

# Efficient Route Guidance in Vehicular Wireless Networks

Yu Stephanie Sun\*, Lei Xie\*, Qi Alfred Chen†, Sanglu Lu\*, Daoxu Chen\*

\*State Key Laboratory for Novel Software Technology, Nanjing University, China

†University of Michigan, USA

Email: \*sunny0401@gmail.com, \*lxie@nju.edu.cn, †alfchen@umich.edu, \*{sanglu,cdx}@nju.edu.cn

**Abstract**—With the rapid proliferation of Wi-Fi technologies in recent years, it has become possible to utilize the vehicular wireless network to assist the route guidance for drivers in a cooperative approach, aiming to mitigating heavy traffic congestion. In this paper, we investigate into the route guidance problem in vehicular wireless network, and then propose two efficient routing algorithms, i.e., centralized route guidance and distributed route guidance, according to different situations. A hybrid framework is then proposed to provide optimized routing decisions in a uniform way. Simulation results in Simulation of Urban MObility (SUMO) indicate that, our route guidance schemes achieve much better performance than traditional GPS-based navigation and randomized routing.

**Index Terms**—Route Guidance; Vehicular Wireless Networks; Energy-Efficiency; Optimization

## I. INTRODUCTION

It is well known that traffic congestions can cause serious problems, such as fuel consumption, air pollution and even economic problems. From Urban mobility report [1], the cost from traffic congestion now is more than \$100 billion, nearly \$750 for every commuter in the U.S. To mitigate this situation, effective route guidance system should be deployed, helping vehicles choosing faster routes to avoid congestions. Traditional route guidance schemes leverage GPS module in vehicles to find the shortest path from the source to destination. However, these methods offer limited help for the current congestion situations, which is mainly because: first, they are unaware of the real-time congestion situations; second, when all vehicles with the same requests are guided to the same shortest paths, this shortest path will face severe congestions and become far from the fastest path. To better solve the congestion problem, a new route guidance system which overcomes these limitations should be designed.

Vehicular wireless network provides opportunities to design a more effective vehicle routing scheme than before to avoid traffic congestions. In this architecture, road-side access points (AP) are widely deployed [2] [3], which can provide wireless access to users in moving vehicles and support data sharing among drivers. Therefore, by sharing dynamic traffic information in this network, it is possible to mitigate the congestion problems and further reduce the travel time of drivers.

In this paper we study the route guidance in vehicular networks, and propose two efficient routing algorithms according to different conditions. The contributions of this paper are summarized as follows,

- 1) Based on the single-source single-destination pattern of the routing requests during peak time, we propose a centralized algorithm, Minimum-Cost Maximum-Flow based Routing (MCMF-R), for the route guidance problem, which can make routing decisions for a group of

vehicles at one time instead of a single one. MCMF-R is designed for heavy traffic such as situation in rush hours. With a central control unit, the routing of MCMF-R can make full use of global traffic information.

- 2) In order to mitigate the possibility of high communication and computation overhead, we propose a distributed algorithm, Traffic Splitting (TS), for the route guidance problem, which only uses local traffic information to help make routing decisions. TS requires few computing and communication resources, thus it can be processed in parallel and more suitable for practical usage.
- 3) We propose a hybrid framework combining MCMF-R and TS, which are mutually complementary to each other. Using the realistic traffic generator Simulation of Urban MObility (SUMO) [4], we evaluate our algorithms in a real-world traffic map, and the result shows that our solutions can reduce the average travel time by 40% than traditional ones.

## II. RELATED WORK

Common vehicular wireless network architecture consists of road side units (access point) which communicate with travelling vehicles and provide information sharing service in local area. Based on this basic design, current research work mainly focus on the routing protocols [5], access association control [6] and information sharing [7].

Traditional route guidance systems are based on GPS module and shortest path algorithm [8]. Realizing their limitation of lacking real-time traffic information, recent research work focus on employing new architectures such as neural network [9], Wireless Sensor Network (WSN) [10] to enhance the information accuracy and routing efficiency. For vehicular wireless network, the infrastructureless route guidance system [11] have been studied in [12] [13], which mainly rely on the inter-communication between vehicles to share information. For infrastructure-based system, latest research work can be found in problems such as travel time prediction [14], congestion avoidance [15], etc. Different from previous work, in this paper we focus on the problem of utilizing road-side APs in the current vehicular wireless network architecture to find the fastest routes at the presence of congestion, and propose two algorithms according to different traffic conditions.

## III. PROBLEM FORMULATION

In a typical traffic routing scenario, vehicles in need of route guidance will send their routing requests including the sources and destinations inside an area to the route guidance system, and wait for the routing result. The area map can be denoted as a directed graph  $G(V, E)$ , where  $V$  and  $E$  are respectively the set of the intersections and the set of roads. We denote  $C$  as the set of vehicles with routing requests. For each vehicle  $c_i \in C$ ,

its routing request contains source  $s_i$  and destination  $d_i$ , where  $s_i \in V, d_i \in V$ . For road  $\langle u, v \rangle \in E$ ,  $M(u, v)$  denotes the capacity of  $\langle u, v \rangle$ , i.e., maximum number of vehicles available for travelling on it.

In regard to the routing result, the routing guidance system will reply to vehicle  $c_i$  with a route  $R_i$  in the form of  $\langle v_0, v_1, \dots, v_k \rangle$ , where  $v_0 = s_i, v_k = d_i$  and  $v_j \in V, j = 0 \dots k$ . If we use  $R_{s_i, d_i}$  to represent the set of all routes from  $s_i$  to  $d_i$ , then  $R_i$  should be in the set  $R_{s_i, d_i}$ .

The routing target is to make the overall/average travel time of all vehicles to be minimum. So the objective of the route guidance problem can be represented as follows,

$$\text{minimize } \sum_{c_i \in C} T(R_i) \quad (1)$$

subject to

$$R_i \in R_{s_i, d_i}, \quad (2)$$

$$\forall \langle u, v \rangle \in E, |\{c_i : c_i \in C, \langle u, v \rangle \in R_i\}| \leq M(u, v). \quad (3)$$

where  $T(\cdot)$  is the time duration of route  $R_i$ . The second constraint requires that the traffic flow on each road  $\langle u, v \rangle$  should be no greater than its capacity.

In order to solve this route guidance problem, in this paper we consider two situations based on the infrastructures in vehicular network: centralized route guidance and distributed route guidance. In the centralized solution, the routing decision is made in a central control module and requires global traffic information. In the distributed solution, there is no central controlling and the algorithm only uses local traffic information for route guidance.

#### IV. SOLUTION

##### A. System Architecture

Fig.1 shows an example of route guidance system architecture in vehicular networks. In this figure, the components are sinks, vehicle-side sensors and a central control unit. Sinks are usually APs in vehicular network, and as discussed in [2], these sinks keep communicating with sensors in the vehicles such as GPS, cameras, mobile phones, etc. The central control unit will exchange data with sinks, but it will only be used in centralized routing. Within this infrastructure, the real-time traffic data and route requests will be shared, and we can utilize these resources to devise efficient route guidance algorithms.

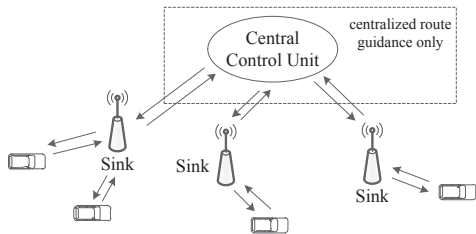


Fig. 1: An example of route guidance system architecture in vehicular network

In centralized routing, the central control unit collects traffic information globally. In this way, not only the static traffic information such as the red light duration time, road length, speed limit, etc., but also the dynamic traffic information like real-time traffic movements can be used in the routing decision for each vehicle. To handle such large amount of data, the central control unit is expected to have abundant storage resources along with powerful computing devices.

On the other hand, distributed routing makes decision locally. There is no central control unit, so the routes are computed locally in the nearby sinks. The dynamic traffic information available is limited to a small range, e.g., adjacent roads. For static traffic information, it can be pre-deployed and thus be global. Therefore, distributed route guidance scheme has a much smaller overhead in storage and computing, and thus more deployable. However, since it lacks global traffic information, its solution is more likely to be a local optimum one. This architecture can also be applied to Vehicular Ad hoc NETWORKS (VANET) since we can choose any point in the ad-hoc network as the sink.

##### B. Analysis

Recall that  $T(\cdot)$  is the key parameter in Eq. (1), in order to measure it, we first explore the travel time a vehicle spends around an intersection. Assume that a vehicle is travelling on road  $\langle u, v \rangle$  and will turn to road  $\langle v, w \rangle$ , the time duration has two parts: the driving time  $D(u, v)$  and the waiting time  $W(u, v, w)$ , which is represented as follows,

$$T(\langle u, v, w \rangle) = D(u, v) + W(u, v, w) \quad (4)$$

$$D(u, v) = \frac{D_0(u, v)}{S(u, v)} \quad (5)$$

In Eq.(4), the driving time refers to the time the vehicle spends on the road without being influenced by other vehicles or traffic lights. As shown in Eq.(5), it can be calculated using distance  $D_0(u, v)$  and speed limit  $S(u, v)$  of road  $\langle u, v \rangle$ .

In regard to the waiting time, we measure it from the number of red lights the vehicle is expected to wait. According to the assumption, the vehicle is on road  $\langle u, v \rangle$  to  $\langle v, w \rangle$ , the number of red lights it has to wait depends on both the set of current vehicles before it,  $F(u, v, w)$ , and the number of vehicles which can leave the road during one green light,  $L(u, v, w)$ . We try to find the upper bound of the waiting time in our measurement as shown in Eq.(6).

$$W(u, v, w) = \left( \left\lfloor \frac{|F(u, v, w)|}{L(u, v, w)} \right\rfloor + 1 \right) \cdot R(u, v, w) \quad (6)$$

$$L(u, v, w) = \frac{G(u, v, w) \cdot S(u, v)}{l} \cdot \rho(u, v, w) \quad (7)$$

In Eq.(6) and Eq.(7),  $R(u, v, w)$  and  $G(u, v, w)$  represent the time of one red light and one green light when turning from road  $\langle u, v \rangle$  to  $\langle v, w \rangle$  at intersection  $v$ .  $l$  is the sum of both (average) vehicle length and the gap between two vehicles.  $\rho(u, v, w)$  denotes the density of vehicles during one green light, which is in the range of  $[0, 1]$ .

With Eq.(4),  $T(R_c)$  can be measured by adding up the travel time of all road-to-road segments in  $R_c$ .

##### C. Baseline Routing Algorithm

To evaluate the route guidance systems, it is essential to construct a baseline routing algorithm whose ability is similar to the routing without any guidance. However, for vehicle set  $C$  from  $s$  to  $d$ , simply randomly selecting routes from  $R_{s, d}$  uniformly is not reasonable, since by using experiences or maps, the majority of drivers' routing decisions without any guidance are more likely to be around the shortest path.

Based on the above understanding, we propose an improved random routing algorithm, IRR for short. In IRR, we first choose the shortest path from  $s$  to  $d$ . From the second path, to mimic the imperfection of human beings, we randomly delete

an edge in the previous path, then find the shortest path again as a new path. This process will continue until the needed number of paths is reached. In this way, these routes are very close to the shortest path but not strictly the top  $|C|$  shortest routes. IRR combines the proximity to the shortest path and imperfection of human beings, which can be viewed as the random routing decisions made by some local taxi drivers.

#### D. Centralized Solution

1) *Motivation:* When the traffic is not heavy, simply using classical guidance techniques such as GPS can handle all routing requests without any congestion. So the centralized route guidance in this paper focuses on the situation when the traffic is very heavy, where large number of vehicles needs to travel on the main roads and severe traffic jams are likely to take place. In these cases, the traffic condition takes on a single-source single-destination pattern. In this way, the two targets of our centralized solution are listed as follows,

- 1) Maximize the number of vehicles that can reach the destination
- 2) Minimize the total time duration from source to destination (refer to the objective in problem formulation)

In this way, we can solve as many routing requests as possible, while sufficiently reducing the overall travel time of vehicular users at the same time.

For vehicle set  $C$  to travel from  $s$  to  $d$ , the routing problem can be represented as Fig.2. If we treat traffic as the flow and the travel time as the cost, centralized route guidance problem with single source and single destination can be solved as a minimum-cost maximum-flow problem, which is the core idea in the centralized solution.

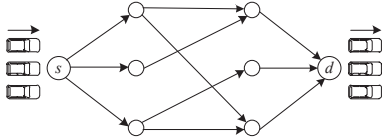


Fig. 2: Centralized routing problem can be reduced to a routing problem from source  $s$  to destination  $d$

#### 2) Minimum-Cost Maximum-Flow based Routing:

a) *Minimum-Cost Maximum-Flow Problem:* In the minimum-cost maximum-flow problem, there is a directed graph  $G\langle V, E \rangle$  with source  $s \in V$  and destination (sink)  $d \in V$ , edge  $(u, v) \in E$  has capacity  $c(u, v) > 0$ , flow  $f(u, v) \geq 0$  and cost  $cost(u, v) \geq 0$ . The total flow is  $\sum_{u \in V} f(s, u)$  and the cost of sending this flow is  $\sum_{(u, v) \in E} f(u, v) \cdot cost(u, v)$ .

The objective of this problem is to find the cheapest possible way of sending a certain amount of flow through a flow network. The solution used in this paper [16] can be viewed as a generalization of the maximum-flow algorithm called Ford-Fulkerson algorithm. The main idea is that while the original Ford-Fulkerson algorithm is seeking an augmenting path, this solution aims to find the one with the minimum cost instead.

b) *Minimum-Cost Maximum-Flow based Routing:* Based on the analysis in Section IV-D1, we propose Minimum-Cost Maximum-Flow based Routing, MCMF-R for short, to solve the centralized route guidance problem. The algorithm is shown in Algorithm 1, where  $MCMF(\cdot)$  means calling the minimum-cost maximum-flow problem solution [16].

- **Flow and Capacity.** In our routing problem, we regard the traffic flow as  $f(u, v)$ , so  $f(u, v) = \sum_{(v, w) \in E, w \neq u} |F(u, v, w)|$ . Capacity  $c(u, v)$  is similar to  $M(\cdot)$ , and is calculated as the maximum number

of vehicles available on the road (shown in line 2 in Algorithm 1, where  $D(\cdot)$  and  $l$  is introduced in Section IV-B,  $L_0(u, v)$  is the number of lanes in road  $\langle u, v \rangle$ ).

- **Cost.** The  $cost(\cdot)$  for minimum-cost maximum-flow problem is the travel time  $T(\cdot)$  for route guidance problem. As shown in Eq.(4), it will be a function of the traffic flow. However, compared to  $cost(\cdot)$ ,  $T(\cdot)$  has an extra parameter  $w$ . To solve this problem, we can first replace  $w$  with a new parameter, the former intersection  $t$  before entering edge  $\langle u, v \rangle$ .  $cost(\cdot)$  will only be used when calculating the shortest path in  $MCMF(\cdot)$ . So if using popular shortest path algorithm such as Dijkstra, SPFA and etc.,  $t$  should be already known when in need of value  $cost(u, v)$ . In this way, value of  $T(\langle t, u, v \rangle)$  can be used as  $cost(u, v)$  in  $MCMF(\cdot)$ .
- **Extracting Paths.** From the result of  $MCMF(\cdot)$ , we can execute graph traversals from source to destination, and extract travel paths one by one until there is no positive flow on any edges. These travel paths then will be assigned to the vehicles with routing request as their routing results.

---

#### Algorithm 1 MCMF-R Algorithm

---

**Input:** routing requests  $\langle s_i, d_i \rangle$  for each  $c_i \in C$

**Output:**  $R_i$  for  $c_i$

- 1:  $\forall \langle u, v \rangle \in E, f(u, v) = 0$ ;
  - 2:  $\forall \langle u, v \rangle \in E, c(u, v) = \frac{D(u, v)}{T} \cdot L_0(u, v)$ ;
  - 3: partition set  $C$  into  $C_j, j = 1, 2, \dots$ , where  $c_i \in C_j$  share the same routing request  $\langle s_j, d_j \rangle$
  - 4: **for all**  $C_j$  **do**
  - 5:     call  $MCMF(s_j, d_j, C_j, cost(\cdot))$  to update  $f(\cdot)$ ;
  - 6: **end for**
  - 7: For each  $c_i \in C$ , extracting path  $R_i$  from  $f(\cdot)$ ;
- 

c) *Multi-Source Multi-Destination Routing:* In the real world, the routing requests will often have multiple sources and multiple destinations. Since minimum-cost maximum-flow algorithm is designed for single source and single destination, before using it we need to first group the vehicles with the same routing request (same  $s_i$  and  $d_i$ ). By doing so, the original vehicle set  $C$  is partitioned into small subsets,  $C_j, j = 1, 2, \dots$ . Vehicles in the same subset will have the same source and destination (line 3 in Algorithm 1). Then we can use minimum-cost maximum-flow algorithm for each subset, as shown in line 4 to 6 in Algorithm 1. When processing, larger subsets will have higher priority in the order. Besides grouping same routing requests, if the sizes of the groups are too small to have good routing performances, we can also merge the sources/destinations which are near to each other geographically into one common source/destination to form larger subsets.

3) *Analysis:* The MCMF-R is meant to achieve the two targets listed in Section IV-D1, which satisfies the objective in Section III. So if the travel time estimation is accurate enough, the routing result of MCMF-R will be exactly the same as the theoretical optimal solution in problem definition. By using MCMF algorithm, MCMF-R can make routing decisions for a vehicle flow at one time instead of a single vehicle, which increases the decision speed by the factor of the flow size.

In multi-source multi-destination routing part, the order of processing vehicle subsets is critical for routing efficiency and fairness. The earlier the subset is fed into  $MCMF(\cdot)$ , the shorter path it will get. So the best way to minimize the overall time duration is to process bigger size subset first. For fairness, this also ensures that the majority of vehicles have the priority to choose better routes.

### E. Distributed Solution

1) *Motivation:* Centralized route guidance solution above requires large storage space and powerful enough computing devices for handling global traffic information and making routing decisions for all requests. However, these resources are sometimes rather limited. Therefore, a distributed solution is essential to be proposed, which is able to make route decision with limited traffic information. As discussed in Section IV-A, the distributed solution can only obtain the dynamic traffic information in adjacent roads collected in one AP.

In  $G(V, E)$ , we denote the vehicles travelling on  $\langle u, v \rangle$  to be a vehicle set  $C_{\langle u, v \rangle}$ . For the vehicles in  $C_{\langle u, v \rangle}$ , when the sink at  $v$  processes their routing requests,  $v$  will be their common start point. So single source shortest path (SSSP) algorithms can be used in the sink at  $v$  for routing, serving as a feasible distributed routing algorithm. Traditional GPS route guidances are mainly based on this idea. However, if every vehicle travel in their shortest paths, extremely heavy traffic jams could take place, and the resulted solutions for each vehicle are very likely to largely deviate from the optimal one (shown in Section V-C).

This huge difference between the expected result and the actual result is due to the missing information of dynamic traffic. With the sinks' dynamic traffic information about adjacent roads, it will be a wise choice to split the traffic to different roads around an intersection and thus avoid traffic jams on one single road.

2) *Traffic Splitting:* For the intersection  $v$ , we assume that there are  $p$  roads adjacent to it, denoted by  $\{\langle v, w_i \rangle | i = 1 \dots p\}$ . Our target is to find a strategy to split the traffic  $C_{\langle u, v \rangle}$  into vehicle sets  $f_i, i = 1 \dots p$  where  $\cup f_i = C_{\langle u, v \rangle}$ . For each vehicle in set  $f_i$ , the next intersection of their routing result will be  $w_i$ . In other words, after these vehicles pass road  $\langle u, v \rangle$ , they will turn to road  $\langle v, w_i \rangle$ .

For a vehicle  $c_k$  which travels on  $\langle u, v \rangle$  to  $d_k$ , if it is classified into set  $f_i$ , setting  $w_i$  as the next intersection of the routing solution should be less time-consuming than any other intersections in the set  $\{w_j : j \in \{1 \dots p\} - \{i\}\}$ . This can be described as follows,

$$T(\langle u, v, w_i, \dots, d_k \rangle) = \min\{T(\langle u, v, w_j, \dots, d_k \rangle)\} \quad (8)$$

where  $i, j = 1 \dots p$ .

In Eq.(8),  $T(\langle u, v, w_i, \dots, d_k \rangle)$  is equal to

$$T(\langle u, v, w_i \rangle) + D(v, w_i) + T(\langle w_i, \dots, d_k \rangle) \quad (9)$$

where  $i = 1 \dots p$ , and  $D(\cdot)$  is the drive time.

In Eq.(9), the time duration from  $u$  to  $d_k$  through  $\langle u, v \rangle$  and  $\langle v, w_i \rangle$  consists of the travel time on road  $\langle u, v \rangle$  and the travel time from road  $\langle v, w_i \rangle$  to  $d_k$ . With the dynamic traffic information of adjacent roads  $F(u, v, w_i)$ , we can calculate  $T(\langle u, v, w_i \rangle)$  using Eq.(4). For the travel time from road  $\langle v, w_i \rangle$  to  $d_k$ , we lack the dynamic information to get a better estimation. So we calculate it using the sum of drive time,  $D(v, w_i)$  defined in Eq.(5), and the travel time of the shortest path from  $w_i$  to  $d_k$ , denoted as  $T(\langle w_i, \dots, d_k \rangle)$ .

In the sink at  $v$ , for each  $c_k \in C_{\langle u, v \rangle}$  and each intersection  $w_i, i = 1 \dots p$  and  $w_i \neq u$ , we have  $T(\langle u, v, w_i, \dots, d_k \rangle)$ . Then we choose the  $w_i$  with the smallest  $T(\langle u, v, w_i, \dots, d_k \rangle)$  to be the next step of  $c_k$ , and add  $c_k$  to  $f_i$ . After all vehicles in  $C_{\langle u, v \rangle}$  have been processed, the traffic splitting of the sink at  $v$  finishes. In the next intersection  $w_i$ , sink at  $w_i$  will carry out the same traffic splitting process, and guide  $c_k$  to the next intersection. This routing will relay in each intersection, and  $c_k$  will finally reach  $d_k$ .

The traffic splitting algorithm in  $v$  discussed above is shown in Algorithm 2. In the algorithm, all edges adjacent to  $v$  should be deleted temporarily. After this operation, it is ensured that the shortest path generated in line 2 to 4 will not include point  $v$ , so that the calculation result excludes the cases that the vehicle travels back to  $v$  after leaving  $v$ .

---

#### Algorithm 2 Traffic Splitting (TS) Algorithm

---

**Input:**  $u, v, C_{\langle u, v \rangle}, w_1 \dots w_p$

**Output:** vehicle sets  $f_1 \dots f_p$

```

1:  $f_i = \emptyset, i = 1 \dots p$ ;
2: for  $i = 1 \dots p$  do
3:   calculate the shortest path from  $w_i$  to all other inter-
   sections using SSSP algorithm such as Dijkstra;
4: end for
5: while  $\sum_{i \in 1 \dots p} |f_i| < |C_{\langle u, v \rangle}|$  do
6:   for all  $c_k \in C_v$  do
7:      $m \in \arg \min_{i \in 1 \dots p} T(\langle u, v, w_i, \dots, D_k \rangle)$ ;
8:      $f_m = f_m \cup \{c_k\}$ ;
9:     update  $T(\langle u, v, w_m, \dots, D_k \rangle)$ ;
10:  end for
11: end while

```

---

We name this algorithm Traffic Splitting Algorithm, TS for short. Its complexity is in  $O(pn^2 + pn)$ , where  $p$  and  $n$  are the number of  $v$ 's adjacent roads and intersections respectively. TS only requires two message exchanges between vehicles and the sink, so the communication overhead is a linear function of the vehicle number, which is quite lightweight.

3) *Analysis:* For the estimation of  $T(\langle w_i, \dots, D_k \rangle)$ , TS uses the shortest paths, which includes no dynamic traffic information. This will lead to mistakes in choosing roads on line 7 to 8 in Algorithm 2, and as we stated before, the difference between the routing of shortest path and reality can be very huge. However, this is actually the limitation from the local information assumption in Section IV-E1. Since the limitation of shortest path estimation will increase with the congestion condition, the performance of TS will degrade when the traffic is heavy.

When splitting traffic in line 6 to 10, the vehicle  $c_k$  which is chosen at an earlier time is more possible to be designated into a shorter path. Considering this, the order of choosing  $c_k$  from  $C_v$  introduces some unfairness in our algorithm. To solve it, we can use some metric such as distance to destination, urgent level of the request, etc. to rank the vehicles in  $C_v$ . For simplification, in the paper all vehicles is considered to have no difference with each other.

4) *Example of Traffic Splitting Algorithm:* For better illustration of TS algorithm, Fig.3 gives an example of the traffic splitting process in one intersection. In the figure, intersection  $v$  has 4 adjacent roads. There are 4 vehicles travelling on road  $\langle u, v \rangle$  and requesting routing to common destination  $d$ , and their next intersection can be  $w_1, w_2$ , or  $w_3$ . Table I gives the initial parameters for routing these vehicles. After the calculation of SSSP algorithm for 3 times, we obtain all three  $T(\langle w, \dots, d \rangle)$  values in table I.

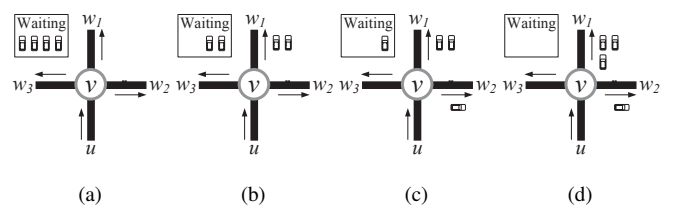


Fig. 3: Example of the Traffic Splitting (TS) algorithm process

As shown in Fig.3a, at the beginning 4 vehicles are placed in the waiting area waiting for routing, and  $f_1, f_2, f_3$  are all empty. Using Eq.(9),  $T(\langle u, v, w_1, \dots, d \rangle) = T(\langle u, v, w_2, \dots, d \rangle) = 9$ , which is the smallest among the 3 candidate roads. Thus we pick  $f_1$  to add the first vehicle. Since  $L(u, v, w_1) = 2$ , the time of waiting red light will increase  $R(u, v, w_1) = 2$  after adding 2 vehicles. So with the same analysis as above, we can add another vehicle into set  $f_1$  (Fig.3b). Now  $f_1$  has traffic volume 2 and  $T(\langle u, v, w_1, \dots, d \rangle) = 11$ , which is no longer the smallest. Now  $T(\langle u, v, w_2, \dots, d \rangle) = 10$  becomes the smallest, and we can add 1 vehicle to  $f_2$  (Fig.3c) and after the value increases to 12. Now  $T(\langle u, v, w_1, \dots, d \rangle) = 11$  is the smallest again, so in Fig.3d the last vehicle in the waiting area is added to  $f_1$ .

In this way, we obtain the routing solution: 3 vehicles travel towards  $w_1$  and 1 vehicle travels towards  $w_2$ .

TABLE I: Parameters Related to the Routing Example in Fig.3

intersection w	$w_1$	$w_2$	$w_3$
$T(\langle u, v, w \rangle)$	2	1	1
$T(\langle w, \dots, d \rangle)$	6	7	12
$D(v, w)$	1	2	1
$L(u, v, w)$	2	1	1
$R(u, v, w)$	2	2	1

### F. Comparison and Hybrid Route Guidance Framework

We summarize the performance of MCMF-R, TS, IRR and traditional GPS routing in Table II. For the time efficiency of the two solutions proposed in this paper, MCMF-R is closer to the optimal solution, but imperfect measurement of travel time makes it underestimate or overestimate the current congestion situation to some extent. TS becomes much worse than MCMF-R when heavy traffic take place. But since it's greedy, it is better than MCMF-R in light traffic condition, where the latter overestimates the congestion. In fairness, TS is better since MCMF-R considers more global optimization, which needs some vehicles to sacrifice. MCMF-R needs global dynamic traffic information and central control, so it has bigger communication overhead compared to other algorithms.

TABLE II: Comparison of MCMF-R, TS, IRR and Traditional GPS Routing

Algorithm	Overhead	Fairness	Time Efficiency	
			Light Traffic	Heavy Traffic
MCMF-R	hard	medium	medium	strong
TS	medium	high	strong	medium
IRR	easy	medium	medium	weak
GPS Routing	easy	low	medium	very weak

### Algorithm 3 Hybrid Route Guidance Framework

**Input:** routing requests  $req$ , current traffic condition  $tc$

- 1: **if** isRushHour( $req, tc$ ) **then**
- 2:     use MCMF-R
- 3: **else**
- 4:     use TS
- 5: **end if**

Considering that MCMF-R has high time efficiency for heavy traffic and TS has low overhead and better fairness for other traffic conditions, we combine them and propose a hybrid framework for route guidance according to different traffic pattern, which are shown in Algorithm 3. In normal period, TS can serve most of the routing requests with high efficiency and low overhead. When the system detects rush hour pattern, the routing algorithm should then be switched to MCMF-R. The hybrid framework balances routing overhead and time efficiency, providing both better deployment and routing decisions than IRR and traditional GPS under different traffic intensity situations.

## V. PERFORMANCE EVALUATION

### A. Evaluation Method

In order to test the routing performances of MCMF-R and TS, we use the realistic traffic generator Simulation of Urban MObility (SUMO) [4] to construct the large road network and generate vehicle traffic. In the simulation, we randomly generate routing requests as input, and then write C++ programs of the route guidance algorithm to make routing decisions for them. To test the performance in heavy traffic condition, we add background traffic in some experiments. After that, we use these routing decisions to generate vehicle traffic in SUMO, and record the travel time for each vehicle.

With the travel time, our evaluation is in two dimensions:

- Time Efficiency: measured using the average travel time of the vehicles.
- Fairness: measured using the standard deviation of the travel time of the vehicles.

### B. Simulation Parameter

For the road topologies, we construct the road network based on the map of Nanjing city, China. We extract the central part of Nanjing from openstreetmap [17], and convert it to the map used in SUMO. For convenience, we assume that vehicles can turn to any roads adjacent to the road they travel on. For traffic light setting, since the extracted map from openstreetmap does not have traffic light information, we manually set the traffic lights at all intersections to be 30 seconds red light duration and 30 seconds green light duration.

TABLE III: Setting of the Vehicles in the Simulation

Parameter	Value	Description
accel	$10(m/s^2)$	acceleration ability of vehicle
decel	$10(m/s^2)$	deceleration ability of vehicle
sigma	0	driver imperfection (between 0 and 1)
length	$3(m)$	The vehicle's length
minGap	1	minimal gap between vehicles
maxSpeed	$80(m/s)$	vehicle's maximum velocity

The parameters about the vehicles are listed in table III. For routing request generation, we first randomly generate 200 source and destination pairs, and then assign random vehicle volumes to them. For background traffic, we use repeated vehicles in SUMO, which have the repeat period to be 2.

### C. Simulation Result and Analysis

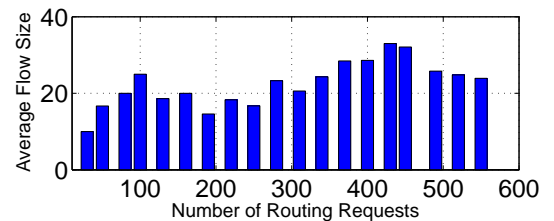


Fig. 5: Average flow size distribution of MCMF-R algorithm

In this section, we will evaluate the performance of centralized routing algorithm MCMF-R and distributed routing algorithm TS. Besides IRR algorithm, we also use the traditional GPS routing as the comparison algorithm, which keeps using shortest path algorithm in routing. Fig.5 shows the traffic grouping performance of MCMF-R, and the simulation result about time efficiency and fairness is shown in Fig.4.

Fig.5 shows the average vehicle flow size managed by MCMF-R for different routing request volumes of one source and destination pair. Between volume 0 to 300, the average flow size is around 15 and between 300 to 550, it increases to 25. This indicates that MCMF-R algorithm can achieve around

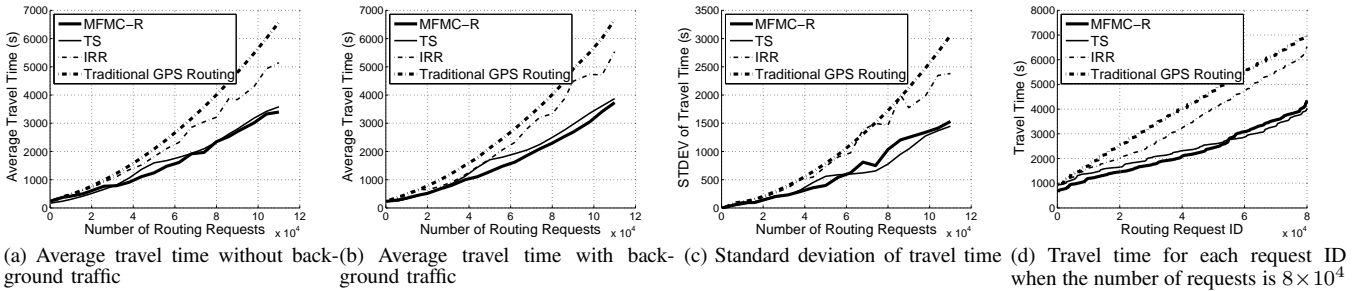


Fig. 4: Time efficiency and fairness comparison among MCMF-R, TS, improved random routing and traditional GPS routing

20 × faster routing decision by considering a group of vehicles at one time.

The first two figures in Fig.4 show the time efficiency comparison result for these four algorithms. Fig.4a shows the average travel time for different routing request volumes with no background traffic flows on roads. In Fig.4b, heavy background traffic is added on some most commonly chosen roads when no background traffic exists, which simulates the heavy traffic cases discussed in Section IV-D1.

The last two figures in Fig.4 show the fairness comparison result. Fig.4c shows the standard deviation of the travel time for different routing request volumes, and Fig.4d shows the sorted travel time for each route request when the routing request volume is  $8 \times 10^4$ .

In both Fig.4a and Fig.4b, the average travel time of MCMF-R and TS are at least 20% better than those of IRR and traditional GPS routing, especially when the traffic volume exceed  $7 \times 10^4$ . For traditional GPS routing, this improvement can achieve 40% on average. This improvement is even more in Fig.4c and Fig.4d, when the vehicle volume approaches  $8 \times 10^4$  to  $10 \times 10^4$ , the standard deviation and max-min difference value of both MCMF-R and TS are about 50% of those of IRR. This indicates that the time efficiency and fairness of both MCMF-R and TS algorithm are improved largely compared to traditional methods and the random routing of local taxi drivers without any route guidance.

IRR is better than traditional GPS routing, especially in routing ability. And this advantage is increasing with the traffic volume. This shows that when the traffic is getting heavy, even local taxi drivers outperform traditional route guidance.

For the time efficiency between MCMF-R and TS, they are quite close, and no one can dominate the other in all conditions. In Fig.4a, when the traffic volume is fewer than  $3 \times 10^4$ , MCMF-R is worse than TS. This is mainly because MCMF-R has more usage of the upper bound of travel time (Section IV-B) than TS, and overestimates the congestion condition. When the traffic volume is more than  $3 \times 10^4$ , the lack of global information makes the performance of TS degrading largely. When traffic volume is around  $5 \times 10^4$ , MCMF-R has a clear advantage over TS, which is about 28% quicker. After that, TS catches up a little, but MCMF-R is still stronger than it in time efficiency. In Fig.4b, TS is influenced more by heavy background traffic than MCMF-R when traffic volume is between  $3 \times 10^4$  to  $11 \times 10^4$ . This indicates that MCMF-R is more time efficient under heavy traffic conditions.

In fairness shown in Fig.4c, TS and MCMF-R are comparable to each other when the traffic volume is less than  $4 \times 10^4$ . Between  $4 \times 10^4$  and  $6 \times 10^4$ , TS is less fair than MCMF-R since it lacks global information. But when the traffic volume exceeds  $6 \times 10^4$ , which is also shown in Fig.4d, TS surpasses MCMF-R and strictly dominates it. It is because in MCMF-R, when the traffic is getting heavy, more vehicles will sacrifice

for the global optimization, causing loss of fairness.

## VI. CONCLUSION

In this paper, we propose two efficient algorithms for route guidance problem in vehicular network: MCMF-R for centralized routing and TS for distributed routing. Experiment results indicate that, both algorithms have better time efficiency and fairness than traditional methods. Between MCMF-R and TS, TS is more deployable, and more suitable in low traffic volume routing than MCMF-R, and when the traffic is heavy, MCMF-R is better in time efficiency but worse in fairness. A hybrid framework is then proposed to provide better deployment and routing decisions under different traffic intensity situations.

## ACKNOWLEDGMENTS

This work is supported in part by National Natural Science Foundation of China under Grant No. 61100196, 61073028, 61321491, 91218302; JiangSu Natural Science Foundation under Grant No. BK2011559.

## REFERENCES

- [1] D. Schrank, T. Lomax, and B. Eisele. (2011, Sep.) Urban mobility report. [Online]. Available: <http://mobility.tamu.edu/ums/report/>
- [2] J. Ott and D. Kutscher, "Drive-thru internet: Ieee 802.11 b for, automobile," *Proc. IEEE INFOCOM*, vol. 1, 2004.
- [3] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ wi-fi networks," *Proc. ACM MOBICOM*, pp. 50–61, 2006.
- [4] Sumo. [Online]. Available: <http://sourceforge.net/apps/mediawiki/sumo/>
- [5] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.
- [6] L. Xie, Q. Li, and et al., "Association control for vehicular wifi access: Pursuing efficiency and fairness," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 22, no. 8, pp. 1323–1331, 2011.
- [7] T. Kitani, T. Shinkawa, and et al., "Efficient vanet-based traffic information sharing using buses on regular routes," *IEEE Vehicular Technology Conference (VTC)*, pp. 3031–3036, 2008.
- [8] F. Zhan and C. Noon, "Shortest path algorithms: an evaluation using real road networks," *Transportation Science*, pp. 65–73, 1998.
- [9] I. Lin and S. Chou, "Developing adaptive driving route guidance systems based on fuzzy neural network," *Proc. IET 4th International Conference on Intelligent Environments*, pp. 1–8, 2008.
- [10] M. Khamfer, M. Guennoun, and H. Mouftah, "Wsn architectures for intelligent transportation systems," *3rd International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–8, 2009.
- [11] M. Khanjary and S. Hashemi, "Route guidance systems: review and classification," *Proc. 6th Euro American Conference on Telematics and Information Systems*, pp. 269–275, 2012.
- [12] R. Bauza and J. Gozalvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, 2012.
- [13] J. Ding, C. Wang, F. Meng, and T. Wu, "Real-time vehicle route guidance using vehicle-to-vehicle communication," *IET. Communications*, vol. 4, no. 7, pp. 870–883, 2010.
- [14] J. Van Lint, "Online learning solutions for freeway travel time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 38–47, 2008.
- [15] S. Dornbush and A. Joshi, "Streetsmart traffic: Discovering and disseminating automobile congestion using vanet's," *Proc IEEE Vehicular Technology Conference (VTC)*, pp. 11–15, 2007.
- [16] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.
- [17] openstreetmap. [Online]. Available: <http://www.openstreetmap.org/>