under the V2X scenario.

5. Conclusion

By fusing the real-time connected vehicle data with static road segment information, an online traffic condition estimation model is proposed and tested in this paper. The OBD data and the traffic evaluation method are applied in the system of connected vehicles on urban road with V2X. Based on the traditional fuzzy synthetic model, the multi-operator synthetic fuzzy and variable membership model is introduced. We determined the scientific model parameters through AHP. In the field experiment, the evaluation results produced by the proposed model are the same with the actual situation of the road, which demonstrates the fidelity and effectiveness of the method. When the OBD data in the vehicle is collected by the V2X, the proposed model has greater advantage over the traditional floating vehicle data evaluation method. In addition, since connected vehicles are supplied with more detailed traffic information, the traffic capacity and safety can be greatly improved in the foreseeable future.

Disclosure

The views are those of the authors alone.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

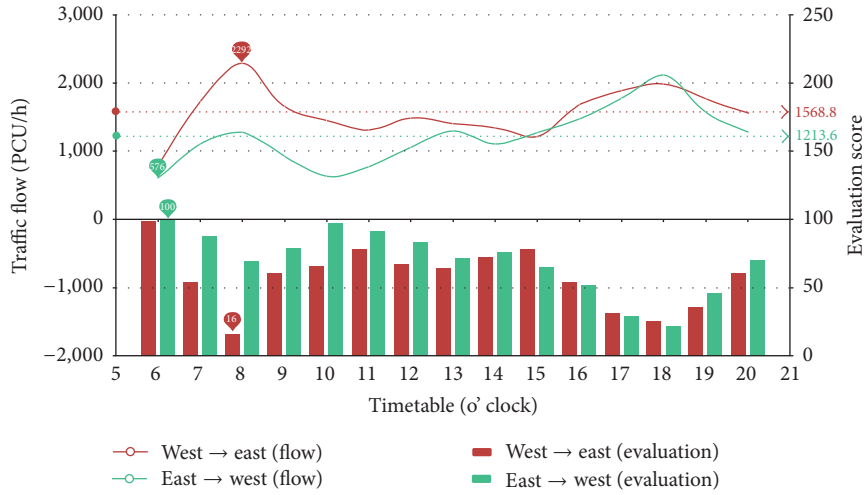


Figure 11: Comparison of evaluation scores and traffic conditions (traffic flows) for connected vehicles.

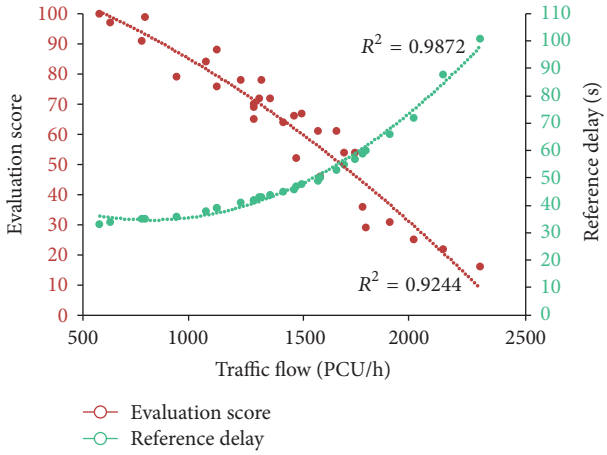


Figure 12: Comparison of evaluation scores and traffic conditions (reference delay) for connected vehicles.

Acknowledgments

This research was supported by grants from the Beijing Natural Science Foundation (4174088), National Natural Science Foundation of China (61603004), and the Scientific Research Project of Beijing Education Committee (Grant nos. PXM2017-014212-000031, PXM2017-014212-000033).

References

[1] J. Hu, B. B. Park, and A. E. Parkany, "Transit signal priority with connected vehicle technology," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2418, pp. 20–29, 2014.

[2] J. Hu, B. B. Park, and Y.-J. Lee, "Coordinated transit signal priority supporting transit progression under Connected Vehicle Technology," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 393–408, 2015.

[3] C. Ding, D. G. Wang, C. Liu, Y. Zhang, and J. W. Yang, "Exploring the influence of built environment on travel mode choice considering the mediating effects of car ownership and travel distance," *Transportation Research Part A: Policy and Practice*, vol. 100, pp. 65–80, 2017.

[4] Q. L. Huang, "Data Fusion of multi-detector in the same traffic detecting section based on NN-FR," *Journal of Highway and Transportation Research and Development*, vol. 23, no. 7, pp. 120–124, 2006.

[5] C. Quek, M. Pasquier, and B. B. S. Lim, "Pop-traffic: A novel fuzzy neural approach to road traffic analysis and prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 133–146, 2006.

[6] N. L. Zhao, L. Yu, Y. B. Geng, and X. M. Chen, "Support Vector Machine-Based Approach to Data-Layer Multi-Source ITS Data Fusion," *Journal of Transportation Systems Engineering and Information Technology*, vol. 7, no. 2, pp. 32–37, 2007.

[7] E. Castillo, Z. Grande, A. Calviño, W. Y. Szeto, and H. K. Lo, "A State-of-The-Art Review of the Sensor Location, Flow Observability, Estimation, and Prediction Problems in Traffic Networks," *Journal of Sensors*, vol. 2015, Article ID 903563, 2015.

[8] K. Thomas and H. Dia, "A neural network model for arterial incident detection using probe vehicle and loop detector data," in *Conference of Australian Institutes of Transport Research (CATIT '2000)*, pp. 6–8, Australian National University, Canberra, Australia, 2000.

[9] Z. S. Yang, S. Wang, and D. S. Ma, "Review of basic traffic data fusion methods," *Journal of Highway and Transportation Research and Development*, vol. 23, no. 3, pp. 111–116, 2006.

[10] Y. W. Ren, J. Lu, H. Y. Wang, and Q. J. Xiang, "Fuzzy fusion arithmetic for adjusting observed traffic data," *Journal of Highway and Transportation Research and Development*, vol. 23, no. 9, pp. 85–89, 2006.

[11] C. Stutz and T. A. Runkler, "Classification and prediction of road traffic using application-specific fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 3, pp. 297–308, 2002.

[12] G.-Y. Jiang, J.-F. Wang, X.-D. Zhang, and L.-H. Gang, "The study on the application of fuzzy clustering analysis in the dynamic identification of road traffic state," in *Proceedings of the*

2003 IEEE International Conference on Intelligent Transportation Systems, ITSC 2003, pp. 1149–1152, October 2003.

[13] A. Rizzi, A. Iacovazzi, A. Baiocchi, and S. Colabrese, "A low complexity real-time Internet traffic flows neuro-fuzzy classifier," *Computer Networks*, vol. 91, pp. 752–771, 2015.

[14] H. F. Guo, G. Y. Jiang, and W. G. Zhu, "Study on identification method for urban road traffic conditions with inductive loop data," in *Proceedings of the 2007 IEEE International Conference on Automation and Logistics, ICAL 2007*, pp. 1736–1740, August 2007.

[15] S. L. He, J. Zhang, Y. Cheng, X. Wan, and B. Ran, "Freeway Multisensor Data Fusion Approach Integrating Data from Cellphone Probes and Fixed Sensors," *Journal of Sensors*, vol. 2016, Article ID 7269382, 2016.

[16] C. Backfrieder, G. Ostermayer, and C. F. Mecklenbrauker, "Increased traffic flow through node-based bottleneck prediction and V2X communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 349–363, 2017.

[17] B. Schünemann, "V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems," *Computer Networks*, vol. 55, no. 14, pp. 3189–3198, 2011.

[18] S. Otsuki and H. Miwa, "Contents Delivery Method Using Route Prediction in Traffic Offloading by V2X," in *Proceedings of the 7th International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2015*, pp. 239–245, twn, September 2015.

[19] J. W. Wedel, B. Schünemann, and I. Radusch, "V2X-based traffic congestion recognition and avoidance," in *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (SPAN '09)*, pp. 637–641, IEEE, Kaohsiung, Taiwan, December 2009.

[20] C. Ding, S. Mishra, G. Q. Lu, J. W. Yang, and C. Liu, "Influences of built environment characteristics and individual factors on commuting distance: a multilevel mixture hazard modeling approach," *Transportation Research Part D: Transport and Environment*, vol. 52, pp. 314–325, 2017.

[21] C. Ding, C. Liu, Y. Zhang, J. W. Yang, and Y. P. Wang, "Investigating the impacts of built environment on vehicle miles traveled and energy consumption: Differences between commuting and non-commuting trips," *Cities*, vol. 68, pp. 25–36, 2017.

Hindawi
Journal of Sensors
Volume 2017, Article ID 5290795, 8 pages
https://doi.org/10.1155/2017/5290795

Research Article

Discovering Public Transit Riders' Travel Pattern from GPS Data: A Case Study in Harbin

Shi An, Lei Wang, Haiqiang Yang, and Jian Wang

School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin, China

Correspondence should be addressed to Lei Wang; yutian@hit.edu.cn

Received 2 March 2017; Accepted 28 May 2017; Published 2 July 2017

Academic Editor: Manel del Valle

This paper proposes a public transit riders' travel pattern measuring method based on divided cells and public transit vehicle's GPS data. The method consists of two parts: detecting urban origin and destination areas and measuring the public transit riders' travel pattern. Moreover, a series of indicators are proposed to reflect the public transit riders' travel pattern. A case study is carried out to evaluate the methods, which use the GPS data collected from taxis and buses in Harbin, China. The study is expected to provide a better understanding of public transit riders' travel patterns.

1. Introduction

The dramatic increase of urban vehicles leads to many serious problems including traffic congestions, road accidents, and air pollutions, which become general conundrums in many metropolitans all over the world, or even in smaller cities. Public transit is considered to be one of the most effective solutions for these general conundrums. Public transit includes various services that provide mobility to the general public, including buses, trains, ferries, shared taxi, and their variations [1, 2]. Public transit has many obvious advantages, such as less expenses, more effective mobility, and saving travel time, which makes more and more urban residents turn to public transit service. The city travelers and commuters proportion of Beijing who takes public transit system (including bus, taxi, and rail transit) has continuously increased in last decade. And this number came up to 54.2% in 2015, which represents more than 15.5 million trips per day [3]. This trend also occurred in other metropolitans and smaller cities all over the world.

Public transit riders usually exhibit a fixed travel pattern. That is to say, at a macrolevel, usually a fixed number of Origin-Destination pairs locate in same place of urban area, and the number of trips between these OD pairs stays steady every day [4–6]. The usage ratios of each kind of public transit stay steady every day. And the trips at morning or evening rush hours occupy a large proportion of the total trips every day [3]. On the other hand, at a microlevel, a single traveler moves from resident area to work place in the morning and moves back in the evening for each day. If the travel pattern of public transit riders could be identified, the urban public transit manager can benefit from it. For example, analyzing those riders who prefer to choose public transit vehicles rather than private cars helps transit authorities to improve the strategies and even make new policy to attract new riders [7]. With a better understanding of the transfer behavior of public riders, transit agencies can adjust the bus route to make transfer easier, which can enhance the riders' satisfaction [8]. By calculating the shortest path lengths between all station pairs, the original-destination matrix, and trip lengths, transit agencies can develop fare change plans to manage demand or raise revenue [9, 10].

The Original-Destination (OD) matrix is a typical representation of residents' travel pattern, which reflects travel demand, trip generation, travel distribution, and so on. The traditional models of establishing OD matrix usually rely on travel behavior survey. In practice, household travel surveys are conducted in many countries [11]. However, the survey data contain many limitations and errors. For example, some metropolitans in Japan (Tokyo, Kyoto, Osaka, etc.) survey residents' travel behavior every 10 years. Since the cities are growing rapidly, the survey data will definitely be out of date [11]. Furthermore, the sampling rate is usually very low, which

brings sampling errors [12]. Meanwhile, many human factors may affect the accuracy of the OD matrix, such as, willfully filtering some trips, forgetfulness, and other related factors [13].

Compared with traditional survey data, GPS data and smart card data exhibit wider coverage, lower cost, and higher accuracy. With the rapid development of data-based technology, various intelligent transportation systems are widely applied in public transit system. These systems could collect residents' mobility data every day, including longitude, latitude, boarding time, and dropping off time [14]. In the last decade, various researches based on these data have been carried out, for example, mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data [14], comparing accessibility in urban slums using smart card and bus GPS data [15, 16], discovering functional zones using bus smart card data [17], and partitioning bus operating hours into time of day intervals based on bus GPS data [18], which makes the data based transportation research to be a hot spot of transportation field [19].

GPS data and smart card data are usually collected from different subsystems of one whole intelligent transportation system, or even from different systems. This is because bus, taxi, and rail transit usually belong to different public transit companies [3]. Therefore, most previous data based researches into transit traveler behaviors utilize smart data [20–22] or GPS data [23, 24], respectively. Accordingly, bus and rail transit riders' travel pattern can be generated [20–22] or travel behavior of taxi riders [23, 24], respectively. To the best of my knowledge, public transit riders' travel pattern researches are barely utilizing both smart card and GPS data. However, smart card data only provide riders' boarding and dropping off information and lack locating information [21, 22]. As a result, only approximate location can be acquired, which causes inaccuracy of origin and destination inference. Moreover, on the microlevel, GPS-equipped public transit vehicle riders' trips are far less than other public transit vehicles [3]. And the insufficient sampling number will definitely lead to inaccuracy of trip distribution. In this light, the smart data and GPS data should be integrated to discover public transit riders' travel pattern.

The aim of this paper is to propose an effective method to explore the public transit riders' travel pattern in an urban area. There are two subgoals identified: detecting urban origin and destination areas at a cell level and measuring public transit riders' travel pattern.

This paper is organized as follows. Section 2 discusses the definition of cells and locating points that will be applied to this research. Section 3 describes the proposed urban public transit riders' travel pattern measuring method. Section 4 applies the proposed public transit riders' travel pattern measuring methodology using taxi and bus GPS data and the urban road network of Harbin. Section 5 provides conclusions and recommendations for future research.

2. Definition of Cells and Locating Points

In this part, we are going to define some parameters of cells and locating points. Firstly, the urban area is divided into $n \times m$ small cells with same size. $C(x, y)$ is one of these cells, where $x = 1, 2, \ldots, n$ and $y = 1, 2, \ldots, m$.

According to the taxi GPS dataset, there are four types of occupation status. In this paper, we are going to study the public transit riders' travel pattern, so we defined two types of locating points according to the boarding and dropping off status, which are described as follows:

Type 1. $P_{\text{taxi.boarding}}$ represents taxi vehicles' locating points whose occupation status value is 768 (i.e. the occupation status is boarding).

Type 2. $P_{\text{taxi.dropping off}}$ represents taxi vehicles' locating points whose occupation status value is 16640 (i.e. the occupation status is dropping off).

The locating points whose occupation status value is 256 (represents the taxi vehicles being vacant) are of no use to the public transit riders' travel pattern, so they are not utilized in our research.

$P_{\text{taxi}}\langle \text{ID}, \text{ts}\rangle$ represents a specific locating point of a taxi vehicle, where ID and ts are the Taxi ID and Timestamp from the taxi GPS datasets.

Passenger only needs to touch smart card once while boarding bus in many cities of China, such as Guangzhou, Xi'an, and Harbin, whereas, in Beijing, passenger needs to touch smart card once again while dropping off. It is easy to know both boarding locating points and dropping off locating points in Beijing. In Harbin, the dropping off locating points can be inferred based on boarding points and time period. In order to simplify the method, in this paper, we use the most periodic commuters' data from the bus datasets. That is to say, if a specific bus rider gets to work every morning and comes back home every evening, only this type of rider's locating points will be included in our research. This type of bus rider generates two trips each day. In this light, we defined two types of bus locating points as follows.

Type 1. $P_{\text{bus.boarding}}$ represents boarding passengers' locating points.

Type 2. $P_{\text{bus.dropping off}}$ represents dropping off passengers' locating points.

$P_{\text{bus}}\langle \text{ID}, \text{ts}\rangle$ represents a specific locating point of a bus vehicle, where ID and ts are the Card ID and Timestamp from the bus GPS datasets.

How to identify the boarding locating points and dropping off locating points from smart card data is described as follows.

Step 1. Extract the locating points with same card ID (i.e., $P_{\text{bus}}\langle \text{ID}_i, \text{ts}_1\rangle, P_{\text{bus}}\langle \text{ID}_i, \text{ts}_2\rangle, \ldots, P_{\text{bus}}\langle \text{ID}_i, \text{ts}_n\rangle$) from the whole smart dataset, where $P_{\text{bus\_}i}\langle \text{ID}_i, \text{ts}\rangle$ is the locating point whose card ID is ID_i and n is the number of this kind of locating points.

Step 2. Extract the locating points whose $n = 2$ and $\text{ts}_1 \in [07{:}00, 10{:}00]$ and $\text{ts}_2 \in [15{:}00, 18{:}00]$.

Step 3. Let $P_{\text{bus.boarding}} = P_{\text{bus}}\langle \text{ID}, \text{ts}_1\rangle$ and $P_{\text{bus.dropping off}} = P_{\text{bus}}\langle \text{ID}, \text{ts}_2\rangle$.

3. Methodology

The methodology for discovering public transit riders' travel pattern is described in this section. Two stepwise methods are proposed to achieve the main goal, including detecting Origin and Destination areas, measuring the commuter pattern between each OD pair.

3.1. Detecting Origin and Destination Areas. In this part, we are going to detect Origin and Destination areas from public transit riders' GPS data (i.e., taxi and bus GPS data). Because of the reasonable urban planning and constructions in recent decades, the urban area is usually divided into many different function zones. Therefore, all the origin points and destination points of passengers' trips will be clustered into several origin areas and destination areas. In this light, we use cluster algorithm to detect origin areas and destination areas in urban area. Moreover, we do not know how many clusters are in advance, and this type of cluster is usually not spherically-shaped. Therefore, we apply a customized Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to solve this problem.

For a specific cell $C(x, y)$, we define a parameter $N(x, y)$ to present the locating point (i.e. P_{taxi} and P_{bus}) number in this cell during a specific period P. Taking a 3×3 cells, for example (as shown in Figure 1(a)), in one period all cells' parameters $N(x, y)$ can be easily calculated. For two specific cells $C(x_1, y_1)$ and $C(x_2, y_2)$, the distance between them $\text{dist}_{[C(x_1,y_1),C(x_2,y_2)]}$ is calculated as follows:

$$\text{dist}_{[C(x_1,y_1),C(x_2,y_2)]} = l \cdot \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \quad (1)$$

where l is the length of the cell. As shown in Figure 1(b), the distance between $C(i-1, j)$ and $C(i-1, j+1)$ is l, and the distance between $C(i-1, j-1)$ and $C(i, j)$ is $l \cdot \sqrt{2}$.

Some relative parameters are defined as follows:

Object o represents the cell.

Core object co: the cell satisfies $N \geq \delta_N$, where δ_N is the threshold of parameter N.

ε-Neighborhood of a core object is the space within a radius ε centered at co.

Figure 2 illustrates the flow chart of the customized DBSCAN algorithm in this research. The Original DBSCAN algorithm is based on density. Comparatively speaking, the density reachable points in our customized algorithm are defined by $N \geq \delta_N$. And we define the *minPts* value as 1 in our paper.

3.2. Measuring Public Transit Riders' Travel Pattern. The public transit riders' travel pattern can be reflected by some indicators, like trip number between each OD pair, the proportion of different transit, path between each OD pair, travel time of the path, and so on. In order to show these indicators, we defined some preliminary terms as follows.

$\text{Tr}_{\text{mode}}(\text{CL}_i \rightarrow \text{CL}_j)$ is the trip number of a specific transit mode (i.e. taxi or bus) from ith cluster to jth cluster.

Based on clusters of origin and destination area, given a specific period P, the public transit OD matrix can be calculated.

$p_{\text{mode}}(\text{CL}_i \rightarrow \text{CL}_j)$ is the proportion of a specific transit mode (i.e. taxi or bus) from ith cluster to jth cluster, which is calculated as follows:

$$p_{\text{mode}}\left(\text{CL}_i \longrightarrow \text{CL}_j\right) = \frac{\text{Tr}_{\text{mode}}\left(\text{CL}_i \rightarrow \text{CL}_j\right)}{\text{Tr}_{\text{MODE}}\left(\text{CL}_i \rightarrow \text{CL}_j\right)} \times 100\%, \quad (2)$$

where $\text{Tr}_{\text{MODE}}(\text{CL}_i \rightarrow \text{CL}_j)$ is the trip number of all transit modes from ith cluster to jth cluster.

$\text{Tt}_{\text{mode}}(\text{CL}_i \rightarrow \text{CL}_j)$ is the travel time by the specific transit mode (i.e. taxi or bus) from ith cluster to jth cluster.

4. Case Study

We apply the proposed methods to Harbin city (China). First of all, the datasets used in this case study are described, and the stepwise methods are implemented one by one.

4.1. Datasets. The smart card data and GPS data are collected from all operating buses and taxis in Harbin. Harbin is the capital of Heilongjiang province in northeast of China. With 4.74 million populations and 7,086 km^2 areas in urban area [25], Harbin is a typical developing city in China. The public transit system of Harbin consists of taxi, bus, metro, and other rail transit. In addition, all taxis are equipped with GPS device and all buses are equipped with smart card system, which make collecting GPS data and smart card data possible.

Approximately 16,000 operating taxis equipped with GPS device are running around Harbin's urban area day and night. The location information is uploaded to the management system every 30 s during the day and 2 min at night. The data are accumulated to 2G in size and around 25 million rows each day. The taxi GPS data collected from 3rd Aug. to 7th Aug. 2015 is used, consisting of taxi ID, timestamp, latitude, longitude, and status, as shown in Table 1. There are 4 kinds of "status" in the table: 17152, 16640, 256, and 768, which represent occupation, dropping off, vacant, and boarding, respectively.

There are nearly 1500 buses traveling around urban area of Harbin, and these buses belong to 100 routes. The sampling frequency is 30 s. The bus IC records collected from 3rd Aug. to 7th Aug. in 2015 are used, consisting of Route ID, Bus ID, Card ID, Timestamp, Latitude, and Longitude, as shown in Table 2.

In our research, another dataset is digital map of Harbin, which consists of most urban areas of Harbin. The research area is nearly 100 square kilometers and just covers the range of 2nd Ring Road of Harbin, as shown in Figure 3. About 80 percent of GPS points locate in this research area. We divided this area into 250 square cells with same size. And each of the cell is 200×200 meters square, which is shown in Figure 3.

4.2. Detecting Origin and Destination Areas. In this paper, we only measure the public transit riders' travel pattern during

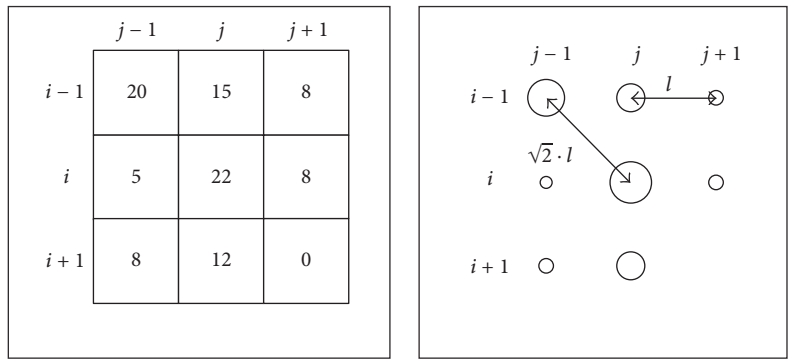


Figure 1: Example of cells with N value.

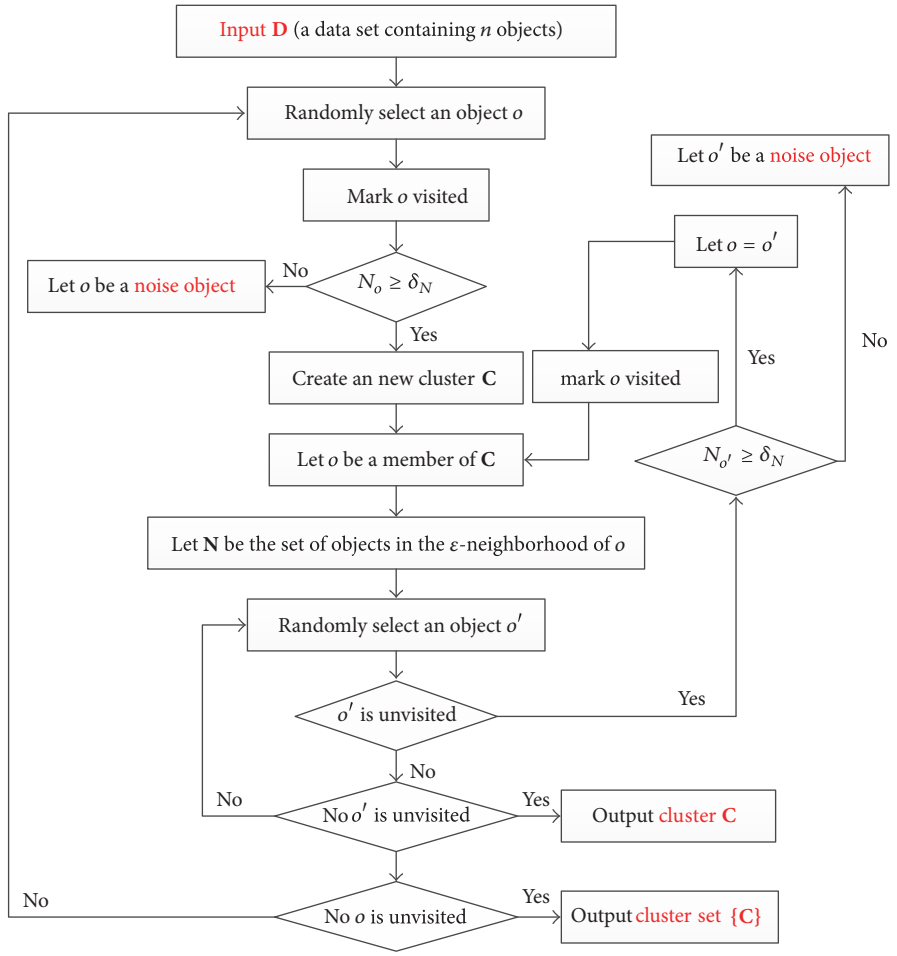


Figure 2: The flow chart of the customized DBSCAN algorithm.

working days. According to Section 3.1, we can calculate the $N(x, y)$ values of all cells. Taking one week's dropping off data (consisting of $P_{\text{taxi.dropping off}}$ and $P_{\text{bus.dropping off}}$) for example, that is, from 3rd Aug. to 7th Aug. in 2015, the relative data are illustrated in Table 3. And the sampling time is from 07:00 am to 19:00 pm.

And then we apply the customized DBSCAN algorithm to measure the origin and the destination clusters. We set ε value as 282.8 meters (i.e. $\sqrt{2} \cdot l$, where l is 200 meters). We apply different δ_N values to our experiment, in order to find out the optimal value. Table 4 illustrates the clustering results by computing with different δ_N values. According to the results,

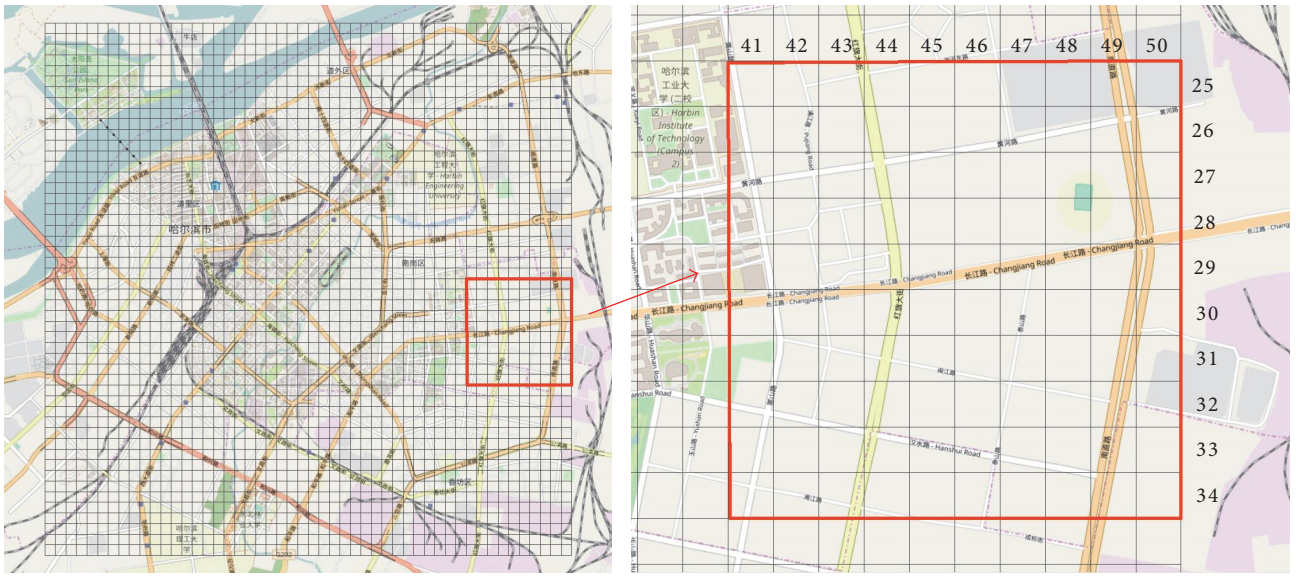


Figure 3: Illustration of divided cells. *The base map is OpenStreetMap*.

(a) Origin clusters (boarding clusters)

(b) Destination clusters (dropping off clusters)

Figure 4: Origin and destination clusters results.

when the value is larger than 1500, the cluster number is going to be stable. So we set the δ_N value as 2,000 each day and 10,000 for 5 days. In this light, we can measure the origin clusters (i.e. boarding clusters) and destination clusters (i.e. dropping off clusters) for the 5 days, as shown in Figure 4.

4.3. Measuring Public Transit Riders' Travel Pattern. For a specific OD pair, we can measure the travel pattern between them, that is, $\mathrm{Tr}_{\mathrm{mode}}(\mathrm{CL}_i \rightarrow \mathrm{CL}_j)$, $p_{\mathrm{mode}}(\mathrm{CL}_i \rightarrow \mathrm{CL}_j)$, and $\mathrm{Tt}_{\mathrm{mode}}(\mathrm{CL}_i \rightarrow \mathrm{CL}_j)$, as mentioned in Section 3.2. Taking a 20×17 cells area, for example, as shown in Figure 5, there are

Table 1: Typical GPS records of a taxi.

Taxi ID	Timestamp	Latitude	Longitude	Status
0100320311	2015-08-21 17:08:21	45.726025	126.69883	17152
0100320311	2015-08-21 17:08:51	45.725937	126.70046	17152
0100320311	2015-08-21 17:09:21	45.72592	126.70041	16640
0100320311	2015-08-21 17:09:51	45.72424	126.70028	256
0100320311	2015-08-21 17:10:21	45.727768	126.678665	256
0100320311	2015-08-21 17:10:51	45.72415	126.70134	768
0100320311	2015-08-21 17:11:21	45.7243	126.70027	17152

Note. The columns from the left to right, respectively, represent the taxi identification; the taxi sampling time; the latitude and longitude of the position; and the occupation status.

Table 2: Typical Bus GPS records of a bus.

Route ID	Bus ID	Card ID	Timestamp	Latitude	Longitude
120	0100320311	313011090	2015-08-11 08:08:21	45.739658	126.643119
120	0100320311	313013116	2015-08-11 08:08:24	45.739658	126.643119
120	0100320311	313008101	2015-08-11 08:08:28	45.739658	126.643119
120	0100320311	313191231	2015-08-11 08:08:33	45.73966	126.643119

Note. The columns from the left to right, respectively, represent the bus route number; bus ID; smart card ID; the card sampling time; the latitude and longitude of the position.

Table 3: The dropping off data in sampling 5 days.

Date	Total	Average	Minimum	Maximum
20150803	1,121,198	448.5	0	26,371
20150804	1,112,459	445.0	0	24,508
20150805	1,112,058	444.8	0	25,100
20150806	1,111,987	444.8	0	24,766
20150807	1,110,522	444.2	0	24,958
Sum	1,113,645	445.5	0	25,140

Note. Total is the number of dropping off points in all cells, *average* is the average number of dropping off points in one cell, *minimum* is the minimum number of dropping off points in one cell, and *maximum* is the maximum number of dropping off points in one cell.

Table 4: Dropping off clustering results by different δ_N variously.

Date	Cluster number					
	$\delta_N = 500$	$\delta_N = 1000$	$\delta_N = 1500$	$\delta_N = 2000$	$\delta_N = 2500$	$\delta_N = 3000$
20150803	38	71	49	30	24	19
20150804	36	75	44	27	20	21
20150805	32	70	47	29	18	19
20150806	35	77	46	29	21	17
20150807	41	75	50	28	20	18
Average	36.4	73.6	47.2	28.6	20.6	18.8
Variance	11.3	8.8	5.7	1.3	4.8	2.2

one typical origin cluster and one typical destination cluster. And the three indicators' values in the survey 5 days can be calculated, which are shown in Table 5.

The Origin Cluster in this case study is the cell $C(48, 22)$, and the Destination Cluster consists of five cells (i.e. $C(35, 33)$, $C(35, 34)$, $C(36, 34)$, $C(36, 35)$, and $C(36, 36)$). The cell $C(48, 22)$ is located nearby Harbin Hongqi Resident District, Heping Resident District, Yuanda Central Park Resident District, and so on. And there is a bus station locating in this cell, which consists of 12 bus routes (e.g. 14, 25, 31, and 44). And the Destination Cluster is located nearby Harbin International Golf Club, Harbin Wanda Plaza, several banks, and so on. This cluster area locates in an important commercial district in Harbin and definitely attracts many residents' travels. In this Cluster, there are several bus stations consisting of more than 20 bus routes (e.g. 17, 27, 34, and 71). From the Origin Cluster to Destination Cluster, there are at least three direct bus routes (i.e. 71, 82, and 209).

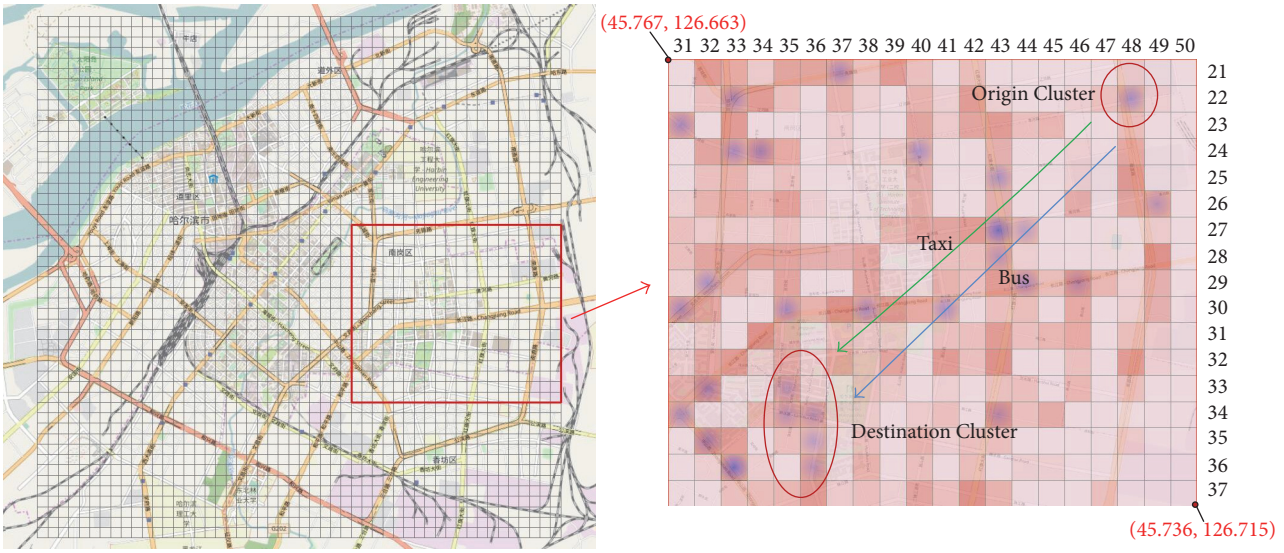


Figure 5: Illustration of a 20 × 17 cells area.

Table 5: Results of the three indicators.

	Taxi	Bus
Tr	367	509
p	41.9%	58.1%
Tt	16 min	40 min

5. Conclusion

This paper presented a cell-based urban public transit riders' travel pattern measurement method. The method used GPS-equipped public transit vehicle's locating data, which is more realistic and easy to collect. We proposed a customized DBSCAN algorithm to detect the origin and destination areas. We computed three indicators for each OD pair, which can reflect the relationship between the origin area and destination area. We carried out a numerical case study to evaluate our method, which uses taxi and bus GPS data in Harbin, China. The results can reflect OD pairs and relationship between each OD pair.

In the future, we will improve the customized DBSCAN algorithm by accuracy and efficiency. The travel pattern is a little simple in this paper, so we will further discover some other indicators in depth. And we plan to extend this research by utilizing more kinds of GPS data from different public transit vehicles.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Project no. 51478151). This work was performed at the Key Laboratory of Advanced Materials & Intelligent Control Technology on Transportation Safety, Ministry of Communications, China.

References

[1] T. Litman, *Evaluating Public Transit Benefits and Costs*, Victoria Transport Policy Institute, British Columbia, Canada, 2015.

[2] C. Ding, Y. Lin, and C. Liu, "Exploring the influence of built environment on tour-based commuter mode choice: a cross-classified multilevel modeling approach," *Transportation Research Part D: Transport and Environment*, vol. 32, pp. 230–238, 2014.

[3] Beijing Transport Annual Report, Beijing Transportation Research Center, Beijing, China, 2015.

[4] M. A. Munizaga and C. Palma, "Estimation of a disaggregate multimodal public transport origin-destination matrix from passive smartcard data from Santiago, Chile," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 9–18, 2012.

[5] C. Ding, S. Mishra, G. Lu, J. Yang, and C. Liu, "Influences of built environment characteristics and individual factors on commuting distance: a multilevel mixture hazard modeling approach," *Transportation Research Part D: Transport and Environment*, vol. 51, pp. 314–325, 2017.

[6] C. Ding, D. Wang, C. Liu, Y. Zhang, and J. Yang, "Exploring the influence of built environment on travel mode choice considering the mediating effects of car ownership and travel distance," *Transportation Research Part A: Policy and Practice*, vol. 100, pp. 65–80, 2017.

[7] A. Imaz, K. M. N. Habib, A. Shalaby, and A. O. Idris, "Investigating the factors affecting transit user loyalty," *Public Transport*, vol. 7, no. 1, pp. 39–60, 2014.

[8] N. Nassir, M. Hickman, and Z.-L. Ma, "Activity detection and transfer identification for public transit fare card data," *Transportation*, vol. 42, no. 4, pp. 683–705, 2015.

[9] Z.-J. Wang, X.-H. Li, and F. Chen, "Impact evaluation of a mass transit fare change on demand and revenue utilizing smart card data," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 213–224, 2015.

[10] C. Ding, X. Wu, G. Yu, and Y. Wang, "A gradient boosting logit model to investigate driver's stop-or-run behavior at signalized intersections using high-resolution traffic data," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 225–238, 2016.

[11] Q. Ge and D. Fukuda, "Updating origin-destination matrices with aggregated data of GPS traces," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 291–312, 2016.

[12] A. Santos, N. McGuckin, H. Y. Nakamoto, D. Gray, and S. Liss, Summary of travel trends: 2009 national household travel survey (No. FHWA-PL-ll-022), 2011.

[13] K. M. Currans and K. J. Clifton, "Using household travel surveys to adjust ITE trip generation rates," *Journal of Transport and Land Use*, vol. 8, no. 1, pp. 85–119, 2015.

[14] S. An, H. Yang, J. Wang, N. Cui, and J. Cui, "Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data," *Information Sciences*, vol. 373, pp. 515–526, 2016.

[15] R. O. Arbex, B. B. Alves, and M. A. Giannotti, "Comparing accessibility in urban slums using smart card and bus GPS data," in *Proceedings of the Transportation Research Board 95th Annual Meeting (No. 16-5614)*, 2016.

[16] J. Hu, B. Park, and A. E. Parkany, "Transit signal priority with connected vehicle technology," *Transportation Research Record*, vol. 2418, pp. 20–29, 2014.

[17] Y. Long and Z. Shen, "Discovering functional zones using bus smart card data and points of interest in Beijing," in *Geospatial Analysis to Support Urban Planning in Beijing*, vol. 116 of *Geo-Journal Library*, pp. 193–217, Springer International Publishing, Beijing, China, 2015.

[18] Y. Bie, X. Gong, and Z. Liu, "Time of day intervals partition for bus schedule using GPS data," *Transportation Research Part C*, vol. 60, pp. 443–456, 2015.

[19] J. Hu, M. D. Fontaine, B. B. Park, and J. Ma, "Field evaluations of an adaptive traffic signal—using private-sector probe data," *Journal of Transportation Engineering*, vol. 142, no. 1, Article ID 04015033, 2016.

[20] X. Ma, Y. J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders travel patterns," *Transportation Research Part C: Emerging Technologies*, vol. 36, no. Part C, p. 12, 2013.

[21] S. Tao, D. Rohde, and J. Corcoran, "Examining the spatial-temporal dynamics of bus passenger travel behaviour using smart card data and the flow-comap," *Journal of Transport Geography*, vol. 41, pp. 21–36, 2014.

[22] L.-M. Kieu, A. Bhaskar, and E. Chung, "A modified density-based scanning algorithm with noise for spatial travel pattern analysis from smart card AFC data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 193–207, 2015.

[23] R. Paleti, P. Vovsha, D. Givon, and Y. Birotker, "Impact of individual daily travel pattern on value of time," *Transportation*, vol. 42, no. 6, pp. 1003–1017, 2015.

[24] J. Cui, F. Liu, J. Hu, D. Janssens, G. Wets, and M. Cools, "Identifying mismatch between urban travel demand and transport network services using GPS data: a case study in the fast growing Chinese city of Harbin," *Neurocomputing*, vol. 181, pp. 4–18, 2016.

[25] Harbin Statistics Yearbook Editorial Department, *Energy Statistics Yearbook*, China Statistics Press, Harbin, China, 2014.