

Towards Efficient Vacant Taxis Cruising Guidance

Y. Hou[†], X. Li[†], *IEEE Member*, Y. Zhao^{*}, X. Jia[†], A. W. Sadek^{*}, K. Hulme[^] and C. Qiao[†], *IEEE Fellow*

[†]Department of Computer Science and Engineering, State University of New York at Buffalo, USA

^{*}Department of Civil, Structural and Environmental Engineering, State University of New York at Buffalo, USA

[^]The New York State Center for Engineering Design and Industrial Innovation, USA

Abstract-Despite the existence of taxi dispatch systems which assign taxis to pick up passengers after receiving dispatch requests in large cities such as New York City or Shanghai, most taxis still look for customers while cruising on the road without receiving any formal guidance from taxi dispatch systems. Different from the conventional operation mode in existing taxi dispatch systems, in this paper, we envision a new Cyber-technology enabled taxi dispatch system which can efficiently provide vacant taxis with cruising route suggestions, *not to respond to any specific pick-up request but instead, hoping to find prospective customers* (and accordingly, this is complementary to the conventional operation mode). We address the *Taxi Cruising Guidance* (TCG) problem with the objective being to minimize the *Global Vacant Rate* (GVR), which is defined as the ratio of traveling miles with no passenger onboard, to the total traveling miles in a given time period. We propose a number of heuristic solutions and conduct comprehensive performance evaluations based on large-scale simulations. A case study is also presented by utilizing real traces collected from taxis in the city of Shanghai. As part of our research, we leverage a well-known microscopic traffic simulator (called TRANSIMS) to demonstrate that the application of TCG is also beneficial to traffic management.

Keywords-Vacant Taxi Cruising; Taxi Dispatch; Intelligent Transportation Systems; Real Case Study; Heuristic Algorithms; Microscopic Traffic Simulator

I. INTRODUCTION

A taxi is an indispensable component of urban transportation systems, especially for large cities such as New York City, Shanghai, Chicago, and many others. Compared to other public transit systems such as bus/metro services, a taxi can serve individuals in a private-service manner and has advantages in terms of time savings and convenience, since individuals can use taxi services between any two locations. In particular, an effective taxi dispatching system is essential for both taxi companies/drivers and traffic authority, because: 1) from a profit or revenue point of view, taxi companies and drivers are always interested in maximizing their revenue/profit and minimizing their cost,

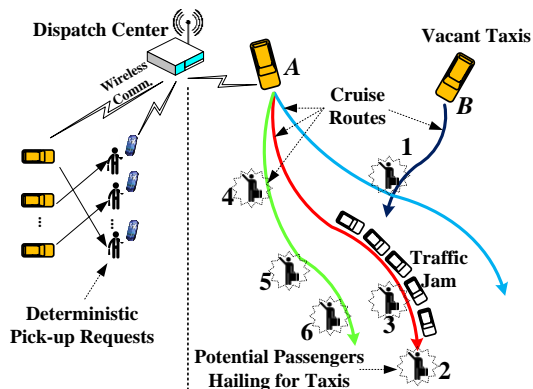


Fig.1. The existing taxi dispatch operation mode (left) and the additional *cruising route suggestion* mode (right) in new taxi dispatching systems.

and cruising for finding customers contributes significantly to the cost, i.e., a vacant taxi needs to find next passenger as soon as possible; 2) from a traffic management point of view, cruising taxis add to the total traffic load on the network, without providing any passenger transport benefit. For example, a published report shows that vacant taxis constituted about 40% of the overall traffic flow along the major roads in Beijing [16], particularly during peak traffic hours.

By investigating how current taxi systems operate, we make the following observations, which serve as motivation for the present work:

Observation 1: Most existing taxi dispatch systems focus on the deterministic case, in which the major technical issue concerns the matching problem between the received pick-up requests from passengers and vacant taxis, as shown in the left side of Fig. 1. Accordingly, the most popular principle is to assign a nearest taxi that can pick up a passenger either with *a)* the minimum fuel cost for the taxi or *b)* the minimum waiting time for the passenger.

Observation 2: Most taxis in large cities, do not rely on taxi dispatch systems because they mainly find their customers when cruising on the road.

Observation 3: Recent research in data mining has studied how to predict where and when prospective passengers may appear by analyzing historical GPS trajectories of taxis [6, 7, 10, 12]. However, if each every vacant taxi individually applies such knowledge when deciding its cruising route, many taxis may cruise along the same streets, competing for passengers.

Observation 4: Most existing dispatch systems do not explicitly consider traffic conditions, so a vacant taxi may be assigned to pick up a passenger, only to end up being stuck in a traffic jam.

In this paper, instead of dealing with the deterministic case (Observation 1), we study a new taxi dispatch paradigm focusing on how to help cruising taxis identify passengers (Observation 2), by not only utilizing the knowledge about the possible passenger appearances (Observation 3) but also explicitly considering the current traffic conditions (Observation 4). In particular, we address a new optimization problem called *Taxi Cruising Guidance* (TCG), whose objective is to minimize taxis' *Global Vacant Rate* (GVR), defined to be the ratio of traveling miles with no passenger onboard to the total traveling miles in a given time period.

The proposed TCG has several technical issues and challenges. As shown on the right side of Fig. 1, vacant taxi *A* has three candidate cruise routes. However, if following the blue route, taxis *A* and *B* may compete for a passenger at location 1. In the meantime, it is not desirable to follow the red route, because taxi *A* could be stuck in a traffic jam, even though it may pick up passengers at locations 2 and 3. By comparison, it is a good choice to follow the green route, because *a*) there are many possible passenger appearances along the route (e.g., at locations 4, 5, 6) so that it is easier (or has higher probability) for taxi *A* to identify prospective passengers, and *b*) traffic conditions on this route are also more favorable than that on the red route.

The major contributions of this paper are as follows:

1. To the best of our knowledge, no existing work has investigated how to assist cruising taxis in finding passengers using cyber technologies. This work, for the first time, has addressed the mutual impacts/interactions between taxi cruising guidance and traffic conditions.

2. We introduce a number of solutions to TCG and perform a large-scale evaluation of the technique.

3. We present a case study in the city of Shanghai by using real trace data collected from taxis, and provide useful insights.

4. We utilize a well-known traffic simulator, called TRANSIMS, to demonstrate that the application of TCG is also beneficial for traffic management.

The rest of the paper is organized as follows. In Sec. II, we present the assumptions and formal problem statement. The detailed solutions are presented in Sec. III, followed by a performance evaluation in Sec. IV. We present a case study in Sec. V, and conduct a field test on a traffic simulator in Sec. VI. We discuss related work in Sec. VII, and Sec. VIII presents conclusions for this paper.

II. THE TAXI CRUISING GUIDANCE (TCG) PROBLEM

In this section we define the proposed TCG problem.

A. System and Operation Overview

We aim to design a solution to be run by a taxi Dispatch

Center (DC) to provide taxi cruising guidance to vacant taxi every T (e.g., $T=2$) hours. More specifically, as long as a taxi becomes vacant (i.e., in unloaded/vacant status), it will send a request to DC for a Cruising Route Suggestion (CRS). Normally, a CRS includes a cruise route and a *short-term* target destination (with a predefined waiting period of t_{hold} time units at that destination). Note that as a special case, the target destination could be a taxi stand (available in some cities), where the taxi can park and wait for a maximum of $t_{hold} < T$ time units for the next passenger to show up.

After it receives such a CRS from DC, the taxi will follow the CRS until *a*) it either finds a passenger (in which case the taxi would notify the DC about its status change, and subsequently, it would not be given any new CRS by the DC until the taxi becomes vacant again) or *b*) it sends another request to the DC for a new CRS after it fails to find any passenger after arriving at the destination indicated by the current CRS (and waiting for t_{hold} time units at that destination).

On the DC side, it is assumed that the DC has the knowledge about real-time traffic conditions and the expected/possible number of passengers that may appear on a given link during the time period T (which could be inferred based on historical GPS trajectories of taxis as studied in the existing works [6, 7, 9, 15]). In addition, DC can receive real-time GPS data from taxis, which indicates their current locations, speed and load (or occupancy) status.

B. System Model and Notations

The road network is modeled by a graph $G=(V, E)$, where V represents the set of intersections, and E the set of links (or road segments), which is denoted by $\{e_i\}$, $i=1,2,\dots,m$. Each link e_i is represented by a 3-tuple $e_i=(l_i, tt_i, tp_i)$, where l_i and tt_i are the length and travel time of e_i according to the current traffic condition, respectively. tp_i is the estimated total number of passengers that may appear on link e_i during the next immediate time period T , which is known to the DC based on the analysis of historical information as mentioned earlier.

Each passenger u_j hailing for a taxi is also represented by a 3-tuple $u_j=(ta_j, mw_j, dst_j)$. Here, ta_j is the time when u_j tries to hail a taxi on a link, which is unknown to the DC, and mw_j is the maximum tolerable waiting time of u_j (also unknown but can be set to an estimated value by the DC). Ideally, the DC would like to send a taxi to pick up passenger u_j before she decides to give up taking taxi and choose other transportation means (e.g, walking or taking bus/metro). Lastly, dst_j is the passenger's destination (although it is not a concern in the TCG problem), which also indicates the location where the taxi backs to vacant.

For a given vacant taxi x_k , it is assumed that it always follows the CRS from the DC. For simplicity, it is assumed that when x_k is cruising on link e_i encountering multiple passengers hailing for a taxi, x_k will randomly choose one such passenger to serve.

C. Problem Statement

Here, we give the formal description of the TCG problem.

First, we note that the driving distance (mileage) is the most essential factor affecting both the cost (in terms of fuel and the taxi driver’s time at work) and revenue. During time period T , for a given taxi x_k , let the cruise mileage be cm_k and the total mileage be tm_k . We define taxi x_k ’s vacant rate (denoted by vr_k) to be:

$$vr_k = \frac{cm_k}{tm_k}$$

Accordingly, the *Global Vacant Rate* (GVR) of all the taxis in the system is the average of all the vr_k ’s.

The Taxi Cruising Guidance (TCG) Problem: Given a set of taxis, decide CRS for each requesting vacant taxi during time period T , such that GVR can be minimized.

It is worth noting that although we introduced a general model in Sec. II.A by defining a predefined waiting period of t_{hold} time units for a vacant taxi after arriving at the destination following the previous CRS, in the following, we mainly consider the situation where $t_{hold}=0$ in order to simplify several practical considerations as follows. First, except at airports, and train/bus stations, it is usually not an effective approach for a vacant taxi to park at a place waiting for prospective passengers. In addition, not all cities have dedicated taxi stands, and the taxis are usually not allowed to park exactly where it drops off the previous passenger to wait for prospective passengers due to strict traffic regulations, especially in the downtown area. Therefore, a vacant taxi would most likely have to cruise along after dropping off a passenger, unless there is already another passenger hailing for the taxi at the same place where the previous passenger is dropped off (in which case, the taxi does not even need a new CRS until later).

III. SOLUTION DESIGN FOR TCG

In this section, we present a list of solutions to TCG, which have different methodologies and design principles. Due to limited space, we only outline the major steps of each algorithm and omit other inessential details.

A. Uncoordinated Cruising Strategy (UCS)

This is a baseline solution, which will be mainly used for comparison. Instead of relying on CRS sent from DC, each vacant taxi individually selects a random destination without any coordination between each other and then follows a shortest path to the destination as her cruising route. Note that, since the destinations are randomly selected among all locations, the issue of excessive supply of taxis for a given link can be alleviated to some extent.

B. Traditional Dispatch-based Strategy (TDS)

Traditionally, the DC will dispatch a nearest vacant taxi to serve a pick-up request. When the DC does not know for sure where exactly a taxi can pick up a passenger, but only has some general knowledge about where prospective

passengers may appear, one may still apply the principle of the traditional dispatch-based strategy in order to come up with a CRS for a taxi as follows: first, route a vacant taxi to an adjacent link which statistically will have higher probability to find passengers, and if it fails to pick up a passenger, route the taxi to the next adjacent link and so on, in a hop-by-hop fashion. Accordingly, in TDS, we consider an adjacent link e_i as a candidate link if the DC has not routed tp_i taxis to e_i (in order to alleviate the excessive supply of taxis for this link). If such candidate links exist, a taxi will be routed to the one with the largest tp_i because the larger the tp_i is, the higher probability that a taxi can find a passenger on that link. If such a candidate link does not exist, the DC will randomly select an adjacent link for the taxi to cruise.

C. Accumulated Probability-based Strategy (APS)

In TDS, this solution always selects an adjacent link as the next destination for a given taxi (i.e., it operates in a one-hop fashion), and as such, TDC cannot guarantee that it can essentially increase the probability that the taxi can find a passenger even after cruising for a long time (in other words, TDC is similar to the classic greedy algorithm which suffers from local maximum issue). By comparison, the proposed *Accumulated Probability-based Strategy* (APS) approaches the problem from a long-term or multi-hop standpoint and aims to suggest a vacant taxi with a relatively remote destination and the corresponding cruise route. In particular, APS will suggest a cruise route for a given taxi, which has the *highest accumulated probability* for finding a passenger along that route. Similar to TDS, APS estimates such a probability based on the value of tp_i of links. However, it uses an elaborate algorithm to determine a destination and route by considering both the traffic condition and competition among the vacant taxis. More specifically, APS works as follows:

Step 1) At DC, the algorithm maintains a *global dispatch graph* (DG) that will be updated when taxis’ status changes from vacant to loaded, or vice versa, and in which the weight of each edge e_i is set as its travel time tt_i , according to the current traffic condition. Note that, the output of every update process is a set of CRS between any two locations. To come up with the set of CRS, the update process considers the following two major aspects jointly: traffic condition and competition among the vacant taxis.

When considering the traffic condition factor, the basic idea is to calculate a cruise route between any two locations with shortest travel time, which can be supported by the well-known Dijkstra’s algorithm. However, a pure Dijkstra’s algorithm cannot handle the non-trivial issue of competing passengers between taxis. Accordingly, we design a TCG component and incorporate it into the existing Dijkstra’s algorithm.

The design of this component has two strategies to alleviate competition among multiple taxis: The first strategy involves defining a dynamic count-down counter c_i for each link e_i to better control the number of taxis that

have been suggested to cruise on link e_i ; initially, c_i is set by $c_i = tp_i$. (How to dynamically update the value of c_i will be discussed in Step 3). Generally speaking, unlike TDS, in which a taxi will not be routed to a link if DC has already suggested tp_i taxis to cruise on that link, APS may keep including link e_i in the cruise routes of different taxis so long as $c_i > 0$. This is because it is not guaranteed that each of tp_i taxis can definitely pick up a passenger on e_i . The second strategy to alleviate taxi competition involves ensuring that the time span between the arrival times of two consecutive taxis on link e_i is longer than a time interval $span_i$. This is because the DC does not know that when passengers actually would appear on e_i during T , so we should avoid suggesting too many taxis to cruise on the same link in a short time. For example, at current time $t \in [0, T]$, a simple way to set $span_i$ is given by $span_i = (T-t)/c_i$ by assuming that the passenger appearances over the remaining time period (i.e., $T-t$) follows a uniform distribution. However, it is worth noting that this procedure can be further improved once we have more detailed information regarding when passengers typically appear on a given link (or its probability distribution).

To implement the above strategies, we incorporate the above TCG component into the Dijkstra's algorithm as follows: In the classical Dijkstra's algorithm, for the current checked node, the algorithm considers all of its unvisited neighbors and calculate their tentative distances. In our modified version, for example, if the current node A is marked with a distance of 10, and the edge connecting it with a neighbor B has length 5, then the distance to B (through A) will be $10+5=15$. Now, if 1) this distance is less than the previously recorded tentative distance of B (which is the original constraint in Dijkstra's algorithm); 2) $c_{A \rightarrow B} > 0$, i.e., there are still passengers to be picked up at the link between A and B ; and 3) the time span between previous and next assigned taxis that arrive at link $A \rightarrow B$ is larger than $span_{A \rightarrow B}$, then overwrite that distance. After Step 1, we will have a CRS for any two given locations.

Step 2) For a vacant taxi x_k , we choose a set of relatively remote destinations for x_k as candidate destinations, e.g., around w minutes travel time from the current location of x_k (w is an empirical parameter, e.g., 15 minutes in our simulation, and how to select an optimal w is out of the scope of this paper and deserves a separate study). Then, we choose a destination for x_k such that the corresponding cruise route (already calculated in Step 1) between x_k and this new destination has the largest value of $\sum c_i$, i.e., such a cruise route having the largest accumulated number of passenger appearances, which is similar to have highest accumulated probability for x_k to find a passenger.

Step 3) Once a new destination and a cruise route are selected for x_k , we will update c_i by $c_i = c_i - 1$ for all the links in CRS (just like a reservation process), then go back to Step 1 to process the next CRS request from another vacant taxi. In addition, we will re-update c_i by $c_i = c_i + 1$ in a later time if x_k fails to find a passenger during cruising on a specific link

e_i (which can be identified from its load status information in real-time GPS data sent to DC).

D. Busy-link Dominant Strategy(BDS)

While APS considers both issues related to the traffic condition and competition among the vacant taxis for every link, in this solution, we propose to differentiate the links by slightly decoupling these two issues in order to put more emphasis on links having a high tp_i (which will be called "busy link" as opposed to "common link"). More specifically, the proposed *Busy-link Dominant Strategy* (BDS) will only consider the traffic conditions on the common links having a relatively low tp_i , and not on the "busy links" since the traffic is expected to be heavy on such busy links, but even so, it is worthwhile for vacant taxis to go there because the chance of being able to pick up a prospective passenger there is high. This tends to favor busy links and accordingly, more vacant taxis may be directed to the busy links. In addition, considering the traffic conditions on the common links along the way to a busy link will also help vacant taxis get to their busy links as quickly as possible.

In the meantime, BDS considers the issue related to the competition among the vacant taxis only for busy links, but will not be concerned about this issue for common links. This is because one still needs to avoid sending too many vacant taxis to any given busy link to prevent the taxis from competing for prospective passengers on these busy links. Accordingly, in BDS, once a vacant taxi is assigned to a new destination among one of the busy links, BDS will then select a cruise route with a short travel time according to the real-time traffic conditions, without considering the competition issue along the way to its destination any more. More specifically, BDS works as the follows:

Step 1) We first decide whether a link is a busy link, which normally depends on historical data. For example, in our simulation, we consider links that are either near business district or transportation center as busy links.

Step 2) We reuse Step 1 of APS to maintain a global dispatch graph DG. While APS considers the issue of multiple taxi competing with one another on each and every link, in BDS, we consider such an issue only for busy links as follows: we modify the Step 1 of APS by setting $c_i = +\infty$ for all the common links, so that only the traffic condition issue will be considered on those links.

Step 3) We select a new destination for a given taxi from a set of busy links (Note that, the route to any destination has already been determined in Step 2, which is similar to Step 1 of APS). While all the busy links are considered as candidate destinations for a given taxi, we consider the following two selection criteria: *a)* in order to send a vacant taxi to a busy link as soon as possible, we rank the busy links in terms of the travel time to them, and will only keep the top β % of the busy links (where β is a configurable parameter and set to ~ 10 in our simulation for example) that have the least travel time from the taxi's current location for further consideration; and then *b)* in an effort to provide fair

Table I: Default values of experiment parameters

Road Network	10×10 Grid
Length of A Link	[500m, 2500m]
Avg. Travel Speed on Each Link	[15, 45] km/h
Time Period (T)	2 Hours
Total Simulation Time for a Given Setting	100 Hours
tp_i on Common Link	[2, 10]
Percentage of Busy Link	~9%
Maximum Tolerable Waiting Time (mw_i)	5 min
Number of Taxi	100

service to prospective passengers, we want to avoid sending all vacant taxis to the same busy link. So, among the remaining candidate busy links selected by a), we will choose the one has the largest value of c_i (as defined in the Step 1 of APS) as the destination. Note that, given the nature of the proposed TCG route, a vacant taxi may pick up prospective passengers along the way to such destination, and hence, prospective passengers appearing on common links (i.e, non-busy links) will also receive taxi service.

Step 4) This step is the same as Step 3 of APS.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed solutions to TCG problem. On a hypothetical scenario (the following sections will describe performance evaluation with real taxi traces (Sec. V) and a detailed microscopic traffic simulator (Sec. VI)). In our simulation here, we focus on a road network in downtown area represented by a 10×10 grid (about 13.5km×13.5km). Table I shows the default value of the parameters in our simulation.

In our experiment, we simulate TCG for every two hours and the total simulation time for a given parameter setting is 100 hours. For a given link e_i , once its tp_i has been decided, the passengers will randomly appear on e_i across the whole time period T . In particular, we select a subset of the links (~9%) as busy links, which represent the links near business district or transportation center, and accordingly have higher tp_i (around 3~5 times more than common links).

Fig. 2(a) shows the performance of proposed strategies with different number of taxis. Generally speaking, BDS performs the best, which indicates that taking the taxi dispatch issue and the traffic condition as the dominant factors on busy and common link respectively is an effective approach. APS also yields reasonable performance, compared to UCS and TDS, both of which do not consider any coordination between taxis or do not apply a sophisticated dispatch strategy in terms of providing good CRS. In the meantime, it can be seen that the GVR of all the solutions increase as the number of total taxis increases because the more taxis in the system, the more difficult it is for vacant taxis to find passengers, i.e., taxi supply is excessive given a limited number of passengers.

Counter to Fig. 2 (a), Fig. 2(b) compares the GVR of the proposed strategies as the upper bound of tp_i increases

(whose maximum default value is 10 as shown in Table I). It is shown that with more passengers, the GVR of all the solutions decrease. In particular, the relative performances between different solutions become less obvious under a large value of tp_i , which means there are always a lot of passengers to be picked up on the road and therefore even a naïve solution may be capable of assisting taxis in finding passengers.

In Fig. 2(c), we investigate how passenger’s maximum tolerable waiting time can affect GVR. Not surprisingly, as the tolerance of passengers to waiting increases, the GVR of all the solutions can be improved, because, in this case, passengers have a higher probability to be picked up by vacant taxis before choosing other transit services.

V. A CASE STUDY IN SHANGHAI

In this section, we evaluate the performance of the solution algorithms to the TCG problem by using real taxi traces (i.e., historical GPS data) collected from Shanghai.

We build our test scenarios as follows. First, to simplify the road network, we plot the entire taxi traces set and build an overlay road network that overlaps with many of those traces. Such an overlay road network *may* still be somewhat different from the real-world arterial road network because many taxis may have chosen to travel on other or lower-class roads. Second, to model realistic passenger appearances, we utilize the load status information included in the GPS data. In particular, for a given taxi and a given time t , if its status changes from vacant to loaded, this means that the taxi has just picked up a passenger on a specific road (which can also be identified through the longitude and latitude information in the GPS data). Since a passenger may already have waited for some time before getting picked up, in our simulation, a passenger will be generated and appear on that link at a point in time, randomly selected from within the time period $[t-mw_i, t]$. In addition, we identify the destination of the current passenger when the taxi status changes from loaded to vacant. Finally, because the real traces of taxis are fixed and hence cannot be directly used in our evaluation and simulation, we needed to artificially generate a number of taxis which can change their routes in any time according to the received CRS. Besides the modifications just noted, the other parameter settings and the simulation setup were the same as the ones presented in Sec. IV, with the exception that the simulation was from 8AM to 1PM in this case, and the GVR was calculated on an hourly basis. Fig. 3 (a)-(c) show how the GVR changed over time, and in response to changes in some of the parameters of the TCG problem.

Fig. 3(a) first shows the total number of passenger appearances we retrieved from the taxi traces at different times during the 8AM-1PM time period. It can be seen that during 9AM-11AM, there are more passenger requests compared to other hours, e.g., 12AM-1PM. Surprisingly, the highest passenger appearances did not come from 8AM-9AM, which is a typical peak hour. This is probably because

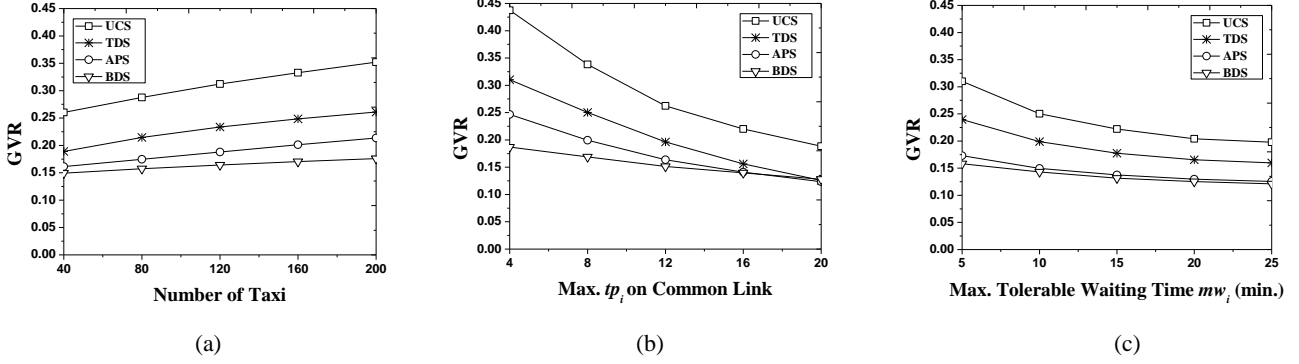


Fig. 2. Testing results (a) Performance under different number of taxi (b) Performance under different maximum tp_i (c) Performance under different mw_i

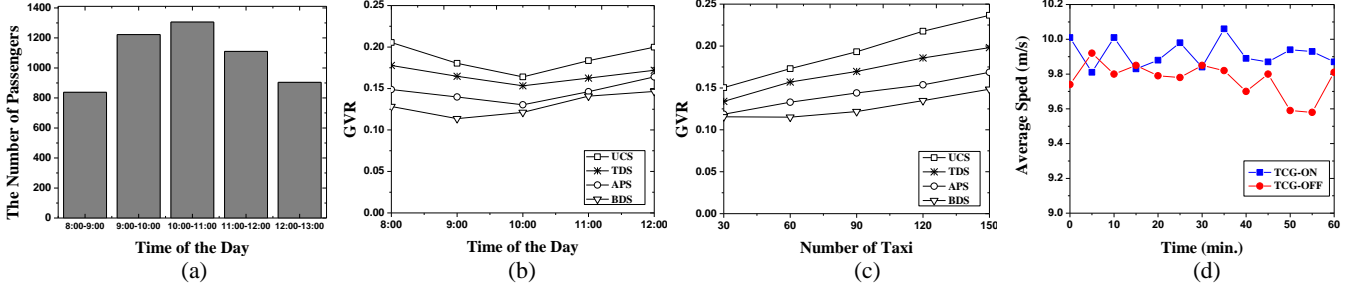


Fig. 3. Testing results in the case study. (a) Number of passengers retrieved from real trace at different times of a day. (b) Performance with real trace at different time of a day. (c) Performance with real trace under different number of taxi. (d) Testing with traffic simulator.

most people choose public transit services (e.g., bus/metro) for their daily commute instead of choosing taxi services.

Fig. 3(b) further shows the GVR which the four different solution algorithms to the TCG problem yielded at different times of the day. The performance fluctuates in different times, but in general BDS and APS still yield better performances in terms of a lower GVR. By looking at Fig. 3 (a), similar to the explanation to Fig. 2 (b), lower GVR can be achieved during the hours having more passenger appearances.

Similar to Fig. 2(a), Fig. 3(c) compares the GVR of the proposed solutions, as a function of the number of taxis during the 8AM-9AM period. Again, BDS always performs best and the value of the GVR of all the solutions increases as the number of total taxis increases.

VI. EVALUATION WITH TRAFFIC SIMULATOR

In this section, the proposed TCG solutions were finally integrated within a well-known microscopic transportation simulator, namely TRANSIMS, to evaluate the transportation system-wide impacts and implications of TCG on a large-scale network. TS advances at the realistic driving behavior, traffic signal settings and so forth.

The technical challenge in evaluating the effectiveness of TCG on the overall transportation system performance is that background traffic (i.e., vehicles other than taxis) in the network are likely to dilute the influence of informed taxi dispatching on the overall system, especially if the taxis

constitute a small percentage of the total traffic volume. Moreover, the impact may not be quite manifest because a) TCG improvements can be made only when a taxi is vacant, which hopefully constitute a relatively small portion of the total taxi operating time; and b) TCG aims at reducing taxi's vacant rate which in turn would lead to less vacant time. On the other hand, a report [16] shows that taxis may constitute a large portion of the total traffic volume in big cities, especially during rush hours. Given the observations above, we focus on a special scenario in which all the vehicles in the network are assumed to be taxis. The same road network as in Sec. IV was deployed to evaluate two scenarios: 1) TCG-ON, in which the target 1000 taxis follow CRS calculated by BDS; 2) TCG-OFF, in which each vacant taxi individually select a nearest busy link and get there by shortest path based on distance.

Fig. 3(d) shows the average road speed on all the network links, as sampled during the simulation period. As can be seen, the average network link speed with TCG-ON is generally higher than the speed with TCG-OFF. This seems to indicate that TCG has helped achieve a more balanced taxi dispatching, and has thus helped improve traffic flow conditions (i.e., minimize congestion as indicated by the higher average speed). Additional research is needed however to confirm this preliminary observation.

VII. RELATED WORK

Taxi dispatch problem has been studied for a while.

Typically, most of the existing works focused on how to match the vacant taxis with the received requests and applied e.g., Integer Linear Programming or other techniques to solve the according optimization problems [8, 13, 14]. For example, [13] provided a solution based on multi-agent architecture and modeled the matching problem as a linear assignment problem.

While in this work, we look into the taxi dispatch problem from a total different angle in the sense that we focus on how to provide vacant taxis with cruising routes to find potential customers, instead of responding to any specific pick-up requests from passengers. Within such a context, on one hand, a taxi may not be able to find a passenger even if when cruising on a link with high possible passenger appearances; on the other hand, it is also possible for a taxi to find a passenger on a link that passengers rarely appear. Therefore, the model considered in this work has the fundamental difference with all the models used in existing works, and therefore brings in total different and new technical issues and challenges.

Although a few works [11] have proposed the concept of dynamic routing for vacant taxis, they did not consider the traffic condition as an essential factor when designing taxi dispatch solutions. While in this work, we explicitly take the traffic condition into consideration during the problem modeling, which makes our problem more interesting as well as more challenging. In the meantime, most of the existing works only conducted testing based on simulation. By comparison, in addition to a comprehensive performance evaluation, we also present a case study by using real taxi trace data from taxis in the city of Shanghai. More than that, we have examined how the proposed TCG problem may affect the transportation system by conducting field testing on a well-known traffic simulator. Overall, such research activities has not been conducted and reported in any existing works.

There are also a number of existing works from data mining area. However, their research challenges and issues are related to massive data processing and knowledge discovery (which could be the basis of our work), but they did not address any issue related to vacant taxi dispatch and cruise route suggestion as considered in this paper.

In addition, many other researches have studied various taxi service-related applications, e.g., how to integrate taxi services with rideshare/carpooling services [1, 2, 3, 4, 5], how to predict or estimate traffic condition based on drivers' routing behaviors [9, 10], etc. Accordingly, some of those aspects can also be considered in TCG problem, and we state it as a part of the future work.

VIII. CONCLUSION

In this paper, we addressed a new problem called *Taxi Cruising Guidance* (TCG), which aims to minimize the *Global Vacant Rate* (GVR) of taxis by trying to provide them with *Cruising Route Suggestions* (CRS). The practical

objective here is mainly to assist vacant taxis in finding passengers as soon as possible during cruising. Note that, this problem is significantly different from all the traditional taxi dispatch problems in the sense that Dispatch Center (DC) guides vacant taxis to find potential customers, instead of responding to any specific pick-up request known in advance. Accordingly, we proposed a number of heuristic solutions based on different methodologies, and performed large-scale evaluation. We also presented a case study in the city of Shanghai by using real traces collected from taxis. A field test has also been conducted on a well-known traffic simulator (TRANSIMS) in order to demonstrate that the application of TCG is also able to benefit the overall transportation system.

REFERENCES

- [1] Y. Hou, X. Li and C. Qiao, "TicTac: From Transfer-Incapable Carpooling to Transfer-Allowed Carpooling," *IEEE GLOBECOM*, 2012
- [2] X. Li, C. Qiao, Y. Hou, Y. Zhao, A. Wagh, A. Sadek, "On-road Ads Delivery Scheduling and Bandwidth Allocation in Vehicular CPS," *IEEE INFOCOM*, 2013.
- [3] P., Lalos, A. Korres, C.K. Datsikas, G.S. Tombras, and K. Peppas, "A Framework for Dynamic Car and Taxi Pools with the Use of Positioning Systems," *COMPUTATION WORLD*, 2009.
- [4] P.I. Philippopoulos, G. Soulos, A. Krukowski, V. Gorgas, D. Kyriazanos, D.I Axiotis and S. Thomopoulos, "ToD: An Intelligent Location-Based Transport Management System," *IEEE IV*, 2007.
- [5] C. Tao, "Dynamic Taxi-Sharing Service Using Intelligent Transportation System Technologies," *IEEE WiCom*, 2007.
- [6] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to Find My Next Passenger", *ACM UbiComp*, 2011.
- [7] J. W. Powell, Y. Huang, F. Bastani, and M. Ji. "Towards Reducing Taxicab Cruising Time Using Spatio-Temporal Profitability Maps," *SSTD*, 2011.
- [8] Y. Ge, C. Liu, H. Xiong, and J. Chen, "A Taxi Business Intelligence System," *ACM KDD*, 2011.
- [9] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. "T-Drive: Driving Directions Based on Taxi Trajectories," *ACM SIGSPATIAL GIS*, 2010.
- [10] Z. Chen, H. Shen and X. Zhou, "Discovering Popular Routes from Trajectories," *IEEE ICDE*, 2011.
- [11] K. Yamamoto, K. Uesugi, "Adaptive Routing of Cruising Taxis by Mutual Exchange of Pathways," *KES*, 2008.
- [12] S. Cheng and X. Qu, "A Service Choice Model for Optimizing Taxi Service Delivery," *ITSC*, 2009.
- [13] K. T. Seow, N. H. Dang and D. Lee, "A Collaborative Multiagent Taxi-Dispatch System," *IEEE Transactions on Automation Science and Engineering*, vol.7, no.3, 2010.
- [14] J. Lu, M. Yeh, Y. Hsu, S. Yang, G. C.-H. and M Chen, "Operating Electric Taxi Fleets: A New Dispatching Strategy with Charging Plans," *IEEE IEVC*, 2012.
- [15] S. Phithakkitnukoon, "Taxi-Aware Map: Identifying and Predicting Vacant Taxis in the City," *AMI* 2010.
- [16] "Six Regulations to Relief Traffic Jam in Beijing", *CCTV News*, 2003.