Employing Opportunistic Charging for Electric Taxicabs to Reduce Idle Time

LI YAN, HAIYING SHEN, ZHUOZHAO LI, ANKUR SARKER, and JOHN A. STANKOVIC, University of Virginia, USA
CHENXI QIU, Pennsylvania State University, USA
JUANJUAN ZHAO and CHENGZHONG XU, Shenzhen Institute of Advanced Technology, China

For electric taxicabs, the idle time spent on cruising for passengers, seeking chargers, and charging is wasteful. Previous works can only save cruising time through better routing, or charger seeking and charging time through proper charger deployment, but not for both. With the advancement of wireless charging techniques, efficient opportunistic charging of electric vehicles at their parked positions becomes possible. This enables a taxicab to get charged while waiting for the next passenger. In this paper, we present an opportunistic wireless charger deployment scheme in a city, which both maximizes the taxicabs' opportunity of picking up passengers at the chargers and supports the taxicabs' continuous operability on roads, while minimizing the total deployment cost. We studied a metropolitan-scale taxicab dataset on several factors important for deploying wireless chargers and determining the numbers of the chargers in the regions: the number of passengers, the functionalities of buildings, and the frequency of passenger appearance in a region, and taxicab traffic flows in a city. Then, we formulate a multi-objective optimization problem and find the solution. Our trace-driven experiments demonstrate the superior performance of our scheme over other representative methods in terms of reducing idle time and supporting the operability of the taxicabs.

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

Additional Key Words and Phrases: Vehicle wireless charging, charger deployment, mobile data analysis, kernel density estimation

ACM Reference Format:
Li Yan, Haiying Shen, Zhuozhao Li, Ankur Sarker, John A. Stankovic, Chenxi Qiu, Juanjuan Zhao, and Chengzhong Xu. 2018. Employing Opportunistic Charging for Electric Taxicabs to Reduce Idle Time. *PACM Interact. Mob. Wearable Ubiquitous Technol.* 0, 0, Article 0 (2018), 25 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Taxicabs are a pivotal component in a modern public transportation system. Their profit is highly reliant on efficient discovery of passengers [28]. Also, due to the foreseen depletion of fossil fuels, gasoline-based taxicabs are being actively replaced by electric vehicles (EVs) [17]. To maximize the profit of electric taxicabs, their idle time (i.e., cruising time for passengers, seeking time for chargers and charging time) must be reduced as much as possible [11, 31].

Authors' addresses: Li Yan, ly4ss@virginia.edu; Haiying Shen; Zhuozhao Li; Ankur Sarker; John A. Stankovic, University of Virginia, USA; Chenxi Qiu, Pennsylvania State University, USA; Juanjuan Zhao; Chengzhong Xu, Shenzhen Institute of Advanced Technology, China.

2474-9567/2018/0-ART0 $15.00
https://doi.org/0000001.0000001

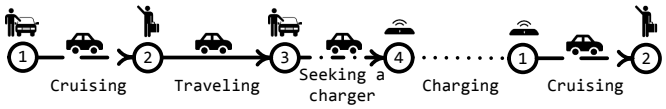

Fig. 1. General operation of a taxicab.

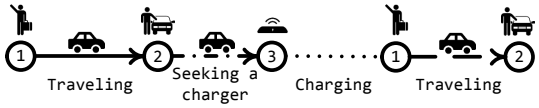

Fig. 2. Expected operation of a taxicab.

In recent years, thanks to the ubiquitous mobile sensing data harvested from GPS-equipped taxicabs in metropolitan cities, many taxicab dispatching methods have been proposed to guide taxicabs to efficiently pick up passengers with reduced cruising miles [28, 29, 31, 33]. Generally, these methods focus on extracting the expected appearance locations of passengers from historical passenger pick-up records or information provided by nearby taxicabs at the current time, and use statistical methods to guide taxicabs to the places with the maximum likelihood of picking up passengers within a certain distance through the shortest routes. However, the taxicabs still have to spend much time driving before picking up the passengers. Moreover, none of the previous taxicab dispatching works considers the time wasted on seeking chargers and charging. It has been reported that the daily average time wasted on seeking the nearest charging station can be almost 1 hour, and the time for charging an EV can be as long as 150 minutes [17]. Such a long idle time greatly degrades the profiting efficiency of the electric taxicabs [4]. Also, since a taxicab cannot be in service all the time due to charger seeking and charging, a metropolitan city needs to put more taxicabs on roads to satisfy taxicab demands, which increases investment cost and traffic congestion on roads.

Meanwhile, driven by the traffic flow and city-wide travel patterns of people reflected in the ubiquitous taxicab movement data, several recent works studied the problem of minimizing average seeking time for the nearest charging station of EVs from the perspective of urban facility planning [17, 21, 26, 32]. These works generally adapt the deployment of charging stations to cover the EV traffic flows so that EVs anywhere can reach the nearest charging stations with the minimal seeking time. However, no matter how well these methods place the charging stations, upon the exhaustion of a battery, the taxicabs must spend extra idle time on seeking a charger and waiting to be charged.

As shown in Figure 1, the traditional operation of a taxicab generally consists of four phases, namely cruising for passengers, traveling with the passengers, seeking a charger, and charging [29]. We see that only traveling contributes to service and making a profit. Reducing the time in other phases helps increase the profit of taxicabs, and reduces taxicab investment and traffic congestion with direct societal and economic impacts.

Experienced taxicab drivers usually prefer to wait at certain places in order to pick up the next passenger with reduced idle cruising miles [28, 31]. Then, if the taxicabs can be offered sufficient opportunity of charging during waiting, as shown in Figure 2, it can enable charging and waiting for passengers to occur simultaneously before picking up the next passenger. Recently, the world has seen a surge in stationary wireless opportunistic charger (Figure 3). Opportunistic charging means that an EV can be charged whenever it is parked over a place with a charger [37]. Since electric energy is transferred from the Transmitter Coil to the Receiver Coil via electromagnetic field, such a charger allows EVs to get charged when they temporarily park somewhere (e.g., at traffic lights, roadside parking lots) without plugging in a cable [10, 12, 22]. The wireless chargers are deployed one time and not portable, and one charger can only serve one EV at a time. The State of Charge (SoC) (i.e., the ratio between the amount

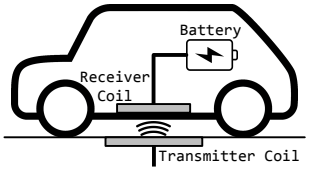

Fig. 3. Stationary wireless charger for EVs.

of charged energy and the battery capacity) of an EV's battery must be maintained above a threshold η (e.g., 20%) for the EV to drive. Different from other public vehicles (e.g., buses or subways), which follow fixed routes every day, the movement of taxicabs is driven by the discovery of passengers. To maximize the service performance of the taxicabs, we must maximally reduce the duration of their idle phases (i.e., cruising for passengers, seeking a charger, and charging). This requires us to design a mechanism that provides opportunities of picking up passengers for the taxicabs, and meanwhile keeps the SoC of taxicabs above a threshold on the road. Motivated by this expectation, we attempt to answer a question in this paper: *how to deploy stationary wireless chargers in a city with the minimum cost (i.e., fewest chargers) to maintain the SoC of taxicabs above a threshold (i.e., can always drive) in the city, and also offer them enough opportunity for picking up passengers while they park for recharging?*

Accordingly, we propose *PickaChu*, a stationary wireless charger deployment scheme that enables the taxicabs to Pick up a passenger with reduced idle time and supports the taxicabs' continuous operability (i.e., always having enough energy to drive) via opportunistic Charging in an urban road network. First, we analyzed a metropolitan-scale taxicab mobility dataset that records the trajectories and passenger pick-up and drop-off activities of 15,610 taxicabs in the year of 2015. There have been previous works proposed for extracting different human mobility patterns (e.g., life patterns, work patterns and commute patterns) for different regions in a city [8]. Its approaches can be used in our work to measure the appearance of passengers. We observed that the building density and building functionalities (or classes) (e.g., professional, residential, commercial buildings) in a city region affect the number of passenger requests. Also, we found that the frequency of the appearance of passenger(s) at one time and the number of pick-ups per unit time vary between different regions. Finally, we analyzed the distribution of the trajectory lengths of the taxicabs to model the traffic in the city.

Fig. 4. Stationary wireless chargers around a mall.

The observations serve as the foundation for the design of *PickaChu*. As shown in Figure 4[1], it determines the regions in a city to deploy wireless chargers and the number of the chargers in each region that offer taxicabs high opportunity of discovering passengers when they are being charged, and help them always maintain a certain level of SoC for continuous operability. As long as a taxicab's SoC is above threshold η, it will pick up nearby passengers during charging. Specifically, we use the average number of passengers around a class of building to determine the weight of this building class, and design a weighted sum of building functionalities as a factor to infer the likelihood of passenger appearance in each region. Moreover, since it is quicker to pick up passengers at the places where passengers appear more frequently, we use the Discrete Fourier Transform (DFT) [23] and AutoCorrelation Function (ACF) [23] to more accurately measure each region's frequency of passenger appearance. Then, to determine a region's priority of being deployed with a charger, we define a scoring mechanism, which jointly considers the average number of passengers that appeared per unit time, the weighted sum of building functionalities, and the frequency of passenger appearance of a region. We use the Kernel Density Estimator (KDE) [27] to model the taxicabs' traffic, and use it to estimate the expected SoC of vehicles in each region. Finally, we formulate a multi-objective optimization problem to minimize the total deployment cost (i.e., the excavation at the charging position, the installation of the charger body, and the wiring to the chargers) of the chargers, maximize the opportunity of picking up passengers at the chargers, and meanwhile ensure a certain level of SoC for the taxicabs in each region.

In summary, our contributions include:

[1]https://www.chargepoint.com/

(1) We comprehensively study a metropolitan-scale, long-term taxicab dataset for analyzing the passenger appearance statistics and taxicab traffic, which serve as the foundation for the design of *PickaChu*.
(2) We propose *PickaChu* that determines the regions for charger deployment and the number of chargers in each region. It minimizes the total deployment cost, maximizes the taxicabs' opportunity of picking up passengers at the chargers, and meanwhile ensures the taxicabs' continuous operability.
(3) We have conducted extensive trace-driven experiments on the SUMO urban mobility simulator to show the effectiveness of *PickaChu* in reducing idle time and supporting operability of the taxicabs compared with a method focusing on minimizing the taxicabs' cruising time for passengers [29] and a method focusing on minimizing the taxicabs' seeking time for chargers [17]. Compared with the previous methods, *PickaChu* reduces the taxicabs' daily average idle time by 81% under the same charger deployment cost. When minimizing the charger deployment cost, *PickaChu* reduces the number of chargers by 27%, but still reduces the taxicabs' daily average idle time by 61%.

In our knowledge, *PickaChu* is the first work for both reducing the idle time and maintaining the SoC of the taxicabs through proper deployment of opportunistic wireless chargers. The remainder of the paper is organized as follows. Section 2 presents literature review. Section 3 presents our metropolitan dataset measurement results. Section 4 presents the detailed design of *PickaChu*. Section 5 presents performance evaluation results. Section 6 concludes the paper with remarks on our future work.

2 RELATED WORK

Taxicab dispatching. Yuan *et al.* [28] introduced a method that schedules the pick-up locations with the shortest routes for taxi drivers and the waiting locations for passengers to reduce the cruising time. Zheng *et al.* [33] modeled the behavior of vacant taxicabs with a non-homogeneous Poisson process to find the optimal waiting positions for passengers. Zhang *et al.* [31] proposed a method to estimate the revenue of each route, and guide the taxicab to the route with the maximum estimated revenue. Zhang *et al.* [29] proposed *pCruise*, in which each taxicab collects the passenger requests from nearby taxicabs and accordingly cruises on the routes with the maximum probability of finding a passenger. Although these works aim to guide taxicabs to pick up the expected passengers with the shortest route, the taxicabs still need to spend much time on driving to the suggested locations without passengers on board. Moreover, the time wasted on seeking chargers and charging is not considered in these works.

Charger deployment. Qin *et al.* [21] scheduled the plug-in charging stations to minimize the time on seeking and waiting in charging stations based on the estimated time and location that each EV needs to be charged. Zhang *et al.* [32] further considered the uncertainty of the EVs' arrival times at the charging stations to shorten the time on seeking chargers and charging. Li *et al.* [17] determined the locations for deploying plug-in charging stations that minimize the time on seeking chargers. Yan *et al.* [26] proposed a method on deploying dynamic wireless chargers based on the features of the positions (i.e., vehicle passing speed, vehicle visiting frequency). Although these works can support the continuous operability of the taxicabs by adapting the deployment of chargers to cover the actual traffic, the taxicabs still have to spend extra idle time on seeking chargers and charging upon the exhaustion of the battery.

Novelty of *PickaChu*. Compared with the above methods, *PickaChu* is novel in two aspects. First, it considers the likelihood of passenger appearance in the regions in deploying the chargers to enable taxicabs to have high likelihood of picking up passengers while being parked for opportunistic charging, which saves the time wasted on cruising, seeking chargers and charging. Second, it supports the taxicabs' SoC by considering their traffic flows, which further reduces the time of seeking chargers and charging.

3 METROPOLITAN-SCALE DATASET MEASUREMENT

3.1 Dataset Description

In this section, we analyze large datasets that record the status of taxicabs in Shenzhen city of China for 12 months (Jan 1 – Dec 31, 2015), with a recording time period of 30 seconds. The datasets include:
(1) **Taxicab data.** It is collected by the Shenzhen Transport Committee, which records the status (e.g., timestamp, GPS position, speed, occupancy) of 15,610 taxicabs. For the occupancy status, 0 means "non-occupied", and 1 means "occupied". The daily size of the uploaded data is around 2 GB.
(2) **Road map data.** The road map of Shenzhen is obtained from OpenStreetMap [5]. According to the municipal information of Shenzhen [17], we used a bounding box with coordinate ($lat = 22.4450, lon = 113.7130$) as the south-west corner, and coordinate ($lat = 22.8844$, $lon = 114.5270$) as the north-east corner, which covers an area of around 2,926 km^2, to crop the road map.

For data management, we utilized a 34 TB Hadoop Distributed File System (HDFS) [1] on a cluster consisting of 10 nodes, each of which is equipped with 16 cores and 64 GB RAM. For data processing, we used Apache Spark [2], which is a fast in-memory cluster computing system running on Hadoop [1].

3.2 Definitions

We first build a *road network*, in which vertices represent landmarks (i.e., intersections or turning points), and edges represent road segments [30, 34]. The movement record of a taxicab is continuous, namely a sequence of GPS positions with corresponding timestamps. We presume that a taxicab has finished its previous trajectory if it stops at a location for more than 10 minutes or its occupancy status changes. Thus, such stopping locations cut the movement record of a taxicab into multiple trajectories. The original GPS positions are scattered around the road segment.

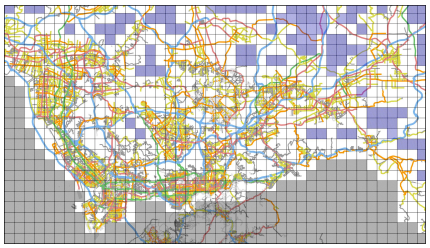

Fig. 5. Gridded road map.

If we apply the optimization on all the GPS positions, we will need to ensure that the taxicabs' SoC on each position is above the threshold. This will be too complex to obtain an optimal solution for the optimization problem. To map them to a uniform road network for the reduction of optimization complexity, we normalize the original GPS positions to their respective nearest landmarks (in Euclidean distance) as in previous methods [25, 28, 29, 36]. Note we only use landmarks in the traffic estimation and optimization of the chargers. For the extraction and analysis of passenger appearance, we still rely on the original GPS positions. We introduce two definitions below.

Definition 3.1. Vehicle Trajectory. A vehicle v_i's trajectory is a sequence of time-ordered landmarks, $Tr_i : \{(\boldsymbol{p_0}, t_0), (\boldsymbol{p_1}, t_1), \ldots, (\boldsymbol{p_r}, t_r)\}$, where each landmark is represented by a latitude and a longitude $\boldsymbol{p_j} = (lat_j, lon_j)$.

Definition 3.2. Region. The road map is partitioned into a set of 557 regions $G = \{\boldsymbol{g_0}, \boldsymbol{g_1}, \ldots, \boldsymbol{g_{M-1}}\}$ with a size of 2,000 m × 2,000 m (Figure 5). Each region is represented by $\boldsymbol{g_i} = \{(lat_i^0, lon_i^0), (lat_i^1, lon_i^1)\}$.

For the ease of analysis, we use a static region size to partition the road map. Some recent works have proved that partitioning the road map with dynamic region sizes can better adapt to the geographical distribution of the passenger appearance [20, 35]. We will use dynamic region size in our future work, but the region size determination does not change the fundamental methods proposed in this paper. The reason we choose 2,000 m × 2,000 m as the region size is to ensure that for the taxicabs within a region, they can reach any position of the region within roughly 6 minutes, which is an acceptable waiting time length for most passengers [28], at the driving speed of 40 km/h (i.e., the approximate average speed

limit of Shenzhen [30]). Combining the taxicabs' trajectories with the changes of their occupancy status, we extracted the pick-up and drop-off locations of the passengers. We calculated the number of passenger pick-ups in each region per unit time (e.g., 30 minutes).

3.3 Motivation

Different from other public vehicles (e.g., buses or subways), which follow fixed routes every day, the movement of taxicabs is driven by the discovery of passengers. To maximize the service performance of the taxicabs, we must maximally reduce the duration of their idle phases (i.e., cruising for passengers, seeking a charger, and charging). This requires us to design a mechanism that provides opportunity of picking up passengers for the taxicabs, and meanwhile keeps the SoC of taxicabs above a threshold. However, the designing of such a mechanism is nontrivial because we need to address two main issues:

(1) **Measuring likelihood of passenger appearance.** The historical average number of passengers that appeared during a unit time (e.g., per hour, per day) can be an indicator of the passenger appearance likelihood. However, using this metric alone for the likelihood measurement may not be accurate for guiding taxicab pick-ups in terms of waiting time. We hope that when a taxicab arrives at a charger and gets charged at a random time, it does not have to wait long before discovering a passenger. For example, in an area mostly consisting of residential buildings, many passengers may appear during rush hours (e.g., 08:00-09:00), resulting in a relatively high hourly average number of appeared passengers. However, this high value does not mean that passengers frequently appear at other times. Thus, we need a new metric that can more accurately reflect the passenger appearance likelihood to guide taxicab pick-ups.

(2) **Supporting taxicabs' continuous operability.** Regions with higher passenger appearance likelihood should have a higher priority to be deployed with a charger in order to offer sufficient passenger pick-up opportunity at the chargers. In addition, we aim to minimize the number of chargers (i.e., deployment cost) while maintaining the taxicabs' continuous operability.

We then analyze the Shenzhen taxicab dataset. For the first issue, we measure the building functionalities and their respective contribution to passenger appearance, and the frequency of passenger appearance in the regions. For the second issue, we measure the distribution of the trajectory lengths of all the taxicabs to model their traffic, which will be used to estimate the expected SoC of taxicabs in each region.

3.4 Dataset Analysis Results

3.4.1 Building Functionality and Passenger Appearance. It was indicated that the passenger appearance in a region is closely related to its composition of buildings, and the likelihood of passenger appearance varies for different classes (i.e., functionalities) of buildings [7, 31, 36]. In this analysis, we attempt to verify if the density and functionalities of buildings (e.g., Residential, Commercial buildings) in a city region influence the number of taxicab passengers in the region.

To study the relation between buildings and the appearance of passengers, we derived the distribution of passenger pick-up events in a part of the road map. As shown in Figure 6, we plot each passenger pick-up event happened in 2015 with a point and drew the heat map. The warmer color a region has, the more concentrated in a short time duration the passenger pick-up events occur. Based on [9, 27] and OpenStreetMap [5], we obtained the class and position of each building in Shenzhen. The building classes include *Residential*, *Commercial*, *Civic*, *Basics*, *Professional* and *Tourism*, as shown in Figure 7. The *Residential* class consists of buildings primarily for residential purposes (e.g., apartments). The *Commercial* class consists of buildings for commercial activities (e.g., supermarkets). The *Civic* class consists of buildings for municipal purposes (e.g., library). The *Basics* class consists of buildings for public

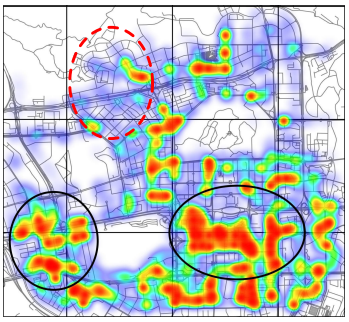

Fig. 6. Heat map of passenger pick-ups.

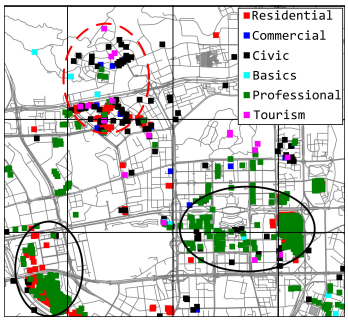

Fig. 7. Distribution of buildings.

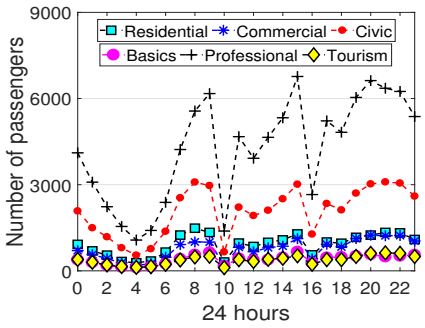

Fig. 8. Num. of passengers for each building class.

service (e.g., garage). The *Professional* class consists of buildings for specific usage (e.g. train/subway stations, airports). The *Tourism* class consists of buildings for recreation (e.g., garden).

By comparing the two figures, we can see that the occurrence of passenger pick-up events generally concentrates at the regions with abundant buildings (e.g., the two regions on the bottom marked with solid circles). In the region on the top marked by red dashed circles, there are much fewer pick-up events though it has many buildings. This is because the majority are residential and civic buildings, where people often have planned travel schedules using private vehicles or public transportation. This result implies that building functionality also influences passenger appearance.

Next, we study the correlation between the building functionalities and the number of passengers. We measured the average number of passengers that appeared within 100 meters around each building in a building class during each hour of a day throughout the 365 days, as illustrated in Figure 8. Though we have already considered the number of passengers that appeared around each building in the measurement, additionally considering building size may further increase the precision of the measurement, which is left as our future work. We further calculated the average, 5^{th} and 95^{th} percentiles of the hourly number of passengers that appeared nearby for each building class, which are illustrated in Figure 9. We see that significantly more passengers appeared nearby the *Professional* buildings than the other building classes during all times. This is because the *Professional* class mostly consists of offices and business buildings that are frequently visited by many people. The *Civic* class has the second most passengers because it mostly consists of libraries and community centers with many public activities. The *Commercial* and *Residential* classes have much fewer passengers than the former two classes because these buildings are not continuously visited by people during a day. The *Basics* and *Tourism* classes have the fewest passengers because there are fewer such buildings. Therefore, the building functionality can be used as a factor to infer the likelihood of passenger appearance in the regions. As the number of pick-ups does not necessarily equal to the number of passenger requests, we need to use other additional factors to more accurately estimate the likelihood of passenger appearance.

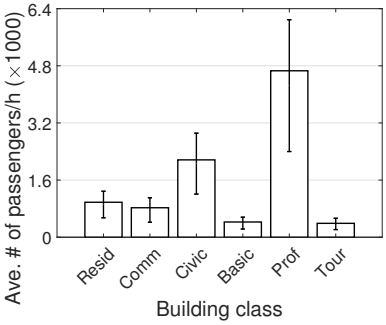

Fig. 9. Average number of passengers per hour.

3.4.2 Frequency of Passenger Appearance. We hope that when a taxicab arrives at a region deployed with a wireless charger, it does not need to wait long before it discovers a passenger. This means that the frequency of passenger appearance in the region must be high, namely the time interval between two consecutive passenger appearances must be short. Note that one passenger appearance means the appearance of passenger(s) at one time. In Figure 10, in Region1, three passengers appearing at one time

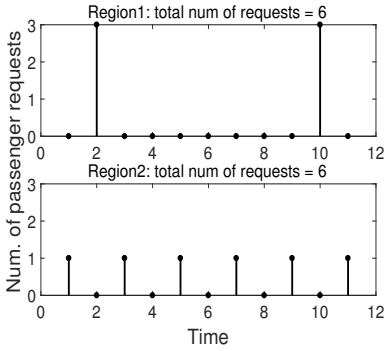

Fig. 10. Frequency comparison of passenger requests' appearance.

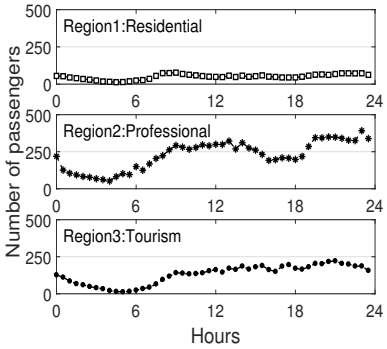

Fig. 11. Passenger time series of regions.

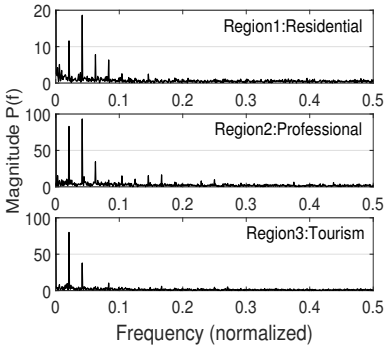

Fig. 12. Periodogram of the passenger time series.

is considered as one passenger appearance, and in Region2, one passenger appearing at one time is also considered as one passenger appearance. Then, the frequency of passenger appearance for Region1 is 1/8, and that for Region2 is 1/2. However, the average number of passengers per unit time (i.e., a day) cannot reflect this frequency. For example, Region1 has 3 passengers in every 8 time units, while Region2 has 1 passenger in every 2 time units. Though both regions have 6 passengers in every 12 time units, Region2 has a higher passenger appearance frequency (1/2) compared with Region1 (1/8), which makes a taxicab wait for a shorter time before it discovers a passenger. As a result, we need to develop a new method to determine the frequency of passenger appearance in a region.

To find the frequency, we draw passenger appearance time series. For each region, we calculated the number of passengers that appeared in every 30 minutes (i.e., a sample) in each day for the 365 days in the dataset. Among the regions mostly consisting of (i.e., more than 50%) *Residential*, *Professional* and *Tourism* buildings, we randomly chose one region respectively, and denote them as *Region1:Residential*, *Region2:Professional* and *Region3:Tourism*. Figure 11 shows the number of passengers that appeared per unit time (i.e., 30 minutes) in the first day of the three regions, respectively. We define a pattern as the periodic occurrence of a certain number of passengers in a certain time period, and its frequency as the number of such occurrences per unit time. If the time series of every region has only one pattern, identifying its frequency is easy. However, the time series may have multiple patterns, which makes it hard to measure the passenger appearance frequency in the region. In the signal processing field, the time series curve in Figure 11 can be considered as a composition of multiple patterns with different frequencies. To find out the frequencies of the patterns, we can decompose the time series to a group of time series with different frequencies using a signal processing technique. Specifically, we applied the Discrete Fourier Transform (DFT) on the passenger time series and got their periodogram [23], as shown in Figure 12. In the figure, the X-axis is the possible frequencies of the patterns in the time series, and the Y-axis reflects the number of passengers in a pattern with a frequency (e.g., 3 and 1 in the above example). We notice that the periodogram of Region2 has relatively more patterns with high frequencies than Region1 and Region3, although the numbers of passengers in the high-frequency patterns are much smaller than that of the low-frequency patterns. This is because the *Professional* buildings are frequently visited by many people, which results in more frequent passenger appearances than the *Residential* and *Tourism* buildings. Compared with Region3, Region1 has more patterns with higher frequencies. This is because people's visiting patterns at the *Tourism* buildings is more likely to follow certain routine (e.g., open and close times) than the *Residential* buildings, which is more randomly visited by people.

Thus far, we have verified that the time series of the passenger appearance of a region can be decomposed to a group of time series with different frequencies. Then, we design a method to combine these frequencies

to measure how frequently passengers appear in a region, which will be introduced in Section 4.2.2. As a result, the region with a higher final frequency metric should have a higher priority to deploy chargers.

3.4.3 Idle Trip Time & Taxicab Traffic. As discussed before, taxicabs may waste much time on cruising for passengers, seeking chargers and getting charged. We then analyzed the Shenzhen dataset to see how much time is spent on these idle operations. We first introduce the definitions for the operations of the taxicabs. We define the cruising time as the time interval between the taxicab dropping off a passenger and picking up the next passenger. From the Shenzhen Transport Committee, we obtained the locations of all the existing plug-in charging stations in Shenzhen. If a taxicab's movement record shows that it has stayed at a charging station for more than 5 minutes, we consider that it was being charged at the station at that time. Therefore, we define the time for seeking a charger as the time interval between the taxicab dropping off its last passenger and entering a charging station to charge. We define the charging time of a taxicab at a charging station as the time duration that the taxicab stayed at the charging station. For each vehicle, we calculated the duration of each idle operation in each day throughout the 365 days, and then calculated the average duration per day. We show the Cumulative Distribution Function (CDF) of the taxicabs in terms of the daily average duration of each operation in Figure 13. We can see that about 50% of the taxicabs spent more than 4.17 hours on cruising per day in average, about 50% of the taxicabs spent more than 2.78 hours on seeking chargers per day in average, and about 50% of the taxicabs spent more than 0.83 hours on charging per day in average. The analytical results indicate that we should try to avoid or reduce the time duration in these idle operation phases when determining the locations to deploy chargers. We can choose the locations where many passengers appear with high frequency, so that when a taxicab is being charged, it has a high probability to quickly pick up a passenger.

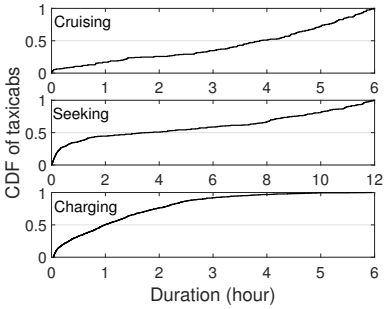

Fig. 13. Distribution of duration of idle trips.

We should also make sure that the deployed chargers can support the continuous operability of the taxicabs considering the taxicabs' traffic flows in the city. Because the taxicabs' trajectories reflect their traffic flows between different locations [27], and the trajectory length generally determines energy consumption of a taxicab, we calculated the lengths of the taxicabs' trajectories to determine the taxicab traffic that the deployed chargers need to support. Figure 14 shows the Probability Density Function (PDF) of the trajectory lengths. If we can describe the taxicabs' trajectory lengths with a certain distribution, we can further determine the deployment of chargers to support these trajectory trips so that the expected SoC of a taxicab at a landmark is always above a certain threshold that allows it to reach its nearest charger. Obviously, the distribution of the trajectory lengths cannot be modeled using a parametric distribution (e.g., Gaussian). Since KDE is a non-parametric method to estimate the PDF of a random variable, we input the trajectory lengths to the KDE model to output a taxicab's probability of reaching each landmark in the road network. The red curve in Figure 14 represents the fitting result from the KDE. We will present more details of this model in Section 4.3.

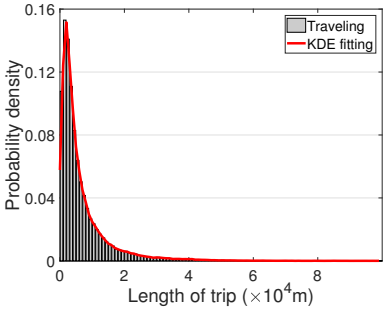

Fig. 14. Traveling trip lengths & KDE estimation.

3.4.4 Summary. Based on the above observations, to deploy the chargers that maximally reduce the idle time of taxicabs, we need to i) consider the density and functionality of buildings and their respective influence weights on the appearance of passengers, ii) measure the passenger frequency in a region from

the region's passenger appearance time series, and iii) estimate the taxicabs' SoC based on the taxicabs' traffic flows. Considering these factors, we will find a solution in Section 4 for the following problem.

Problem: Given a road network comprised of a set of regions G, and taxicabs' trajectory datasets $\{Tr\}$, how to select regions to deploy chargers with the minimum cost so that the expected SoC of the taxicabs at each landmark is no less than a threshold, and the taxicabs have high probability of discovering passengers while being charged?

4 SYSTEM DESIGN OF PICKACHU

4.1 Framework of PickaChu

PickaChu consists of the following three stages as shown in the three dashed boxes in Figure 15.

1. **Map gridding & information derivation**. First, the entire city area is partitioned into a *Gridded Roadmap* consisting of several equal-sized regions. Also, the taxicab dataset is cleaned up (e.g., filtering out positions out of the actual range of Shenzhen, redundant positions). Then, based on the cleaned data, we derive the *Taxicab Trajectories*, which will be used for extracting passenger requests and building traffic models. From the taxicabs' change of occupancy status from 0 to 1, we extract the *Passenger Appearance Records* (i.e., location and time). Finally, based on the *Gridded Roadmap* and the *Passenger Appearance Records*, we calculate the *Passenger Appearance Time Series* for each region.
2. **Measuring likelihood of passenger appearance** (Section 4.2). Based on the output from the first stage, we consider the *Number of Passengers Per Unit Time*, the *Building Functionality*, and the *Passenger Appearance Frequency* for each region to assign *Region Scores* to regions to measure their likelihood of passenger appearance.
3. **Charging position determination** (Section 4.3 and Section 4.4) We first use the lengths of the trajectories to model the *Continuous Operability Support* using KDE, which is used to estimate the taxicabs' expected SoC at different regions. Then, we formulate a multi-objective optimization problem to solve the wireless charger deployment problem, and its solution is the *Charger Position Determination* (i.e., where and how many wireless opportunistic chargers we should deploy).

4.2 Measuring Passenger Appearance

In the following, we firstly introduce how *PickaChu* estimates the likelihood of passenger appearance via a weighted sum of building functionalities. Then, we elaborate how *PickaChu* measures the frequency of passenger appearance in a region. Finally, we design a scoring mechanism for measuring each region's likelihood of passenger appearance.

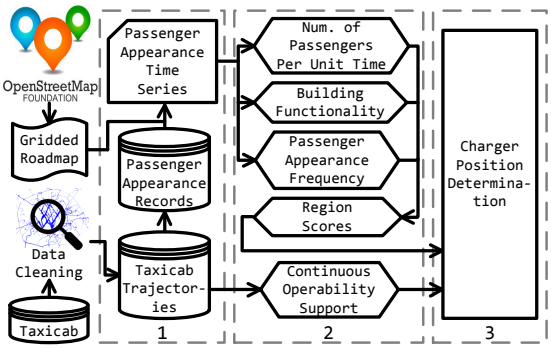

Fig. 15. Framework of PickaChu.

4.2.1 Building Functionality. Different regions have different densities of buildings with different functionalities (e.g., *Residential* buildings, *Commercial* buildings). For example, the region in a central business district is likely to be filled with office buildings and shopping centers where passengers frequently appear, while the region in a residential area is likely to be filled with dwellings where a large number of passengers only appear during specific hours. Correspondingly, we use a weighted sum of building functionalities within a region to measure the buildings' potential contribution to passenger appearance.

We set the weight of a building class as the hourly average number of passengers that appeared nearby (e.g., within 100 meters) each building in the class throughout the dataset. For example, according to

Figure 9 of our trace analysis, the weights of the building classes are: *Residential*=0.9, *Commercial*=0.7, *Civic*=2.0, *Basics*=0.2, *Professional* =4.4, and *Tourism*=0.2.

Suppose C is the set of the building classes in a region $\boldsymbol{g_i}$, and $\mathrm{P}_i(c)$ is the probability function of building class c, i.e., the percentage of buildings with building class c in $\boldsymbol{g_i}$. $\mathrm{w}(c)$ is the passenger appearance weight of building class c. We define the weighted sum of the building functionalities in $\boldsymbol{g_i}$ as:

$$\bar{H}_i = \frac{B_i}{B_{max}} \sum_{c \in C} \mathrm{w}(c)\mathrm{P}_i(c) \tag{1}$$

where B_i is the total number of buildings in $\boldsymbol{g_i}$, and B_{max} is the maximum number of buildings in a region among all the regions, i.e., $B_{max} = \max_{\boldsymbol{g_i} \in G} B_i$. Suppose a region has the following composition: {*Residential* (20%), *Commercial* (5%), *Civic* (20%), *Basics* (5%), *Professional* (10%), *Tourism* (40%)}, and $B_i = 100$, $B_{max} = 500$. Its weighted sum of the building functionalities is $\frac{100}{500} \times (0.9 \times 0.2 + 0.7 \times 0.05 + 2.0 \times 0.2 + 0.2 \times 0.05 + 4.4 \times 0.1 + 0.2 \times 0.4) = 0.23$. From Section 3.4.1, we know if a region has more heavy-weighted buildings, it has a larger H_i, meaning it tends to have more passengers.

4.2.2 Frequency of Passenger Appearance. When deploying chargers, we hope that when a taxicab arrives at a charger at a random time, it has a high probability of discovering a passenger nearby. It means that the region has a high frequency of passenger appearance and the number of passengers should be high at a time. As shown in Section 3.4.2, a passenger appearance time series can be considered as being composed by a group of patterned time series with different frequencies. We call the area size (i.e., the number of passengers) of a pattern (i.e., Y value in Figure 12) the *magnitude* of the pattern. Thus, we need to i) derive passenger appearance frequency, ii) derive the patterns with a high magnitude, and iii) find a way to measure the global frequency given multiple patterns. For tasks i) and ii), we use the approach introduced in [23]. For task iii), we design a metric. The details are introduced below.

In Section 3.4.2, we show that we can detect the potential patterns and their frequencies of a region's passenger appearance time series through DFT. However, DFT may generate false frequencies in the periodogram [18]. AutoCorrelation Function (ACF), another method for detecting repeated patterns, can avoid false detection of frequencies of a time series [23], but may result in the detection of integer times of true periods (i.e., reciprocal of the frequencies) [18]. For example, in addition to the true frequency of a pattern, say 1/30, the frequencies, which are integer multiples of 1/30 (i.e., {1/60, 1/90, ...}), are also falsely considered as the frequencies of this pattern. Therefore, solely using DFT or ACF cannot accurately determine the true frequencies in a time series. To more accurately find the patterns, we adopt the approach in [23] that combines the results from DFT and ACF to identify frequencies.

Below, we first present how to derive patterns with significant magnitude [23] from the periodogram generated by DFT. We then present how to get the intersection of the two groups of frequencies from DFT and ACF as the final detected frequencies. Finally, we propose a method that combines all the frequencies to get a global metric to evaluate the frequency of passenger appearance in a region.

(1) *Deriving patterns with significant magnitude.* As shown in Figure 12, some patterns have an extremely low magnitude. Therefore, we first determine the base magnitude and then derive the patterns with magnitude larger than the base magnitude [23]. Considering that any random time series has patterns with certain magnitudes [23], we use its maximum magnitude (denoted by p_i^{max}) as the base magnitude. In a region $\boldsymbol{g_i}$, the passenger appearance time series is defined as: $x_i(n), n = 0, \ldots, N-1$, where N is the total number of samples and $x_i(n)$ is the value of the n^{th} sample. To create random time series, we randomly shuffle the original $x_i(n)$ into a new sequence $\tilde{x}_i(n)$. To ensure 99% confidence level on the selection of the base magnitude, we repeat the shuffling for 100 times and record the maximum magnitude each time. Finally, we choose the 99^{th} value as the base magnitude.

(2) *Determining global frequency.* In step (1), for $\boldsymbol{g_i}$, we select potential significant patterns with frequencies denoted by $F_i^{DFT} = \{f_i^1, f_i^2, \ldots, f_i^{m'}\}$. Then we use ACF to identify the patterns with frequencies denoted by $F_i^{ACF} = \{f_i^1, f_i^2, \ldots, f_i^{m''}\}$. The final frequency set is calculated by: $F_i = F_i^{DFT} \cap F_i^{ACF}$.
(3) *Measuring passenger request frequency in a region.* Suppose the magnitudes of the significant patterns with frequencies $F_i = \{f_i^1, f_i^2, \ldots, f_i^m\}$ in the time series are $P_i = \{p_i^1, p_i^2, \ldots, p_i^m\}$. Since the magnitude of a pattern reflects how significant this pattern is to the entire time series, we use the weighted sum of the frequencies of the significant patterns to describe passenger appearance frequency in each region. We call it region $\boldsymbol{g_i}$'s weighted frequency of passenger appearance and denote it by $\bar{F}_i$.

$$\bar{F}_i = \sum_{k=1}^{m} \frac{p_i^k}{\sum_{j=1}^{m} p_i^j} \cdot f_i^k. \tag{2}$$

For example, consider a time series which has two significant patterns with magnitudes of 2 and 3, and frequencies of 1/10 and 1/20, respectively. The weighted frequency of this region is calculated as $\frac{2}{5} \times \frac{1}{10} + \frac{3}{5} \times \frac{1}{20} = \frac{7}{100}$.

4.2.3 Likelihood of Passenger Appearance. *PickaChu* assigns scores to the regions to show their likelihood of passenger appearance considering the above metrics. We favor the regions with more passengers, and higher frequency of passenger appearance. Therefore, we define the score of a region, say $\boldsymbol{g_i}$, as:

$$\rho(\boldsymbol{g_i}) = (\frac{\bar{x}_i}{\bar{x}_{min}})^{\alpha} \cdot \bar{F}_i^{\beta} \cdot \bar{H}_i^{\gamma} \tag{3}$$

where $\bar{x}_i = \frac{\sum_{n=0}^{N-1} x_i(n)}{N}$ is the average number of passengers over all the N samples of $\boldsymbol{g_i}$, $\bar{x}_{min}$ is the minimum average number of passengers in a region among the regions, $\bar{F}_i$ is $\boldsymbol{g_i}$'s weighted frequency of passenger appearance, $\bar{H}_i$ is the weighted sum of building functionalities in $\boldsymbol{g_i}$, and α, β, and γ are constants that control the respective influence of the three metrics. We scale $\bar{x}_i$ by $\bar{x}_{min}$ to constrain the scores of the regions that have few passengers, which have almost no contribution on increasing the score. To find the best values for α, β, and γ, we vary each variable within a certain range (e.g., [1, 5]) and test different combinations of the values. Specifically, we use each combination to determine the deployment of the chargers and run our experiment for 1 hour randomly chosen among the 24 hours of a day. Then, we choose the combination of the values that results in the minimum time duration of the idle phases on the vehicles (i.e., cruising, seeking for chargers and charging) as the final setting. We find $\alpha = 1.2$, $\beta = 2$ and $\gamma = 1$ is the best combination for the case of Shenzhen.

Table 1. Comparison between two regions.

	$\boldsymbol{g_1}$	$\boldsymbol{g_2}$
$\bar{x}_i$	110	22
$\bar{F}_i$	1/10	1/2
Composition	1 airport 1 barn	2 houses
Building wgt	Professional=4.4 Basics=0.2	Residential=0.9
W/o building	11	11
W/ building	25.3	9.9

Note that the region scores calculated by Equation (3) is not the optimal way to reflect the likelihood of passenger appearance, and it is only a heuristic approach. It is difficult to find the optimal way to describe the distribution of the likelihood of passenger appearance in different regions. In order to make the scores more accurately reflect the likelihood of passenger appearance, in addition to using parameters $\bar{x}_i$ and $\bar{F}_i$, we further consider the weighted sum of the building functionality ($\bar{H}_i$) that also reflects the number of passengers in a region. In other words, parameter $\bar{H}_i$ enlarges the difference between the regions with higher likelihood of passenger appearance and the regions with low likelihood of passenger appearance. That is, the distribution of region scores calculated by Equation (3) is closer to the actual distribution of the likelihood of passenger appearance in different regions. In spite of the simplicity of this approach, it is helpful for differentiating the likelihood of passenger appearance in the regions. We use a simple example to show the effectiveness of additionally considering parameter $\bar{H}_i$. Suppose we have two regions, say $\boldsymbol{g_1}$

and $\boldsymbol{g_2}$, with different compositions of buildings, of which details are summarized in Table 1. We can see that in $\boldsymbol{g_1}$, there is an airport (50% of the buildings, classified as *Professional*), and a barn (the other 50% of the buildings, classified as *Basics*); while in $\boldsymbol{g_2}$, there are 2 houses (100% of the buildings, classified as *Residential*). Suppose the weights of the building classes are: *Professional*=4.4, *Basics*=0.2, and *Residential*=0.9. For simplicity for this example, we set $\alpha = \beta = \gamma = 1$. Without considering buildings, the region scores are $\rho(\boldsymbol{g_1}) = 110 \times \frac{1}{10} = \rho(\boldsymbol{g_2}) = 22 \times \frac{1}{2} = 11$, which means we cannot differentiate which region is better for picking up passengers. The reason that $\boldsymbol{g_1}$ has the same region score as $\boldsymbol{g_2}$ even though $\boldsymbol{g_1}$ has an airport (i.e., *Professional* building class), which has a high frequency of passenger appearance, is because the low frequency of passenger appearance of the barn makes the frequency of passenger appearance of region $\boldsymbol{g_1}$ low. The region scores calculated with considering buildings are $\rho(\boldsymbol{g_1}) = 110 \times \frac{1}{10} \times (\frac{1}{2} \times 4.4 + \frac{1}{2} \times 0.2) = 25.3$, and $\rho(\boldsymbol{g_2}) = 22 \times \frac{1}{2} \times 0.9 = 9.9$. The result shows that $\boldsymbol{g_1}$ is better than $\boldsymbol{g_2}$ for picking up passengers, which is consistent with our intuition that regions with airports are more likely to have high and constant flows of passenger appearances. This example shows that the additional consideration of buildings in Equation (3) can help more accurately reflect the likelihood of passenger appearance.

4.3 Supporting Continuous Operability

One of our objectives in the charger deployment is to ensure that the taxicabs can reach a nearby charger when their SoC is about to be exhausted (e.g., below 20%). To this end, we need to infer the taxicabs' expected SoC at each region given certain regions are installed with wireless chargers. KDE can be used to describe the taxicabs' probability of reaching a region from another region based on their distance in the road network. Also, the SoC of a taxicab is a function of the distance from the taxicab's source landmark to the destination landmark. Then, the expected SoC of a taxicab at a landmark in the road network can be calculated. We present the details below.

Since taxicabs' mobility patterns imply their traffic flows between certain locations [27], we feed their trajectories into a KDE model to infer the Probability Density Function (PDF) of the distribution of the trajectory lengths as in Equation (4). Given a trajectory length d, the KDE model outputs the probability that a taxicab takes a trajectory with length d.

$$\hat{f}(d) = \frac{1}{R \cdot h} \sum_{t=0}^{R-1} K(\frac{d - d_t}{h});\ -\infty < d < \infty, \tag{4}$$

where R is the total number of the taxicab trajectories, d_t is the length of the t^{th} trajectory, h is the smoothing parameter influencing the estimation accuracy of the KDE and is determined according to the MISE criterion [24], $K(\cdot)$ is the kernel function whose value decays with the increasing of d, which is set to the Gaussian function based on [15, 16].

According to the state-of-the-art EV energy consumption model [14], the energy consumption of a taxicab (E_c) is primarily determined by air drag (E_{air}) and rolling resistance (E_{roll}):

$$\begin{aligned} \Delta E_c &= \Delta E_{air} + \Delta E_{roll} \\ &= c_w v^2 \Delta l + c_e \kappa g \Delta l \end{aligned} \tag{5}$$

where c_w is the air drag coefficient determined by vehicle front surface area; v is the driving speed; Δl is the distance that the taxicab has moved; c_e is the rolling resistance coefficient; κ is the taxicab's mass; and g is the gravity acceleration.

Suppose the taxicabs have the same battery capacity, E_0, and each taxicab gets fully charged before leaving a charger. We define the shortest distance between two regions as the distance of the shortest route between their respective central landmarks, which are the landmarks located the nearest to the

middle of the two regions, respectively. Given a taxicab starting from a charger, based on Equation (5), its residual energy at a location, which is d distance away from the charger through the shortest route, can be estimated as $E_r^d = E_0 - \sum_{t=0}^{R'-1}(c_w v_t^2 + c_e \kappa g) l_t$ [14], where R' is the number of road segments of the shortest route, and v_t and l_t are the speed limit and length of the t^{th} road segment, respectively. The taxicab's SoC at the location can be represented as:

$$SoC(d) = \begin{cases} E_r^d / E_0, & \text{if } E_r^d \geqslant 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We use a natural number μ_i to denote the number of chargers deployed in region $\boldsymbol{g_i}$. We set $b_i = 0$, if $\mu_i = 0$; $b_i = 1$, if $\mu_i \geqslant 1$. Then, the expected SoC of the taxicabs at a region $\boldsymbol{g_j} \in G$ is:

$$\overline{SoC}(\boldsymbol{g_j}) = \sum_{i=0}^{M-1} \hat{f}(d_{i,j}) SoC(d_{i,j}) b_i, \tag{7}$$

where M is the total number of regions, and $d_{i,j}$ is the distance of the shortest route from $\boldsymbol{g_i}$ to $\boldsymbol{g_j}$.

4.4 Optimization Problem

Our objective is to minimize the total deployment cost of the chargers, maximize the opportunity of picking up passengers at the charger positions, and meanwhile ensure that at each region, the expected SoC of a taxicab is higher than a threshold η (e.g., 20%). η is determined so that a taxicab can reach the nearest charger with η SoC left. We can set η to be a relatively high value, so that the taxicabs are always operable with high confidence. Meanwhile, the charging rate of the deployed chargers must be able to support the power demands from all the taxicabs. According to Equation (5), we can derive the battery consumption rate for each taxicab as $\phi = \frac{\Delta E_c}{\Delta t} = c_w v^3 + c_e \kappa g v$. Hence, the battery consumption rate depends on the speed limit of every road segment. That is, as the speed limit v increases, the battery consumption rate increases. To derive the maximum battery consumption rate ϕ_{max}, we use the maximum speed limit v_{max} of the entire road map. Finally, the optimization problem is formulated as:

$$\begin{aligned} \text{minimize} \quad & \sum_{\boldsymbol{g_i} \in G} \omega_0 \mu_i \\ \text{maximize} \quad & \sum_{\boldsymbol{g_i} \in G} \rho(\boldsymbol{g_i}) \mu_i \\ \text{subject to} \quad & \overline{SoC}(\boldsymbol{g_i}) \geqslant \eta, \forall\, \boldsymbol{g_i} \in G \\ & \mathcal{C} \sum_{\boldsymbol{g_i} \in G} \mu_i \geqslant \phi_{max} V \\ & \mu_i \in \mathbb{N}, \forall\, \boldsymbol{g_i} \in G, \end{aligned} \tag{8}$$

where ω_0 is a constant representing the unit cost of deploying a charger, $\mathcal{C}$ is the charging rate of one charger, and V is the total number of taxicabs driving in the road map. This problem tries to minimize the total deployment cost of the chargers and maximize the total region scores covered by the chargers with two constraints: i) the expected SoC at any region is no less than threshold η, and ii) the total charging rate of the deployed chargers is not less than the total battery consumption rate of the electric taxicabs. Given source location $\boldsymbol{g_i}$ and destination location $\boldsymbol{g_j}$, the coefficient $\hat{f}(d_{i,j}) SoC(d_{i,j})$ in Equation (7) is determined. Therefore, we can use a constant λ_{ij} to represent $\hat{f}(d_{i,j}) SoC(d_{i,j})$. As a result, the optimization problem (8) is actually a classic Multi-objective Integer Programming (MIP) problem, and its optimal solutions can be found through a branch-and-bound search [6]. We can use an existing solver (i.e. JuMP [19], MultiJuMP [3]) to obtain its integer-feasible solution. After solving the optimization problem, we obtain the number of chargers (μ_i) in each selected region for charger deployment. For each

selected region, we rank the landmarks within the region by their daily average number of passenger requests in descending order, and assign the μ_i chargers to the top ranked μ_i landmarks accordingly.

Note that the more passengers appear in a region (i.e., larger $\rho(\boldsymbol{g_i})$), the more opportunity the taxicabs will have in picking up the passengers in the region [33]. Meanwhile, the chargers will attract vacant taxicabs to wait in the region, namely create the opportunity of picking up passengers for the taxicabs. Therefore, our optimization problem has considered maximizing the opportunity of picking up passengers for the regions. In our future work, we will explore the accurate relationship between the likelihood of passenger appearance and the distribution of vacant taxicabs to better describe the opportunity of picking up passengers in the regions.

4.5 Taxicab Dispatching

During the driving process, the taxicabs follow the rules below in order to quickly discover passengers.

1. If a vacant taxicab finds that its SoC is below certain level θ (e.g., 80%), it moves to the nearest charger and randomly selects a period of waiting time (e.g., 5 to 30 minutes), which is the usual waiting time of taxicab drivers [33].
2. When a taxicab's SoC is below η (e.g., 20%), it seeks the nearest charger to get a full recharge. Note the charging rate of the state-of-the-art vehicular wireless charging system, say $\mathcal{C}$, is 150 kW [10]. This means for a taxicab with a battery capacity of 75 kWh, it can be charged with 20% of SoC within around 7 minutes, which is consistent with the length of time that the taxicabs usually spend on waiting for passengers [31]. In this case, the time required for fully recharging a taxicab is around 30 minutes. Note a full charge can only support a taxicab to drive for around 300 km. However, a taxicab in a metropolitan road network usually needs to drive 800 km in one day [31]. This means that a taxicab needs to charge around 3 times (i.e., roughly 2 hours) to support its daily operation, during which it cannot serve any passenger. Therefore, instead of letting a taxicab be idle for such a long time, we let it charge opportunistically while waiting for passengers.
3. When a vacant taxicab's SoC is above θ, it cruises between chargers to seek passengers.
4. When a taxicab receives a passenger request before or during charging, if its SoC is above η and is sufficient for the travel and subsequent charging, it will stop charging and pick up the passenger; otherwise, it declines the request since maintaining operability has the highest priority. Note that once the taxicab starts to serve a passenger request, it won't stop to charge again until it completes the current request. For the detailed scheduling of the taxicabs, we refer to existing taxicab dispatching methods [28, 29, 33].
5. Our charger deployment ensures that there are chargers in less popular regions because the deployed chargers need to maintain the taxicabs' SoC to be above the threshold. However, the taxicabs may not want to serve in less popular regions. To motivate the taxicabs to stay in less popular regions more often, we specify the unit charging price in less popular regions to be lower (e.g., \$0.11 per kWh), and the unit charging price in popular regions to be higher (e.g., \$0.22 per kWh). We leave the exploration of the optimized pricing strategy for balancing the taxicabs to our future work.

Since a taxicab only waits for 5 to 30 minutes at a charger, cruises between chargers to seek passengers, and meanwhile our charger deployment makes it very likely for a taxicab to pick up a passenger while waiting, the taxicabs are moving around and able to serve the passengers widely distributed in the city. Note the above parameters can be adjusted according to different service requirements. *PickaChu* can easily adopt the taxicab dispatching strategies in previous works [28, 29, 33], which is not our focus in this paper. In a centralized dispatching system, when the system receives a passenger's request, it will find the nearest vacant taxicab and notify it of the pick-up location [33]. In a distributed dispatching

system [28, 29], a taxicab receives passenger request from nearby taxicabs through vehicle-to-vehicle communication, and decides the route to the location.

Though we allocate different numbers of chargers to different regions, it is still possible that when a taxicab arrives at a charger, it must wait in a waiting queue. Currently, the number of chargers in each region is determined based on the likelihood of passenger appearance. Therefore, each taxicab can quickly pick up a passenger and leave the charger. Namely the case of a taxicab waiting for an available charger should be rare. Moreover, we let each taxicab start looking for an available charger as long as its SoC is below 80%, and it will keep moving between the chargers until it finds an available charger. Thus, the taxicab will not just wait at a charger position for its turn of charging. Therefore, the possible waiting time caused by an unavailable charger is included in the seeking time for charger. We will further study how to optimize the number of chargers at a charging position so that the taxicabs' seeking time caused by looking for an available charger can be minimized.

In the current design of PickaChu, we mainly focus on regular passenger appearance with stable periods (e.g., airport passenger flows, daily rush hours). For disruptions or unplanned events, we rely on existing taxicab dispatching methods [28, 29, 33] to guide the taxicabs to adapt to the variation of passenger appearance frequency. We leave the comprehensive solution of this problem as our future work. The chargers are deployed one time. When the long-term traffic flows in the city change significantly, the wireless chargers need to be re-deployed.

5 PERFORMANCE EVALUATION

5.1 Comparison Methods

To evaluate *PickaChu*'s performance in reducing the idle time and supporting the continuous operability of electric taxicabs in a city, we compare it with a representative charging station deployment algorithm: Optimal Charging Station Deployment [17] (*OCSD* in short), and a representative taxicab guiding system: cruising on purpose (*pCruise* in short) [29]. We also evaluate the performance of existing deployment of plug-in charging stations in Shenzhen (*Baseline* in short) as the baseline.

To make the methods comparable, we assume that they all use the same wireless chargers. In *OCSD*, based on the analysis of taxicab mobility, the chargers are deployed to minimize the taxicabs' average seeking time for the nearest charger. To make methods comparable, in *PickaChu*, the deployment of chargers is determined by our optimization solution with the same cost as in *OCSD*. To demonstrate that *PickaChu* can further reduce the deployment cost, we also evaluated *PickaChu* with its optimization problem solution that minimizes deployment cost (denoted by *OptPickaChu*). We let *OCSD*, *PickaChu*, *OptPickaChu*, and *Baseline* all use the centralized taxicab dispatching system explained in Section 4.5. As *OCSD* and *Baseline* do not have a strategy to guide pick-ups, the taxicabs wander around in the road network to discover passengers before receiving notifications. In *pCruise*, the taxicabs share passenger information and cruise on the route with the most passenger requests. By communicating with its nearby taxicabs, each taxicab creates a cruising graph, which is the taxicab's nearby road network with vertices representing intersections, and edges weighted by the probability of finding a potential passenger. The probability is calculated as the number of unserved passenger requests over the total number of passenger requests found on the route. Then it uses the cruising graph to select the route that has the maximum probability of finding a passenger. In *pCruise*, we use the same charger deployment as that in *OCSD*. In all the methods, when a taxicab's SoC is below 20%, it drives to the nearest charger to get a full recharge, during which they won't serve passengers.

Parameters	Setting
Charging rate $\mathcal{C}$	150 kW
Charger unit price ω_0	$2,000
Air drag coefficient c_w	0.3
Rolling resistance coefficient c_e	0.01
Mass of a taxicab κ	2,020 kg
Gravity acceleration g	9.8 m/s^2
Battery capacity of a taxicab E_0	75 kWh
SoC threshold η	20%
Vacant SoC threshold θ	80%
Maximum speed limit v_{max}	60 mph

Table 2. Table of parameters.

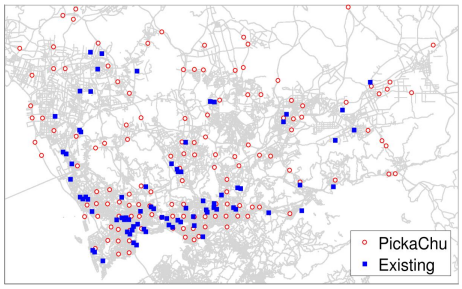

Fig. 16. Comparison of deployed chargers.

5.2 Experiment Settings

Parameter Settings: The parameters related to chargers, vehicles, and batteries are listed in Table 2. As BYD e6 is a widely used vehicle model among the taxicabs in Shenzhen [17], we use it to determine the parameters for taxicabs. After solving the optimization problem, *OptPickaChu* selects 93 regions out of 557 regions to deploy 350 wireless opportunistic chargers. *PickaChu* selects 125 regions to deploy 480 chargers, as shown in Figure 16. We observe that *OptPickaChu* results in fewer chargers than *PickaChu* since *OptPickaChu* additionally aims to minimize deployment cost. We can see that *PickaChu*'s deployment is generally consistent with the distribution of the existing 81 charging stations, which means it is extensible from the current charger deployment scheme. As for the calculation of revenue and cost, the unit electricity cost of driving was set to $0.5/mile, the unit service loss cost of charging (which is the possible loss of revenue earning opportunity) was set to $0.1/hour, and the unit revenue of traveling with passengers was set to $1.5/mile.

Simulation Settings: With the deployment schedule, we use SUMO [13] to simulate the operation of 1,000 taxicabs on Shenzhen's road network for 24 hours. In SUMO, taxicabs drive by following the traffic model we built in Section 4.3. The location and time of passenger requests follow the actual passengers' requests happened on July 15, 2015. We converted OpenStreetMap road network of Shenzhen to a SUMO road network file. We assume that each taxicab can only serve one passenger in a travel [29].

The metrics we measured are:

- *Ratio of an operation phase*: the average hourly ratio of the time duration of respective operation phase (i.e., cruising, travel, seeking chargers, charging) of all the taxicabs. For an operation phase, we first record the average hourly ratio of each taxicab during the day. Then, we calculate the average of the ratios of all the taxicabs. We also show the CDF of vehicles in terms of the time duration for each operation phase.
- *Revenue*: the daily average revenue earned by all the taxicabs through traveling with the passengers. It is calculated by multiplying all the taxicabs' daily traveling distance with the unit revenue of traveling with passengers. We also show the CDF of vehicles in terms of the daily revenue for traveling phase.
- *Cost*: the sum of the daily average cost of the electricity consumed by all the taxicabs through driving (i.e., cruising, seeking chargers, and traveling) and the daily average service loss cost caused by charging. The cost of cruising, seeking chargers and traveling is calculated by multiplying the driving distance of respective phase with the unit electricity cost of driving. The cost of charging is calculated by multiplying the charging time with the unit service loss cost. We also show the CDF of vehicles in terms of the daily cost for each idle operation phase.
- *Vehicle SoC*: we measure the SoC of each taxicab at each hour during a day, and calculate the median, 5^{th} percentile and 95^{th} percentile values to compare the performance of the methods in supporting the continuous operability of taxicabs.

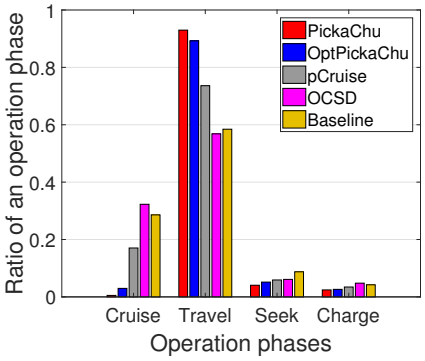

Fig. 17. Ratio of each operation phase.

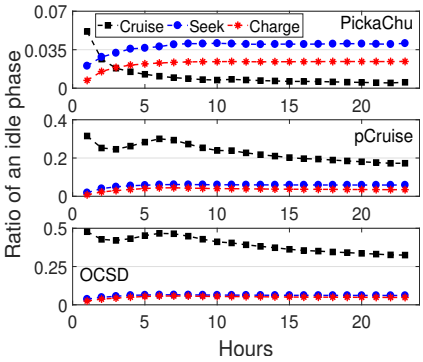

Fig. 18. Ratio of each idle phase ratio over time.

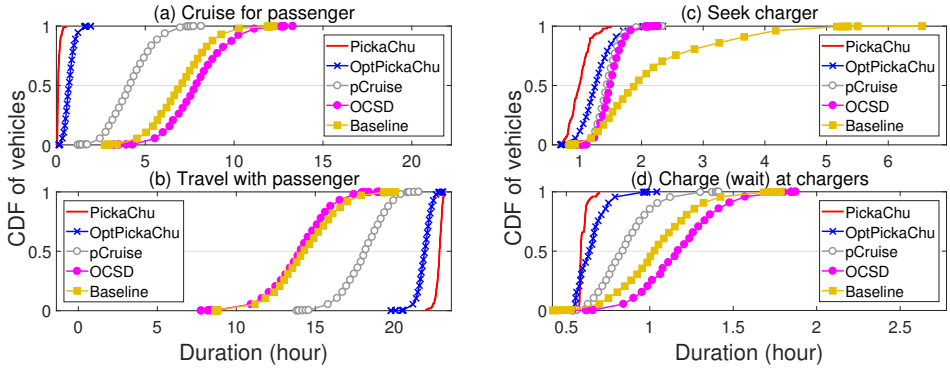

Fig. 19. CDF on the durations of different operation phases.

• *Overall energy supply overhead*: the energy supply overhead on all chargers in kWh. We measure it under different hours during a day to observe different methods' charging pressure on the power grid.
• *The number of served passengers*: the number of passengers served by the taxicabs. We measure it under different hours during a day to compare the performance of the methods in serving passengers.

5.3 Experimental Results

5.3.1 Ratio of Each Operation Phase. Figure 17 shows the average hourly ratio of each operation phase of all the taxicabs throughout a day. We see that for all the idle operation phases (i.e., cruising, seeking and charging), *PickaChu* has the lowest ratio. We also see that compared with *pCruise*, *OCSD* and *Baseline*, the cruising time in *PickaChu* is greatly reduced. Correspondingly, the time that *PickaChu*'s taxicabs spend on traveling with passengers on board (92%) is about 15% higher than that of *pCruise* (77%), 35% higher than that of *OCSD* (57%), and 33% higher than that of *Baseline* (59%). In *OCSD*, to discover passengers, the taxicabs must wander around in the road network, which increases cruising time. What's worse, with more time spent on cruising, the taxicabs have to charge more frequently to remain operable, which leads to higher ratios of seeking phase and charging phase than the other methods. As for *pCruise*, the taxicabs are always guided to the route with the highest probability of discovering passengers, which greatly reduces cruising time. However, the effective discovery of passengers still causes the taxicabs to waste much time on approaching the potential passengers. Compared with *Baseline*, only *OCSD* spends more time on cruising, which is caused by its inefficient discovery of passengers. But we also notice that *Baseline*'s time of seeking chargers ranks the highest, which means Shenzhen's current deployment of charging stations needs improvement in accessibility. In *PickaChu*, since the taxicabs

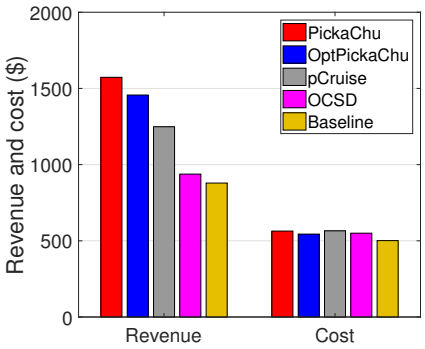

Fig. 20. Revenue and cost of each method.

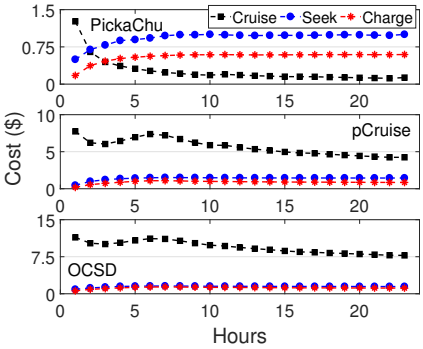

Fig. 21. Cost variation of each idle phase over time.

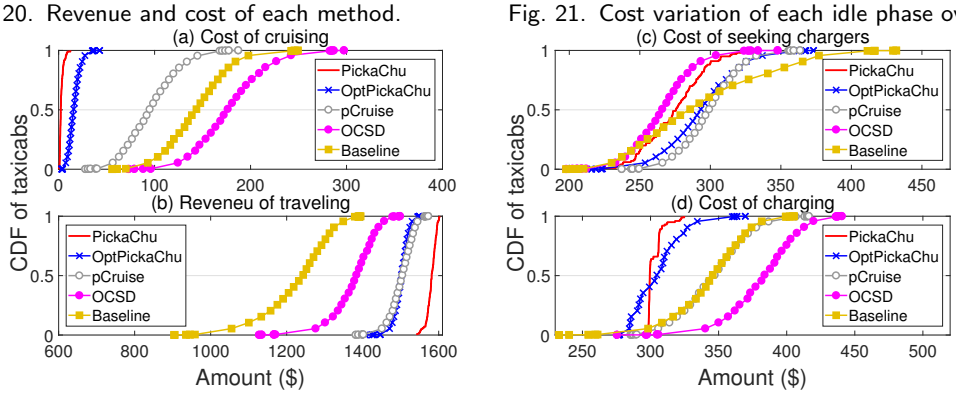

Fig. 22. CDF on the cost/revenue of different operation phases.

are allowed to stay at an opportunistic charger for some time, and the regions with chargers have high likelihood of passenger appearance, many taxicabs find passengers during their stay. Compared to *pCruise*, this strategy further reduces the time wasted on cruising for passengers and saves energy for the taxicabs. We notice that compared with other methods, *PickaChu* also reduces the charger seeking time. This is because that taxicabs cruise between chargers, and are less likely to exhaust their power.

Figure 18 shows the variation of the ratios of the idle phases by hour throughout a day. We can see that in *pCruise* and *OCSD*, the taxicabs have to spend a large portion of time on cruising during each hour. Also note that there are small bumps on the cruising time curves in *pCruise* and *OCSD*. This is because there are not enough passenger pick-up requests appearing between 05:00 and 07:00, so the ratios of cruising phase in *pCruise* and *OCSD* are increased during these hours. In contrast, in *PickaChu*, except for the first few hours, during which most of the taxicabs do not need to get charged, and keep cruising between the regions with opportunistic chargers, the time on cruising is largely replaced with the time of seeking chargers and charging in the following hours.

Figure 19 demonstrates the CDF of the taxicabs on the time durations of different phases. Figure 19 (a) shows that the taxicabs' cruising phase durations in *PickaChu* ($<$ 1 hour) are much shorter than those of the other methods. Figure 19 (b) shows that the taxicabs' traveling phase durations in *PickaChu* and *OptPickaChu* ($>$ 20 hours) are significantly longer than those of the other methods. This is caused by their difference in operation strategies. Figure 19 (c) shows that due to the same deployment of chargers in *pCruise* and *OCSD*, they have similar distributions of seeking phase durations (1 hour $\sim$ 2.5 hours). *Baseline* has much longer seeking phase durations (1 hour $\sim$ 6.5 hours), which means that the current charger deployment needs improvement. In Figure 19 (d), all the taxicabs in *PickaChu* have much shorter charging phase durations ($<$ 0.8 hours) than the others, which means it also reduces the need for recharge.

Except for the seeking phase, the distribution of other operation phase durations in *PickaChu* is much more concentrated than the others, which further proves the consistency of *PickaChu*'s effectiveness on all the taxicabs. We also see that *OptPickaChu* is slightly worse than *PickaChu* in reducing idle operation time, though it still outperforms other methods. This shows that *PickaChu* can achieve better performance on operation efficiency even with relatively lower deployment cost than the others.

In addition, we also measured the average revenue resulted from the traveling phase, and the average cost resulted from the other idle phases during the day, which are shown in Figure 20. We can see that compared with *pCruise*, *OCSD* and *Baseline*, *OptPickaChu* and *PickaChu* can increase the average revenue of all the taxicabs' by approximately more than \$250, \$500 and \$600 per day, respectively, with almost the same average cost. We also measured the changes of the costs of different methods under various hours, which are shown in Figure 21. The reason is the same as that of Figure 18. We also measured the distribution of the revenues, and the distribution of the costs of the taxicabs, which are shown in Figure 22. We can see that most of the taxicabs in *PickaChu* and *OptPickaChu* spend less than \$50 on cruising and less than \$300 on seeking chargers, which are less than those of the other methods. The taxicabs' costs spent on seeking chargers in *PickaChu* and *OptPickaChu* are comparable to those of the other methods. However, the revenues of the taxicabs in *PickaChu* and *OptPickaChu* ($>$ \$1,400) are conspicuously higher than those of the other methods.

5.3.2 SoC Maintenance of Taxicabs. We measured the SoCs of all the taxicabs at each hour throughout a day. As we cannot show all the SoCs in a figure, we plot the median, 5^{th} and 95^{th} percentiles of SoCs of all the taxicabs at a few time points in Figure 23. We see that *OptPickaChu* and *PickaChu* maintain almost the same SoC levels as the other methods. However, as observed in Section 5.3.1, the taxicabs in the other methods spend more time on idle phases, which results in a lower energy efficiency and lower profits. Note that *OptPickaChu* provides a comparable SoC as the other methods during most of the time, although it has fewer deployed chargers. This demonstrates its effectiveness on minimizing the deployment cost while still guaranteeing the SoC of taxicabs.

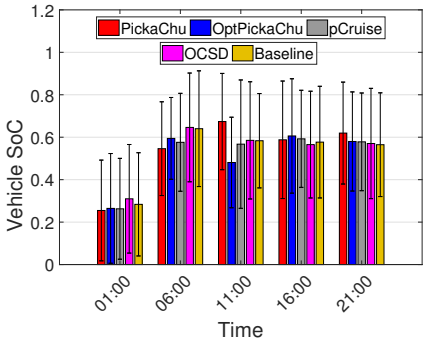

Fig. 23. Vehicle SoC.

5.3.3 Overall Energy Supply Overhead. Figure 24 shows the overall energy supply overhead of different methods under different hours throughout a day. The results follow: *OCSD*>*Baseline*>*pCruise*>*PickaChu* ≈*OptPickaChu*. We can see that *PickaChu* and *OptPickaChu* result in the least pressure on the power grid given the same number of taxicabs. Rather than cruising for passengers as *pCruise*, *Baseline* and *OCSD*, the taxicabs in *PickaChu* and *OptPickaChu* can wait at the chargers for their next passengers. Moreover, since the taxicabs in *OCSD* and *Baseline* cannot effectively harvest passengers from chargers, they drive more idle trips and require more charging.

It is worth mentioning that in the first few hours, the energy supply overhead increases significantly. For *pCruise*, *Baseline* and *OCSD*, the peak that appears between 05:00 and 07:00 is caused by the lack of passengers. Taxicabs start with 100% SoC. Since there are few passengers during 00:00-07:00, the taxicabs in *pCruise*, *Baseline* and *OCSD* keep cruising for passengers and their SoC keeps decreasing. Finally, all taxicabs exhaust their SoC and recharge at about the same time, resulting in a peak in charging overhead. After then, their SoC exhausts at different times caused by different passengers. Therefore, they charge at different times, resulting in no peaks in charging overhead. The energy supply overhead in *PickaChu* and *OptPickaChu* stabilize more quickly, which reflects their resilience against the variation of passengers.

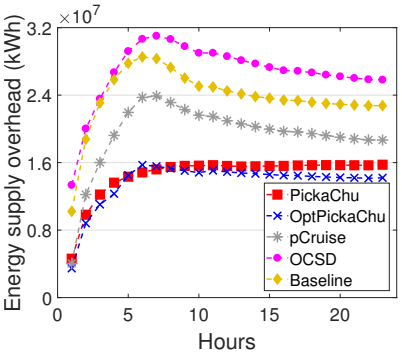

Fig. 24. Energy supply overhead.

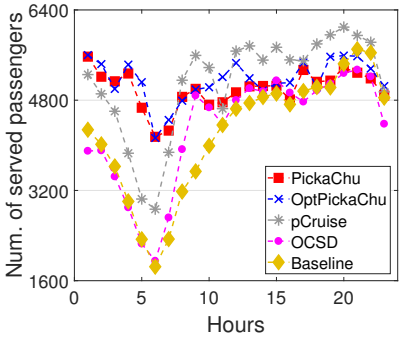

Fig. 25. Num. of served passengers.

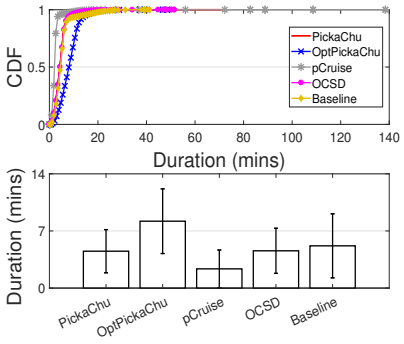

Fig. 26. Waiting time of passengers.

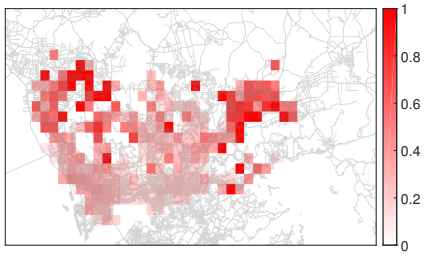

Fig. 27. Heat map of service rates.

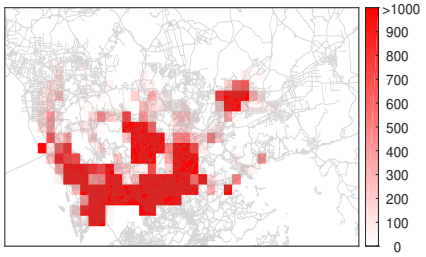

Fig. 28. Heat map of passengers.

5.3.4 Service Performance. Figure 25 shows the numbers of served passengers of different methods during different hours throughout a day. We see that during the hours with relatively fewer requests (01:00-08:00), the results follow: *PickaChu*>*OptPickaChu*>*pCruise*>*OCSD*≈*Baseline*. After then, *pCruise* can serve slightly more passengers (< 1,000) than *PickaChu*, *OCSD* and *Baseline*. Figure 26 shows the distribution of the waiting time of the passengers (upper part), and the average, 5^{th} and 95^{th} percentiles of the waiting time of the passengers (lower part) in different methods. We can see that the passengers' average waiting time in *OptPickaChu* (8 minutes) is longer than the other methods. Since there are fewer chargers in *OptPickaChu*, so the chargers are more sparsely distributed in the road network. Since the vacant taxicabs cruise between the chargers, the waiting time of the passengers in the regions without chargers is usually very long, which results in the longer average waiting time of passengers in *OptPickaChu*. We also find that the results of *PickaChu* (5 minutes), *OCSD* (5 minutes), and *Baseline* (6 minutes) are comparable to each other, and the result of *pCruise* is the shortest (2 minutes). In *PickaChu*, most of the passengers are picked up in the regions with chargers. Since the distance from the taxicabs to the passengers is bound by the region size (i.e., 2,000 meters), the passengers' waiting time is not very long. In *OCSD* and *Baseline*, the taxicabs randomly cruise in the road network, which means most of the passengers are picked up during the cruising of the taxicabs. Thus, the passengers' waiting time is comparable to that of *PickaChu*. In *pCruise*, the taxicabs are always cruising on the routes with the maximum probability of finding a passenger, so the passengers have the shortest waiting time.

When there are few pick-up requests, *PickaChu* serves more passengers than *pCruise*. This is because in *pCruise*, through vehicle-to-vehicle communication, a taxicab may not discover sufficient passengers to generate an effective cruising graph for guidance. On the contrary, the taxicabs in *PickaChu* wait at the regions with high likelihood of passenger appearance, which helps the taxicabs efficiently discover passengers. When there are many pick-up requests, the taxicabs in *pCruise* can easily discover requests. Hence, *pCruise* can serve more passengers than *PickaChu* during this time, but at the cost of more energy consumption, as mentioned in Section 5.3.3. We see that *PickaChu* always outperforms *OCSD* and

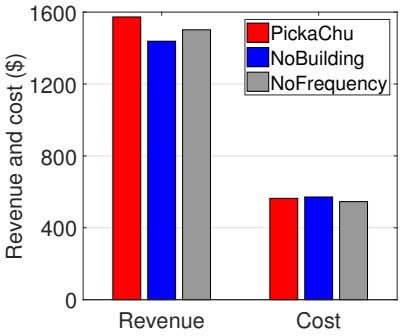

Fig. 29. Effectiveness of components.

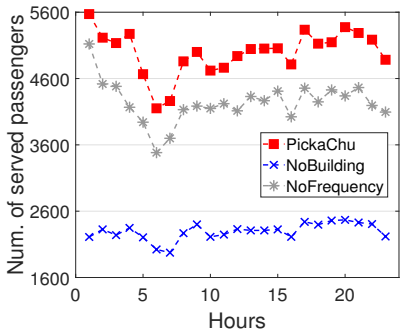

Fig. 30. Served passengers.

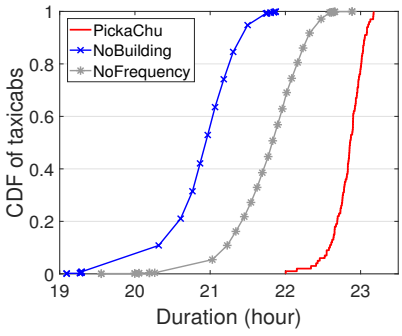

Fig. 31. Travel phase durations.

Baseline, which spend more time on cruising, seeking and charging. In addition, *OptPickaChu* provides service performance comparable to *PickaChu*, although *PickaChu* deploys more chargers. This is because the redundant chargers do not significantly benefit the discovery of passenger requests. This shows the effectiveness of *OptPickaChu* on minimizing the deployment cost while achieving our objectives.

Figure 27 shows the service rate (i.e., ratio between the number of served passengers and the total number of passenger requests) of each region. Figure 28 shows the distribution of daily average passenger requests in the regions. We can see that even for the distant regions with rare appearance of passengers (e.g., northwestern regions), the service rates were kept at high levels. Namely, the distribution of service rates is balanced among the regions. Note the service rates in the southern regions are relatively low. This is because in the simulation, the 1,000 taxicabs, which is limited by the simulator, cannot serve all the passenger requests.

5.3.5 Effectiveness of Components. As discussed in Section 4.2.3, the additional consideration of building functionalities ($\bar{H}_i$) in calculating the region scores in Equation (3) can help more accurately reflect the likelihood of passenger appearance in different regions, and then better guide the deployment of wireless chargers. Additionally considering the frequency of passenger appearance in Equation (3) serves the same purpose. To demonstrate the effectiveness of these two components, we recalculated the score of each region ($\rho(\boldsymbol{g_i})$) without multiplying the weighted sum of the building functionalities (denoted as *NoBuilding*), and without multiplying the weighted sum of the passenger appearance frequencies (denoted as *NoFrequency*). Based on the new region scores, we redetermined the deployment of chargers, and measured the average costs and revenues of the taxicabs during the day as shown in Figure 29. In addition, we also measured the number of passengers served by the taxicabs in different methods under different hours as shown in Figure 30, and the distribution of the travel phase durations of the taxicabs as shown in Figure 31 and the distribution of the revenues of the taxicabs as shown in Figure 32.

We can see that compared with *NoBuilding* and *NoFrequency*, *PickaChu* increases the average revenue by \$150 and \$75 per taxicab, respectively, while the costs are almost equal. Also, *PickaChu* can serve at most 1,000 more passengers than *NoFrequency*, and at most 2,500 more passengers than *NoBuilding*. This is because with considering these two components, the region scores can more accurately reflect the likelihood of passenger appearance and the resultant charger deployment in *PickaChu* can provide higher opportunity of picking up passengers for the taxicabs.

5.3.6 Impact of the Number of Chargers. Our optimization problem outputs the selected regions for deploying chargers, and the number of chargers at each selected region. To illustrate the impact of the total number of chargers on the taxicabs' operation efficiency, from the optimally selected regions, we randomly picked 10 to 90 regions to deploy the chargers, while the number of chargers per region remains

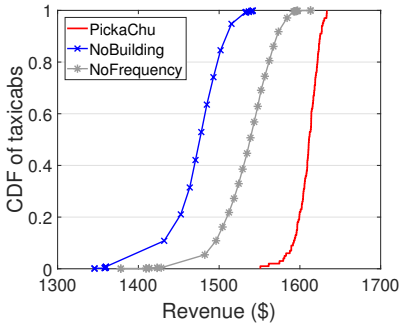

Fig. 32. Distribution of taxicabs' revenues.

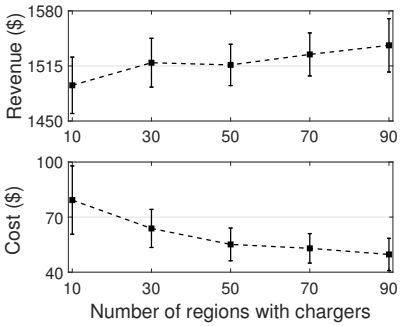

Fig. 33. Impact of the number of regions with chargers.

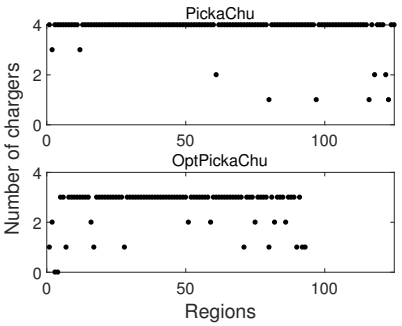

Fig. 34. Distribution of the number of chargers in regions.

the same as that in the optimization output. We then measured the average, 5^{th} and 95^{th} percentiles of the revenues and the costs of the taxicabs under various total numbers of chargers, which is shown in Figure 33. We can see that along with the increasing of the total number of chargers, the average revenue of the taxicabs keeps increasing, and the average cost of the taxicabs keeps reducing. This is because the more chargers deployed, the less idle miles the taxicabs need to drive in seeking the chargers, which reduces the taxicabs' cost. Meanwhile, the taxicabs' opportunity of picking up passengers also increases with the increased number of chargers, which increases the taxicabs' revenue.

We also measured the distribution of the numbers of the chargers in the regions, which is shown in Figure 34. We can see that there are fewer regions deployed in *OptPickaChu* than in *PickaChu* (93 vs. 125), and the majority of the regions in *OptPickaChu* have 3 chargers, and the majority of the regions in *PickaChu* have 4 chargers. This is because *OptPickaChu* has smaller total deployment cost, it must deploy fewer chargers per region so that the chargers can be deployed to a sufficient number of regions to support the SoC of the taxicabs. While *PickaChu* has a higher budget, so it can select more regions to deploy wireless chargers and deploy more chargers per region.

6 CONCLUSION

The idle time of electric taxicabs is wasteful against making profits and energy consumption. Wireless charging techniques enable EVs to be charged at their parked positions. Our proposed *PickaChu* is the first work that aims at both maximally reducing the taxicabs' idle time and supporting the continuous operability of the taxicabs through proper deployment of wireless opportunistic chargers. Our analytical results on a metropolitan-scale taxicab dataset lay the foundation of the design of *PickaChu*. We assign scores to regions to represent the likelihood of passenger appearance in the regions, and model taxicab mobility to calculate the expected SoC of the taxicabs in each region. We design a multi-objective optimization problem to minimize the total deployment cost of chargers, maximize the passenger pick-up opportunity at the chargers, and ensure the continuous operability of the taxicabs. We conducted trace-driven experiments on SUMO to verify the performance of *PickaChu*. Compared with the previous methods, *PickaChu* reduces the taxicabs' daily average idle time by 81% and increases the taxicabs' daily revenue by more than 50% under the same charger deployment cost. When minimizing the charger deployment cost, *PickaChu* reduces the number of chargers by 27%, but still reduces the taxicabs' daily average idle time by 61% and increases the taxicabs' daily revenue by more than 40%.

The components of *PickaChu* can also be used for the planning of many existing charging facilities, such as fast charging stations, and battery swapping stations. In future work, we will explore some other region partitioning methods to improve the charger deployment (e.g., considering building size, city layout

plan, and possible waiting queue length). We will also consider the pattern of passenger appearance to guide the taxicab pick-ups proactively before receiving passenger requests.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants OAC-1724845, ACI-1719397 and CNS-1733596, and Microsoft Research Faculty Fellowship 8300751. IBM Ph.D. fellowship award 2017. We would like to thank Dr. Hongning Wang for his valuable discussions and comments.

REFERENCES

[1] 2017. Apache Hadoop 2.7.3. http://hadoop.apache.org/. (2017). Accessed in August, 2017.
[2] 2017. Apache Spark 1.5.2. http://spark.apache.org/. (2017). Accessed in August, 2017.
[3] 2017. MultiJuMP multiobjective optimisation package. https://github.com/anriseth/MultiJuMP.jl. (2017). Accessed in August, 2017.
[4] 2017. National household travel survey. http://nhts.ornl.gov/briefs/EVFeasibility20160701.pdf. (2017). Accessed in August, 2017.
[5] 2017. OpenStreetMap. http://www.openstreetmap.org/. (2017). Accessed in August, 2017.
[6] Maria Joao Alves and João Clímaco. 2007. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research* 180, 1 (2007).
[7] Federica Bogo and Enoch Peserico. 2013. Optimal Throughput and Delay in Delay-Tolerant Networks With Ballistic Mobility. In *Proc. of Mobicom*.
[8] Zipei Fan, Xuan Song, and Ryosuke Shibasaki. 2014. CitySpectrum: A Non-Negative Tensor Factorization Approach. In *Proc. UbiComp*.
[9] Desislava Hristova, Matthew J Williams, Mirco Musolesi, Pietro Panzarasa, and Cecilia Mascolo. 2016. Measuring Urban Social Diversity Using Interconnected Geo-Social Networks. In *Proc. of WWW*.
[10] Y. J. Jang, E. S. Suh, and J. W. Kim. 2015. System Architecture and Mathematical Models of Electric Transit Bus System Utilizing Wireless Power Transfer Technology. *IEEE Systems Journal* PP, 99 (2015).
[11] Lei Kang, Bozhao Qi, Dan Janecek, and Suman Banerjee. 2015. EcoDrive: A Mobile Sensing and Control System for Fuel Efficient Driving. In *Proc. of Mobicom*.
[12] Aristeidis Karalis, John D Joannopoulos, and Marin Soljačić. 2008. Efficient wireless non-radiative mid-range energy transfer. *Annals of Physics* 323, 1 (2008).
[13] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. 2012. Recent Development and Applications of SUMO - Simulation of Urban MObility. *Int. Journal On Advances in Systems and Measurements* 5, 3&4 (2012).
[14] Tamás Kurczveil, Pablo Álvarez López, and Eckehard Schnieder. 2013. Implementation of an Energy Model and a Charging Infrastructure in SUMO. In *Proc. of SUMO User Conference*.
[15] Ruimin Li, Huajun Chai, and Jin Tang. 2013. Empirical Study of Travel Time Estimation and Reliability. *Mathematical Problems in Engineering* 2013 (2013).
[16] Ruimin Li, Geoffrey Rose, and Majid Sarvi. 2006. Using automatic vehicle identification data to gain insight into travel time variability and its causes. *JTRB* 1945 (2006).
[17] Yanhua Li, Jun Luo, Chi-Yin Chow, Kam-Lam Chan, Ye Ding, and Fan Zhang. 2015. Growing the charging station network for electric vehicles with trajectory data analytics. In *Proc. of ICDE*.
[18] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining periodic behaviors for moving objects. In *Proc. of SIGKDD*.
[19] Miles Lubin and Iain Dunning. 2015. Computing in Operations Research Using Julia. *INFORMS Journal on Computing* 27, 2 (2015).
[20] Fei Miao, Shuo Han, Abdeltawab M Hendawi, Mohamed E Khalefa, John A Stankovic, and George J Pappas. 2017. Data-driven distributionally robust vehicle balancing using dynamic region partitions. In *Proc. of ICCPS*.
[21] Hua Qin and Wensheng Zhang. 2011. Charging Scheduling With Minimal Waiting in a Network of Electric Vehicles and Charging Stations. In *Proc. of VANET*.
[22] Ankur Sarker, Chenxi Qiu, Haiying Shen, Andrea Gil, Joachim Taiber, Mashrur Chowdhury, Jim Martin, Mac Devine, and AJ Rindos. 2016. An Efficient Wireless Power Transfer System To Balance the State of Charge of Electric Vehicles. In *Proc. of ICPP*.
[23] Michail Vlachos, S Yu Philip, and Vittorio Castelli. 2005. On Periodicity Detection and Structural Periodic Similarity.. In *Proc. of SIAM*.

[24] Matt P Wand and M Chris Jones. 1994. *Kernel smoothing*. CRC Press.
[25] Mengwen Xu, Dong Wang, and Jian Li. 2016. DESTPRE: A Data-Driven Approach to Destination Prediction for Taxi Rides. In *Proc. UbiComp*.
[26] Li Yan, Haiying Shen, Juanjuan Zhao, Chengzhong Xu, Feng Luo, and Chenxi Qiu. 2017. CatCharger: Deploying Wireless Charging Lanes in a Metropolitan Road Network through Categorization and Clustering of Vehicle Traffic. In *Proc. of INFOCOM*.
[27] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proc. of SIGKDD*.
[28] Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie, and Guangzhong Sun. 2011. Where to Find My Next Passenger?. In *Proc. of UbiComp*.
[29] Desheng Zhang, Tian He, Shan Lin, Sirajum Munir, and John A Stankovic. 2015. Online Cruising Mile Reduction in Large-Scale Taxicab Networks. *IEEE TPDS* 26, 11 (2015).
[30] Desheng Zhang, Jun Huang, Ye Li, Fan Zhang, Chengzhong Xu, and Tian He. 2014. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proc. of Mobicom*.
[31] Daqing Zhang, Lin Sun, Bin Li, Chao Chen, Gang Pan, Shijian Li, and Zhaohui Wu. 2015. Understanding Taxi Service Strategies From Taxi GPS Traces. *IEEE TITS* 16, 1 (2015).
[32] Tian Zhang, Wei Chen, Zhu Han, and Zhigang Cao. 2014. Charging Scheduling of Electric Vehicles With Local Renewable Energy Under Uncertain Electric Vehicle Arrival and Grid Power Price. *IEEE TVT* 63, 6 (2014).
[33] Xudong Zheng, Xiao Liang, and Ke Xu. 2012. Where to Wait for a Taxi?. In *Proc. of SIGKDD*.
[34] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM TIST* 6, 3 (2015).
[35] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York city's noises with ubiquitous data. In *Proc. of UbiComp*.
[36] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. 2011. Urban computing with taxicabs. In *Proc. of UbiComp*.
[37] Stephen Zoepf, Don MacKenzie, David Keith, and William Chernicoff. 2013. Charging Choices and Fuel Displacement in a Large-Scale Demonstration of Plug-In Hybrid Electric Vehicles. *Transportation Research Record: Journal of the Transportation Research Board* 2385 (2013).

Received August 2017; revised November 2017; accepted Jan 2018