Exploring the Ridesharing Efficiency of Taxi Services

WEILIANG ZENG1,2,3, YU HAN1, WEIJUN SUN1, AND SHENGLI XIE1,4,5, (Fellow, IEEE)
1Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou 510006, China
2Guangdong Provincial Key Laboratory of Intelligent Transportation System, Guangdong 510006, China
3School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China
4Joint International Research Laboratory of Intelligent Information Processing and System Integration of IoT, Ministry of Education, Guangzhou 510006, China
5Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China

Corresponding authors: Weiliang Zeng (weiliangzeng@gdut.edu.cn) and Weijun Sun (14341569@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61803100 and Grant U1911401, in part by the Guangdong Provincial Key Laboratory of Intelligent Transportation System under Grant 202005003, in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2019B010121001, Grant 2019B010118001, and Grant 2019B01019001; in part by the Industrial Internet innovation and development project of Ministry of Industry and Information Technology of the People’s Republic of China, under grant TC190A3X9-2-2, and in part by the National Key Research and Development Project, China, under Grant 2018YFB1802400.

ABSTRACT

The application of ridesharing strategy to autonomous taxi system holds great promise for improving the efficiency in the future on-demand ride-hailing services. Prior to the implementation of dispatching strategies to the autonomous taxi system, it is necessary to gain insight into the performance of the dispatching strategies. This study aims to solve the dynamic ridesharing problem and conduct a comprehensive quantity analysis of the ridesharing efficiency for various demand levels and carrying capacities in a metropolitan area. We quantify the success rate of serviced requests, the trip travel time, the discount rate of taxi fare, and total energy consumption for different carrying capacities and demand levels in the road network of Shenzhen city. The simulation results show that the ride-matching success rate within 3 minutes enables to increase by more than 13% in the ridesharing mode, and over 80% of the passengers can be served within 6 minutes if the carrying capacity is set to four. The trip travel time and energy consumption also show a significant downward trend as the capacity of the taxi increases in the ridesharing mode.

INDEX TERMS

Dispatching strategies, dynamic ridesharing, ridesharing efficiency, carrying capacity.

I. INTRODUCTION

Ride-hailing service has been regarded as one of the most vital components of the public transport system in a city. As the travel demand increases in metropolitan areas, hailing a taxi during rush hours has long troubled people. According to the Economic Daily, Didi Chuxing reported that the average success rate of taxi-hailing was only 60 percent during the 2019 Spring Festival.

Recently, the rise of autonomous and connected vehicles [1] and the extension of the concept of the sharing economy [2] may provide a perfect solution to on-demand ride-hailing service by a shared autonomous taxi system [3]. Simulation studies [4]–[9] have reported that traffic congestion can be alleviated by implementing the shared autonomous vehicle (SAV) systems, e.g., T-share system [11] and the MoD system [12], [13]. However, most of the previous research did not investigate the carrying capacity and the demand level comprehensively. Most of them only set the carrying capacity as two passengers at a normal travel demand level for the sake of simplicity of calculation. They usually ignored the potential carrying capacity for a multiple-seat taxi or minibus, which can be an important traffic tool in the shared public transport mode. Some local transportation sectors in China have encouraged the usage of ridesharing. In October 2017, the seven-seat taxis were launched in Shanghai. Similarly, Shenzhen Transportation Bureau changed the standard of taxi carrying capacity to “special vehicles with less than seven seats” in 2019. There is no doubt that the high-capacity taxi may appear in the ridesharing mode in the near future. However, few studies considered the efficiency of ridesharing serving over two rides with a trip. Thus, it is necessary to investigate whether the overall benefit necessarily increases when a taxi is allowed to share with more than two passengers.
on a combined trip. On the other hand, the problem of “the minimum taxi fleet in the city” has been raised [14] since the traffic demand increases rapidly with the development of urbanization. Autonomous dispatching and ridesharing strategies might be a potential way to reduce the vehicle density in the road network.

In this study, we propose an autonomous taxi dispatching system allowing for setting various ridesharing carrying capacities. Different from previous studies, the main contributions made by this paper are twofold. (a) A modified T-shared system [11] is proposed, which allows for dynamic insertion of trips and flexible carrying capacity. (b) The ridesharing efficiency is simulated under various settings of travel demand levels and carrying capacities in a real-world network, which are helpful to the policymakers.

II. LITERATURE REVIEW

The shareability of taxi trips has been confirmed by several studies [15], [16]. With the development of online ride-hailing service, the idea of ridesharing [11], [17], [18] gradually arouses extensive research interest in the operation efficiency [19], [20], [21], [22]. The ridesharing can be divided into two types, i.e., static and dynamic sharing. The static ridesharing [23], [24] requires all taxi trips to be determined beforehand, while the dynamic ridesharing [25], [26] reroutes in real time. The former considers a static scenario in which the sets of rider requests and drivers are known ahead of time, which can be achieve in the pre-trip phase. The latter one needs an on-demand response to the request, which requires a perfect en-route ride-matching mechanism.

In early studies, the static ridesharing problem was usually formulated as a dial-a-ride problem (DARP), which referred to the matching of one driver to multiple riders with specified trip requests. To solve this problem, Desrosiers et al. [27] proposed an improved labeling scheme of ridesharing on the basis of Psaraftis [28]. However, time windows were not considered. Madsen et al. [29] proposed a solution to the static DARP routing and scheduling problem with time windows. Later, more effective and exact solution algorithms [30]–[32] were emerged following their study. However, they were not available to provide immediate response to the passenger request in real time.

Dynamic or demand-driven ridesharing services are paid more attention in recent studies. The dynamic ridesharing problem can be formulated as the dynamic vehicle routing problem (DVRP) [35], [36], in which the constraint that the order of the passing node in the current taxi schedule must keep intact when the next passenger joins in the taxi trip is required [11], [37], [38]. Wang et al. [33] proposed a ridesharing strategy that allowed a vehicle to change its route at most once while it was serving a passenger to respond to another ad hoc request. Maciejewski et al. [25] proposed a real-time taxi scheduling strategy to find the global optimal passenger matching scheme in each decision-making period. To guarantee the on-time service quality (i.e., pick-up and drop-off time delay), Ma et al. [11] designed a T-share service model with three spatial-temporal grid cell lists while Huang et al. [26] proposed an effective kinetic tree model. Both of them relied on a very complicated index structure that incurred a high maintenance cost but lack of consideration of the traffic flow. Instead, Wang et al. [34] proposed a cooperative scheduling model for multiple taxis based on real-time information and traffic flow data, in which the drivers’ benefit, i.e., expected revenue, was also considered as the one of the optimization objectives of the scheduling strategy.

For the era of SAVs and online taxi-hailing, a wide spectrum of studies investigates the benefits of SAV services on various fields. For example, some studies evaluated the direct or indirect environmental benefits by questionnaire survey [39], [40] or analyzing the request order [41]. Some studies maximized the reduction of vehicle miles/kilometers traveled (VMT/VKT) by accommodating as many riders as possible under a set of constraints [34]. Other impacts of shared mobility services such as cost, convenience, vehicle usage and vehicle ownership were also quantified in some regions [40]. Many works had confirmed the effectiveness of ridesharing in travel time and cost-saving. But few of them considered the vehicle carrying capacity and the rapid inflation of the travel demand. Alonso-Mora et al. [3] examined the comparison of the vehicle carrying capacity of 1,2,4 and 10, but they did not consider the increase of the order requests.

Thus, this study aims to conduct a comprehensive quantity analysis of the ridesharing efficiency for various demand levels and carrying capacities. First, we propose a practical and easy-to-maintain autonomous ridesharing framework for dynamic taxi dispatching system by considering the flexible trip combination and carrying capacity. Then, we do a simulation study on how the taxis with varying capacities influence the system performance in the different travel demand levels. Since our system runs in real time, it is particularly suited to autonomous vehicle fleets that can continuously reroute based on real-time requests.

III. METHODOLOGY

A. VARIABLE DEFINITION

The variables are defined in TABLE 1.

Order (O): An order (O) is defined as a passenger’s ride-hailing request associated with the pickup point O_i, the delivery point O_i,d, the moment of initiating a request T_{O_i,1}, the moment of ride-matching T_{O_i,2} when O is successfully matched to a taxi, the boarding time T_{O_i,3}, the alighting time T_{O_i,4}, and the latest arrival time T_{O_i,5}.

Vehicle (v): A vehicle (v) refers to its status including the information about the number of available seats, real-time location v_i,l and the trip schedule defined by a temporally-ordered sequence of pickup and delivery points and their corresponding arrival time.

Road (r): The road network used in our research is extracted from the Shenzhen City, China. It includes the current road traffic status (travel time) and the future traffic status prediction by using the floating car data. Since the
traffic prediction is not the focus of this paper, we use the hourly average link travel time for the routing strategy.

Grid (g): A grid is defined as a small piece of the road network. We divide road map into dozens of 500m×500m grids to speed up the dispatching algorithm. As shown in Fig.1, each grid g serves as an independent traffic domain. The traffic information was recorded within its domain area, including static and dynamic information. The static information refers to all road nodes and links. The node closest to the grid’s geographical center is defined as the anchor node of g (represented by a). Then, the distance and the travel time between two grids can be obtained by searching the shortest path of their anchor node (a). The dynamic information refers to the list of taxi fleet in each g, in which the existing and scheduled taxis within a short time are included. Each taxi is tagged with the exact arrival time, indicating when the taxis will enter the grid. And they are sorted in ascending order of the arrival time.

### B. Modelling of Ridesharing System

In this section, a practical dynamic taxi ridesharing framework is proposed. As shown in Fig. 2, the framework involves three core modules: Searching, Scheduling and Waiting. All of the active requests in very short period (1s) will be sorted by a scheme first-in-first-out (FIFO). And server constantly updates the vehicles’ locations and status. In each calculation period, the searching module retrieves the appropriate taxis with the constraints for the requests. After that, the scheduling module matches the orders with the selected taxis and compares all of the possible paths. Finally, the system updates the route and schedule of the best-matched vehicle. The order will be put into the waiting module if any vehicles are failed to match during the above process. The three core modules are described in the following sections.

1) VEHICLE SEARCHING MODULE

The purpose of the searching module is to screen out candidate vehicles that satisfy the specific constraints. However, checking all of the taxi around is unpractical because of the high computation complexity. Most passengers only focus on the time they spend from their pickup points to the delivery points, hoping that the delay will be as short as possible. Here, we separate the delay into the waiting time $t_w$ and the en-route detour time $t_d$. $t_w$ is given by the difference between $T_{O_{i3}}$ and $T_{O_{i1}}$, and $t_d$ is the extra time due to detouring to meet multiple orders. Thus, we use $t_m$ to represent a maximum delay allowed for an order which always satisfies (1)

\[ t_w + t_d \leq t_m \] (1)

The searching module searches suitable taxis with constraints via two main steps. The first step is to select the grids. For the sake of clarity of description, we give an example in Fig.3. Suppose O.o is a passenger’s boarding point (represented by a red dot in Fig.3) and it belongs to $g_7$. The module searched the grids that could arrive $g_7$ within $t_m$ according to the temporal closeness. $g_7$ is the first grid found by the module. Any other arbitrary grids are selected as the candidates by the searching module if (2) holds, where $t_{g_i,g_7}$ means the travel time from $g_i$ to $g_7$. In this example, $g_7$, $g_8$, $g_5$, $g_4$ and $g_9$ can be found.

\[ t_{g_i,g_7} < t_m \] (2)

The second step is to pick out the appropriate taxis in the selected grids. For each selected $g$, the module retrieves the

| Notation | Definition |
|----------|------------|
| Order or request of passenger $i$ | $O_i$ |
| The origin and destination of passenger $i$ | $O_i, o_i, d_i$ |
| The moment of initiating a request $O_i$ | $T_{O_{i1}}$ |
| The moment of ride-matching for $O_i$ | $T_{O_{i2}}$ |
| The boarding time of $O_i$ | $T_{O_{i3}}$ |
| The alighting time of $O_i$ | $T_{O_{i4}}$ |
| The latest arrival time of $O_i$ | $T_{O_{i5}}$ |
| A vehicle or a taxi | $v_i$ |
| The current location of $v_i$ | $v_i, l_i$ |
| A road segment or link | $r_i$ |
| A road node | $n_i$ |
| A grid | $g_i$ |
| The anchor node of $g_i$ | $a_i$ |
| A set of successful matching vehicle | $S_{o_i}$ |
| A set of unsuccessful matching vehicle | $S_{f_i}$ |
| The arrival time | $T_i$ |
| The current moment | $T_{now}$ |
| The extra delay time for a new insertion | $t_e$ |
| The maximum delay | $t_m$ |
| The waiting time | $t_w$ |
| The detour time | $t_d$ |
| The time buffer | $t_b$ |
| The travel time from node $i$ to node $j$ | $t_{ij}$ |
dynamic taxi list, and selects the suitable taxis satisfying two constraints: (a) Arrival before \( t_m \); (b) with enough vacant seats. Fig. 4 shows how taxis are selected from the selected grids step by step. Firstly, two empty sets are initialized. One is for the list of successful matching taxis \( S.o \) and another is for the list of unsuccessful matching taxis \( S.f \). Then, the suitable taxis selected from the taxi list of \( g_7 \) are added to the set \( S.o \). Equation (3) can be used to calculate whether the vehicle \( v \) can arrive \( O.o \) on time, where \( v.l \) is the taxi’s current location. As shown in Fig. 4 (A), \( v_1, v_3 \) and \( v_4 \) are going to enter \( g_7 \) before \( T_{now} + t_m \), where \( T_{now} \) represents the current time. However, these vehicles will not be selected because they are full of passengers. In this case, the result set \( S \), calculated by (4), is empty. Naturally, we continue to retrieve the next grid \( g_8 \). As shown in Fig. 4 (B), we find the appropriate taxis \( v_{12} \) and \( v_{13} \), so we add them into the set \( S.o \). \( S \) has several elements after the second step. In this step, it is necessary to address a problem that whether the algorithm should search continuously. Even though it is possible to obtain the globally optimal vehicle by searching all grids, it takes relatively long calculation time. Furthermore, we empirically found that the globally optimal vehicle emerging in the top 3 grids with suitable vehicles has 90% probability. Thus, we heuristically speed up the algorithm by searching the two most likely grids (e.g., \( g_5 \) and \( g_4 \)) after searching \( g_8 \), as shown in Fig. 4 (C). Finally, \( S \) has three elements (\( v_{12}, v_{13} \) and \( v_{11} \)) which will be sent to the scheduling module.

\[
v_{11}, O.o < t_m \tag{3}
\]

\[
S = S.o - S.f \tag{4}
\]

Meanwhile, if all of the selected taxis fail to match the requests in the scheduling module, they will be added into a set \( S.f \). The searching module will search again in order to improve the success rate and save the passengers’ waiting time.

FIGURE 2. The framework of the ridesharing system.

FIGURE 3. Grids selection.

FIGURE 4. Vehicle selection.
The pseudo code of the searching module is presented in Algorithm I. In Line 1, the algorithm finds a grid to which the origin point belongs. In Line 2, a string of grids is obtained temporally according to the specified delay. The grids are stored in an ordered list and we can use it directly in next calculation interval. It is worth mentioning that the grids are neighbors in the lists may not be true neighbors in real-world space. From Line 7 to Line 15, the algorithm searches the first suitable grid and two consecutive grids. Finally, Line 16 returns the candidates of matching vehicle.

2) SCHEDULING MODULE

Given a set of candidates of matching vehicle from the searching module, the scheduling module inserts the new request into the schedule of the alternative vehicles. In this section, we discuss how to compare all of the insertions fast and seek out the insertion with minimum additional travel time increasing.

Without changing the sequence of the passing nodes in the original schedule for each vehicle [1], [11], [37], the procedure of insertion can be divided into two stages: (a) Inserting the origin of O(O.o); (b) Inserting the destination of O(O.d); Fig. 5 illustrates an example of how the scheduling module works. Given an order O with origin O.o and destination O.d, the scheduling module inserts O.o and O.d into the original schedule of one of the vehicles in S. Among all possible routes of the alternative vehicles, the algorithm chooses to insert O into the route that minimally increases the travel time. For instance, there are four plans in the original schedule in Fig. 5(A). The scheduling module tries to make a new path arrangement n1 → O.o → n2 by inserting O.o between node n1 and node n2 firstly. It will check whether the selected taxi is able to arrive at O.o before T_{now} + t_m. The insertion fails if the new path can not satisfy the constraint of the maximum delay. Otherwise, the module updates the travel time delay after the insertion of O.o. Since the time required for the new path and the original path are respectively \( t_{n1,o.o} + t_{o.o,n2} \) and \( t_{n1,n2} \), the updated delay \( t_e \) can be given by (5)

\[
\begin{align*}
  t_e &= t_{n1,o.o} + t_{o.o,n2} - t_{n1,n2} \\
  \text{where } T_{od} &= \text{minimum travel time from the origin to the destination.}
\end{align*}
\]

An appointment between each order and the alternative taxi is that passengers must be guaranteed to drop off before the latest arrival time \( T_{o.o} \) given by (6).

\[
\begin{align*}
  T_{o.o} &= T_{o.o} + T_{od} + t_m \\
\end{align*}
\]

FIGURE 5. Schedule update.
### Algorithm 2 Scheduling Module

**Input:** An order $O$
- A vehicle $v$ returned from the search module
- The upper limit of delay time $t_m$

**Output:** Return new_schedule if the insertion succeeds, otherwise return False

1. **for** next unchecked insertion position $i$ in the taxi schedule of $v$ **do**
2.  **if** $v$ can not arrive the origin $O.o$ within $t_m$ while inserting $O.o$ at the position $i$ **then**
3.    **break**
4.  **if** the time delay incurred by the insertion of $O.o$ is larger than the buffer time of any plan after position $i$ in the schedule **then**
5.    **continue**
6.  **else**
7.    **for** next unchecked insertion position $j$ after $i$ in the taxi schedule of $v$ **do**
8.     **if** $v$ can not arrive the destination $O.d$ within $t_m$ while inserting $O.d$ at the position $j$ **then**
9.      **break**
10.    **if** the time delay incurred by the insertion of $O.d$ is larger than the buffer time of any plan after position $j$ in the schedule **then**
11.       **continue**
12.    **else**
13.       `poss_schedule ← store the possible insertion positions i and j`
14.    **if** `poss_schedule` is not empty **then**
15.       `new_schedule ← find the time-least insertions from the poss_schedule`
16.    **return** new_schedule
17.  **return** False

### Algorithm 3 Waiting Module

**Input:** A waiting queue $Q$ with a series of orders
**Output:** A set of orders $S$ needed to be rematch

1. sort the orders in $Q$ according to the waiting time from largest to smallest
2. **for** next unchecked order $O$ in the $Q$ **do**
3.    **if** the waiting time of $O$ is larger than 1200 **then**
4.      delete $O$
5.    **continue**
6.    **if** the waiting time of $O$ is less than 60 **then**
7.      break
8.    **if** the waiting time of $O \% 60$ is 0 **then**
9.      $S ← add O$
10. **return** $S$

- speed up the calculation [37], (b) Replacing with minimizing the distance [11], (c) Replacing with maximizing the carrying capacity [15], (d) Replacing with optimizing the passengers’ cost and driver’s income [33], and (e) Fixing the origin of the vehicle [43]. Though these strategies are crucial in some parts, it is not clear how much extra time produces for each passenger. Alternatively, we adopt the scheme of Insertion Feasibility Check [11] to pick out the best insertion position with the minimum additional travel time. Insertion Algorithm II shows the core of the scheduling process. It returns the insertion position with minimal increment of travel time. From Line 1 to Line 21, it searches feasible insertion positions meeting the constraints and stores the candidates in a list. From Line 22 to Line 26, the optimal schedule is picked out by comparing the path travel time of all the newly feasible schedules.

3) **WAITING MODULE**

Some orders might be failed to match a suitable vehicle due to the constraint of maximum delay. In order to improve the success rate, we introduce a waiting module with a first-in-first-out (FIFO) scheme to prioritize those passengers who have waited for a long time. The system will serve the order with earlier birth time first in each processing period. To save the calculation resource, the orders in the waiting list will be retrieved only once within 60s. An order waiting for more than 20 minutes will be removed as a failed order. Fig. 6 illustrates an example of queuing procedure. There are five orders $O_1$ to $O_5$ in the queue and their birth time satisfies (10). The waiting module takes out $O_1$ first at the moment $T_{now}$. Because $O_1$ satisfies (11) and (12), $O_1$ is stored in an ordered set waiting to rematch. Then, the module takes out $O_2$ with creation time $T_2 = T_{now} - 30s$ which obviously not satisfies (11). According to (10), the subsequent orders do not satisfy (11) too. Thus, the waiting module finally sends $O_1$ to the search module.

$$T_i = \min(T_1, T_2, \ldots, T_n)$$ (10)
\[ T_{\text{now}} - T_i > 60s \]  
\[ (T_{\text{now}} - T_i) \%60 = 0 \]

The primary process of the waiting module is shown in Algorithm III. From Line 1 to Line 2, the waiting orders are sorted according to the creation time. Then, the algorithm removes the orders waiting for over 20 minutes. If the earliest creation time is within the latest 60 seconds, the algorithm will stop the programs in Line 7 and Line 8. Then, the orders with 60s-interval waiting time will be picked out and added to a set. Finally, the algorithm returns the set containing the orders to be re-matched.

IV. NUMERICAL ANALYSIS

In this section, experiments are designed to study the efficiency of ridesharing based on real-world taxi booking data in the city of Shenzhen, China. We set the acceptable maximum delay time \( t_m \) for each passenger as 5 minutes. The effects of different constraints of the taxi booking service are evaluated, including the difference of the available seats of taxis and the increasing travel demand. The results indicate that ridesharing could significantly improve the success rate, decrease the passengers’ cost, travel time, and the energy consumption.

A. EXPERIMENT SETUP

The experiment uses real taxi booking requests of a working day on September 20, 2016 in Nanshan district of Shenzhen as the input of the simulation. Orders are collected by a time window (e.g., 1 min) in our experiment, which will be assigned in batch to the taxis fleet. The road network contains 1426 nodes and 2715 links. To speed up the routing plan, we divide the map into \( 15 \times 11 \) grids and the nodes in the map are classified into the grids. The travel time on each road segment is estimated by the real taxi trip data.

Floyd algorithm is applied to precompute the least travel time paths of all the nodes in different periods and store them into a lookup table. Taxis in our simulation experiment will run following the lookup table. The ride-hailing service is simulated from 8:00 am to 23:00 pm and evaluate the performance of ridesharing. To keep the characteristics of the realistic scenario, the taxis are distributed throughout the road network based on the real taxi distribution at 8:00 am initially, which was extracted from the real taxi trajectory data.

B. RESULTS AND DISCUSSION

In the experiment, the number of orders is extracted in every hour. Considering the inflation of traffic demand in the future, we set different ratios of passengers to taxis (\( PTR \)) to investigate the benefits of the ridesharing strategies. As shown in Fig. 7, we take the actual average statistics per hour (\( PTR=2.5 \)) in the experimental area as the baseline, then increase the number of orders proportionally (\( PTR=2.5 \sim 5.0 \)). The benefit of ridesharing with the inflated travel demands and varied carrying capacities is evaluated in the following sections.

1) SUCCESS RATE (SR)

\( SR \) is a ratio of the matched orders to the total orders. It is an essential criterion to measure the effectiveness of ridesharing. Previous studies have confirmed that \( SR \) can be significantly improved by ridesharing [11], [33]. Different from previous studies, we not only evaluate the overall \( SR \) but also analyze the change of \( SR \) with various waiting time. Such analysis enables to give us some insight into the ridesharing efficiency. The passenger waiting time is set to 0~3 minutes, 3~6 minutes, 6~9 minutes, 9~12 minutes, 12~15 minutes, and more than 15 minutes, respectively. Then, we compare the \( SR \) for the non-ridesharing and ridesharing modes with various the carrying capacities and \( PTR \). As shown in Fig. 8, ridesharing has a considerably higher total success rate than no-ridesharing. Especially in the high demand conditions, such as \( PTR=4.5 \) and \( PTR=5 \), ridesharing can improve the success rate of matching by more than 20%. In the non-ridesharing mode, the difference of the total \( SR \) between \( PTR=5 \) and \( PTR=2.5 \) is nearly 30%, while it is less than 1% in the ridesharing mode with carrying capacity setting to four persons. From Fig. 8 (a) to Fig. 8 (f), it is found that the success rate of ride matching reduces sharply (up to 30%) as the \( PTR \) increases from 2.5 to 5 in the non-ridesharing mode. However, the success rate of ride matching keeps a relatively high value as the \( PTR \) increases in the ridesharing modes. For example, the success rate is lower than 25% in the non-ridesharing mode, while it is always higher than 60% when the \( PTR \) is set to 5. It indicates that the ridesharing modes can satisfy the increased travel demand and save the passenger waiting time.

Fig. 8 also indicates that \( SR \) gradually converges to a steady level rather than increases significantly as the carrying capacity increases with a fixed size of vehicle fleet. As shown in TABLE 2, we set different values of carrying capacity in simulation to find out the optimal passenger number of ridesharing. We regard the carrying capacity as the optimal value when the change of \( SR \) is less than 0.5%. It is found that the optimal carrying capacity is 2 for the real-world demand level (\( PTR=2.5 \)) while it is 4 for the high demand level (\( PTR=5 \)). A weak positive correlation can be found between \( PTR \) and the carrying capacity. That means the more seats should be set as the travel demand increases if the size of vehicle fleet does not increase. According to the simulation result, the maximum number of ridesharing setting to four is suitable for the travel demand level in current and near future situation.

2) TRAVEL TIME EFFICIENCY

Some people worry about that ridesharing might cause longer travel time due to ride matching and detouring. We explore this problem by evaluating the average trip travel time (\( TTT \)) of the orders assigned to the vehicles successfully. As shown in Fig. 9, it demonstrates that the average \( TTT \) reduces as the carrying capacity increases. The average \( TTT \) can be save up to 20%, especially in the high travel demand condition.
It can be concluded that ridesharing not only enables the taxi to serve multiple passengers but also reduces the average travel time.

Ridesharing is also beneficial to the boot time that refers to the time from request submission to successful ride-matching. Once the booting time exceeds the expected
W. Zeng et al.: Exploring the Ridesharing Efficiency of Taxi Services

3) TAXI FARE DISCOUNT (TFD)

In order to attract more people to participate in ridesharing, a taxi fare discount for passengers is essential. It should be noted that two properties for a pricing scheme are necessary: (a) The taxi fare in the ridesharing mode cannot be higher than that in the non-ridesharing mode; (b) The taxi fare of shared distances is evenly split among the riding passengers. Thus, more people share a ride, more discount can be obtained.

Without considering the extra fee, taking a taxi in Shenzhen basically follows the billing strategy that the starting price is 10 RMB for the first 2 km, and the additional kilometer price is 2.6 RMB per kilometer. Combining the current taxi billing strategy in Shenzhen and the above principles, we provide a simple yet effective pricing scheme for our taxi ridesharing service as shown in equation (13)-(15).

\[
\begin{align*}
f_1 &= 2.6k_1 + 2.6 \frac{k_2 - 2}{num} + \frac{10}{num} \\
f_2 &= 2.6(k_3 - 2) + 10 \\
fare &= \min(f_1, f_2)
\end{align*}
\]

where \(k_1\) is the non-ridesharing distance, \(k_2\) is the ridesharing distance, \(k_3\) is the distance of the least travel time path from origin to destination, \(num\) is the number of passengers sharing the trip.

Thus, the taxi fare discount for each passenger can be obtained by (16).

\[
TFD = \frac{fare}{f_2}
\]

By applying the pricing scheme above, we compare the TFD between the non-ridesharing mode and ridesharing mode. As shown in Fig. 11, passengers can easily get a discount of lower than 90% in the ridesharing mode. It is found that the TFD steadily drops as the PTR increases. It is likely because the taxi ridesharing opportunities surge as the number of orders increases. This result suggests that ridesharing can provide monetary incentives to passengers.

4) ENERGY CONSUMPTION

We consider applying taxi ridesharing in battery electric vehicles (BEVs). Ridesharing is also of great help in saving electric energy. With regards to BEVs, energy consumption is determined by the structure of electricity production, the efficiencies of the process of power generation, and the efficiencies of electricity transmission and distribution [41]. For the sake of description simplicity, we use 15 kilowatts per 100 kilometers, according to the current average performance.
of BEVs, to estimate the energy consumption of the fleet at 15 rush hours. We can observe in Fig. 12 that approximately 15% of direct energy saving can be achieved in the ridesharing mode. Furthermore, Yu et al. [41] found ridesharing also benefits from emission from indirect impacts of industrial production. It indicates a substantial possibility for achieving more considerable environmental benefits from trips made by ridesharing. Thus, promoting ridesharing among passengers will substantially mitigate environmental pressure in the transport sector and bring enormous social benefits.

V. CONCLUSION

This paper introduces a practical real-time shared taxi system for assigning passenger requests to a fleet of taxis of varying capacity. Our system consists of three modules, each of which runs independently and can be combined with other feasible approaches. The advantage of the system is that it enables to reducing computation largely through selective retrieval, improving the success rate and minimizing the travel time as much as possible. What’s more, it records the route of each taxi and predicts the near future traffic situation.

We explore the effects of ridesharing with a heavy transportation demand. A series of experiments are designed and conducted to evaluate the ridesharing efficiency for the taxi fleet of varying capacity based on a real taxi data for the city of Shenzhen. Success rate, travel time, taxi fare discount and energy consumption are quantified in this paper. The experimental results demonstrate the effectiveness and efficiency of our system in dynamic ridesharing. Firstly, ridesharing can enhance the capacity of the taxi fleet so as to serve the commute of more people. For instant, when $PTR=5.0$, ridesharing can serve additional 25% passengers than non-ridesharing. In addition, the passengers’ waiting time and their total travel time decreases. Over 85% of the passengers can be picked up within 6 minutes when $PTR$ is less than 4.5. Thirdly, it is found that a taxi fare discount of 90% at least can be obtained for most ridesharing passengers. Compared to non-ridesharing, our analysis also shows that ridesharing can reduce the travel distance and energy consumption. It demonstrates that as the travel demand increases, the advantage of ridesharing is more significant. Thus, we recommend to apply ridesharing service for the city transportation.

The limitation of the simulation study is that we ignore the way of dividing grids and the assumption of the passenger number due to lack of data. In the future, a simulation study based on various shapes and sizes of grids and varying passenger number is expected to make this study more general.

REFERENCES

[1] Z. Liu, T. Miwa, W. Zeng, M. G. H. Bell, and T. Morikawa, “Dynamic shared autonomous taxi system considering on-time arrival reliability,” Transp. Res. C, Emerg. Technol., vol. 103, pp. 281–297, Jun. 2019.

[2] Z. Liu, T. Miwa, W. Zeng, M. G. H. Bell, and T. Morikawa, “Shared autonomous taxi system and utilization of collected travel-time information,” J. Adv. Transp., vol. 2018, pp. 8919721.1–8919721.13, Aug. 2018.

[3] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment,” Proc. Nat. Acad. Sci. USA, vol. 114, no. 3, pp. 462–467, Jan. 2017.

[4] N. D. Chan and S. A. Shaheen, “Ridesharing in north america: Past, present, and future,” Transp. Rev., vol. 32, no. 1, pp. 93–112, Jan. 2012.

[5] S. H. Jacobson and D. M. King, “Fuel saving and ridesharing in the US: Motivations, limitations, and opportunities,” Transp. Res. D, Transp. Environ., vol. 14, no. 1, pp. 14–21, Jan. 2009.

[6] A. Amey, J. Attanucci, and R. Mishalani, “Real-time ridesharing: Opportunities and challenges in using mobile phone technology to improve rideshare services,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2217, no. 1, pp. 103–110, Jan. 2011.

[7] D. J. Fagnant and K. M. Kockelman, “The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios,” Transp. Res. C, Emerg. Technol., vol. 40, pp. 1–13, Mar. 2014.

[8] A. S. Shaheen and T. E. Lipman, “Reducing greenhouse emissions and fuel consumption: Sustainable approaches for surface transportation,” IATSS Res., vol. 31, no. 1, pp. 6–20, 2007.

[9] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, “Ridesharing: The state-of-the-art and future directions,” Transp. Res. B, Methodol., vol. 57, pp. 28–46, Nov. 2013.

[10] S. A. Shaheen and A. P. Cohen, “Carsharing and personal vehicle services: Worldwide market developments and emerging trends,” Int. J. Sustain. Transp., vol. 7, no. 1, pp. 5–34, Jan. 2013.

[11] S. Ma, Y. Zheng, and O. Wolfson, “T-share: A large-scale dynamic taxi ridesharing service,” in Proc. IEEE Int. Conf. Data Eng., Apr. 2013, pp. 410–421.

[12] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, “Robotic load balancing for mobility-on-demand systems,” Int. J. Robot. Res., vol. 31, no. 7, pp. 839–854, Jun. 2012.

[13] G. H. D. A. Correia and B. van Arem, “Solving the user optimum privately owned automated vehicles assignment problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility,” Transp. Res. B, Methodol., vol. 87, pp. 64–88, May 2016.

[14] B. Cici, A. Markopoulou, E. Frias-Martínez, and N. Laouaritis, “Assessing the potential of ride-sharing using mobile and social data.” 2013, arXiv:1305.3876. [Online]. Available: https://arxiv.org/abs/1305.3876

[15] P. Santi, G. Resta, M. Szell, S. H. Strogatz, and C. Ratti, “Quantifying the benefits of vehicle pooling with shareability networks,” Proc. Nat. Acad. Sci. USA, vol. 111, no. 37, pp. 13290–13294, Sep. 2014.

[16] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, “Addressing the minimum fleet problem in on-demand urban mobility,” Nature, vol. 557, no. 7706, pp. 534–538, May 2018.

[17] Z. Xu, Y. Yin, and L. Zha, “Optimal parking provision for ride-sourcing services,” Transp. Res. B, Methodol., vol. 105, pp. 559–578, Nov. 2017.

[18] J. Ke, H. Zheng, H. Yang, and X. Chen, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” Transp. Res. C, Emerg. Technol., vol. 85, pp. 591–608, Dec. 2017.

[19] P. Chen and Y. Nie, “Connecting e-hailing to mass transit platform: Analysis of relative spatial position,” Transp. Res. Part C: Emerg. Technol., vol. 77, pp. 444–461, Apr. 2017.

[20] Y. Nie, “How can the taxi industry survive the tide of ridesourcing? Evidence from Shenzhen, China,” Transp. Res. C, Emerg. Technol., vol. 79, pp. 242–256, Jun. 2017.
L. Zha, Y. Yin, and Y. Du, “Surge pricing and labor supply in the ride-sourcing market,” Transp. Res. Procedia, vol. 23, pp. 2–21, Jan. 2017.

L. Zha, Y. Yin, and H. Yang, “Economic analysis of ride-sourcing markets,” Transp. Res. C, Emerg. Technol., vol. 71, pp. 249–266, Oct. 2016.

S. A. Shaheen, “Mobility and the sharing economy,” Transp. Policy, vol. 2016, pp. 141–142, Oct. 2016.

Y. Lin, W. Li, F. Qiu, and H. Xu, “Research on optimization of vehicle routing problem for ride-sharing taxi,” Procedia-Social Behav. Sci., vol. 43, pp. 494–502, 2012.

M. Maciejewski, J. Bischoff, and K. Nagel, “An assignment-based approach to efficient real-time city-scale taxi dispatching,” IEEE Intell. Syst., vol. 31, no. 1, pp. 68–77, Jan. 2016.

Y. Huang, F. Bastani, R. Jin, and X. S. Wang, “Large scale real-time ridesharing with service guarantee on road networks,” Proc. VLDB Endowment, vol. 7, no. 14, pp. 2017–2028, Oct. 2014.

J. Desrosiers, Y. Dumas, and F. Soumis, “A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows,” Amer. J. Math. Manage. Sci., vol. 6, nos. 3-4, pp. 301–325, Feb. 1986.

H. N. Psaraftis, “A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem,” Transp. Sci., vol. 14, no. 2, pp. 130–154, May 1980.

O. B. G. Madsen, H. F. Ravn, and J. M. Rygaard, “A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives,” Ann. Oper. Res., vol. 60, no. 1, pp. 193–208, Dec. 1995.

H. N. Psaraftis, “An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows,” Transp. Sci., vol. 17, no. 3, pp. 351–357, Aug. 1983.

J.-F. Cordeau, “A branch-and-cut algorithm for the dial-a-ride problem,” Oper. Res., vol. 54, no. 3, pp. 573–586, Jun. 2006.

Z. He-Ping, Z. Bi-Qiang, and P. Xia-Hua, “The route choice and rate optimization model of taxi-pooling,” J. Changsha Univ. Sci. Technol., vol. 8, no. 4, pp. 20–24, 2011.

Y. Wang, B. Zheng, and E.-P. Lim, “Understanding the effects of taxi ride-sharing—A case study of singapore,” Comput. Environ. Urban Syst., vol. 69, pp. 124–132, May 2018.

Y. Wang, B. Liang, W. Zheng, L. Huang, and H. Liu, “The development of a smart taxicab scheduling system: A multi-source data fusion perspective,” in Proc. IEEE 16th Int. Conf. Data Mining (ICDM), Dec. 2016, pp. 1275–1280.

M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, “Real-time path planning based on hybrid-VANET-enhanced transportation system,” IEEE Trans. Veh. Technol., vol. 64, no. 5, pp. 1664–1678, May 2015.

S. Rahili, B. Riviere, S. Olivier, and S.-J. Chung, “Optimal routing for autonomous taxis using distributed reinforcement learning,” in Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW), Singapore, Nov. 2018, pp. 556–563.

M. E. T. Horn, “Fleet scheduling and dispatching for demand-responsive passenger services,” Transp. Res. C, Emerg. Technol., vol. 10, no. 1, pp. 35–63, Feb. 2002.

J.-F. Cordeau and G. Laporte, “A tabu search heuristic for the static multi-vehicle dial-a-ride problem,” Transp. Res. B, Methodol., vol. 37, no. 6, pp. 579–594, Jul. 2003.

P. Minett and J. Pearce, “Estimating the energy consumption impact of casual carpooling,” Energies, vol. 4, no. 1, pp. 126–139, Jan. 2011.

E. W. Martin and S. A. Shaheen, “Greenhouse gas emission impacts of carsharing in north america,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1074–1086, Dec. 2011.

B. Yu, Y. Ma, M. Xue, B. Tang, B. Wang, J. Yan, and Y.-M. Wei, “Environmental benefits from ridesharing: A case of beijing,” Appl. Energy, vol. 191, pp. 141–152, Apr. 2017.

S. Narayanan, E. Chaniotakis, and C. Antoniou, “Shared autonomous vehicle services: A comprehensive review,” Transp. Res. C, Emerg. Technol., vol. 111, pp. 255–293, Feb. 2020.

Y. Li, R. Chen, L. Chen, and J. Xu, “Towards social-aware ridesharing group query services,” IEEE Trans. Services Comput., vol. 10, no. 4, pp. 646–659, Jul. 2017.

WEILIANG ZENG received the B.Sc. and M.Sc. degrees from the School of Engineering, Sun Yat-sen University, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Civil Engineering, Nagoya University, Japan, in 2016. He is currently an Associate Professor with the Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, China. His research interests include sharable autonomous vehicle, route planning, machine learning, traffic simulation, and transportation dispatch systems.

SHENGLI XIE (Fellow, IEEE) received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, China, in 1997. He is currently a Full Professor and the Head of the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou. He has authored or coauthored two books and over 150 scientific papers in journals and conference proceedings. His research interests include wireless networks, automatic control, and blind signal processing. He was a recipient of the Second Prize in China’s State Natural Science Award, in 2009, for his research on blind source separation and identification.