

Research Article

A Dynamic Travel Time Estimation Model Based on Connected Vehicles

Daxin Tian,^{1,2,3} Yong Yuan,¹ Honggang Qi,⁴ Yingrong Lu,¹ Yunpeng Wang,^{1,2,3} Haiying Xia,⁵ and Anping He⁶

¹Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems & Safety Control, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

²Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Beijing 100191, China

³Key Lab of Urban ITS Technology Optimization and Integration, The Ministry of Public Security of China, Hefei 230088, China

⁴School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

⁵Research Institute of Highway, Ministry of Transport, Beijing 100088, China

⁶School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China

Correspondence should be addressed to Anping He; hapetis@gmail.com

Received 4 April 2015; Accepted 27 April 2015

Academic Editor: Xiaoyu Song

Copyright © 2015 Daxin Tian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With advances in connected vehicle technology, dynamic vehicle route guidance models gradually become indispensable equipment for drivers. Traditional route guidance models are designed to direct a vehicle along the shortest path from the origin to the destination without considering the dynamic traffic information. In this paper a dynamic travel time estimation model is presented which can collect and distribute traffic data based on the connected vehicles. To estimate the real-time travel time more accurately, a road link dynamic dividing algorithm is proposed. The efficiency of the model is confirmed by simulations, and the experiment results prove the effectiveness of the travel time estimation method.

1. Introduction

For the past decades, many densely populated cities suffered from traffic congestion in road systems. As is known to all, traffic congestion causes many serious problems, such as the waste of time for vehicle drivers, fuel consumption, and air pollution. To alleviate the problems caused by traffic congestion, widening existing roads and constructing new roads are the main methods. However, this kind of method cannot satisfy the requirement of the increasing number of vehicles. A number of studies have indicated that route guidance system (RGS) can effectively increase the traffic capacity of roads [1].

The RGS can be classified into static or dynamic route guidance. Static route guidance systems are designed to compute the shortest path with Dijkstra algorithm to direct

a vehicle from its origin to destination. Nevertheless, these paths are identified based on predetermined set of network attributes and not combined with real-time traffic conditions on roadways. So the shortest path may not be the quickest path. Advancements made in connected vehicle technology which is integrated with wireless communications and global position system (GPS) now allow much greater access to real-time roadway data. Access to the data, along with computational power available, has made implementation of dynamic route guidance systems (DRGS) feasible and one of the most promising technologies for alleviation of traffic congestion, which will reduce the travel time of drivers, avoid congested road segments, and raise road network efficiency.

However, some challenges have to be confronted in DRGS. For example, the amount of spatial data collected is large and how to transfer it to road weight is a problem. And



FIGURE 1: The vehicles clustering on the road.

the traffic state of the road network changes frequently; the system should be able to quickly obtain the accurate real-time travel time. In the recent years, researches have been devoted to design various models to calculate the travel time of road link accurately. Yuan et al. [2] proposed a method to adaptively split a day into different time segments, based on the variance and entropy of the travel time between landmarks. Tu et al. [3] proposed a method to build time varying road networks. A congestion index is introduced to quantify the traffic state of each snapshot, and the shortest path of origin-destination (OD) pair at different departure time was calculated by A* shortest path algorithm. Zheng and van Zuylen [4] presented a new descriptive model for estimating hourly average of urban link travel times using taxicab OD trip data. The model estimated the link travel times by minimizing the error between the expected path travel times and the observed path travel times. Li et al. [5] discussed the speed-based travel time estimation models, where the travel time estimation errors were quantified against actual travel times measured by a timed number plate survey and time-stamped toll tag data. El Esawey and Sayed [6] studied travel time estimation using buses as probes. The model was calibrated and validated using real-life traffic volumes and travel time data, and the neighbor links travel time estimation accuracy using bus probes data was assessed using the Mean Absolute Percentage Error (MAPE).

In summary, previous research efforts focused on obtaining travel times and speeds of the link itself using the mean metric. However, as the distribution of the vehicles on the road is uneven, the way to estimate average travel time based on the mean metrics such as mean time and mean speed without considering the influence of any variation due to the stochastic nature of the traffic cannot adequately reflect the actual degree of congestion of each road. Hence, we have to develop appropriate methodologies to accurately estimate the performance metric of interest at the link, path, and network level.

The vehicles on the road have the characteristics that adjacent vehicles usually share the same traffic condition, so vehicles can be clustered into different cells (see Figure 1). Therefore each road segment can be dynamically divided into several cells depending on the traffic state. Accounting for time-dependent traffic characteristics and DRGS's challenges, this paper proposes a road link dynamic dividing (RLDD) model to process spatial and temporal traffic data for estimating travel time more accurately.

The rest of this paper is organized as follows. Section 2 presents the RLDD model for DRGS. Section 3 discusses the dynamic travel time estimation algorithm in detail.

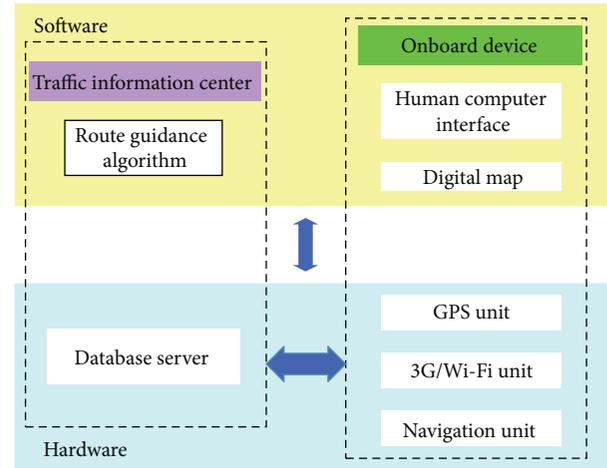


FIGURE 2: The real-time dynamic route guidance system.

The performance of the method and experimental results are presented in Section 4, and the conclusions are given in Section 5.

2. Road Link Dynamic Dividing Model

The main components of the RLDD based on the connected vehicle have been shown in Figure 2. The system includes two parts: the software part and the hardware part. In the software part, the traffic information center uses route guidance algorithm to find the quickest route for the vehicle which sends request including its current position and desired destination for reasonable route. The desired destination can be input for vehicle drivers through human computer interface of onboard device, which incorporates some multimedia technologies, such as graphical user interface and speech recognition to provide a user-friendly interface. And the digital map of onboard device can provide map information of the road networks for navigation. In the hardware part, database server is used to receive and send information to specified vehicles. And the more important function is to save real-time traffic state data collected by the GPS unit and 3G/Wi-Fi unit from the online vehicles and geoinformation data, such as road net information. The real-time vehicle route guidance system can implement interaction between vehicles and traffic information center so as to provide optimal routes from dynamic traffic state.

In the system, one of the most important parts is the data collection model. Data collection approaches in the dynamic routing guidance system can be classified into two types: infrastructure-based and infrastructure-free. In the former approach, a large number of wireless sensors and communication equipment are deployed along the roadside to monitor the traffic condition of each road. Then, the local traffic information can be transmitted to traffic information center (TIC). After the process of the collected information,

vehicles can obtain the required information from TIC via 3G or other wireless networks. Usually the loop detectors data are the major data source of infrastructure-based method [7–9].

In the infrastructure-free method, roadside sensors and roadside communication equipment are no longer needed. The sensors onboard are the main equipment to probe the traffic condition. One of the applications is floating car technology, which has been used for travel time data collection by some researches [10, 11]. Floating cars can be commercial fleet vehicles, taxis, buses, and so on. They move on a road network and the travel time of the individual vehicles over each link in their route can be measured and recorded. Another kind of infrastructure-free method is using vehicle-to-vehicle (V2V) links and multihop relay to collect real-time traffic information. Several V2V based traffic information systems have been developed in [12–14]. Wischhof et al. [12] develop a SeU-Organising Traffic Information System (SOTIS) based on the intervehicle communication. In SOTIS, each vehicle monitors the local traffic situation, and a traffic situation analysis process runs in each vehicle and the results are broadcasted to all vehicles within the transmission range. Ding et al. [13] developed a real-time route guidance algorithm, called V2R2 (vehicle-to-vehicle real-time routing). In V2R2, the vehicle that needs route guidance sends route query along the shortest path to the destination via V2V communication. Then, the destination will calculate the travel time of the shortest path traversed by the RQ packet and send a route reply packet to the vehicle that needs route guidance. Jerbi et al. [14] proposed a distributed and infrastructure-free mechanism for road density estimation. It divided a road into multiple cells and employs cell density data packet to collect density information on each road.

In view of the above data collection methods, the real-time traffic state data is collected by the onboard hardware from the online vehicles and saved in the data sever. The real-time state of vehicles (e.g., location and speed of vehicles) at located time obtained from the route guidance system is stored in a database ordered by the GPS ID. However, the data formats cannot be applied to the route guidance algorithm. So map-matching method [15] is adopted to combine them to the right road links. The data will be restored in accordance with each road link in the form

< roadid, (startpoint: longitude & latitude), (endpoint: longitude & latitude), (carid, location, speed, time),... >, where

roadid is road identifier;

startpoint, *endpoint* are two natural nodes of the road link;

(*carid*, *location*, *speed*, *time*) denotes real-time information of vehicles and is arranged by the distance from the startpoint, where

carid is the identifier of vehicle;

location is location information of the vehicle (e.g., its latitude and longitude);

speed is the velocity of vehicle;

time is the local time.

Finally this type of data is inputted to the route guidance algorithm, which is the core part of the system. In the algorithm, data is classified further and a graph will be built to represent the current traffic state. The detailed steps will be discussed in the next section.

3. Dynamic Route Guidance Algorithm

3.1. Road Link Structure. For a DRGS, it should provide dynamic routing planning based on real-time traffic information and traffic states. To resolve the problem, it is crucial to build a dynamic network graph of roadway where the real-time traffic information is contained in the graph. The road network is defined with a graph (V, E) where V is a set of vertices and E is a set of edges. Each edge $e \in E$ has the following fields: two endpoint vertices e_{v_1} and e_{v_2} and weight e_{cost} . The topological structure of road can be obtained from the digital map, while the most important task that needs to be done is how to calculate weight of each road from the real-time data. So, we have to define a metric to measure it. Bovy and Stern [16] showed that more factors could influence route choice, such as travel time (TT), travel distance (TD), width of the road, delays, road safety, and traffic density. van Vliet and van Vuren [17] assumed that distance or time minimization was the only criterion for drivers' route choice. Generally, TT and TD are usually considered as the main factors to decide the route selection in current onboard route guidance system. Here, we adopt the TT as the main metric and the TD as the auxiliary metric to build the graph.

In the graph, we define E as the road link. Traditionally, a link is defined to be the road section divided by nodes, and nodes are often defined as the center of the intersections or the road section between the stop lines of two adjacent intersections. However the traditional road dividing method causes error because the traffic state on different parts of one link may vary greatly, especially when the link is long. Taking into account adjacent vehicles usually sharing the same traffic condition, a link can be divided into one or more segments. And the travel time can be estimated by the sum of all the segments' travel times.

3.2. Road Link Dynamic Dividing Algorithm. From the database presented in Section 2, we can get a table which records road links with real-time traffic data. The following work we need to do is to calculate the link travel time. Usually the road links have the character that vehicles are not evenly distributed along the road, but they may cluster in some segments of a road link. Based on this character, road can be divided into different segments dynamically. In the following, we will illustrate the way to divide road links. Firstly, some definitions will be given to describe the above attributes of road segment.

Definition 1 (adjacent vehicle). It is determined by a distance function (e.g., Euclidean distance) for location points of vehicles p and q , denoted by $\text{dist}(p, q)$.

```

Input:
Data set LinkD = {P1, P2, ..., Pn}
Output:
Cell Labels (CellID or "noise") of Data set LinkD
Initialization:
Eps and MinPts;
Mark all points P with "untouched";
CellID = 0;
Program:
for each point Pj in LinkD
  if Pj is not marked as "touched" then
    Mark Pj as "touched";
    Calculate the Number (denotes as Numpj) of points in the Eps-neighborhood of Pj;
    if Numpj < MinPts then
      Mark each point in Eps-neighborhood of Pj as "noise";
    else
      CellID = CellID + 1;
      Mark each point in Eps-neighborhood of Pj with "CellID";
      Insert the points {Pj, Pj+1, ..., Pj+Numpj} without "touched" in Eps-neighborhood of Pj into a queue;
      while the queue is not empty do
        Select the head object Pj in the queue;
        Mark Pj as "touched";
        Calculate the Number (denotes as Numpj) of points in the Eps-neighborhood of Pj;
        Delete Pj from the queue;
        if Numpj ≥ MinPts then
          Mark each point without "touched" in Eps-neighborhood of Pj with "CellID";
        end if
      end while
    end if
  end if
end for

```

ALGORITHM 1: Road link dynamic dividing algorithm.

The location point of vehicle is usually recorded as (latitude, longitude) on the World Geodetic System (WGS). But for the simulation, we always convert it into the Cartesian coordinate system. Then the location can be denoted as a point like (x, y) . Therefore, in this paper, Euclidean distance is used to determine whether two vehicles are adjacent. Euclidean distance of two points is

$$\text{dist}(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (1)$$

where $i = (x_i, y_i)$ and $j = (x_j, y_j)$ are two location data points.

Definition 2 (Eps-adjacent vehicles). The Eps-adjacent vehicles of a vehicle p are defined by $\{q \in D \mid \text{dist}(p, q) \leq \text{Eps}\}$, where Eps is a given threshold distance.

Definition 3 (core vehicle). A core vehicle refers to such point that its neighborhood of a given radius (Eps) has to contain at least a minimum number (MinPts) of other vehicles.

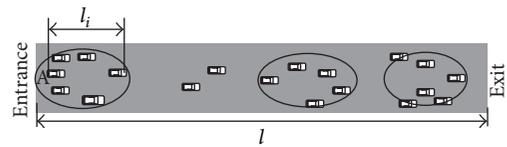


FIGURE 3: Example of the clustering algorithm.

Definition 4 (cell and noise vehicle). In a given data set, all vehicles nearby a core vehicle form a cell. And the vehicle which could not be contained by any cell is called noise vehicle.

Based on the above definition, the RLDD will use the DBSCAN clustering algorithm to cluster the real data for each link and the cells will be drawn out to reflect the traffic state. The process is shown in Algorithm 1.

Figure 3 gives an example to show the result of the above algorithm. In this figure, the road link consists of two nodes, the entrance and exit. Between the nodes, N vehicles from

a data set LinkD are distributed on it. First of all, two parameters must be determined, namely, Eps and MinPts. The Eps denotes the threshold to identify the adjacent vehicles and the MinPts means the minimal number of adjacent vehicles in the neighborhood. RLDD algorithm starts with the first point P_1 (denotes the position of vehicle A in Figure 3) in data set LinkD and retrieves all points density reachable from P_1 , which formulates a local density denoted as density (P_1) in the neighborhood of P_1 , as the total number of vehicles in its neighborhood: $\text{density}(x_1) = \text{Count}(N_{\text{Eps}}(P_1))$, where $N_{\text{Eps}}(x_1)$ denotes neighboring vehicles in the neighborhood of a radius Eps, $N_{\text{Eps}}(P_1) = \{P_j \in \text{LinkD} \mid \forall j, \text{distance}(P_j, P_1) < \text{Eps}\}$.

We assume that the MinPts is 4 and the number of the neighbors of vehicle A is 5, so A is identified as a core vehicle by the line density ($P_{\text{core}} \geq \text{MinPts}$) and a cell is formed. After the process, the algorithm outputs three cells and two noise vehicles. Finally, we can get 3 subsegments with several vehicles. Here, we denote the cluster number of each link as CNum.

3.3. Travel Time Estimation. After the road link dividing process, to estimate the real-time travel time, the road link weight needs to be calculated with the parameters: CNum and cluster contents of each link. We will present the calculation method with two types of link.

Type 1. The CNum of a link is greater than 1, which means the link has been divided into several subsegments.

Type 2. The CNum is equal to 1 or 0, which means the link has a congestion state that vehicle has distributed with high density or few vehicles are traveling on the link.

For Type 1, according to the cluster numbers, each link can be divided into several segments. The length of the i th segment l_i is the distance between the head and tail vehicle of one cell; see Figure 3. Then the average speed in each segment can be calculated as

$$v_m^i = \frac{1}{D_i} \sum_{g=1}^{D_i} v_g^i, \quad (2)$$

where v_g^i is the speed of the g th vehicle in the i th segment and D_i is the number of vehicles in the i th segment. Then the travel time t_i of the i th segment in the link can be calculated as

$$t_i = \frac{l_i}{v_m^i}. \quad (3)$$

The travel time of the other parts in the link can be defined as t_{others} :

$$t_{\text{others}} = \frac{l - \sum_{i=1}^{\text{CNum}} l_i}{V_{\text{max}}}, \quad (4)$$

where l is the length of the link; l_i is the length of the i th segment; V_{max} is the maximum travel speed of vehicles allowed on the link; and CNum is the number of the segments in the link. Then, we can get the weight of the link as follows:

$$w_{rs} = \sum_{k=1}^{\text{CNum}} t_k + t_{\text{others}}. \quad (5)$$

For Type 2, since the link needs not to be divided, the average speed of road link is defined as

$$v_m = \begin{cases} \frac{1}{n} \sum_{i=1}^n v_i & \text{CNum} = 1 \\ V_{\text{max}} & \text{CNum} = 0, \end{cases} \quad (6)$$

where V_{max} is the maximum travel speed of vehicles allowed on the link, v_i is the speed of the i th vehicle on the link, and n is the number of the vehicles on the link. So, the travel time of the road link can be calculated with

$$t = \frac{l}{v_m}, \quad (7)$$

where l denotes the length of the road link. The weight of the road link is $w_{rs} = t$.

Finally, the real-time road network graph $G(N, E, W)$ is built, where $N = \{1, 2, \dots\}$ is the node set, $E = \{(r, s) \in N \times N\}$ is the road link set, and $W = \{w_{rs}\}$ is the weight set. After we get the origin and destination at any given start time from drivers, based on the Dijkstra algorithm, we can draw a subgraph $G1$ from the real-time road network G as the selected quickest path.

4. Experiments

4.1. Simulation Setup. To validate the RLDD algorithm, we use the microscopic road traffic simulation package ‘‘Simulation of Urban MObility’’ (SUMO) to model the urban traffic [18]. Two applications in the SUMO package are used to generate the road network and vehicle routes. One is NETCONVERT tool which can import digital road networks from different sources and converts them into the SUMO format. Another is DUAROUTER which can generate routes and generate vehicles into road networks.

First of all, we choose a downtown area road network in Beijing, which is downloaded from the OpenStreetMap (OSM) [19] as a simulation case; see Figure 4(a). Then the OSM XML file is edited in Java OpenStreetMap Editor (JOSM) in order to remove the railway, pedestrian, buildings, and so on, which are not used in the simulation; the result is shown in Figure 4(b). The simplified version of network consists of 10 nodes, 16 links, and 4 traffic lights. The road properties, for example, street name, speed limit, and phase assignment of traffic light, are all included in the OSM data.

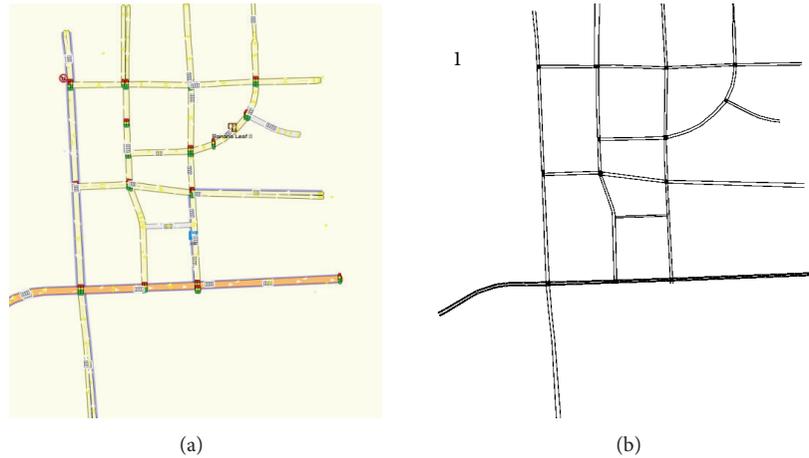


FIGURE 4: Simulated road network: (a) image from the OpenStreetMap and (b) simplified diagram from the SUMO.

Since OSM data works with the WGS instead of the Cartesian coordinate system used by SUMO, the OSM file needs to be converted into the network file through the NETCONVERT. Traffic light logics, speed limits, and other properties are also encoded in the network file.

Once the road network is ready, the next step is to generate the traffic. First, the properties of vehicles such as acceleration, maximum allowed speed, and speed variation are defined. Then, we populated the roads with random routes of vehicles which are generated for a specific time interval using the DUAROUTER. Since the limited simulation network makes the traveling vehicles leave the network fast (typically not more than 250 s), in each time step a given number of vehicles are emitted to the network in order to achieve traffic equilibrium. We set the simulation time to 1000 seconds, and 575 vehicles have been generated with random trips. From the SUMO generated data, we can get the real-time travel information of vehicles at each 1 s interval, which can be used to validate the algorithm.

The RLDD algorithm is realized in MATLAB, and Figure 5 shows the result of the collected data in the road network. It can be found that vehicles cluster at different parts of road link is really evident. The simulated data agrees with the characteristics we proposed in Section 3. Suppose that we focus on the travel time of links 1, 2, 3, 4, and 5. These links' length ranges from 174 m to 348 m. We will extract the data of the above links and store it accordingly.

4.2. Travel Time Estimation. In RLDD algorithm, the input includes not only the data set, but also two parameters Eps and MinPts needed for clustering; in the simulation, Eps = 10 and MinPts = 2. Figure 6 shows the data distribution and clustering results for each selected link at 300 s. From the results it can be found that the road is divided into several subsegments.

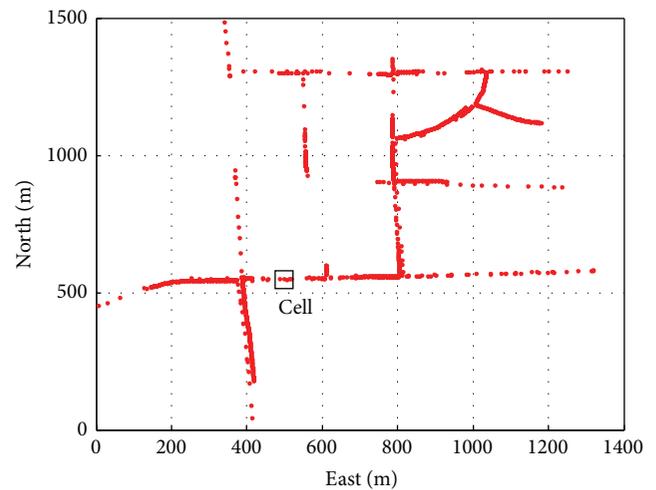
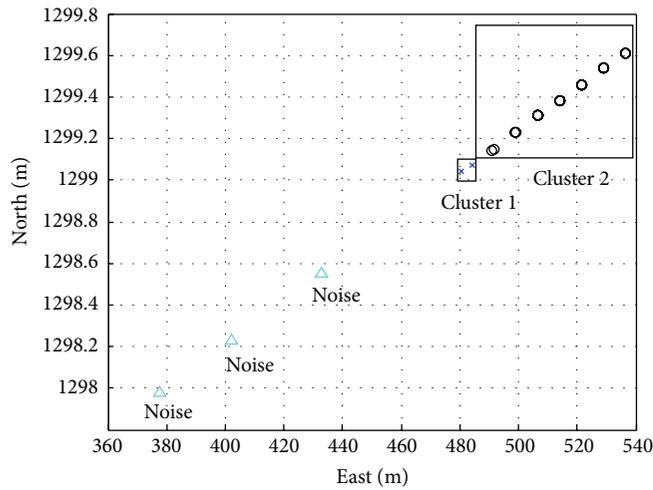


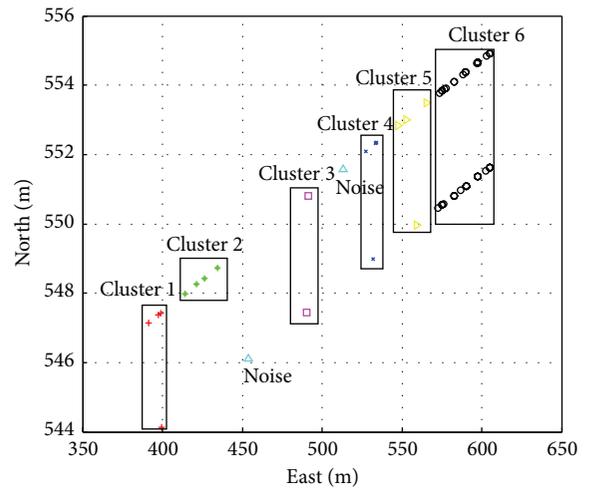
FIGURE 5: Travel data distribution on the road network.

In the simulation, we collected the traffic information every 60 s. Figure 7 shows the cluster number and the estimated travel time for each link. From the results it can be found that when the number of cluster decreases, the travel time will increase accordingly. This phenomenon is consistent with reality, because when the number of the cell reduces, the density of the vehicle will rise. It proves the algorithm can divide the link dynamically according to the traffic condition.

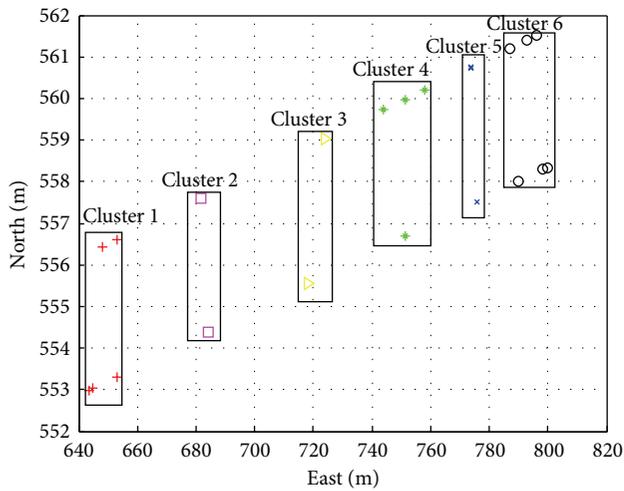
To validate the accuracy of travel time estimation, we select vehicles in the simulation which first enters link 2 at the time intervals [0–60], [60–120], [120–180], [180–240], and [280–320]. In the SUMO, we can get the “real” travel time by the time when car enters and exits the link. SUMO can also output the estimated average travel time of each link, so we



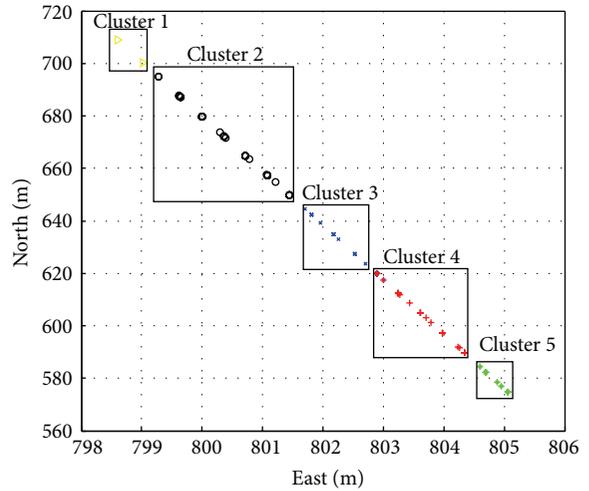
(a) Link 1



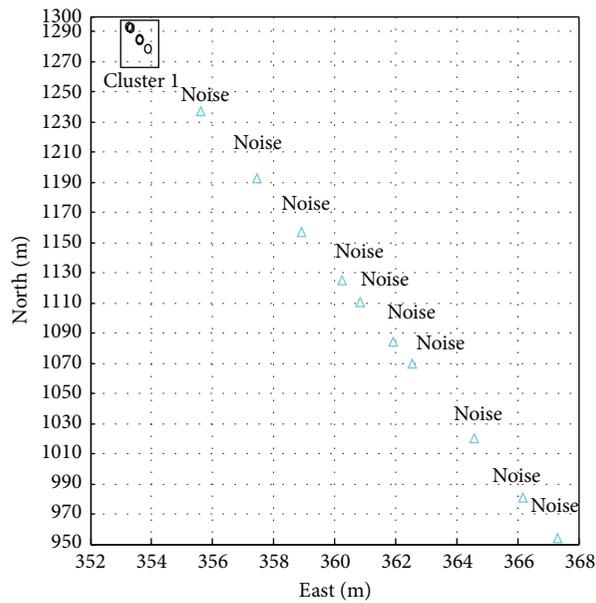
(b) Link 2



(c) Link 3

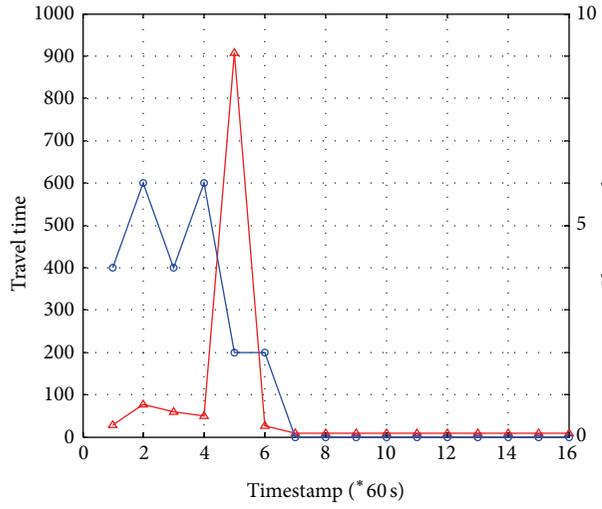


(d) Link 4

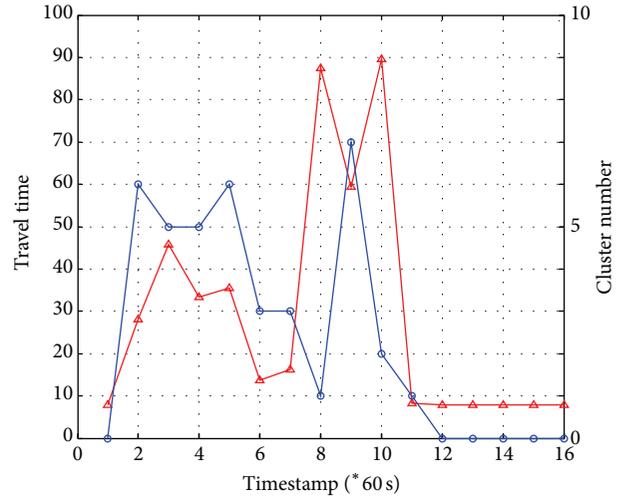


(e) Link 5

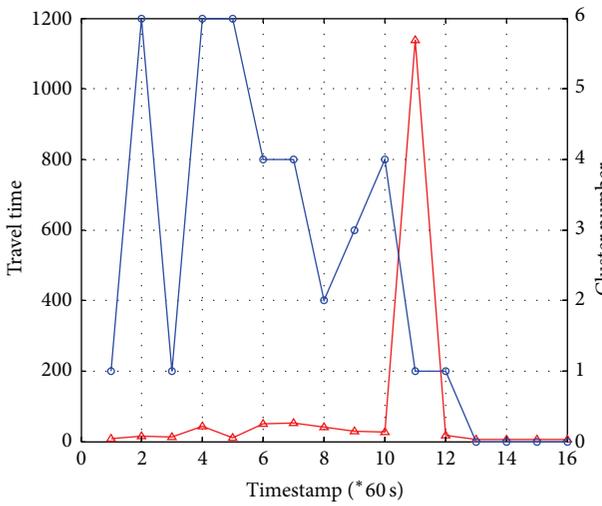
FIGURE 6: Data distribution and the clustering results on the selected links.



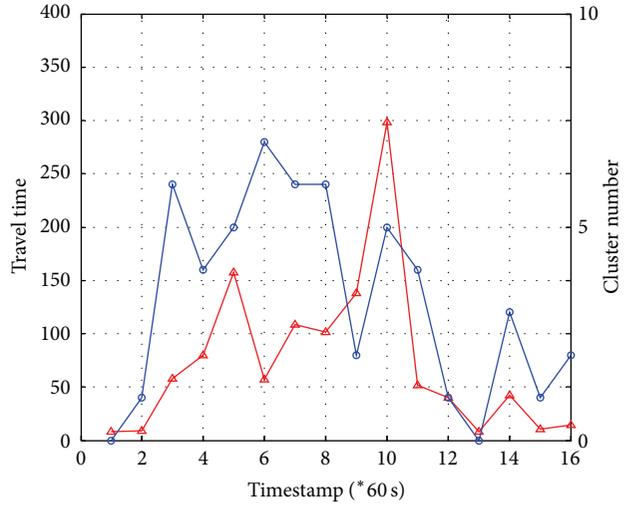
(a) Link 1



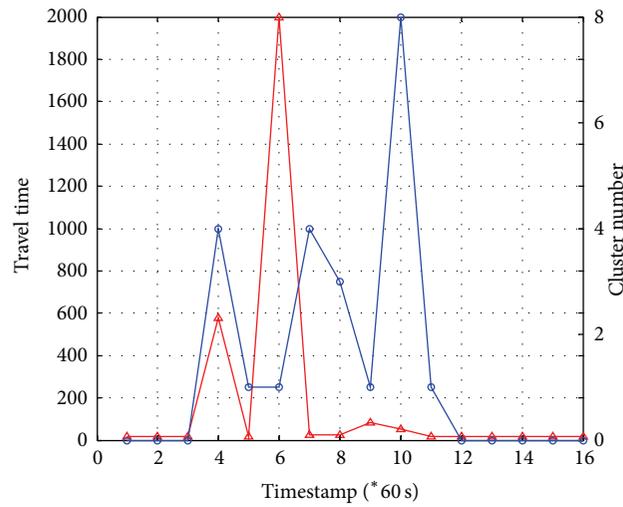
(b) Link 2



(c) Link 3



(d) Link 4



(e) Link 5

FIGURE 7: Travel time and cluster numbers of links 1–5.

TABLE 1: Optimal path and cost by traditional routing algorithm and the dynamic routing algorithm.

Method	Cost	Route
Traditional routing algorithm	6780.62 m	1 → 10 → 11 → 18 → 17 → 16 → 22 → 32 → 40 → 31 → 30 → 46
	00:00:00	419.05 s
	09:00:00	481.95 s
The dynamic routing algorithm	13:00:00	605.56 s
	17:00:00	851.92 s

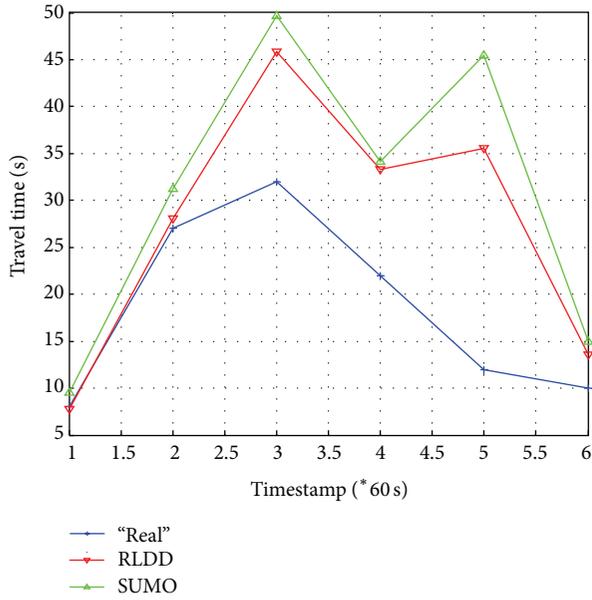


FIGURE 8: Travel time estimation.

compare it with the travel time estimated by the RLDD. The three kinds of travel time curves are shown in Figure 8. It can be found that the estimated travel time by RLDD is closer to the “real” travel time.

4.3. Dynamic Route Guidance. In order to test our dynamic route guidance algorithm, we use a real data set and road networks in Haidian District, Beijing (see Figure 9), which consists of 46 nodes and 148 links. To calculate the optimal routes, the nodes information used in the map is extracted from Google Earth and the graph structure is drawn with MATLAB Mapping Toolbox; see Figure 10.

We set node 1 as the origin and node 46 as the destination; the optimal routes in different time are shown in Table 1 and Figure 11. At 00:00:00, 09:00:00, and 13:00:00, the optimal routes are the same, but the travel time of each route varies. At 17:00:00, the optimal route is different from the above and consumes the most travel time; because it is a traffic peak,



FIGURE 9: The road network in Haidian District, Beijing.

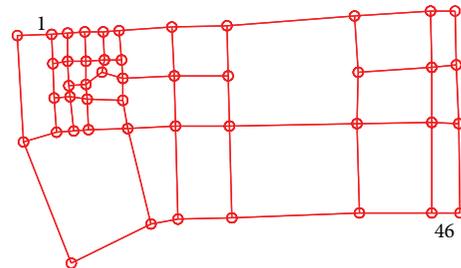


FIGURE 10: Road networks generated by MATLAB.

the traffic jam is relatively obvious. These results validate that the dynamic route guidance algorithm can use the real-time traffic information to find out optimal route with the least travel time.

5. Conclusions

Real-time dynamic route guiding is an effective method for alleviating traffic congestion problems in urban areas. In this paper we propose a road link dynamic dividing (RLDD) model to calculate the real-time traffic information. This method can be realized with the help of the connected vehicle technology which is integrated with wireless communications and global position system (GPS). The experimental results with simulation prove that the dynamic travel time

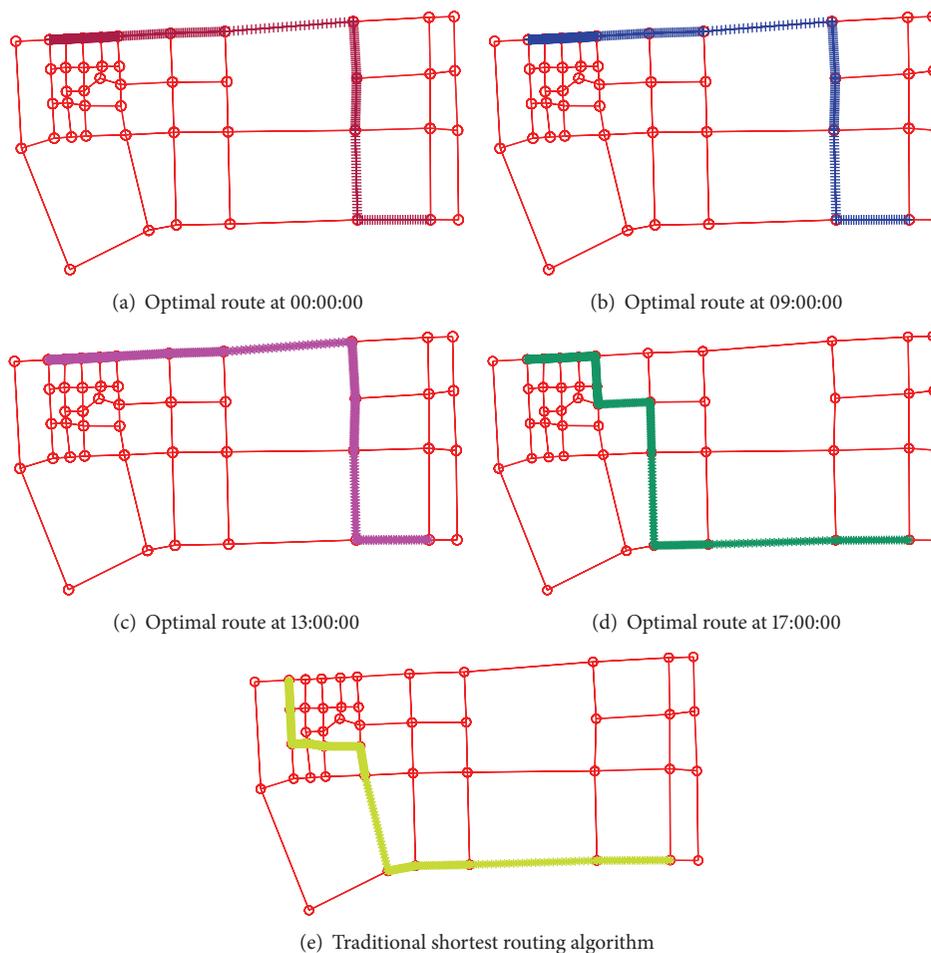


FIGURE 11: The optimal path selected by dynamic routing algorithm at different time ((a)–(d)) and traditional routing algorithm (e).

estimation model is effective and efficient in real-time traffic information acquisition and optimal route finding.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant no. 91118008 and the Foundation of Key Laboratory of Road and Traffic Engineering of the Ministry of Education in Tongji University.

References

- [1] A. Bose and P. Ioannou, "Evaluation of mixed semi-automated/manual traffic," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 868–872, September 1998.
- [2] J. Yuan, Y. Zheng, C. Zhang et al., "T-drive: driving directions based on taxi trajectories," in *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*, pp. 99–108, ACM Press, New York, NY, USA, November 2010.
- [3] W. Tu, Z. Fang, and Q. Li, "Exploring time varying shortest path of urban OD pairs based on floating car data," in *Proceedings of the 18th International Conference on Geoinformatics*, pp. 1–6, IEEE, Beijing, China, June 2010.
- [4] F. Zheng and H. van Zuylen, "Urban link travel time estimation based on sparse probe vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 31, pp. 145–157, 2013.
- [5] R. Li, G. Rose, and M. Sarvi, "Evaluation of speed-based travel time estimation models," *Journal of Transportation Engineering*, vol. 132, no. 7, pp. 540–547, 2006.
- [6] M. El Esawey and T. Sayed, "Using buses as probes for neighbor links travel time estimation in an urban network," *Transportation Letters*, vol. 3, no. 4, pp. 279–292, 2011.
- [7] T. Yamashita, K. Izumi, and K. Kurumatani, "Car navigation with route information sharing for improvement of traffic efficiency," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC '04)*, pp. 465–470, October 2004.

- [8] X. Shi, J. Hu, Y. Xu, and J. Song, "A simulation study on agent-network based route guidance system," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 263–268, September 2005.
- [9] X. Zhang, J. Hong, S. Fan, Z. Wei, J. Cao, and Y. Ren, "A novel real-time traffic information system based on wireless mesh networks," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 618–623, October 2007.
- [10] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing, (UbiComp '11)*, pp. 89–98, ACM Press, New York, NY, USA, September 2011.
- [11] O. Wolfson, Y. Zheng, and S. Ma, "T-share: a large-scale dynamic taxi ridesharing service," in *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE '13)*, pp. 410–421, IEEE Computer Society, Brisbane, Australia, April 2013.
- [12] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS—a self-organizing traffic information system," in *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC '03)*, vol. 4, pp. 2442–2446, April 2003.
- [13] J.-W. Ding, C.-F. Wang, F.-H. Meng, and T.-Y. Wu, "Real-time vehicle route guidance using vehicle-to-vehicle communication," *IET Communications*, vol. 4, no. 7, Article ID ICEOCW000004000007000870000001, pp. 870–883, 2010.
- [14] M. Jerbi, S.-M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, "An infrastructure-free traffic information system for vehicular networks," in *Proceedings of the IEEE 66th Vehicular Technology Conference (VTC '07)*, pp. 2086–2090, Baltimore, Md, USA, September–October 2007.
- [15] K. Ghys, B. Kuijpers, B. Moelans, W. Othman, D. Vangoidsenhoven, and A. Vaisman, "Map matching and uncertainty: an algorithm and real-world experiments," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 468–471, ACM, New York, NY, USA, November 2009.
- [16] P. H. L. Bovy and E. Stern, *Route Choice: Wayfinding in Transport Networks*, Studies in Operational Regional Science, Kluwer Academic Publishers, Norwell, Mass, USA, 1990.
- [17] D. van Vliet and D. van Vuren, *Route Choice and Signal Control*, Avebury ITS Institute for Transport Studies, Ashgate, Farnham, UK, 1992.
- [18] SUMO, <http://sumo.sourceforge.net>.
- [19] OpenStreetMap, <http://www.openstreetmap.org>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

