Traffic Routing Guidance Algorithm based on Backpressure with a Trade-off between User Satisfaction and Traffic Load

Rui Zhang, Zhijun Li, Cheng Feng, Shouxu Jiang
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China

Abstract—**Traffic routing guidance algorithms which only consider user satisfaction will result in road density unbalance. On the contrary, the route algorithms will not meet user request if they merely focus on the traffic load balance. So it is important to take user satisfaction and traffic load into consideration at the same time. Although a few works focus on this combination, they ignore the difference between the user requests. We propose a traffic dispersion routing algorithm on VANET naming BPR-US which is based on backpressure theory. It tries to satisfy the user demand and also separates the traffic flow. At last, we make simulation experiments to compare BPR-US algorithm and other route guidance algorithms, proving that BPR-US algorithm can have a better effect on both user satisfaction and traffic load.**

Keywords- VANET; routing guidance; backpressure; user satisfaction; traffic load

I. INTRODUCTION

As the fast increasing of vehicles in urban area, traffic congestion is becoming a problem that cannot be ignored [1]. Traffic routing guidance system has applied in our daily life to make car driving convenient. It is an important part of Intelligent Transportation System (ITS). The routing guidance system collects vehicle information (such as traffic density, vehicle speed and traffic flow) through Vehicular Ad-hoc Network (VANET). Then the information is used by routing algorithm which can return recommended routes to users. Besides routing vehicles to optimal paths, traffic routing guidance system can also reduce the environment pollution and energy consumption [2]. So it is essential for the routing guidance algorithm to achieve a great performance on both users and traffic system.

The routing guidance algorithms which only consider user satisfaction try to make sure that the user can reach the destination in shortest time. These algorithms are mostly based on shortest path or improved shortest path algorithm [1]. They predict the traffic flow by using the current road information [3], and return the shortest path from the perspective of time. Paper [2] extracts the travel time on each road by GPS raw data, and applies the Floyd-Warshall algorithm to compute the time shortest path. The Particle Swarm Optimization (PSO) algorithm [4] is introduced to reduce the time to find the shortest path. But these algorithms cannot balance the traffic load, because they lead all users to the same shortest path, then the shortest path will get congested.

The method considering traffic load on each road can have a good effect on balancing traffic density and reducing traffic congestions. It is necessary to take traffic load and user satisfaction into consideration at the same time, especially when the road density is high. Some researches have focused on this combination. The Ant Dispersion Routing (ADR) algorithm [5] is proposed to disperse vehicles into different roads in order to achieve the traffic load balance. But it ignores the user time request to arrive the destination and leads to low user satisfaction. Zhe Cong et al. figure out a modified ant algorithm to trade off between user optimum and system optimum in [6]. It can provide a fast route to users while average the road density in some degree. But the approach does not notice the different requests of users, so it cannot lead the sum of all users' satisfaction to maximum.

The remainder of this paper is organized as follows. Section II describes the traffic model and defines the "user satisfaction-traffic load" problem. In section III, the Back Pressure Routing (BPR) control method is introduced in traffic system and an improved BPR considering user satisfaction algorithm naming BPR-US is proposed. Section IV makes several experiments to compare the performances of shortest path algorithm, BPR algorithm and BPR-US algorithm from the perspective of average travel time, road density and user satisfaction. Finally, we conclude the whole paper and discuss the future work in Section V.

II. MODEL AND PROBLEM DEFINITION

In this section, a simple traffic model is introduced at first. Then we give the definition of user satisfaction and traffic load. After that, we propose the "user satisfaction-traffic load" problem in mathematical form.

A. *Traffic Model*

We suppose that there are n routes from source s to destination d. The routes are remarked as R_1, R_2,..., R_n. t_s is the time slice duration. Given that a user named u starts from s at

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

T_0 which satisfies $T_0 \in [kt_s, (k+1)t_s)$. If u chooses the route R_m, the travel time cost $\varphi_{u,m}(k)$ is calculated as follow:

$$\varphi_{u,m}(k) = \sum_{\forall(i,j)\in R_m} t_{i,j}(k_i) \quad \forall m = 1,2,...,n \tag{1}$$

i, j are the adjacent intersections of route R_m. $t_{i,j}(k_i)$ stands for the time cost of traveling through from intersection i to intersection j when the user enters i at step k_i. $t_{i,j}(k_i)$ can be expressed as:

$$t_{i,j}(k) = L_{i,j} \big/ v_{i,j}(k) \tag{2}$$

Where $L_{i,j}$ is the length of link (i,j). $v_{i,j}(k)$ is the average vehicle speed of link (i,j) at step k. Average speed on the link is a variable associated with road density. A simple linear model called Greenshields [7] is used to describe the relation between average vehicle speed and road density:

$$v_{i,j}(k) = v_f \cdot (1 - \rho_{i,j}(k) \big/ \rho_{i,j}{}^{upper}) \tag{3}$$

$\rho_{i,j}(k)$ is the average road density of link (i,j) at step k, $\rho_{i,j}{}^{upper}$ is the highest density of link (i,j). v_f is the maximum speed that a car could drive through link (i,j).

Traffic flow, road density and average speed must satisfy the following formula:

$$q_{i,j}(k) = \rho_{i,j}(k) \cdot v_{i,j}(k) \tag{4}$$

The road density is an iteration variable that related with the driving-in and driving-out flow, that is:

$$\rho_{i,j}(k) = \rho_{i,j}(k-1) + q_{in,i,j}(k) \cdot t_s \big/ L_{i,j} - q_{out,i,j}(k) \cdot t_s \big/ L_{i,j} \tag{5}$$

Where $q_{in,i,j}(k)$ and $q_{out,i,j}(k)$ are driving-in and driving-out flow respectively.

At any time, the driving-in flow of a link should be a part of the driving-out flow at the link entrance, just as:

$$q_{in,i,j}(k) = \beta_{i,j}(k) \cdot \sum_{\forall l \in I(i)} q_{out,l,i}(k) \tag{6}$$

$\beta_{i,j}(k)$ is the splitting rate of cars at the intersection i driving to intersection j. $I(i)$ is the set of intersections those are adjacent with i and can drive to i.

If a car drives out of a link, it must choose another link to drive in immediately. To ensure this, we have:

$$\sum_{\forall l \in O(i)} \beta_{i,l}(k) = 1 \tag{7}$$

$O(i)$ is the set of intersections those are adjacent with i and can be arrived from i.

B. User Satisfaction

The user u has an expecting travel time from source s to destination d, marked as a. He also has a deadline b, in other words, the longest time he can tolerate spending on the road. The user satisfaction function is defined as the same in [8]:

$$F(t,a,b) = \begin{cases} 1 & t \le a \\ \dfrac{b-t}{b-a} & a < t \le b \\ 0 & t > b \end{cases} \tag{8}$$

t is the real time the user u spends on the road. Formula (8) implies that if a route result is less than the user's expecting time cost, he or she will completely satisfy this result. But if the result exceeds the deadline, the user will not satisfy the route at all. To be mentioned here, even for the same source and same destination, different user has different expecting travel time a and deadline b. This is because the difference between the user personalization and characteristics.

C. Traffic Load

Here we define the variance of all links' density as traffic load. If the variance is large, we say the traffic load is heavy; otherwise, the load is light. Traffic load $H(\Phi, \overline{\rho})$ is defined as:

$$H(\Phi, \overline{\rho}) = \sum_{i=1}^{M} \left(\rho_i - \overline{\rho}\right)^2 \Big/ M \tag{9}$$

Where $\overline{\rho}$ is the average density of all links. ρ_i is the average density of link i during a specific time period T. M is the number of links which are on the routes from source s to destination d. Φ is a M-dimensional vector of all links' average density during T, that is $\Phi=(\rho_1, \rho_2,..., \rho_M)$. $\rho_i^{(j)}$ is the density of link i at time j. It can be collected through VANET application. $\overline{\rho}$ and ρ_i are calculated as:

$$\overline{\rho} = \sum_{i=1}^{M} \rho_i \Big/ M \tag{10}$$

$$\rho_i = \sum_{j=1}^{T} \rho_i^{(j)} \Big/ T \tag{11}$$

D. "User Satisfaction-Traffic Load" Problem

Now we define the "user satisfaction-traffic load" problem. Supposed that there are n routes from source s to destination d, the routes are marked as $R_1, R_2,..., R_n$. There are N users at source s and their destination is d. The problem is described as follows.

$$\max \beta \cdot \left[\sum_{i=1}^{N}\sum_{j=1}^{n} x_{i,j} \cdot F(\varphi_{i,j}, a_i, b_i)\right] - (1-\beta) \cdot H(\Phi, \overline{\rho}) \tag{12}$$

The problem is to maximize all users' satisfaction and minimize the traffic load of the road system. $\varphi_{i,j}$ means the time cost of user i choosing route j. a_i and b_i are user i's expecting travel time and deadline respectively. At the same time, the constraints below must be satisfied.

$$x_{i,j} \in \{0,1\} \quad \forall i = 1...N, j = 1...n \tag{13}$$

$$\sum_{j=1}^{n} x_{i,j} = 1 \quad \forall i = 1,2,...,N \tag{14}$$

$$\rho_k = \sum_{j \in S(k)} \sum_{i=1}^{N} x_{i,j} \Big/ (l_k \cdot T) \tag{15}$$

Formula (13) indicates that $x_{i,j}$ is a single value variable of 0 or 1. If user i choose route j, then the value is 1; otherwise it is 0. Formula (14) means that a user must choose only one route from source s to destination d. Formula (15) tells how to calculate the average density of link k during time period T, while l_k is the length of link k. T is a constant that must be large enough to ensure all cars can get their destination. $S(k)$ is the set of links those include link k, that is, $S(k)=\{j \mid \textit{route } j \textit{ includes link } k\}$. Formula (15) is established if and only if all cars take part in the routing guidance. Otherwise, formula (5) is used instead of (15).

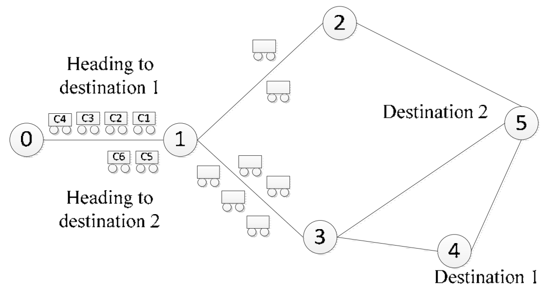
III. Backpressure Based Routing Algorithm

This section we propose a method that trades off between user satisfaction and traffic load. The algorithm is based on backpressure control theory [9-11] which is widely used in control system. Firstly, we apply the backpressure method into traffic system. And then we propose our algorithm to solve the problem in section II.

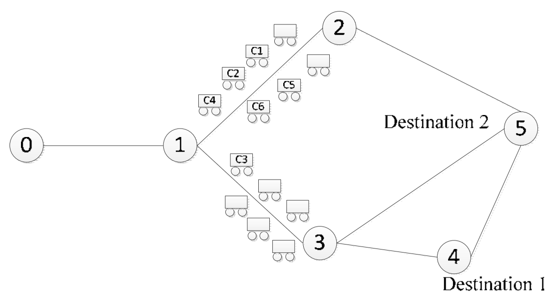
A. *Basic Idea of Backpressure Control*

Backpressure routing refers to an algorithm for dynamically routing traffic over a multi-hop network by using congestion gradients. The algorithm can be applied to wireless communication networks, product assembly systems and processing networks. In traffic system, we define the difference of car numbers in each link as traffic pressure, that is, the pressure between the link *i* and the link *j* is calculated as*:*

pressure[i,j]=car number of link i - car number of link j (16)



(a)



(b)

Figure 1. Traffic Backpressure Exmaple.

As we can see in Fig. 1(a), there are six cars on link (0,1), where two cars on link (1,2) and five cars on link (1,3). So the pressure between link (0,1) and link (1,2) is 4, while pressure between link (0,1) and link (1,3) is 1. The pressure stands for the spare capacity of a link. So we route the cars on link (0,1) to link (1,2) and link (1,3) as a proportion of 4:1. Then we get the result of traffic backpressure method showing in Fig. 1(b).

B. *Traffic Dispersion Method BPR*

After introducing the basic ideas in part A of section III, we introduce the Back Pressure Routing (BPR) algorithm applied in traffic system. The basic principle of BPR algorithm is that higher route probability with higher link spare capacity.

Supposed that a car named *C* is on the link *(p,i)*, *j* is adjacent intersection of *i*. The pseudo code of BPR algorithm is listed below.

BPR (Back Pressure Routing) Algorithm

Input: Car numbers in every adjacent link of intersection *i*, car *C*'s candidate route to its destination.
Output: Car *C*'s choosing probability of next intersection j, naming *pro[i,j]*.
Step 1: Get the set of *i*'s neighbor intersection *N(i)*.
Step 2: Get the subset of *N(i)*, which satisfy that its element is in *C*'s candidate route. The subset is called *S(i)*.
Step 3: For any $k \in S(i)$, calculate the pressure difference *bp[p,i,k]* between link *(p, i)* and link *(i, k)*.
Step 4: Assign cars in link *(p, i)* to choose adjacent link *(i, j)* with the probability $pro[i,j] \leftarrow bp[p,i,j]/(\sum_{k \in S(i)} bp[p,i,k])$

Because BPR algorithm does not take travel time cost into consideration, it cannot satisfy users' request. So the BPR-US algorithm which is based on BPR algorithm and considers user satisfaction is proposed.

C. *BPR-US Algorithm*

BPR-US (Back Pressure Routing with User Satisfaction) algorithm is an improved method of BPR. It considers user satisfaction and traffic load at the same time. The basic idea of BPR-US is that link choosing probability is the weight sum of link spare capacity and user urgent degree to the destination. The pseudo code of BPR-US algorithm is listed below.

BPR-US (Back Pressure Routing with User Satisfaction) Algorithm

---

Input: Car numbers in every adjacent link of intersection *i*, car *C*'s candidate route, *C*'s expecting arrival time, *C*'s current traveled time.
Output: Car *C*'s choosing probability of next intersection *j*, naming *pro[i,j]*.
Step 1: Get the set of *i*'s neighbor intersection *N(i)*.
Step 2: Get the subset of *N(i)*, which satisfy that its element is in *C*'s candidate route. The subset is called *S(i)*.
Step 3: For any $k \in S(i)$, calculate the pressure difference between link *(p, i)* and link *(i, k)*. The pressure difference is marked as *bp[p,i,k]*.
Step 4: Choosing adjacent link *(i, j)* with the backpressure probability: $bp\_pro[i,j] = bp[p,i,j]/(\sum_{k \in S(i)} bp[p,i,k])$
Step 5: Calculate *C*'s urgent degree to destination:
urgent ← min(1, traveled_time / expecting_time)
Step 6: If j is on shortest candidate path of C
*pro[i,j]← α*bp_pro[i,j]+(1-α)*urgent*
else
*pro[i,j]←α*bp_po[i,j]+(1-α)*(1-urgent)*

Take Fig. 1 for example. In Fig. 1(a), the car *C1* can get its destination (intersection 4) through link (1,2) or link (1,3). Supposed that *C1* has spent 85s on the road, and its expecting arrival time is 100s, so its urgent degree to the destination is *urgent*=85/100=0.85. The probability of *C1* choosing link (1,3) *p{next=3|current=1}=α*bp_pro(1,3)+(1-α)*urgent*=*α**0.2+(1-*α*)*0.85, because intersection 3 is on the shortest path from intersection 1 to intersection 4. The probability choosing link (1,2) is calculated as *p{next=2|current=1}=α*bp_pro(1,2)+(1-α)*(1-urgent)*= *α**0.8+(1-*α*)*0.15. We set the weighting factor *α*=0.5. Therefore, we have *p{next=3|current=1}*=0.525， while *p{next=2|current=1}*=0.475*,* then *C1* should have a higher probability to choose link (1,3) as the next link.

IV. Simulation Results and Analysis

In this section, we compare the shortest path algorithm, BPR algorithm and BPR-US algorithm from the perspective of average travel time, road density and user satisfaction.

A. Experiment Scene

We choose Singapore expressway mentioned in [5] and [6] to conduct our experiment. The Singapore expressway has 11 intersections and 28 links (both directions). The structure of the expressway is shown in Fig. 2.

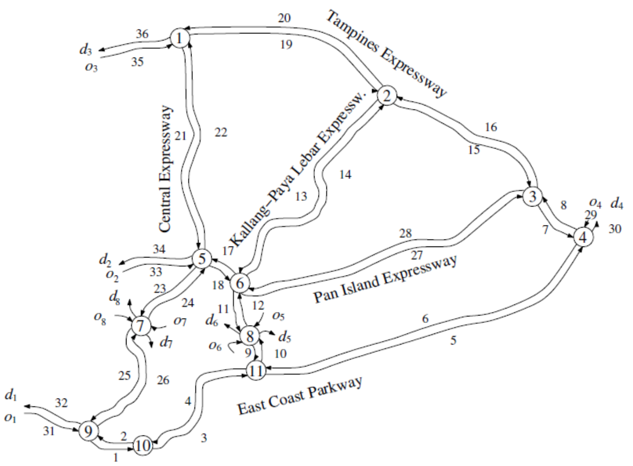


Figure 2. Singapore expressway(central and eastern parts)

Traffic simulation software "sumo" is used to carry our experiments. At first, 1000 cars are deployed on the expressway. These cars have different sources and destinations. The detail route of each car is described in Table I.

TABLE I. ROUTES OF CARS ON THE EXPRESSWAY

Vehicle ID	Group	Source (Intersection No.)	Destination (Intersection No.)	Route
0~99	1	9	1	9to10 10to11 11to8 8to6 6to5 5to1
100~199	2	10	1	10to11 11to8 8to3 3to2 2to1
200~299	3	11	2	11to4 4to3 3to2
300~399	4	9	2	9to7 7to5 5to6 6to2
400~499	5	5	3	5to1 1to2 2to3 3out
500~599	6	6	3	6to8 8to11 11to4 4to3
600~699	7	9	4	9to7 7to5 5to1 1to2 2to3 3to4
700~799	8	7	4	7to5 5to6 6to8 8to11 11to4
800~899	9	2	10	2to6 6to5 5to7 7to9 9to10
900~999	10	4	9	4to3 3to8 8to6 6to5 5to7 7to9

We pick every 5 cars from the vehicle ID sequence to implement the routing algorithm. Shortest path, BPR and BPR-US are applied in route choosing respectively. Average travel time, road density and user satisfaction are the evaluation indicators of the three routing algorithms.

B. Simulation Results

At first, we compare the results of average travel time cost of shortest path, BPR and BPR-US. Cars which participate in the route guidance in group 1, grout 2, grout 4, group 6, and grout 9 are chosen. We rename these groups as O-D (Origin-Destination) pair 1 to 5, as illustrated in Fig. 3.

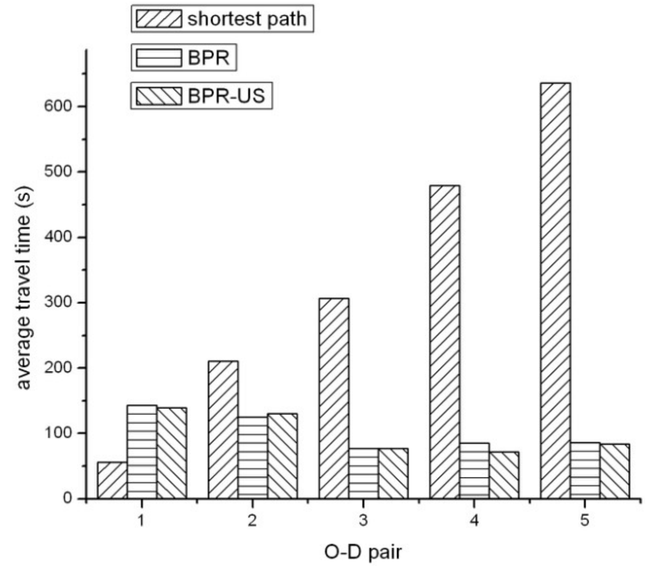


Figure 3. Shortest path, BPR and BPR-US performance comparison on average time cost

As shown in Fig. 3, BPR and BPR-US have a much better performance than shortest path algorithm. Because the shortest path algorithm usually picks main roads as the route result, which leads that all the guided cars travel to the same main road. Thus the road becomes congested, and cars suffer from high time cost. But if the congestion does not happen on the shortest path, the cars could arrive at their destinations very fast, just as the O-D pair 1 in Fig. 3 shows. BPR and BPR-US can disperse the cars into different roads. They avoid traffic jam happening in some degree, so the cars can get their destination faster. We can also conclude that time cost of BPR-US is less than the cost of BPR most of the time. This is because BPR is lack of time control. However, BPR-US raises the congestion risk, especially when the cars are hurry to their destination. So the cost of BPR-US is a little higher than the cost of BPR shown in O-D pair 2.

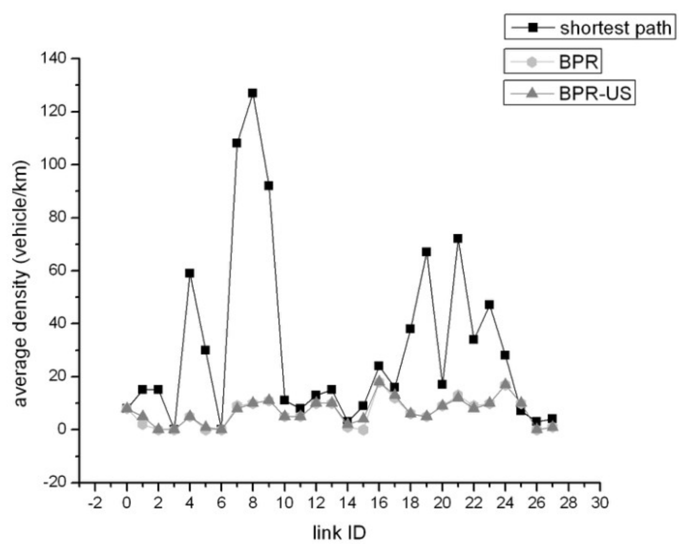


Figure 4. Shortest path, BPR and BPR-US performance comparison on road density

There are 28 links on the expressway. We have counted the density of each link for 1100s. Link average density is calculated according to (11). From Fig. 4, it is clear to demonstrate that BPR and BPR-US could average the traffic density to all links while shortest path algorithm could not. Shortest path algorithm leads to a peak density on the main road, and the density of each link is of great difference. As for BPR algorithm and BPR-US algorithm, they both have great and similar performance on road density, in other words, they both reduce the traffic load effectively.

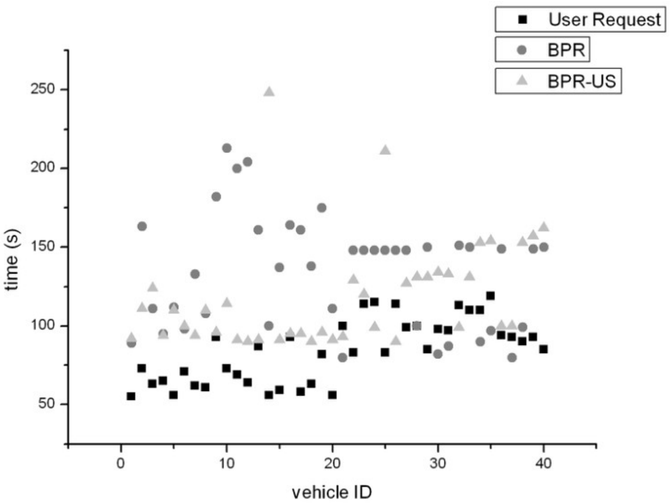


Figure 5. Shortest path, BPR and BPR-US performance comparison on user satisfaction

Every user has own travel time demand to arrive the destination. Shortest path algorithm tries to provide a route with shortest time. It is also a way to consider user satisfaction. BPR algorithm takes no notice of the user time request, while BPR-US can consider user demand especially when the expecting arrival time is getting close. Fig. 5 is the comparison of BPR algorithm and BPR-US algorithm in the performance at user satisfaction.

In Fig. 5, vehicle IDs from 1 to 20 and IDs from 21 to 40 are from O-D pair 1 and O-D pair 2 in Fig. 3 respectively. The black point is the user expecting travel time. From the compare result, we can see that BPR-US algorithm can better fit user request better than BPR does for most of the time, although sometimes the user demand is difficult to meet. As for ID 14 and ID 25, they ask for less time to get their destination in their own group. Then BPR-US provides them the shortest path with a high probability. It is easy to route the cars to a main road with high density. That is the reason why the time cost of BPR-US is higher than the one of BPR for vehicle 14 and 25. Because of the existing of this kind of cars, the average time cost of BPR-US even higher than the cost of BPR, just as the O-D pair 2 in Fig. 3 shows.

V. CONCLUSIONS AND FUTURE WORK

This paper analysis the disadvantage of current traffic routing guidance system, and points out it is necessary to combine the user satisfaction and traffic load at the same time. Then we define the “user satisfaction-traffic load” problem. After that, backpressure control method is applied in traffic system and we propose the BPR-US algorithm to solve the problem mentioned before. Finally, simulation experiments are carried out to prove the effectiveness of BPR-US in meeting user demand and averaging road density.

Because the intersection can only control the area of its own, and they cannot share backpressure information from each other, BPR-US algorithm would stuck into the local optimization sometimes. Our future work will focus on providing a coordination control method to solve the local optimization problem.

ACKNOWLEDGMENT

The paper is sponsored by National Natural Science Foundation of China (No. 60803148 and No. 60973124).

REFERENCES

[1] Yang Jianren, “A Dynamic Route Guidance Algorithm Based on Modified Ant Colony Optimization,”. International Symposium on Computer Network and Multimedia Technology, pp. 1-3, 2009.

[2] Sung-Soo Kim and Jong-Hyun Lee, “A study on design of dynamic route guidance system using forecasted travel time based on GPS data and modified shortest path algorithm,” IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp. 44-48, 1999.

[3] Peng Chen, Zhao Lu and Junzhong Gu, “Vehicle Travel Time Prediction Algorithm Based on Historical Data and Shared Location,” Fifth International Joint Conference on INC, IMS and IDC, pp 1632 – 1637, 2009.

[4] Chen Qun. “Dynamic Route Guidance Method Based on Particle Swarm Optimization Algorithm,” Second International Conference on Intelligent Computation Technology and Automation, pp. 267 – 270, 2009.

[5] Diogo Alves, Jelmer van Ast, Zhe Cong, Bart De Schutter and Robert Babuška. “Ant Colony Optimization for Traffic Dispersion Routing,” 13th International IEEE Conference on Intelligent Transportation Systems(ITSC), pp. 683 – 688, 2010.

[6] Zhe Cong, Bart De Schutter and Robert Babuška. “A New Ant Colony Routing Approach with a Trade-Off between System and User Optimum,” 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1369–1374, 2011.

[7] B. D. Greenshields, “A study of traffic capacity,” Highway Research Board Proceedings, vol. 14, pp. 448-477, 1935.

[8] Zhihua Xiong, Chunfu Shao and Zhisheng Yao. “Fuzzy Travel Time Reliability with Uncertainty,” Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 6, pp. 277 – 280, 2009.

[9] Lei Ying, Sanjay Shakkottai, Aneesh Reddy and Shihuan Liu, “On combining shortest-path and back-pressure routing over multihop wireless networks,” Proceedings of IEEE INFOCOM 2009, pp. 1674 – 1682, 2009.

[10] Tassiulas L., Ephremides A., “Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput,”. Proceedings of the 29th IEEE Conference on Decision and Control, vol. 4, pp. 2130 – 2132, 1990.

[11] Bo Ji, Changhee Joo and Shroff N.B., “Delay-based Back-Pressure scheduling in multi-hop wireless networks,” Proceedings of IEEE INFOCOM 2011, pp. 2579 – 2587, 2011.