TOP: Optimizing Vehicle Driving Speed with Vehicle Trajectories for Travel Time Minimization and Road Congestion Avoidance

LI YAN and HAIYING SHEN, University of Virginia, USA

Traffic congestion control is pivotal for intelligent transportation systems. Previous works optimize vehicle speed for different objectives such as minimizing fuel consumption and minimizing travel time. However, they overlook the possible congestion generation in the future (e.g., in 5 minutes), which may degrade the performance of achieving the objectives. In this article, we propose a vehicle Trajectory–based driving speed OPtimization strategy (*TOP*) to minimize vehicle travel time and meanwhile avoid generating congestion. Its basic idea is to adjust vehicles' mobility to alleviate road congestion globally. *TOP* has a framework for collecting vehicles' information to a central server, which calculates the parameters depicting the future road condition (e.g., driving time, vehicle density, and probability of accident). Based on the collected information, the central server also measures the friendship among the vehicles and considers the delay caused by red traffic signals to help estimating the vehicle density of the road segments. The server then formulates a non-cooperative Stackelberg game considering these parameters, in which when each vehicle aims to minimize its travel time, the road congestion is also proactively avoided. After the Stackelberg equilibrium is reached, the optimal driving speed for each vehicle and the expected vehicle density that maximizes the utilization of the road network are determined. Our real trace analysis confirms some characteristics of vehicle mobility to support the design of *TOP*. Extensive trace-driven experiments show the effectiveness and superior performance of *TOP* in comparison with other driving speed optimization methods.

17

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**;

Additional Key Words and Phrases: Vehicular networks, driving speed optimization, game theory

ACM Reference format:
Li Yan and Haiying Shen. 2019. TOP: Optimizing Vehicle Driving Speed with Vehicle Trajectories for Travel Time Minimization and Road Congestion Avoidance. *ACM Trans. Cyber-Phys. Syst.* 4, 2, Article 17 (November 2019), 25 pages.
https://doi.org/10.1145/3362162

1 INTRODUCTION

In recent decades, Intelligent Transportation Systems (ITSs) have received much attention. The ITSs summarize advanced applications aiming at providing innovative services related to different modes of transportation and traffic management. To support the operation of various ITS

This research was supported in part by U.S. NSF grants ACI-1719397 and CNS-1733596 and Microsoft Research Faculty Fellowship 8300751.

Authors' address: L. Yan and H. Shen, University of Virginia, USA; emails: {ly4ss, hs6ms}@virginia.edu.

2378-962X/2019/11-ART17 $15.00
https://doi.org/10.1145/3362162

applications, traffic congestion control is very important for urban road networks [16, 22, 25] when trying to maximize their utilization. For example, the road management authorities hope that the density of vehicles simultaneously passing through each road is lower than a threshold so that the overall road network keeps operable. Also, the public transit service vehicles require their covered routes to be non-congested so that they can follow their schedule on time. However, due to the high mobility of vehicles and difficulty in controlling vehicle speeds, congestion control in urban road networks is a non-trivial task.

In recent years, many methods have been proposed to reduce vehicles' travel time by adaptively controlling traffic signal [21, 32, 48] or suggesting optimal speeds to vehicles for different objectives such as minimizing fuel consumption and travel time [5, 11, 15, 31, 38, 46]. In the former group of methods [21, 32, 48], the controller at a road intersection properly schedules the passing of vehicles to minimize the vehicles' total travel time caused by red lights or long queues. In the latter group of methods [5, 11, 15, 31, 38, 46], the optimal driving speed of a vehicle is determined based on the vehicle's real-time driving information (e.g., fuel consumption, traffic state). However, these methods all focus on optimizing the vehicle's driving speed on the single vehicle's perspective but overlook the possible road congestion generation in the future (e.g., in 5 minutes), which may degrade the performance of achieving the objectives. In other words, these methods cannot avoid the generation of road congestion globally in the road network in the future. By "in the future," we mean in a future time during a vehicle's driving time period. For example, before "rush hours," arterial roads may be non-congested, because most vehicles are distributed to smaller road segments. At this moment, the methods will conclude there is no congestion ahead, so the vehicles can drive by their optimal speeds determined upon current traffic state. However, if legions of vehicles drive by the currently "optimal speeds" in their individual routes, they may crowd into the arterial roads simultaneously, which results in congestion. Therefore, we need a system to coordinate the driving speed of all the vehicles in a global manner through considering their future movement, so that each vehicle can drive to their destination with the minimum delay, and the road network will be free from traffic congestion.

However, designing such a system is non-trivial. The road congestion is measured by vehicle density; a higher vehicle density increases the utilization of the road network but generates congestion and decreases vehicle speed and vice versa. Therefore, it is a challenge to maximize the utilization of the road network while proactively avoiding congestion and maximizing the vehicle speed. In this article, we aim to tackle this challenge by proposing a vehicle Trajectory–based driving speed OPtimization strategy (*TOP*) that uses game theory to let vehicle drive as fast and safely as possible and meanwhile proactively avoid generating road congestion in the future. Its basic idea is to periodically adjust vehicles' mobility to alleviate road congestion globally. The vehicles report their information to a central server through road-side-units (RSUs) located alongside the roads. The central server calculates each vehicle's trajectory in the next time slot (denoted by $\mathcal{T}_{c+1}$) and determines the parameters depicting the future utilization of the road network (e.g., vehicle density, driving time, and probability of accident). This is based on the previous observation that vehicles' trajectories can soundly illustrate the future mobility of the vehicles [17–19, 41, 42]. In addition, in the estimation of the vehicles' travel time to each position of the trajectory, we take into account the delay caused by red traffic signals according to the changing schedule of the traffic signals on the road network in the near future (e.g., 5 minutes). Based on the collected trajectories, we first let the central server determine the routines for each vehicle, which are the trajectories that a vehicle frequently drives on, for each vehicle. Then, the central server measures social closeness (i.e., friendship) among each pair of vehicles by calculating how many similar routines the two vehicles share in common. Specifically, if the spatial overlap of two vehicles' routine positions is higher than a threshold (e.g., 60%) and the temporal deviation of the vehicles' routine start and end time is less than a threshold (e.g., 5 minutes), then we view these two routines are similar. If the similar routines take up more than some threshold (e.g., 40%) over each of the two vehicles' total routines, then the central server views the two vehicles as friends. Finally, based on the probability of a vehicle's presence on each position of the trajectory (determined from the estimated travel time), we use its closeness with friends to help estimating the probability that its friends will also simultaneously appear at the same road segment and thereby the vehicle density of the road segment. To maximize the utilization of the road network while minimizing the probability of road congestion, the central server formulates a non-cooperative Stackelberg game, in which each vehicle aims at minimizing its travel time and maximizing safety while avoiding generating congestion in the future. After the Stackelberg equilibrium is reached, when vehicles follow their optimal speeds, it also proactively avoids generating congestion in the future. Moreover, the road network can be fully utilized without imbalanced or high vehicle density (i.e., congestion) in road segments. In summary, our contributions include the following:

(1) Our analysis on two real vehicle traces [4, 33] confirms the characteristics of vehicle mobility, distribution of the travel time of road segments, and social closeness in terms of the vehicles' common routines, which lay the foundation for the design of *TOP*.
(2) We propose a non-cooperative Stackelberg game-based vehicle speed optimization strategy to find the optimal speed for each vehicle that enables it to drive as fast and safely as possible, while avoiding the generation of congestion in the future.
(3) We have conducted extensive trace-driven experiments to show the effectiveness of *TOP* in maximizing the utilization of road network, avoiding congestions, and satisfying drivers' need of driving as fast and safely as possible. Additionally, we also demonstrate the respective effectiveness of considering red traffic signal delay and vehicle social closeness on improving the driving speed of the vehicles and the vehicle flow rate of the road segments.

Within our knowledge, this work is the first to provide vehicle speed optimization that aims at letting vehicles drive as fast and safely as possible, while proactively avoiding the generation of road congestion in the future. The remainder of the article is organized as follows. Section 2 presents related works. Section 3 presents the trace analysis and findings that support *TOP*. Section 4 presents the detailed design of *TOP*. Section 5 presents trace-driven experimental results. Section 6 concludes the article and marks future directions.

2 RELATED WORK

Real-time traffic-based vehicle speed optimization. Several methods for vehicle speed optimization with different objectives have been proposed. Kouvelas et al. [21] proposed a hybrid approach for traffic signal control considering the saturation status of the road. Zhao et al. [48] proposed to utilize discrete dynamic programming (DPP) with variable optimization time step size to reach the optimal tradeoff between precision and computational cost in vehicle driving speed optimization. Pandit et al. [32] proposed a vehicular network-based method that collects and aggregates real-time traffic information to optimize signal control. Tseng et al. [38] proposed a vehicle density estimation scheme using neighbor tables communicated between vehicles. Chen et al. [11] proposed to use vehicular ad hoc networks (VANETs) to send queries between source and destination back and forth and select the path with the shortest time. Ozatay et al. [31] used cloud computing [10, 34] in optimizing vehicle speed profile by solving a dynamic programming problem. Asadi and Vahidi [5] proposed a control algorithm to enable vehicles to approach traffic light at green as much as possible, thereby saving fuel and reducing travel time. Ye et al. [46] proposed to utilize vehicle-to-infrastructure (V2I) communication and driving state information of vehicles to optimize the timing of traffic lights to avoid minimize the vehicles' driving delay. Groot et al. [15] proposed to model vehicle-congestion relationship as reverse Stackelberg games to optimally distribute traffic over road network and meanwhile ensure that each vehicle can finish travel within its expected travel time. For the vehicles with the same origin-destination, the central server uses different pricing of freeways (e.g., longer route has lower price) to induce these vehicles to choose different routes to distribute the traffic. However, this method overlooks that the vehicles with different origin-destination pairs may compete for the same road segment. Also, it does not aim to minimize the travel time of the vehicles. As indicated previously, the above methods do not consider whether the currently suggested speed will cause congestion to certain road segments in the future. To solve this problem, *TOP* first utilizes vehicles' trajectories to extract the parameters of future road traffic and then uses the parameters in formulating a non-cooperative Stackelberg game that aims to let vehicles drive as fast and safely as possible and meanwhile avoid the generation of congestion in future.

Vehicle future mobility-based routing. Many works [28, 30] focus on using vehicles' current or historical mobility statistics to predict the vehicles' future mobility. Some other works [17–19, 41, 42] found that utilizing vehicles' GPS trajectory to deduce the vehicles' future mobility is reliable for data delivery in vehicular networks. Wu et al. [41] found the spatio-temporal correlation in vehicle mobility and noted that the future trajectory of a vehicle is correlated with its past trajectory. In Trajectory-based Data Forwarding Scheme (TBD) [18], Trajectory-based Statistical Forwarding Scheme (TSF) [17, 19], and Shared-Trajectory-based Data Forwarding Scheme (STDFS) [42], trajectory information of vehicles is collected through access points and used to predict the vehicle mobility for data forwarding. Our work is based on the observations in these works that vehicles' trajectories can soundly illustrate the future mobility of vehicles, which is used to estimate road vehicle density in the future.

3 TRACE ANALYSIS

In this section, we present our trace analysis on the Rome trace [4] and the San Francisco trace [33], which demonstrates the characteristic of vehicle mobility in urban area and provides the rationale of the design of *TOP*. The details of the datasets are presented as follows.

Rome Taxi Trace. It is a 30-day Global Positioning System (GPS) coordinate record of 320 taxis driving in the center of Rome from February 1, 2014, to March 2, 2014. In the trace, each taxi uses an Android tablet to receive the GPS position and uploads it with the driver ID and the timestamp to a central server every 7 seconds. To make the trace fit our analysis, we first extracted the road intersections and turning points as landmarks and normalized the movement of the taxis to the nearest landmarks. We also filtered out the nodes with few occurrences (<500) or low precisions (<20 m) and merged repeated records. Finally, we obtained 315 taxis and 4,638 landmarks.

San Francisco Taxi Trace. It is also a 30-day GPS coordinate record of 536 taxi traces driving in the San Francisco Bay Area from May 17, 2008, to June 15, 2008. In the trace, each taxi is equipped with a GPS receiver and uploads the position record of each taxi (identifier, timestamp, geo-coordinates) with a time interval less than 10 seconds. For analysis, we also normalized the movement of the taxis to the nearest landmarks. Each taxi has abundant records in this trace, so we filtered out the wrong records (e.g., positions out of the actual range of San Francisco, redundant positions). Finally, we obtained 536 taxis and 2,508 landmarks.

When a vehicle stays at one position for more than 5 minutes, we call this position an *anchor position* and consider it as the ending position of the previous trajectory and the starting position of the next trajectory. Thus, the anchor positions separate each vehicle's trace into several trajectories. The characteristics of the traces are summarized as in Table 1.

Table 1. Characteristics of Taxi Mobility Traces

	Rome	San Francisco
Number of taxis	315	536
Number of landmarks	4,638	2,508
Duration	30 days	30 days
Number of trajectories	572,143	469,223
Number of trajectories per day	19,071	15,640

3.1 Concepts and Problem Introduction

We define a road segment (denoted by s_i) as the road link between two neighboring intersections (i.e., landmarks). Vehicle density of road segment s_i (denoted by d_i) is defined as the average number of vehicles per meter in the road segment (veh/m), and the flow rate of road segment s_i (denoted by r_i) is defined as the average number of vehicles driving through the segment per unit time [6, 15] (veh/h). The vehicle flow rate of segment s_i equals to the product of vehicle density and average vehicle speed on s_i (denoted by v_i) [43],

$$r_i = d_i \cdot v_i. \tag{1}$$

The road congestion is measured by vehicle density, and the utilization of the road network is measure by flow rate. Therefore, to increase the utilization of the road segment s_i (i.e., r_i), we need to increase vehicle density (d_i) and/or vehicle speed (v_i). However, a higher vehicle density may lead to congestion and hence lower vehicle speed. Therefore, it is a challenge to maximize the utilization of the road network and meanwhile maximize vehicle speed, which is the objective of this article.

3.2 Concurrent Competition for Road Usage

Previous methods locally control traffic or compute suggested speed based on current traffic state on each vehicle's scheduled route. If a vehicle follows the speed individually optimized for it, due to the ignorance of other vehicles' mobility, then some arterial roads may become crowded with many vehicles, that is, the vehicles concurrently compete for these roads. To confirm this conjecture, we measured the Cumulative Distribution Function (CDF) of the vehicle density and the CDF of the flow rate on all road segments as shown in Figure 1 and Figure 2. We calculated the vehicle density and vehicle flow rate for 30 days and draw their average values. The vehicle density and vehicle flow rate are sampled every 30 minutes on every road segment per day. We see that for the Rome trace, the vehicle density of 90% of the road segments is less than 0.5 veh/m, and the vehicle flow rate of 90% of the road segments is less than 10 veh/h. But the other 10% of the road segments have vehicle density and vehicle flow rate as high as 3 veh/m and 60 veh/h, respectively. For the San Francisco trace, the vehicle density of 95% of the road segments is less than 3 veh/m, and the vehicle flow rate of 95% of the road segments is less than 25 veh/h. But the other 5% of the road segments have vehicle density and vehicle flow rate as high as 24 veh/m and 50 veh/h, respectively. These results demonstrate that in the urban road network, vehicles usually concurrently compete for usage on few popular roads, resulting in their excessive utilization. Therefore, we can try to distribute traffic evenly in the road network, i.e., achieve similar vehicle density in all road segment, to avoid the congestion and increase the utilization of the road network. The cause to repeated congestion on arterial roads is excessive concurrent utilization of vehicles [15]. Therefore, we further analyze vehicles' temporal preference on driving roads.

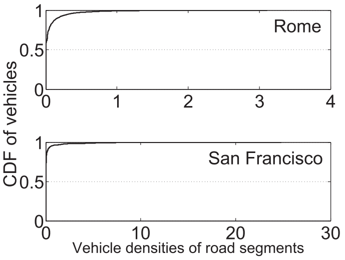


Fig. 1. Vehicle densities.

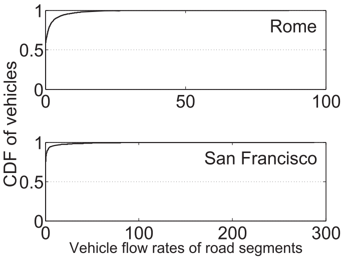


Fig. 2. Vehicle flow rates.

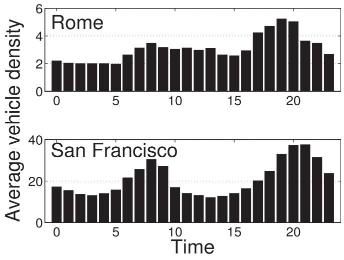


Fig. 3. Vehicle densities over time.

3.3 Vehicles' Temporal Preference on Roads

If the vehicle density on a road exceeds a threshold, then the driving speed of vehicles on the road is likely to be affected due to congestion. This is especially true for arterial roads, since they are quite likely to be over-utilized during rush hours. To verify such intuition, we measured the average vehicle density and average vehicle speed on the most highly utilized road segment of the two traces (road segment 4,433 in the Rome trace and road segment 0 in the San Francisco trace) hourly during each day in the 30 days, which are shown in Figure 3 and Figure 4, respectively. We see that for the Rome trace, the vehicle densities during 06:00-13:00 and 17:00-20:00 are higher than the other hours. In contrast, the average vehicle speeds during these two periods are lower than the other hours. For the San Francisco trace, the vehicle densities during 06:00-09:00 and 18:00-22:00 are higher than the other hours. In contrast, the average vehicle speeds during these two periods are generally lower than the other hours. These results demonstrate that excessively high vehicle density deteriorates road driving condition, which causes reduced driving speed. The results confirm that avoiding congestion is important to increasing driving speed and reducing travel time, especially in rush hours.

3.4 Distribution of Travel Time of Road Segments

When a vehicle is driving on a trajectory, the accumulated travel time of the composing road segments on the trajectory determines its position on the trajectory at a certain specific time. If the vehicle's travel time of the composing road segments follows some probabilistic distribution, then we may use the distribution to estimate the probability of the vehicle's presence on each road segment, which can be used to estimate the vehicle density of the road segment. To support this conjecture, we collected the historical travel time of each road segment in the Rome trace and San Francisco trace, respectively, so each road segment has several data samples of travel time. We first filtered out the road segments with too few samples of travel time (i.e., less than 100). Then, we applied the One-sample Kolmogorov-Smirnov test (K-S test in short), which verifies whether the population CDF of the data is equal to the hypothesized CDF [29] on each road segment to identify whether the collected travel time data of each road segment follow the normal distribution with the mean and the standard deviation of the road segment's historical travel time data as the parameters. If the test approves the hypothesis that the CDF of the travel time data of the road segment is similar to the normal distribution's CDF at the significance level (measures the reliability of this test) of 5%, then we view the data follow the normal distribution. Finally, we counted the total number of road segments that passed the test in the Rome trace and San Francisco trace, respectively, and calculated the test pass rate for the two traces. The test pass rate results are illustrated in Figure 5. We can see that almost 80% of the historical travel time of the road segments in Rome and San Francisco traces can be viewed as normally distributed.

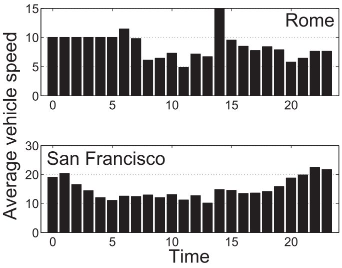


Fig. 4. Vehicle speeds over time.

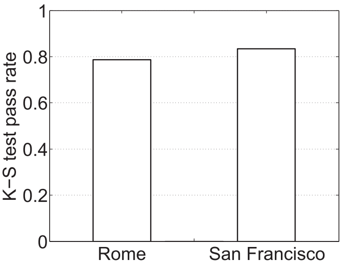


Fig. 5. Pass rate of K-S test.

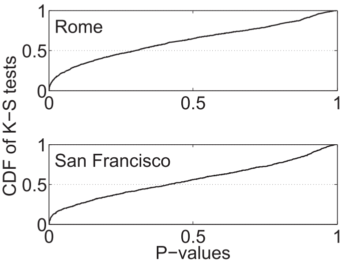


Fig. 6. P-values of the passed tests.

However, the test results only have a certain level of confidence to be reliable. Therefore, for the passed tests, we further measured their p-values. The p-value is the probability that the data samples will actually follow the target normal distribution, which measures the doubt on the test result. Small p-values (e.g., less than 0.1 in our case) mean that even if the road segment's travel time data passed the K-S test, the validity of the test is in doubt [29]. The p-values of the tests are shown in Figure 6. We can see that more than 80% and 85% of the passed tests resulted in a p-value larger than 0.1 in Rome and San Francisco, respectively. This means most of the test results are reliable. Therefore, the travel time of most road segments can be estimated with some normal distribution determined based on their historical travel time data. In Section 4.2, we will explain how we utilize the distribution of the travel time of the road segments to estimate the presence of the vehicles on each position of their respective trajectories.

3.5 Demonstration of Similarity between Vehicle Routines

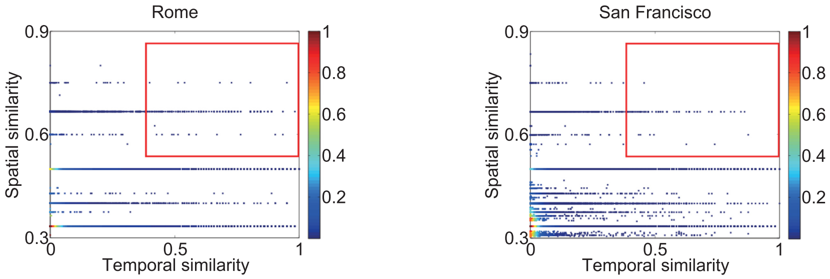
It has been confirmed that most vehicles repeat the same trajectories during certain time periods each day [45]. Among all the trajectories that a vehicle has driven by, if the ratio of a trajectory is higher than a threshold (e.g., 20%), we view this trajectory is a *routine* of the vehicle. Intuitively, some vehicles may share common routines during certain times. For example, certain people usually take the same routine routes to commute from their home to working place at around 08:30-09:00 every morning. In this case, we may use the similarity of the vehicle routines to deduce the presence of certain vehicles at specific times. To verify this conjecture, we measure the similarity between the vehicles' routines. Since the similar routines of two vehicles will not be completely identical, we use the spatiotemporal overlap of the routines to measure their similarity. Suppose we have two routines: R_i of vehicle V_i and R_j of vehicle V_j. R_i covers positions: $r_i = \{p_i(0), \ldots, p_i(m)\}$. Its start time range is T_i^s (e.g., 08:00–08:15), and its end time range is T_i^e (e.g., 09:00–09:10). R_j covers positions: $r_j = \{p_j(0), \ldots, p_j(m')\}$. Its start time range is T_j^s (e.g., 08:10–08:20), and its end time range is T_j^e (e.g., 08:50–09:15). We use $\bar{T}^s$ and $\bar{T}^e$ to denote the means of T^s and T^e, respectively. Then, the spatiotemporal similarity between R_i and R_j is measured as:

$$\tau_t = \exp\Big(- \max \Big\{ \Big|\bar{T}_i^e - \bar{T}_j^e\Big|, \Big|\bar{T}_i^s - \bar{T}_j^s\Big| \Big\} \Big)$$

$$\gamma_s = \frac{\Big|r_i \bigcap r_j\Big|}{\Big|r_i \bigcup r_j\Big|}, \tag{2}$$

where τ_t is the overlapping of the start times and end times of the two routines, γ_s is the overlapping of the positions covered by the two routines.

Next, we calculated the spatial and temporal similarities of every two trajectories generated by any pairs of two vehicles, respectively, in the two traces. We filtered out the trajectories that have too low spatial similarity (i.e., the overlapping of routine positions is less than 0.3). The results are



(a) Rome.

(b) San Francisco.

Fig. 7. Density scatter of temporal deviation w.r.t. spatial deviation.

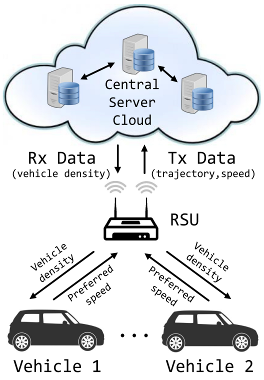


Fig. 8. System structure.

illustrated in Figure 7 as the density distribution of the temporal similarity with respect to (w.r.t) the spatial similarity. Each point represents a comparison result between two trajectories coming from two different vehicles. We use color heat to demonstrate the density of the distribution of the results. The warmer color the points have, the more concentrated these results are on their corresponding spatial and temporal similarities. We can see that the routines with both high spatial similarity (0.5) and high temporal similarity (0.4, namely time deviation is less than 1 minute), which means the two vehicles having these routines are likely to simultaneously appear on certain road segments, take up a small portion within the square circle. This motivates us to properly measure the vehicles' social closeness (i.e., friendship in terms of the spatial and temporal similarities of their routines) between each other. Then, given the presence of a vehicle, we can use the friendship to estimate the probability of its friends' presence on the same road segment. Based on the presence probability of the vehicles, we can further estimate the vehicle density for each road segment. We will introduce the details of the determination of the friendship among the vehicles in Section 4.2.1.

4 SYSTEM DESIGN

4.1 Overview

ITSs support the installation of RSUs alongside road segments to provide communication between vehicles and the central server [12, 17, 26]. As shown in Figure 8, we establish a three-layer information collection and dissemination framework, which consists of vehicles as the service layer, RSUs as the communication backbone and a central server as the computation layer. Each vehicle contacts the central server through RSUs. As in the traffic management papers in Reference [8], we consider road segments have equal vehicle density limits. In this article, we focus on optimizing the vehicles' speed on their original route. We leave the optimal route selection as future incremental work. To let vehicles drive as fast and safely as possible while avoiding generating congestion on the road network, we use the Stackelberg game [36] between the vehicles and the central server to determine the expected vehicle density that maximizes the utilization of the road network and optimal driving speed for each vehicle. The gaming process is executed periodically with period $\mathcal{T}$ (e.g., 5 minutes). Note that the value setting of the period determines the tradeoff between traffic optimization timeliness and computation overhead. If the period is relatively larger, then the gaming process will be executed less frequently but reduces more computation pressure on the central server, and vice versa. Thus, the value of the period should be set manually according to different city road networks' requirement on traffic optimization timeliness and computation overhead. The gaming process can be summarized as follows:

1. Through a nearby RSU, the vehicle reports its current position and intended destination to the central server.
2. Based on the information collected from vehicles, the central server calculates the trajectory of each vehicle in $\mathcal{T}_{c+1}$ and predicts the vehicle density in each road segment at the next time slot. Then, a gaming process is conducted between the central server and each vehicle.
3. Based on the predicted average vehicle density per road segment in the road network in $\mathcal{T}_{c+1}$, the central server determines a set of expected average densities that are achievable by vehicle speed adjustment.
4. Based on each expected average density, each vehicle determines its speed that maximizes its utility (speed and safety) and reports the speed to the central server.
5. The central server determines the final expected average density that maximizes its utility (maximizing flow rate of the road network) and notifies all vehicles.
6. Each vehicle chooses its speed corresponding to the final expected average density.

With the optimal speeds, the vehicle density of each road segment will be approximately the determined vehicle density. Thus, the total traffic in the road network is well balanced with no congestion and its utilization is maximized. We will first explain how the central server predicts the vehicle density of road segments (Section 4.2) and then present the non-cooperative Stackelberg gaming (Section 4.3).

4.2 Future Road Vehicle Density Prediction

The gaming process runs after each time slot $\mathcal{T}$ (e.g., 5 minutes). For example, when the central server starts the game at 00:00, it needs to estimate the vehicle density of each road segment in [00:00, 00:05] to determine an achievable vehicle density in the entire road network for vehicles to choose their optimal speeds.

In this section, we present how to estimate the vehicle density of each road segment in $\mathcal{T}_{c+1}$ with current vehicle speeds. First, the central server needs to determine each vehicle's trajectory in $\mathcal{T}_{c+1}$. It consists of the road segments it will pass in $\mathcal{T}_{c+1}$ and their corresponding travel times $\{(s_i, \widetilde{T_i}) | i = 1, 2, \ldots, M\}$, where s_i denotes the ith road segment, $\tilde{T}_i$ denotes the estimated travel time from current position to s_i and M denotes the number of road segments that the vehicle will pass in $\mathcal{T}_{c+1}$.

Then, by modeling the arrival times as normal random variables, *TOP* sums up the vehicles' probabilities of appearance on each road segment as its vehicle density at the next time slot. The average vehicle density per road segment will be used in the driving speed optimization gaming

Table 2. Table of Friends

Vehicle ID	*Friends*
V_1	$V_2(0.5), V_3(0.3), V_4(0.2), V_5(0.1)$
V_2	$V_1(0.4), V_2(0.2)$
V_3	$V_8(0.6), V_1 0(0.2), V_1 5(0.1)$
...	...

presented in Section 4.3. After each vehicle determines its speed in gaming, the vehicle density will be updated and used for the next gaming process.

4.2.1 Measuring Friendship among Vehicles. In Section 3.5, we have defined the routines of a vehicle as certain trajectories it frequently drives on at specific times. In Figure 7, we also demonstrated that some vehicles are likely to simultaneously drive on the same road segment during specific time periods. Given two routines, say, R_i of vehicle V_i and R_j of vehicle V_j, their similarity is measured by Equation (2). To identify whether R_i and R_j are similar, we first determine whether they overlap enough with each other in spatial positions. Specifically, we will further consider the two routines' temporal similarity only if their spatial similarity $\gamma_s \geqslant 0.6$, namely they have more than 60% of spatial positions in common. Then we determine whether their temporal similarity is high enough. For metropolitan cities, relatively high temporal deviation should be tolerated. Therefore, we use a temporal similarity threshold of $\tau_t \geqslant 0.0067$ (i.e., maximum difference in the starting time and ending time is less than 5 minutes) for identifying temporal similarity. Finally, the principle for determining the similarity between the routines of two vehicles can be represented as follows: For R_i and R_j whose spatial similarity is greater than 0.6, if the maximum difference in the start time and end time of R_i and R_j is no more than 5 minutes, we view the two routines are spatiotemporally similar to each other.

Next, we specify how to determine the friendship among two vehicles in terms of the ratios of their similar routines over the total routines of each vehicle. We define two vehicles are friends if for each vehicle, the ratio of the two vehicles' similar routines over the vehicle's total routines (i.e., closeness ratio β) is higher than a threshold, say, β_{th}. For example, suppose vehicle V_i has routines: R_i^1 and R_i^2 in total, vehicle V_j has routine: R_j^1, and the closeness ratio threshold β_{th} is 0.4. Suppose R_i^1 of V_i and R_j^1 of V_j are similar. Thus R_i^1 takes up $\beta_i^1 = 50\%$ of V_i's total routines, and R_j^1 takes up β_j^1 =100% of V_j's total routines. Since both β_i^1 and β_j^1 are higher than the threshold $\beta_{th} = 0.4$, we conclude these two vehicles are friends. In *TOP*, routine extraction and friendship determination are conducted by the central server with a relatively long period (e.g., 3 months) and maintained in a table as shown in Table 2. Note that each friend is denoted with a closeness ratio (i.e., β), respectively.

If V_i and V_j have lots of routines in common, then it means that the two vehicles are likely to simultaneously drive on the same road segments at specific times. Therefore, given the presence of one vehicle, its friendship may offer us some hint on where its friends will possibly be, which can be an auxiliary method for estimating vehicle density. For example, suppose Alice and Bob live in the same suburban community, every morning they drive through the same highway to downtown. Given Alice is on a certain road segment at 08:30, Bob has certain probability to be on the same road segment at that time. Based on such observation, we take into account the friendship among vehicles in the estimation of vehicle density on the road segments, which will be elaborated in Section 4.2.4.

4.2.2 Trajectory Calculation. A vehicle periodically reports its current position and its destination to the central server. To generate the vehicle's trajectory in $\mathcal{T}_{c+1}$, the central server first determines the sequence of road segments connecting the current position and the destination based on road topology [40]. It then calculates the travel time of each road segment that will be passed in $\mathcal{T}_{c+1}$ by the vehicle. After a gaming process, a vehicle's optimal speed on s_i is determined, denoted by v_i. Then, for each road segment s_i, the estimated travel time on s_i (denoted by $\widetilde{t_i}$) can be calculated by

$$\widetilde{t_i} = l_i / v_i, \tag{3}$$

where l_i is the length of s_i. A problem is how to estimate the travel time of s_i initially when no game has been played. To handle this problem, we use the current vehicle density of the road segment to estimate the speed for the vehicle as in traditional vehicle density-based speed estimation methods [43]. It has been indicated that for a road segment s_i, its reachable speed is related to a vehicle density limit d_i^m. When the vehicle density is below d_i^m, vehicles on the road segment can drive with the free flow speed (i.e., speed limit, denoted by v_i^{max}). If the vehicle density exceeds d_i^m, then the road segment will be congested and the vehicles have to drive with the congested speed (denoted by v_i^{min}). d_i^{jam} is the vehicle density that will cause s_i to be completely jammed. d_i^m can be obtained from field observation, and d_i^{jam} can be obtained from the road network's designed capacity [43]. Currently, the vehicle density of each road segment can be well monitored [14, 16, 20, 35, 37]. Then, we can roughly estimate the allowed vehicle speed under current vehicle density for each road segment as below:

$$\widetilde{t_i} = \begin{cases} l_i / v_i^{max}, & 0 \leqslant d_i < d_i^m \\ l_i / v_i^{min}, & d_i^m \leqslant d_i < d_i^{jam} \\ \infty, & d_i \geqslant d_i^{jam} \end{cases}. \tag{4}$$

The trajectories generated by GPS do not consider the road congestion condition and hence may not be sufficiently accurate. In contrast, *TOP* calculates the trajectories of vehicles considering future road congestion.

4.2.3 Travel Time Estimation Considering Traffic Signal. On the one hand, according to previous works of road segment travel time modeling [23, 24], the travel time of a road segment can be described by normally distributed and statistically independent random variables with acceptable precision. But from the K-S tests in Section 3.4, we also notice that the travel time of some road segments cannot be modeled with normal distribution. For these road segments, we apply Kernel Density Estimator (KDE) [47] to learn the Probability Density Function (PDF) and use the PDF to estimate the vehicles' travel time on the road segment. In Section 3.4, we have also shown that the distribution of the travel time of the road segments can be described with the normal distribution with the mean and the standard deviation of the travel time data as parameters. Therefore, the estimated travel time from the vehicle's current position to some road segment s_i is the sum of the respective travel times of the composing road segments from current position to s_i, namely $\widetilde{T_i} = \sum_{m=1}^{M_i} \widetilde{t}_m$, where M_i is the number of road segments from the vehicle's current position to s_i. When $\widetilde{T_i} \geq \mathcal{T}$, the trajectory for $\mathcal{T}_{c+1}$ has been generated. Based on the historical records of the travel time of s_m and real travel time on s_m of all vehicles, the central server can calculate the variance of each road segment σ_m^2. Then, the standard deviation of $\widetilde{T_i}$ is calculated by summing the variances of the composing road segments, $\Delta_i^2 = \sum_{m=1}^{M_i} \sigma_m^2$.

However, when driving through the road segments, the vehicle may be intermittently blocked by red traffic signals. Thus, when estimating the travel time of a trajectory, we also need to consider the possible waiting time caused by traffic signals on the way. With the estimated travel time of respective road segments and current time, we can deduce the vehicle's arrival time at each intersection. For example, in Figure 9, given the starting time (t_s) at the vehicle's current position,

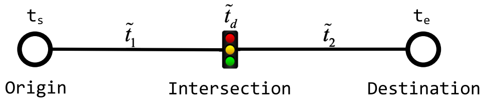


Fig. 9. Travel time considering traffic signal.

Table 3. Table of Traffic Signal Time Sequence

Traffic signal ID	00:00:01	00:00:02	00:00:03	00:00:04	...
S_1	r	g	g	g	...
S_2	r	r	r	g	...
S_3	g	g	g	g	...
...					

the vehicle will arrive at the intersection at around $t_s + \widetilde{t_1}$. The changing schedule of the traffic signals, which is as shown in Table 3, is extracted from the real traffic signal schedule trace of Rome and San Francisco [4, 33]. In this table, "r" represents the red signal, and "g" represents the green signal. At the intersection, by referring to the traffic signal table, we estimate that the vehicle will encounter a red traffic signal, which blocks it for a duration of $\widetilde{t_d}$. Thus, in addition to the travel time of the composing road segments, the delay caused by traffic signals should also be considered if we want to estimate where the vehicle will be in the near future. In the next section, we will introduce how we utilize the estimated positions of the vehicles to calculate road vehicle density in the near future.

4.2.4 *Road Vehicle Density Calculation.*

The estimated travel time in $\{(s_i, \widetilde{T_i})|i = 1, 2, \ldots, M\}$ only has a certain probability to be accurate. Then, we have two steps to calculate the vehicle density of each road segment in $\mathcal{T}_{c+1}$. First, we use a vehicle's trajectory in $\mathcal{T}_{c+1}$ to estimate the probability that the vehicle will appear at each road segment in its trajectory in $\mathcal{T}_{c+1}$. Then, we calculate the sum of all the vehicles' appearance probabilities at a road segment in $\mathcal{T}_{c+1}$ as the vehicle density of the road segment in $\mathcal{T}_{c+1}$.

Suppose the next time slot is the jth time slot in a day, represented by $\mathcal{T}_{c+1} = [t_j^s, t_j^e]$ (e.g., [00:00, 00:05]), where t_j^s and t_j^e are the starting time and ending time of the time slot, respectively. Within the short time interval of $\mathcal{T}_{c+1}$, the total delay caused by the traffic signals on the trajectory of each vehicle (T_i^d) can be viewed as a constant [22]. Thus, the sum of the travel time of the composing road segments and the traffic signal delay still follows normal distribution with a mean of $\widetilde{T_i} + T_i^d$ and a variance of Δ_i^2. For each vehicle, *TOP* uses its estimated travel time to s_i to measure its appearance probability at s_i during $[t_j^s, t_j^e]$. Therefore, the vehicle's appearance probability at s_i during $[t_j^s, t_j^e]$ is

$$P\left(T_i \leqslant t_j^e - t_j^s\right) = \Phi\left(\frac{t_j^e - t_j^s - \widetilde{T_i} - T_i^d}{\Delta_i}\right) - \Phi\left(\frac{-\widetilde{T_i} - T_i^d}{\Delta_i}\right), \tag{5}$$

where T_i denotes the vehicle's actual travel time from current position to s_i, and $\Phi(\cdot)$ is the CDF of the standard normal distribution with mean $\widetilde{T_i}$ and standard deviation Δ_i. The CDF for each vehicle on each road segment s_i is calculated based on the historical records of all vehicles' travel times on the road segment. Then, for each road segment s_i, the central server estimates its vehicle density in $\mathcal{T}_{c+1}$ by summing up the appearance probabilities of vehicles (P_k) on s_i during $\mathcal{T}_{c+1}$:

Table 4. Table of Accident Probability of Road Segment *College Ave*

Time	*Accident probability*
00:00∼00:05	0.05
00:05∼00:10	0.02
...	...

$$d_{c+1}^{s_i} = \sum_{k=1}^{N} P_k\left(T_i \leqslant t_j^e - t_j^s\right) + \sum_{f=1}^{F_N} \max\left\{\beta_f^1, \ldots, \beta_f^N\right\} P_f\left(T_i \leqslant t_j^e - t_j^s\right), \tag{6}$$

where N is the number of vehicles that will pass s_i during $[t_j^s, t_j^e]$. F_N is the total number of friends of the N vehicles. Since the N vehicles may have overlap in friendship, we use $\max\{\beta_f^1, \ldots, \beta_f^N\}$, which is the maximum social closeness ratio of the fth friend vehicle among the N vehicles, as the weight of the fth friend vehicle. $P_f(T_i \leqslant t_j^e - t_j^s)$ is the fth friend vehicle's appearance probability at s_i during $[t_j^s, t_j^e]$. For example, given current time 00:00, the estimated vehicle density of *College Ave* for $\mathcal{T}_{c+1}$, namely 00:00–00:05, is 26.16 vehicles per meter.

4.2.5 *Safety Estimation.* Each road segment has a probability of accident occurrence. The probability depends on the structure feature of the road segment (e.g., the degree of straightness, sharp turn, road surface bump) and the traffic condition. It has been verified that traffic conditions (e.g., heavy traffic volume, speeding) affect the likelihood of accident occurrence [3]. The traffic condition of a road segment has a long-term pattern, that is, the vehicle flow rate at each time slot remains similar irrespective of days. For example, people are likely to encounter congestion on their way to work during morning rush hours every workday.

TOP relies on historical records of accidents to depict the likelihood of accident for each road segment in each time interval in a day [27, 44]. Considering that the probability of accident is time-varying (e.g., some road segments are more likely to have accidents in Winter than in Spring), *TOP* uses a time window to control the number of days for consideration. The larger the window size is, the more accident events that can be captured. Specifically, from the historical records, we can know that during the jth time interval of the wth day, the time duration that road segment s_i is affected by accident is T_j^w. Thus, given the total number of days for consideration W (i.e., time window size), the cumulative time duration that s_i is affected during the jth time interval by accident is $\sum_{w=1}^{W} T_j^w$. Moreover, the total time duration covered by this window size is $W(t_j^e - t_j^s)$, where $[t_j^s, t_j^e]$ is the start and end time of the jth time interval. Finally, the accident probability of s_i during the jth time interval of the W days is calculated as the cumulative time duration that s_i is affected during the jth time interval (i.e., $\sum_{w=1}^{W} T_j^w$) over the total time duration covered by this window size (i.e., $W(t_j^e - t_j^s)$):

$$p_i^j = \frac{\sum_{w=1}^{W} T_j^w}{W(t_j^e - t_j^s)} \tag{7}$$

where p_i^j is the accident probability of s_i during the jth time interval. For example, we found that during the time slot [00:00, 00:05] in 3 days, the time durations that a road segment is affected by accident are 3 minutes, 0 minutes, and 5 minutes. Thus, based on the 3-day-long time window, the accident probability of the road segment is calculated as $\frac{3+0+5}{3*5} = 0.53$. Finally, for each road segment, the central server generates a table summarizing accident probability during each time interval, as shown in Table 4. Since higher vehicle density leads to shorter distance between vehicles, which renders higher risks of accident, we relate p_i^j with vehicle density in determining the utility of drivers in gaming in Section 4.3.

4.3 Driving Speed Optimization Gaming

4.3.1 Overview. On one hand, traveling quickly (i.e., short driving time and no congestion) and safely (i.e., no accident) is desired by drivers. On the other hand, the transportation authority hopes to maximize the utilization of the road network (i.e., maximum vehicle flow rate). Based on Equation (1), to increase road network utilization, we need to increase the vehicle density, which however may lead to road congestion. Then, the vehicle speed drops down and results in low flow rate and hence low utilization of the road network.

It is found that drivers may drive slower given a higher specified vehicle density to keep a safe inter-vehicle distance to keep safety, especially for the drivers with high probability of accidents [3]. Therefore, the drivers will adjust speeds in response to a given vehicle density. Thus, we can formulate the speed optimization as a non-cooperative Stackelberg game [9, 15] between the central server and the drivers, where the central server is the leader and the drivers are followers.

In the Stackelberg game, the leader considers the predicted average vehicle density of a road segment (introduced in Section 4.2.4), and then chooses a set of expected vehicle densities (denoted by $D = \{d_1, d_2, \ldots, d_n\}$) that are achievable by vehicle speed adjustment. The central server hopes to evenly distribute the vehicles over all road segments by properly assigning a d value. The followers receive D from the leader and picks a speed in response to each d_i to maximize its own utility (driving as fast and safely as possible while minimizing the risk of congestion). Next, the central server selects the vehicle density that maximizes its utility (i.e., vehicle flow rate of the road network), denoted by d_l and then the vehicles choose their speeds corresponding to the selected d_l. Finally, we solve the Stackelberg equilibrium of the game, i.e., the game reaches a state that the road network utilization is maximized while the drivers are satisfied with the driving status (judged by driving speed and associated risk of congestion). The gaming is executed periodically. In the following, we first introduce the utility of a driver and the utility of the central server, and then introduce the gaming between them.

4.3.2 Utility Function of Drivers. For drivers, we define a utility function to quantify the level of benefit that a driver obtains from driving by a speed on road segment s_i. It is calculated by subtracting the potential risk of congestion ($U_r(\cdot)$) from a vehicle's satisfaction degree ($U_s(\cdot)$), as shown in Equation (9),

$$F\left(v_i, \alpha_i, p_i^j\right) = U_s\left(v_i, \alpha_i, p_i^j\right) - U_r\left(d, v_i, p_i^j\right) \quad \text{s.t.} \quad v_i \leqslant v_i^{max}, \tag{8}$$

where v_i is the vehicle's speed for optimization; α_i is a scale factor to make $U_s(\cdot)$ and $U_r(\cdot)$ comparable; p_i^j is calculated by Equation (7).

$U_s(\cdot)$ ought to be non-decreasing as each driver desires high speed (i.e., short driving time). Meanwhile, the marginal satisfaction degree of the driver is non-increasing, because the driver's satisfaction degree gradually gets saturated when the vehicle speed increases to some level [43]. Moreover, $U_s(\cdot)$ is inversely related with the probability of accident, because a lower possibility of accident corresponds to higher level of satisfaction [13]. Considering these properties, we design $U_s(\cdot)$ as a concave function. Since the Natural Logarithmic Functions are representative concave functions [7], we define

$$U_s\left(v_i, \alpha_i, p_i^j\right) = \alpha_i \cdot \ln\left(v_i + p_i^{j^{-1}}\right). \tag{9}$$

A driver's potential risk of congestion is determined by the accident probability of the road segment (p_i^j) and vehicle flow rate [3] (Equation (1)),

$$U_r\left(d, v_i, p_i^j\right) = p_i^j d v_i. \tag{10}$$

The utility of a driver decreases with a higher vehicle density and vice versa. Combining Equation (9) and Equation (10) into Equation (9), we have the following:

$$\begin{aligned} F\left(v_i, \alpha_i, p_i^j\right) &= \alpha_i \cdot \ln\left(v_i + p_i^{j-1}\right) - p_i^j d v_i \\ \text{s.t.} \quad v_i &\leqslant v_i^{max}. \end{aligned} \tag{11}$$

Note the gaming is executed periodically, so it is possible that a vehicle may enter other road segments during the current time slot. We use γ_i to denote the percentage of $\mathcal{T}$ that the vehicle will spend on segment s_i. Then, the utility of the vehicle is calculated by

$$\begin{aligned} &\sum_i \gamma_i F\left(v_i, \alpha_i, p_i^j\right) \\ &\text{s.t.} \quad v_i \leqslant v_i^{max}. \end{aligned} \tag{12}$$

4.3.3 Utility Function of Central Server. The central server always aims at maximizing the vehicle flow rate on overall road network:

$$L(d) = \sum_{i=1}^{N_s} d_i \cdot v_i, \tag{13}$$

where N_s is the total number of road segments.

4.3.4 Optimal Driving Speed Selection. Recall that based on Equation (6), the central server predicts the vehicle densities of all road segments. It then calculates the average estimated vehicle density of the road network during next period of gaming: $\overline{d_{c+1}} = \sum_{k=1}^{N_s} d_{c+1}^{s_k} / N_s$. Based on $\overline{d_{c+1}}$, the central server determines a range of expected vehicle densities that are achievable by vehicle speed adjustment and offers these densities to each vehicle for selection, which is defined as

$$d_u = \ln(u+1) \cdot \overline{d_{c+1}}, \ u \in [1, \ldots, n]. \tag{14}$$

We have designed Equation (14) using a natural logarithm as they are slowly varying functions with respect to variable changes. This means that the central server prefers the drivers to select the vehicle density approximate to the $\overline{d_{c+1}}$. Arranging the density selections with a logarithm can more quickly enable the vehicles to select their respective driving speeds that can achieve the expected average vehicle density $\overline{d_{c+1}}$ or a similar value during the gaming process. We use $D = \{d_1, d_2, \ldots, d_n\}$ to denote the n levels of expected vehicle densities for $\mathcal{T}_{c+1}$. The reason for providing n levels of expected vehicle densities for selection is that the $\overline{d_{c+1}}$ provided by the central server may not be applicable for every vehicle all over the road network, and needs to be relaxed for general vehicles. In practice, n should be at least larger than the exponent so that the vehicle has multiple selections around $\overline{d_{c+1}}$. The central server notifies drivers of the D. If $\overline{d_{c+1}}$ leads to an increased expected vehicle density (d_u), then the drivers will be encouraged to decrease driving speed to drive safely. Otherwise, the drivers will be encouraged to increase driving speed to increase benefit while maintaining driving safety. Note the increment rate of $U_s(\cdot)$ (Natural Logarithmic Function) is slower than $U_r(\cdot)$ (Linear Function) when speed v_i increases. Therefore, according to Equation (9), increasing driving speed on current road segment (v_i) over the optimal expected driving speed will reduce a driver's utility, because $U_r(\cdot)$ will increase faster than $U_s(\cdot)$. Thus, driving at a slower speed can prevent the vehicle density of the road network from further increasing, i.e., prevent traffic congestion.

For each $d_u \in D$, if a driver will drive in its current road segment s_i during the next time slot, it chooses a new speed that maximizes its utility $F(\cdot)$, denoted by v_{iu}, as shown in Equation (15),

$$v_{iu} = \underset{v_i \leqslant v_i^{max}}{\arg\max} F\left(v_i, \alpha_i, p_i^j\right). \tag{15}$$

If a driver will drive through more than one road segment s_i, $s_j, \ldots$, then it chooses a set of speeds in each of the segments to maximize its utility $F(\cdot)$, denoted by $\{v_{iu}, v_{ju}, \ldots\}$ as shown in Equation (16),

$$\{v_{iu}, v_{ju}, \ldots\} = \underset{v_k \leqslant v_k^{max}}{\arg\max} \sum_k \gamma_k F\left(v_k, \alpha_k, p_k^j\right). \tag{16}$$

Finally, the driver reports the n candidate speeds to the central server. The central server finalizes the expected vehicle density (d_l) that maximizes its utility $L(\cdot)$ based on the candidate speeds from all drivers,

$$d_l = \underset{d_u \in D}{\arg\max} L(d_u) = \underset{d_u \in D}{\arg\max}\, d_u \sum_{N_s} v_{iu}. \tag{17}$$

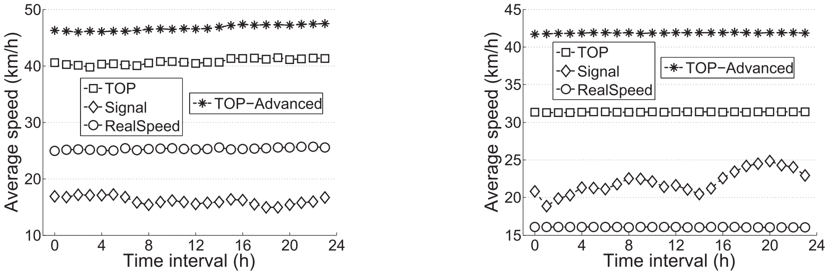
Since each vehicle's local driving state may have changed at the time when d_l is finalized, we let each vehicle finalize its driving speed based on its driving status in a distributed manner to avoid excessive information exchange on the negotiation between each vehicle's respective driving speed and the central server [39]. Specifically, the central server then notifies all drivers of the new expected vehicle density d_l. Then, among the n candidate speeds, each driver picks the speed corresponding to d_l.

5 PERFORMANCE EVALUATION

5.1 Experimental Settings

We conducted trace-driven experiments based on the Rome [4] and the San Francisco [33] traces introduced in Section 3. Unless otherwise specified, the experiment settings are the same as those in Section 3. The number of accidents occurred in Rome and San Francisco in each month are obtained from Reference [1] and Reference [2], respectively. The window size W was set to 7 days and T_j^w =1 hour. The scale factor α_i was set to 2.85 for Rome and 5 for San Francisco. The social closeness ratio threshold β_{th} is 0.4 in both Rome and San Francisco. We measure a driver's satisfaction degree when (s)he drives on road segment s_i with speed v_i by $\ln(v_i + p_i^{j-1})/\ln(v_i^{max} + p_i^{j-1})$ (deduced from Equation (9)). The gaming procedures are launched every 15 minutes in the two traces. To simulate that vehicles drive by their optimal speeds, we dynamically update the timestamps of arrivals at landmarks according to the vehicles' optimal speeds. Therefore, in the experiment, the vehicles follow the movement paths recorded in the traces but with modified timestamps.

We compared *TOP* with a representative traffic signal control method [32] (*Signal* in short) and a representative vehicle speed optimization method [31] (*RealSpeed* in short). *Signal* also focuses on optimizing the driving speed of the vehicles but via the control of traffic signals. It uses VANETs to formulate vehicles into platoons. The controller at each intersection uses the oldest-arrival-first scheduling algorithm to arrange the passing of platoons so that the vehicles' total travel time is minimized. *RealSpeed* tries to optimize the driving speed of vehicles by considering the current traffic on each vehicle's driving route. By aiming at reducing fuel consumption and satisfying driver with reduced travel time, the vehicle speed is optimized by dynamic programming constrained by speed limit, real-time traffic, and driver's destination. To make *RealSpeed* comparable to the other methods, we excluded its fuel consumption constraint in our experiments. *Signal* and *RealSpeed*



(a) Rome.

(b) San Francisco.

Fig. 10. Speed optimization performance for vehicles over time.

cannot proactively avoid the generation of road congestion in the future. To demonstrate the effectiveness of considering the friendship among the vehicles and the delay caused by traffic signals on increasing the accuracy of vehicle density on the road segments, we extend *TOP* with the two advanced components in the experiment, which is named as *TOP-Advanced*. We also measured the performance improvement brought by these two components, respectively. Each experiment is for 30 days. In each hour throughout each day, we measured the following metrics and report the average value in each hour for the 30 days.

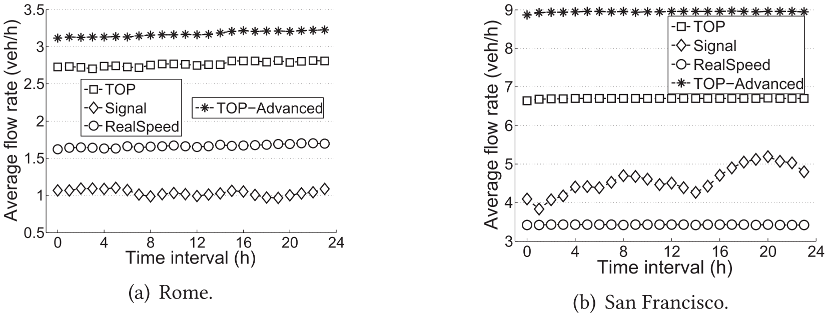
- *Average vehicle speed*: The average speed of all the vehicles determined by the games during an hour.
- *Average flow rate*: After each game, we calculate the flow rate per road segment by $\sum_{i=1}^{N_s} d_i \cdot v_i / N_s$. Then, we calculate the average flow rate per road segment in all the games during an hour.
- *Average driving time*: The average driving time on each road segment for all the travels on segments during an hour.
- *Average driver satisfaction*: The average satisfaction degree of the drivers after travels per hour.
- *Runtime of optimization processes*: The CDF of the runtime of all the optimization processes.

5.2 Experimental Results

5.2.1 Average Vehicle Speed. Figure 10(a) and Figure 10(b) show the average speed of vehicles during different time intervals with the Rome and San Francisco traces, respectively. We see that for Rome, the average vehicle speeds follow *TOP-Advanced*> *TOP*> *RealSpeed*> *Signal*. While for San Francisco, the average vehicle speeds follow: *TOP-Advanced*> *TOP*> *Signal*> *RealSpeed*.

TOP always has much higher average vehicle speed than others in both traces. Before optimization, the future traffic on the scheduled route has been deduced by the central server from the vehicle trajectories. Although the results might have deviation from the true results, they are still effective in predicting the future movement of vehicles. With gaming using the predicted vehicle density, *TOP* enables the central server to maximally avoid road congestion caused by confluent vehicle flows. Meanwhile, each vehicle can drive by a speed as fast and safely as possible. As a result, *TOP* generates the highest average speed of vehicles. Both *RealSpeed* and *Signal* do not proactively avoid generating congestions in the future and congestions decrease vehicle speeds, thus producing lower average speed of vehicles.

RealSpeed has the secondary performance in Rome, but the lowest performance in San Francisco. In *RealSpeed*, for a vehicle, the server generates a route based on the collected information of the intended travel. Then, the server collects the associated traffic and geographical information,



(a) Rome.

(b) San Francisco.

Fig. 11. Vehicle flow rate optimization performance for road segments over time.

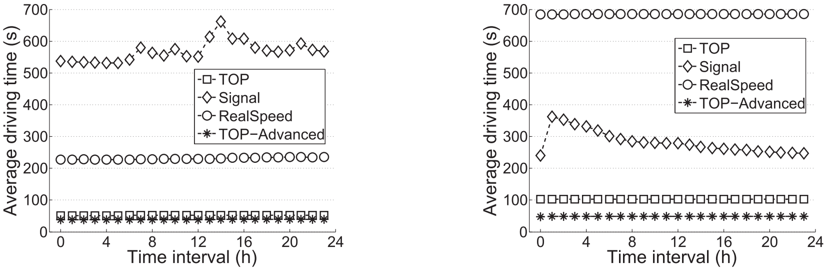
and calculates optimal vehicle speed aiming at reducing travel time through dynamic programming. However, since San Francisco has many uniformly distributed road segments with short lengths [33], the vehicle flow on the vehicle's scheduled route can be easily congested by vehicle flows from other intersected road segments. In contrast, the road segments in Rome have fewer intersections [4]. Therefore, the vehicle traffic in the road network is less likely to be congested than that in San Francisco.

Signal has the lowest average vehicle speed in Rome, but the second lowest performance in San Francisco. This is because *Signal* aims at reducing vehicles' travel time near intersections rather than in the global road network. In *Signal*, the vehicle flows on the road network are partitioned into several platoons of vehicles. By viewing each platoon as a job, the traffic management problem is formulated as a job scheduling problem at intersections. To minimize the time of vehicles passing the intersection, *Signal* utilizes the oldest-arrival-first scheduling algorithm. However, Rome's road segments are quite crowded at popular sites and have short distance [4], which make streets near popular sites heavily utilized. Locally minimizing the time at certain intersections inevitably exacerbates congestion at the other intersections. Therefore, *Signal* cannot achieve an optimal solution in the whole road network in Rome. In contrast, the road segment distribution of San Francisco is more uniform than that in Rome [33]. Therefore, *Signal* can better schedule vehicles passing through the intersections in San Francisco than in Rome, resulting in the better performance of *Signal* in San Francisco.

In *TOP-Advanced*, on the one hand, with the analysis result of the friendship among the vehicles, we can better estimate the probability of certain vehicles' presence on some road segments. Therefore, the vehicle densities on the road segments are more accurately estimated, which prevents much congestion from being generated in advance and meanwhile provides the optimal driving speed for the vehicles. However, by taking the delay caused by the traffic signals into account, the vehicles' arrival time on the positions of their trajectories are also more accurate, which further increases the accuracy of vehicle density estimation. Thus, the vehicles can drive with a relatively faster speed compared with the other methods and don't suffer from congestion.

5.2.2 Average Flow Rate. Figure 11(a) and Figure 11(b) show the average flow rate during different time intervals with the Rome and San Francisco traces, respectively. We see that for Rome, the average vehicle flow rates follow *TOP-Advanced> TOP> RealSpeed> Signal*. While for San Francisco, the results follow *TOP-Advanced> TOP> Signal> RealSpeed.*

The average vehicle flow rates follow the same trend as that of the average vehicle speed. Higher speed means that the vehicle flow can move faster on road segment as long as the vehicle density does not result in congestion. Although some road segments may be too crowded to let vehicles maintain high speeds, their flow rate is still large as long as their vehicle density does not exceed



(a) Rome.

(b) San Francisco.

Fig. 12. Driving delay optimization performance for vehicles over time.

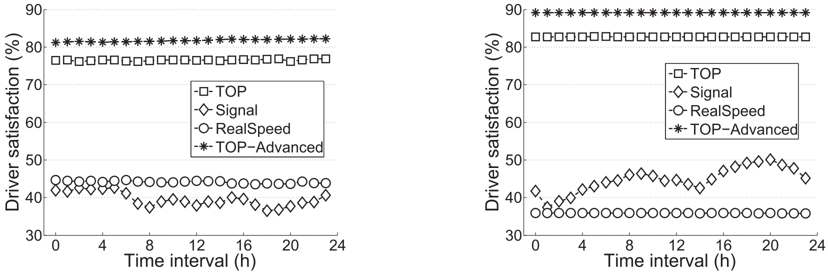
the jam level. Through comparing Figure 10(a) with Figure 11(a) and Figure 10(b) with Figure 11(b), respectively, we can see that although the average speed keeps above 15 km/h, the vehicle flow rate can be as low as 1 vehicle per hour. This shows that when the road network is non-congested, the vehicles in *Signal* and *RealSpeed* can drive as fast as possible (i.e., speed limit), which results in acceptable average driving speed. While without proactively avoiding congestion, the vehicle flows may generate congestion. As for *TOP-Advanced*, since the friendship among the vehicles helps increasing the accuracy of the estimated vehicle density on the road segments, the gaming process can more effectively avoid the generation of future road congestion. So the vehicles can drive with a high speed allowed by the road network in the near future. Meanwhile, since *TOP-Advanced* also considers the possible delay caused by traffic signals, the vehicle density on the road segments can be estimated in advance. As a result, the vehicles are constantly driving on the road segments with certain density, but no congestion. Therefore, the flow rate of *TOP-Advanced* is the highest among all the methods.

5.2.3 Average Driving Time. Figure 12(a) and Figure 12(b) show the average driving time during different time intervals with the Rome and San Francisco traces, respectively. We can see that for Rome, the average vehicle driving time follows *Signal>RealSpeed> TOP≈TOP-Advanced*. While for San Francisco, the average vehicle driving time follows *RealSpeed>Signal> TOP> TOP-Advanced.*

TOP always has the lowest driving time, because each vehicle can drive by a fast speed with low probability of suffering from congestion. *Signal* has the highest driving time in Rome, and the second highest driving time in San Francisco. Correspondingly, *RealSpeed* has the second highest driving time in Rome, and the highest driving time in San Francisco. These results are consistent with those of the average vehicle speed due to the same reasons. It is noticeable that in San Francisco, there is a heap between 0th hour and 1st hour. This is because there is a drop of speed during this time interval. When multiple vehicles simultaneously enter an intersection, but traffic signals cannot schedule their passing in time, the vehicles then wait in queues at the intersection. In *TOP-Advanced*, with considering the friendship among the vehicles and the possible delays caused by the traffic signals, the vehicles drive faster with higher densities on the road segments than the other methods, which results in shorter average driving time for all the road segments. But the improvement of the average driving time of the road segments in Rome is smaller than that in San Francisco. This is because the improvement of average driving speed in Rome is less than that in San Francisco, and Rome has longer average length of the road segments.

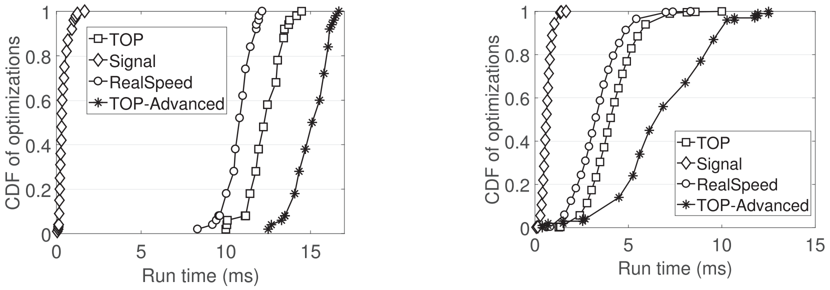
Recall

5.2.4 Average Driver Satisfaction. Figure 13(a) and Figure 13(b) show the average driver satisfaction during different time intervals with the Rome and San Francisco traces, respectively. We



(a) Rome. (b) San Francisco.

Fig. 13. Satisfaction of drivers over time.



(a) Rome. (b) San Francisco.

Fig. 14. Runtime of optimization processes.

can see that for Rome, the average satisfaction follows *TOP>RealSpeed>Signal*. While for San Francisco, the average satisfaction follows *TOP>Signal>RealSpeed*.

Driver satisfaction is jointly determined by vehicle speed and accident probability. Since the accident probability is calculated offline and does not change during vehicles' movement, drivers' satisfaction is solely determined by the vehicle speed. Since *TOP* generates the highest vehicle speed, it always ranks the highest and achieves over 80% satisfaction in both traces. In *TOP-Advanced*, since the vehicles' average driving speed is increased compared to *TOP*, it achieves the highest driver satisfaction. The satisfaction results are consistent with the average vehicle speed results due to the same reasons.

5.2.5 Runtime of Optimization Processes. Figure 14(a) and Figure 14(b) show the CDF of the runtime of all the optimization processes under different methods with the Rome and San Francisco traces, respectively. We can see that for both traces, the results generally follow *TOP-Advanced>TOP>RealSpeed>Signal*.

Since in *Signal*, each time when a vehicle passes an intersection with traffic signal, the central server applies the oldest-arrival-first scheduling algorithm to locally minimize the vehicle's traveling time on its route, it has the shortest runtime delay. In *TOP*, the vehicles exchange driving status information with the central server for three rounds in each gaming process. Therefore, under the same optimization period, it has higher runtime delays than *RealSpeed*. Since *TOP-Advanced* needs to further consider traffic signal and vehicle friendship for estimating vehicle density, it results in the highest runtime delay. However, considering that *TOP* generates a better vehicle flow rate optimization performance, the overhead is worthwhile.

5.3 Effectiveness of Each Enhancement Method

In this section, we test on the effectiveness of considering friendship among vehicles and traffic signal delay in improving vehicle density estimation and gaming efficiency (i.e., vehicle driving speed, road segment flow rate) with and without using the components. Specifically, in Section 5.3.1, we evaluate the performance of *TOP-Advanced* with and without considering the friendship among vehicles, respectively. In Section 5.3.2, we evaluate the performance of *TOP-Advanced* with and without considering traffic signal delays, respectively. In each of the following figures, the top figure is for Rome and the bottom figure is for San Francisco.

5.3.1 Friendship among Vehicles. To verify the effectiveness of considering the friendship among vehicles, we measured the CDF of the average driving speed of all the vehicles with and without utilizing the vehicles' friendship in gaming, respectively, which is shown in Figure 15. For Rome, we can see that about 50% of the vehicles have average driving speed higher than 45 km/h without considering vehicle friendship, and 70% of the vehicles have average driving speed higher than 45 km/h with considering vehicle friendship. We also note that with considering vehicle friendship, the fastest average driving speed that the vehicles can reach is around 60 km/h, while the one for the vehicles without considering vehicle friendship is only around 50 km/h. For San Francisco, we can see that about 50% of the vehicles have average driving speed higher than 20 km/h without considering vehicle friendship, and almost 90% of the vehicles have average driving speed higher than 20 km/h with considering vehicle friendship. With considering vehicle friendship, a small portion of the vehicles (i.e., less than 10%) can reach the average driving speed as high as around 90 km/h. Without considering vehicle friendship, the highest average driving speed the vehicles can reach is only around 50 km/h. This is because the vehicles' friendship offers hint on the presence probability of the vehicles on certain road segments, which improves the accuracy of the estimated vehicle density on the road segments. Then the gaming process driven by the improved estimation of the vehicle traffic can let the vehicles drive faster without facing congestion.

We also further measured the vehicle flow rate of the road segments, which is as shown in Figure 16. For Rome, about 50% of the road segments have vehicle flow rate higher than 50 veh/h without considering vehicle friendship, and around 90% of the road segments have vehicle flow rate higher than 50 veh/h with considering vehicle friendship. For San Francisco, about 50% of the road segments have vehicle flow rate higher than 40 veh/h without considering vehicle friendship, and around 70% of the road segments have vehicle flow rate higher than 40 veh/h with considering vehicle friendship. This is because the vehicles drive on the road segments more densely with increased speeds, resulting higher vehicle flow rates. The above observations confirm the effectiveness of utilizing vehicle friendship in both improving road network utilization efficiency and fulfilling the vehicles' travel expectations.

From Figure 16, we can also see that around 10% and 20% of the road segments in Rome and San Francisco have vehicle flow rate lower than 50 veh/h and 70 veh/h, respectively. This is because that the travel time of only around 78% of the road segments in Rome and around 85% of the road segments in San Francisco follows normal distribution (Section 3.4). Both *TOP* and *TOP-Advanced* cannot estimate the travel time with a sufficient accuracy on the road segments whose travel time does not follow normal distribution, which causes the vehicles' driving speed to be less optimized. We leave the development of more sophisticated road segment travel time estimation methods as our future work.

5.3.2 Traffic Signal Delays. To verify the effectiveness of considering the delays caused by red traffic signals in the estimation of the vehicles' arrival time to the positions of their trajectory, we measured the CDF of the average driving speed of all the vehicles with and without considering red traffic signal delay, respectively, which is shown in Figure 17. For Rome, we can see that about

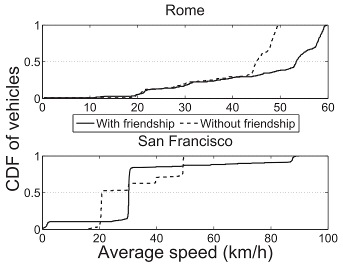


Fig. 15. Effectiveness of considering vehicle friendship on average vehicle driving speed.

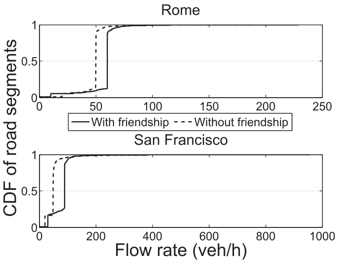


Fig. 16. Effectiveness of considering vehicle friendship on average road segment flow rate.

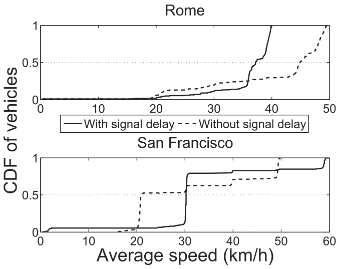


Fig. 17. Effectiveness of considering traffic signal delay on average road segment flow rate.

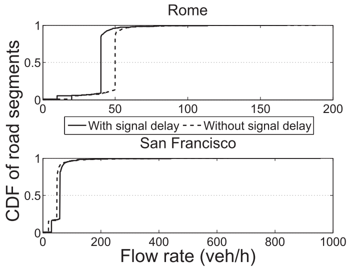


Fig. 18. Effectiveness of considering traffic signal delay on average vehicle driving speed.

50% of the vehicles have average driving speed higher than 38 km/h with considering red traffic signal delay, and about 75% of the vehicles have average driving speed higher than 38 km/h without considering red traffic signal delay. For San Francisco, we can see that about 25% of the vehicles have average driving speed higher than 30 km/h with considering red traffic signal delay, but around 50% of the vehicles have average driving speed higher than 30 km/h without considering red traffic signal delay. We can observe that considering the delay caused by red traffic signal makes the highest average driving speed achieved by the vehicles lower than that without considering red traffic signal delay. This is because the red traffic signal completely stops the vehicles' driving speed to 0, which prevents the vehicles from driving too fast on the road segments. We also notice that after considering red traffic signal delay, more vehicles are able to drive with average speed higher than 20 km/h in both Rome and San Francisco. This demonstrates that although the traffic signal reduces the vehicles' driving speed, considering its delay is beneficial for improving the gaming efficiency, thereby increasing the driving speed of overall vehicles.

In addition, we also measured the vehicle flow rate of the road segments, which is as shown in Figure 18. For Rome, about 50% of the road segments have vehicle flow rate higher than 40 veh/h with considering red traffic signal delay, and around 90% of the road segments have vehicle flow rate higher than 40 veh/h without considering red traffic signal delay. This is primarily caused by the reduced speed under red traffic signal. For San Francisco, the results of the two cases are quite close, which means through estimating vehicle density with taking into account the traffic signal, the gaming can achieve a high efficiency for the road network.

6 CONCLUSION

Previous works for speed optimization does not proactively avoid the generation of congestion in the future. We proposed *TOP*, a vehicle trajectory–based driving speed optimization strategy aiming at minimizing each vehicle's travel time while avoiding generation of congestion. Our analysis on the vehicle mobility and congestion based on two real-world traces support the motivation for the design of *TOP*. *TOP* uses vehicle trajectories to estimate the vehicle density of each road segment in the near future. Additionally, we consider the friendship among the vehicles and the delay caused by red traffic signal in the estimation of the vehicle density of the road segments. Then, by using a non-cooperative Stackelberg game between each vehicle and the central server, the vehicle's driving speed is optimized so that it can drive as fast and safely as possible while proactively avoiding generating congestion. We have conducted extensive experiments based on the two traces. The experiment results validate the high effectiveness of *TOP* and its superior performance compared to previous methods in terms of the utilization of road network, congestions, and driver satisfaction. In our future work, we plan to consider vehicles' social relationship in avoiding road congestion and develop more reasonable schemes to motivate vehicle cooperation.

REFERENCES

[1] 2015. Rome Accident Statistics. Retrieved July 29, 2015 from http://www.telegraph.co.uk.

[2] 2015. San Francisco Accident Statistics. Retrieved July 29, 2015 from http://www.city-data.com.

[3] Mohamed A. Abdel-Aty and A. Essam Radwan. 2000. Modeling traffic accident occurrence and involvement. *Accident Analysis & Prevention* 32, 5 (2000).

[4] Raul Amici, Marco Bonola, Lorenzo Bracciale, Antonello Rabuffi, Pierpaolo Loreti, and Giuseppe Bianchi. 2014. Performance assessment of an epidemic protocol in VANET using real traces. In *Proceedings of the International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT'14)*.

[5] Behrang Asadi and Ardalan Vahidi. 2011. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Trans. Contr. Syst. Technol.* 19, 3 (2011).

[6] Fan Bai and Bhaskar Krishnamachari. 2009. Spatio-temporal variations of vehicle traffic in VANETs: Facts and implications. In *Proceedings of the Vehicular Internetworking (VANET'09)*.

[7] Kenneth George Binmore. 1982. *Mathematical Analysis: A Straightforward Approach*. Cambridge University Press.

[8] Elmar Brockfeld, Robert Barlovic, Andreas Schadschneider, and Michael Schreckenberg. 2001. Optimizing traffic lights in a cellular automaton model for city traffic. *Phys. Rev. E* 64, 5 (2001).

[9] Kang Chen, Haiying Shen, and Li Yan. 2015. Multicent: A multifunctional incentive scheme adaptive to diverse performance objectives for DTN Routing. *IEEE Trans. Parallel Distrib. Syst.* 26, 6 (2015).

[10] Liuhua Chen and Haiying Shen. 2014. Consolidating complementary VMs with spatial/temporal-awareness in cloud datacenters. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'14)*.

[11] Wenping Chen, Sencun Zhu, and Deying Li. 2010. VAN: Vehicle-assisted shortest-time path navigation. In *Proceedings of the Mobile Ad-hoc and Sensor Systems (MASS'10)*.

[12] Kakan C Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrur Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. 2016. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). *IEEE Trans. Intell. Transport. Syst.* 17, 2 (2016).

[13] E. Freitas, Catarina Mendonça, Jorge A. Santos, Carla Murteira, and J. P. Ferreira. 2012. Traffic noise abatement: How different pavements, vehicle speeds and traffic densities affect annoyance levels. *Transport. Res.* 17, 4 (2012).

[14] Cyril Furtlehner, Jean-Marc Lasgouttes, and Arnaud de La Fortelle. 2007. A belief propagation approach to traffic prediction using probe vehicles. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC'07)*.

[15] Noortje Groot, Bart De Schutter, and Hans Hellendoorn. 2015. Toward system-optimal routing in traffic networks: A reverse Stackelberg game approach. *IEEE Trans. Intell. Transport. Syst.* 16, 1 (2015).

[16] Ryan Herring, Aude Hofleitner, Pieter Abbeel, and Alexandre Bayen. 2010. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC'10)*.

[17] Jaehoon Jeong, Shuo Guo, Yu Gu, Tian He, and David HC Du. 2010. TSF: Trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'10)*.

[18] Jaehoon Jeong, Shuo Guo, Yu Gu, Tian He, and David HC Du. 2011. Trajectory-based data forwarding for light-traffic vehicular ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.* 22, 5 (2011).

[19] Jaehoon Jeong, Shuo Guo, Yu Gu, Tian He, and David HC Du. 2012. Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery. *IEEE Trans. Mobile Comput.* 11, 10 (2012).

[20] Qing-Jie Kong, Qiankun Zhao, Chao Wei, and Yuncai Liu. 2013. Efficient traffic state estimation for large-scale urban road networks. *IEEE Trans. Intell. Transport. Syst.* 14, 1 (2013).

[21] Anastasios Kouvelas, Konstantinos Aboudolas, Markos Papageorgiou, and Elias B. Kosmatopoulos. 2011. A hybrid strategy for real-time traffic signal control of urban road networks. *IEEE Trans. Intell. Transport. Syst.* 12, 3 (2011).

[22] Li Li, Ding Wen, and Danya Yao. 2014. A survey of traffic control with vehicular communications. *IEEE Trans. Intell. Transport. Syst.* 15, 1 (2014).

[23] Ruimin Li, Huajun Chai, and Jin Tang. 2013. Empirical study of travel time estimation and reliability. *Mathematical Problems in Engineering* 2013 (2013). https://scholar.googleusercontent.com/scholar.bib?q=info:3cFKWMZCU0kJ:scholar.google.com/&output=citation&scisdr=CgXIhcG2ELWP2EWYFII:AAGBfm0AAAAAXaidDIIUWM8gcZfgb0np4wuT278OH4Cx&scisig=AAGBfm0AAAAAXaidDOKMJ8fHtRK-pOrtqEwuclczdiuC&scisf=4&ct=citation&cd=0&hl=en.

[24] Ruimin Li, Geoffrey Rose, and Majid Sarvi. 2006. Using automatic vehicle identification data to gain insight into travel time variability and its causes. *J. Transport. Res. Board* 1945, 1 (2006), 24–32.

[25] Shu Lin, Bart De Schutter, Yugeng Xi, and Hans Hellendoorn. 2013. Integrated urban traffic control for the reduction of travel delays and emissions. *IEEE Trans. Intell. Transport. Syst.* 14, 4 (2013).

[26] Yuhua Lin and Haiying Shen. 2016. VShare: A wireless social network aided vehicle sharing system using hierarchical cloud architecture. In *Proceedings of the ACM/IEEE International Conference on Internet of Things Design and Implementation (IoTDI'16)*.

[27] Jinwei Liu, Lei Yu, Haiying Shen, Yangyang He, and Jason Hallstrom. 2015. Characterizing data deliverability of greedy routing in wireless sensor networks. In *Proceedings of the International Security Exhibition and Conference (SECON'15)*.

[28] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. 2008. Data aggregation and roadside unit placement for a VANET traffic information system. In *Proceedings of the Vehicular Internetworking workshop (VAINET'10)*.

[29] Frank J. Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *J. Am. Stat. Assoc.* 46, 253 (1951).

[30] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. 2004. TrafficView: Traffic data dissemination using car-to-car communication. *SIGMOBILE* 8, 3 (2004).

[31] Engin Ozatay, Simona Onori, James Wollaeger, Umit Ozguner, Giorgio Rizzoni, Dimitar Filev, John Michelini, and Stefano Di Cairano. 2014. Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution. *IEEE Trans. Intell. Transport. Syst.* 15, 6 (2014).

[32] Kartik Pandit, Dipak Ghosal, H. Michael Zhang, and Chen-Nee Chuah. 2013. Adaptive traffic signal control with vehicular ad hoc networks. *IEEE Trans. Vehic. Technol.* 62, 4 (2013).

[33] Michał Piórkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. A parsimonious model of mobile partitioned networks with clustering. In *Proceedings of the International Conference on Communication Systems and Networks (COMSNETS'09)*.

[34] Chenxi Qiu, Haiying Shen, and Liuhua Chen. 2015. Towards green cloud computing: Demand allocation and pricing policies for cloud service brokerage. In *Proceedings of the Annual Conerence on Big Data*.

[35] Ankur Sarker, Chenxi Qiu, Haiying Shen, Andrea Gil, Joachim Taiber, Mashrur Chowdhury, Jim Martin, Mac Devine, and A. J. Rindos. 2016. An efficient wireless power transfer system to balance the state of charge of electric vehicles. In *Proceedings of the International Conference on Parallel Processing (ICPP'16)*.

[36] Haiying Shen and Zhuozhao Li. 2014. New bandwidth sharing and pricing policies to achieve a win-win situation for cloud provider and tenants. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'14)*.

[37] Sha Tao, Vasileios Manolopoulos, Saul Rodriguez, and Ana Rusu. 2012. Real-time urban traffic state estimation with A-GPS mobile phones as probes. *J. Telemed. Telecare* 2, 01 (2012).

[38] Yu-Tian Tseng, Rong-Hong Jan, Chien Chen, Chu-Fu Wang, and Hsia-Hsin Li. 2010. A vehicle-density-based forwarding scheme for emergency message broadcasts in VANETs. In *Proceedings of the Mobile Ad-hoc and Sensor Systems (MASS'10)*.

[39] Wantanee Viriyasitavat, Ozan K. Tonguz, and Fan Bai. 2011. UV-CAST: An urban vehicular broadcast protocol. *IEEE Commun. Mag.* 49, 11 (2011).

[40] Bo Wu, Haiying Shen, and Kang Chen. 2015. Exploiting active sub-areas for multi-copy routing in VDTNs. In *Proceedings of the International Conference on Computer Communications and Networks (ICCCN'15)*.

[41] Yuchen Wu, Yanmin Zhu, and Bo Li. 2011. Trajectory improves data delivery in vehicular networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'11)*.

[42] Fulong Xu, Shuo Guo, Jaehoon Jeong, Yu Gu, Qing Cao, Ming Liu, and Tian He. 2011. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'11)*.

[43] Yanyan Xu, Qing-Jie Kong, Shu Lin, and Yuncai Liu. 2012. Urban traffic flow prediction based on road network model. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC'12)*.

[44] Li Yan, Haiying Shen, and Kang Chen. 2015. TSearch: Target-oriented low-delay node searching in DTNs with social network properties. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'15)*.

[45] Li Yan, Haiying Shen, and Kang Chen. 2017. MobiT: A distributed and congestion-resilient trajectory based routing algorithm for vehicular delay tolerant networks. In *Proceedings of the ACM/IEEE International Conference on Internet of Things Design and Implementation (IoTDI'17)*.

[46] Ziqi Ye, Kailai Li, Michael Stapelbroek, Rene Savelsberg, Marco Günther, and Stefan Pischinger. 2018. Variable step-size discrete dynamic programming for vehicle speed trajectory optimization. *IEEE Trans. Intell. Transport. Syst.* 99 (2018).

[47] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[48] Junfeng Zhao, Wan Li, Junmin Wang, and Xuegang Ban. 2016. Dynamic traffic signal timing optimization strategy incorporating various vehicle fuel consumption characteristics. *IEEE Trans. Vehic. Technol.* 65, 6 (2016).

Received October 2018; revised August 2019; accepted September 2019