Growing the Charging Station Network for Electric Vehicles with Trajectory Data Analytics

Yanhua Li [#], Jun Luo [†#], Chi-Yin Chow [*], Kam-Lam Chan [*], Ye Ding [§], and Fan Zhang [†]

[#] *HUAWEI Noah's Ark Lab, Hong Kong,* [*] *City University of Hong Kong, Hong Kong*
[†] *Shenzhen Institutes of Advanced Technology, CAS, China,* [§] *HKUST, Hong Kong*
li.yanhua1@huawei.com, jun.luo@siat.ac.cn, chiychow@cityu.edu.hk
klchan85-c@my.cityu.edu.hk, valency@ust.hk, zhangfan@siat.ac.cn

***Abstract*—Electric vehicles (EVs) have undergone an explosive increase over recent years, due to the unparalleled advantages over gasoline cars in green transportation and cost efficiency. Such a drastic increase drives a growing need for widely deployed publicly accessible charging stations. Thus, how to strategically deploy the charging stations and charging points becomes an emerging and challenging question to urban planners and electric utility companies. In this paper, by analyzing a large scale electric taxi trajectory data, we make the first attempt to investigate this problem. We develop an optimal charging station deployment (OCSD) framework that takes the historical EV taxi trajectory data, road map data, and existing charging station information as input, and performs optimal charging station placement (OCSP) and optimal charging point assignment (OCPA). The OCSP and OCPA optimization components are designed to minimize the average time to the nearest charging station, and the average waiting time for an available charging point, respectively. To evaluate the performance of our OCSD framework, we conduct experiments on one-month real EV taxi trajectory data. The evaluation results demonstrate that our OCSD framework can achieve a 26%–94% reduction rate on average time to find a charging station, and up to two orders of magnitude reduction on waiting time before charging, over baseline methods. Moreover, our results reveal interesting insights in answering the question: "Super or small stations?": When the number of deployable charging points is sufficiently large, more small stations are preferred; and when there are relatively few charging points to deploy, super stations is a wiser choice.**

I. Introduction

The fast development of sustainable energy technology enabled a drastic increase of **E**lectric **V**ehicles (EVs) in recent years. Electric vehicles (including plug-in electric cars, hybrid electric cars, etc.) have unparalleled advantages over gasoline cars in green transportation and cost efficiency. With zero total emissions, we can achieve significant environmental savings by transferring to EVs. A research [6] has shown that if we switched from gasoline cars to EVs, we would see a 42 percent average reduction in carbon dioxide (CO2) emissions, which is a primary culprit in the global warming. On the other hand, the fuel (electricity) costs for EVs are significantly lower than for similar gasoline cars. As a result, EV sales have undergone an explosive increase over recent years. As highlighted in [2], in US, hybrid vehicle sales in 2013 grew by 14% over 2012, and sales of electric cars grew by 83%!

A new challenge raised by such fast development of EV market is the explosive increase of new charging station deployment. It is important to understand where to place new charging stations with how many charging points, so as to catch up the needs, while maintaining a reasonable utilization rate of the deployed charging resources. For instance, since 2010, the number of electric taxis in Shenzhen in China keeps increasing, and there were in total 780 plug-in electric taxis by November 2013. However, only 25 public charging stations were built, with a large variance in sizes, say, some with more than one hundred charging points, and some with only two or three. From the EV taxi trajectory data we collected (in Nov 2013), a taxi on average spends four minutes to find a charging station, and waits in queue for 15 minutes before charging.

In operations research, the station sitting problem has been studied for deploying gas stations and hydrogen filling stations [14], [21], [32], [29]. The problem is primarily formulated as facility location on a network of roads [14], [21] or as a subset of the existing gasoline station network [32]. However, these facility location models cannot be applied for charging station sitting, because of the following two reasons. (1) Differing from the gasoline cars, the charging durations of EVs are very long, i.e., around a few hours, which yields long waiting time for incoming EVs (when charging points are all occupied). Existing models do not capture such (long) charging time and waiting time. (2) The existing facility location models all require trip origin-destination data as an input, which is in general hard to obtain. To our best knowledge, there is no systematic work so far to study how to strategically deploy *charging stations* for EVs.

Hence, in this work, we are motivated to investigate this problem[1]: Given a city with L existing charging stations and their locations and numbers of charging points, if a total of K new charging stations and M new points are available to deploy, where to deploy those new stations, and how to assign the number of charging points to each charging station, so as to minimize the average time needed for an EV driver to *find* and *wait* for an EV charging point. We develop an **O**ptimal **C**harging **S**tation **D**eployment (OCSD) framework, which takes a historical EV taxi trajectory data, road map data, and existing charging station information as input, and performs **O**ptimal **C**harging **S**tation **P**lacement (OCSP) and

[1]Note that we are interested in the deployment of public charging stations, that are available to every EV. The deployment of private charging stations should follow a different deployment manner.

Optimal **C**harging **P**oint **A**ssignment (OCPA). These OCSP and OCPA optimization components are designed to minimize the average time to travel to charging station, and the average waiting time for an available charging point, respectively. Our main contributions are summarized as follows.

- In our OCSD framework, we design a behavior extraction method, that can extract sub-trajectories for EV taxis from their trajectory data. The sub-trajectories represent three typical behaviors of EV taxis, namely, *seeking for charging*, *on charging*, and *traveling*.
- We formulate the charging station placement problem using integer programming, which is NP-hard. We provide a polynomial time approximation algorithm to solve the problem with provable error bound.
- Given the selected locations for charging stations, we formulate the charging point assignment problem as an average utilization minimization problem, and obtain a closed form optimal solution.
- To evaluate the performance of our OCSD framework, we conduct experiments on one-month real EV taxi trajectory data. The evaluation results demonstrate that our OCSD framework can achieve a 26%–94% reduction rate on average time to find a charging station, and six times reduction on waiting time before charging, over baseline methods. Moreover, our results reveal interesting insights in answering the question: "Super or small stations?": When the number of deployable charging points is sufficiently large, more small stations are preferred; and when there are relatively few charging points to deploy, super stations is a wiser choice.

The rest of the paper is organized as follows. Section II formally defines the problem, presents the overview and outlines the key components of our framework. Section III provides detailed methodology of OCSD. Section IV presents evaluation results on a large-scale EV taxi trajectory data. Section V discusses two extensions of our OCSD framework. Related works are discussed in Section VI, and the paper is concluded in Section VII.

II. Overview

In this section, we define the charging station planning problem, describe the dataset we have, and outline the framework of our methodology.

A. Problem Definition

An electric vehicle (EV) uses one or more electric motors for propulsion, and can be directly powered from an external charging station. Nowadays, a full charging of an EV takes around a few hours. Once fully charged, the mileage that the EV can drive varies for different EV models. In our dataset, since all EVs are taxis with the same model, they usually can drive 200 kilometers with a full charge, and a complete charging takes around 1.5 to 2 hours. Due to the fast development of the sustainable energy technology, the number of EVs grows drastically these years, which drives the demands for building more charging stations. In Shenzhen, China, there were in total 780 EV taxis (by the statistic in November 2013). However, only 25 charging stations are available in Shenzhen, with a huge variance on numbers of charging points among the stations, e.g., some stations have more than one hundred charging points, and some other have only two or three charging points. In this paper, by utilizing the historical GPS trajectory data of EV taxis in Shenzhen, we make the first attempt to investigate how to strategically deploy charging stations and points across the city to facilitate the public EV charging.

Over time, a GPS-equipped electric vehicle reports its GPS locations, that form a trajectory as defined below.

Definition 1 (Trajectory). *A trajectory is a sequence of spatial points that a moving object follows through space as a function of time. Each point thus consists of a trajectory ID, latitude, longitude, and a time stamp.*

A trajectory of an EV taxi can be divided in three types of sub-trajectories, representing a sequence of time-stamped GPS locations, where the EV taxi is seeking for charging stations, on charging, or traveling. We define each of them as follows.

Definition 2 (Seeking sub-trajectory). *A seeking sub-trajectory represents that an EV taxi is going to a charging station to charge, which includes the trace that the EV* seeks *the charging station, and the trace (if any) that the EV* waits *in a queue for an available charging point.*

Definition 3 (Charging sub-trajectory). *A charging sub-trajectory represents that an EV taxi is charging at a charging point.*

Definition 4 (Traveling sub-trajectory). *A traveling sub-trajectory represents that an EV taxi is traveling after a charging and before a seeking sub-trajectory, where the EV may drive with or without a passenger, or park at a restaurant for lunch, etc.*

These sub-trajectories basically label an EV trajectory into three possible behaviors[2]. We will elaborate how we label the sub-trajectories on our EV taxi GPS data in Section III. Note that the three sub-trajectories can be viewed as three states, where an EV traverses among them over time. For example, Figure 1 illustrates two taxi trajectories, where they are divided and labeled into the above three sub-trajectories. It indicates that each EV trajectory follows a loop of traveling, seeking, and charging. Moreover, a charging sub-trajectory is always a sequence of the same location at a charging station.

We are interested in the average time duration elapsed during the seeking sub-trajectories, which indicates the average time an EV taxi driver spends for *seeking* and *waiting at* a charging station. We define the starting GPS record of a seeking sub-trajectory as **a seeking event** of an EV taxi, which indicates that the EV taxi starts seeking for a charging station. We thus define the idle time of a seeking event as follows.

[2]Note that these sub-trajectories are exclusive, namely, each EV GPS record belongs to one and only one sub-trajectory.

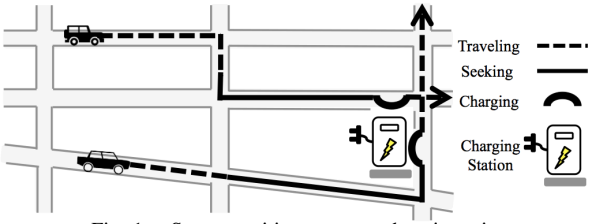

Fig. 1. State transition among sub-trajectories

Definition 5 (Idle time). *The idle time of a seeking event is the time elapsed during the seeking sub-trajectory, which includes the* seeking time *for the EV to reach the charging station, and the* waiting time *(if any) for the next available charging point.*

Problem Definition. Given a set of locations of L existing charging stations with their numbers of charging points, a set of seeking events, and a budget of K new charging stations with a total of M new charging points, we aim to find the optimal locations to open the new charging stations and optimal assignment of new charging points to the stations, so as to minimize the average idle time of each EV taxi to find and wait at a charging station.

B. Data Description

Given the problem defined above, we describe the datasets we use in this paper, including (1) the EV taxi trajectory data, (2) road map data, (3) existing charging station data. Note for consistency, we choose these datasets aligned within the same time domain. Below, we describe each of them, respectively.

EV Taxi Trajectory Data. We have an EV taxi GPS dataset from Shenzhen City during November 1st–30th, 2013. There were in total 780 registered EV taxis, where 490 of them were equipped with GPS sets. Hence, we have the GPS trajectories for those 490 EV taxis. For each EV taxi, the average recording frequency for GPS location is about 40 seconds. The data contains $23,967,501$ GPS records of EV taxis. Each records contains five useful fields for our study, including the taxi ID, time stamp, latitude, longitude, passenger load indicator.

Road Map Data. In our study, we use the Google Geocoding API [3] to retrieve a bounding box of Shenzhen City, specified by the south-west and north-east corners as $(22.447203, 113.769263)$ and $(22.70385, 114.33991)$ in latitude and longitude, which covers an area of roughly $1,804$ km^2 in Shenzhen City.

With the bounding box, Shenzhen road map data were obtained from OpenStreetMap [4], which contains all road segments and their road types. There are in total six levels of roads in Shenzhen, specified in OpenStreetMap data. Table I lists all the road types and their numbers of roads. In Figure 2, the top five levels of roads are drawn, with the width and darkness indicating their levels.

Charging Station Data. By November 2013, there were in total 25 charging stations within the bounding box of Shenzhen city. Figure 2 indicates the spatial distribution of those charging stations, with marker size indicating the number of charging points deployed in the stations. Charging stations

TABLE I
SHENZHEN ROAD MAP DATA

Level	Type	Counts	Level	Type	Counts
1	Motorway	563	4	Secondary	868
2	Trunk	258	5	Tertiary	1,393
3	Primary	745	6	Unclassified	16,829
				Total number of all roads:	20,656

with more than 50 charging points are super stations marked with large circle symbols, where those with less than 50 charging points are marked as small stations. We observe that there are three super stations.

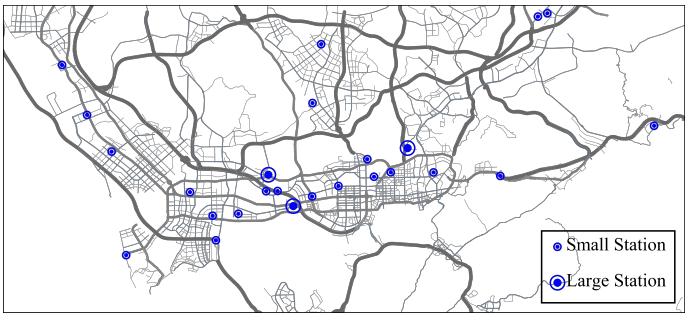

Fig. 2. The distribution of charging stations in Shenzhen

C. Solution Framework

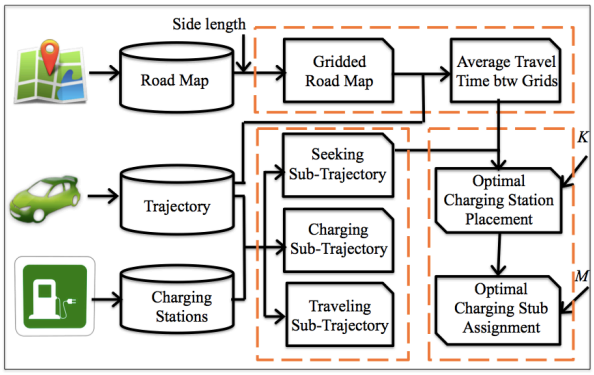

Fig. 3. Framework of our solution

Figure 3 presents our optimal charging station deployment (OCSD) framework. It takes three datasets as inputs, including EV trajectory data, city road map data, and existing charging station and charging point distribution. The whole framework consists of three stages (highlighted as three dashed boxes): (1) road map griding, (2) extracting sub-trajectories, (3) optimal charging station placement and charging point assignment.

- **Stage 1 (Road map griding):** First, the road map is divided into n equal size grids. Then, to estimate the average travel time between neighboring grids, we take all taxi traveling events that traverse between the two neighboring grids in the trajectory data, and calculate the average as the estimate of travel time between them. Thus, an n by n travel time matrix T of adjacent grids is formed. Using the adjacent travel time matrix, we compute the shortest path travel time across all grid pairs.
- **Stage 2 (Extracting sub-trajectories):** This stage extracts charging, seeking, and traveling sub-trajectories

from the raw EV taxis trajectories, by combining the EV taxi trajectory data with the charging station information.

- **Stage 3 (Optimal Charging Station Deployment, OCSD):** Given a constraint of K new charging stations and M charging points to deploy, we propose a two-step optimization framework to tackle the optimal charging station placement (OCSP) problem and the optimal charging point assignment (OCPA) problem. The OCSP problem is formulated as an integer linear programming problem, with an objective of minimizing the average time for EV drivers to find a charging station. A polynomial time approximation algorithm is provided to find the solution. The OCPA problem is formulated to minimize the average utilizations of charging points, where a close-form optimal solution is derived.

Table II provides notations used throughout the paper.

TABLE II
KEY NOTATIONS AND TERMINOLOGIES

Notations	Description
$G_0 = \{g_i\}$, $1 \leq i \leq n_0$	G_0 is the grid set of the gridded road map. there are in total n_0 grids.
$T = [T_{ij}]$	n_0 by n_0 adjacent average travel time matrix.
$G \subseteq G_0$	Subset of G_0, containing n grids in giant strongly connected component of G_0.
$C = [C_{ij}]$	n by n shortest path travel time matrix.
W_i, W	W_i: the number of seeking events in grid g_i; $W = \sum_{g_i \in G} W_i$
K, L, M	K: the number of new charging stations; L: the number of existing charging stations; M: the number of new charging points.
$y = [y_j]$, $1 \leq j \leq n$	Station placement vector, indicating a grid g_j has a charging station, if $y_j = 1$, and no station, otherwise.
$X = [X_{ij}]$ $1 \leq i, j \leq n$	Grid assignment matrix, indicating grid g_i is assigned to grid g_j for charging, if $X_{ij} = 1$, and Otherwise, $X_{ij} = 0$.
λ_ℓ	Arrival rate of seeking events in the regions that station ℓ covers.
μ_ℓ	Service rate of a charging point in station ℓ, namely, the average number of electric vehicles being served per hour.
ρ_ℓ	The charging point utilization in station ℓ.

III. METHODOLOGY

In this section, we elaborate each stage of our framework outlined in Figure 3.

A. Road Map Griding

Since the deployment of a charging station depends on many factors, such as the availability of the location, the topology of the underlying power grid network, and etc, it is not necessary to find the exact locations to deploy new charging stations. Therefore, we aim to locate the most suitable regions. At the initial stage, our approach divides the road map into equal size grids with a given side length s in latitude and longitude. Then, the station placement problem turns into finding the right grids to deploy charging stations, by transferring the continuous spatial space into discrete grid ID space. This griding method is flexible in adjusting the side-length of the grid, which enables us to identify good candidate regions in different granularities during the charging station placement stage. Moreover, we take the grid-based method for the ease of implementation in practice (which is adopted by many works [44], [25]), instead of other partitioning methods, such as voronoi cell [45] or road network based method [12].

With gridded road map, we estimate the average travel time between grids using the collected taxi trajectory data, and compute the shortest grid paths among grid pairs. The grid level shortest path distance is taken as an important input for the optimization stage to decide which grid to deploy a new charging station. Below, we elaborate how we construct the shortest path travel time matrix.

Adjacent average travel time matrix T. Let $G_0 = \{g_j\}$ denote the set of grids on the road map, with respect to a side length s, where $n_0 = |G_0|$ is the total number of grids. We define a grid transition event as a taxi sub-trajectory, that travels within a grid g_i until the first time it traverses to an adjacent grid g_j. The time duration of the sub-trajectory is thus the travel time of the grid transition event. Suppose that in the historical trajectory dataset, there are in total v_{ij} grid transition events from g_i to its neighboring grid g_j, each of which takes $t_{ij}(k)$ travel time with $1 \leq k \leq v_{ij}$. The average travel time from grid g_i to g_j is computed as follows:

$$T_{ij} = \sum_{k=1}^{v_{ij}} t_{ij}(k)/v_{ij},$$

where the n_0 by n_0 square matrix $T = [T_{ij}]$ represents the average travel time among adjacent grids. Note that the diagonal entries of T indicates the travel time within each grid. In our study, we estimate within-grid travel time as the average travel time inside each grid.

Shortest path travel time matrix C. Given the grids and the adjacent average travel time matrix T. The road map can be represented as a graph, with grids as nodes. There is a directed edge from grid g_i to g_j, if the corresponding entry $T_{ij} > 0$. Since only grids that cover some road segments have historical taxi GPS data, we first extract the giant strongly connected component (GSCC) [16] denoted as $G \subseteq G_0$ from all grids. Then, based on the graph of GSCC, we can compute the shortest path between any pair of grids, with the sum of the weights of its constituent edges minimized, which can be obtained using Dijkstra's algorithm or Bellman-Ford algorithm. In the rest of the paper, we will primarily work on this GSCC G, instead of G_0, since the GSCC components are grids covering roads, non-GSCC grids are off the road network and have no traffic. We denote the shortest path distance from grid g_i to g_j as C_{ij}, and $C = [C_{ij}]$ thus form the shortest path travel time matrix among grids. Figure 4 highlights the gridded road map of Shenzhen city (of total $1,508$ grids), with a giant connected component of 760 grids.

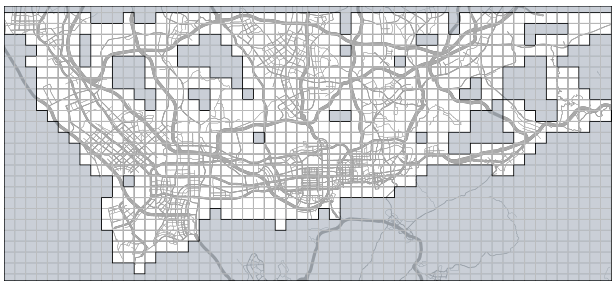

Fig. 4. Griding road map and GSCC extraction

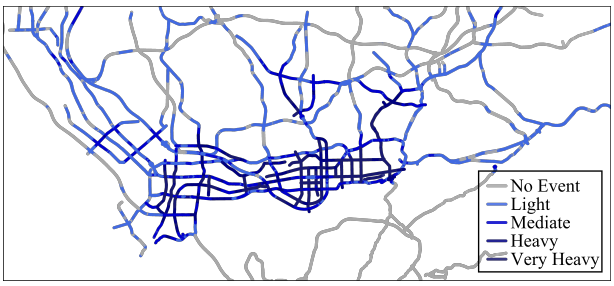

Fig. 5. The spatial distribution of seeking events

B. Extracting Sub-Trajectories

As elaborated earlier (in Figure 1 in Section II), each EV taxi periodically takes three actions in loop, namely, traveling, seeking, and charging. We now present how we extract these actions (i.e., sub-trajectories) from our data.

Charging Sub-trajectory extraction. Given the fact that the time for charging an EV taxi is usually within a certain charging interval, say, 30 minutes to 150 minutes for the current electric vehicles, we can detect charging sub-trajectories by combining the trajectory data and the existing charging station information. To be precise, if a sub-sequence of GPS records indicates the same location at an existing charging station, and the time staying there is within a charging interval, then these GPS records are labeled as a charging sub-trajectory.

Seeking and Traveling Sub-trajectory extraction. We detect seeking events based on two common sense rules, (1) prior to each charging sub-trajectory, there is a seeking sub-trajectory, which determines the ending point of the sub-trajectory; (2) a seeking sub-trajectory starts from dropping the last passenger before the next charging sub-trajectory. After finding the charging and seeking sub-trajectories, we label all other un-labeled records as traveling sub-trajectories.

Visualization. Consider the starting point of a seeking sub-trajectory as a seeking event, we highlight the geo-distribution of seeking events in our data in Figures 5, namely, the road segments colored by the numbers of seeking events.

C. Optimal Charging Station Deployment (OCSD)

We solve the OCSD problem by proposing a two-stage optimization framework. We first formulate the optimal charging station placement (OCSP) problem as an integer programming problem, which is NP-hard. We provide an LP-rounding based approximation algorithm to solve it with provable error bounds. Then, we model the optimal charging point assignment (OCPA) problem as a minimization problem of average charging point utilization, which minimizes the average proportion of time each charging point is occupied. Below, we elaborate these two components in detail.

1) Optimal Charging Station Placement (OCSP): We first formulate the problem of placing K charging stations, given L initial charging stations, with the objective to minimize the average travel time for an EV to find the nearby charging station.

Given the gridded road map $G = \{g_i\}$ $(1 \leq i \leq n_0)$, we denote W_i as the total number of seeking events within grid g_i in the trajectory dataset, and $W = \sum_{i=1}^{n_0} W_i$ as the total number of seeking events in all grids. Moreover, taking the shortest path travel time between grids, $C = [C_{ij}]$, as an input, we are now in a position to formulate the OCSP problem. Given a total number of K charging stations to deploy, let $y = [y_j]$ denote the deployment configuration, with each $y_j = 0$ or 1 representing whether or not a charging station should be deployed in grid g_j. Given that there exist L charging stations in L grids in grid set G_L, Obviously, $\sum_{g_j \in G} y_j \leq K + L$ should hold. Let $X = [X_{ij}]$ be a 0/1 indicator, representing whether the EVs which seek for charging stations inside g_i, will step to grid g_j for charging.

We aim to find y, indicating the best grids to deploy up to K stations, and X, the assignment of each non-station grid to a station grid, such that the *average seeking time*, i.e., the travel time to find the nearest charging station, is minimized. This problem is formally formulated as below.

$$\min : \frac{1}{W} \sum_{g_i \in G} \sum_{g_j \in G} W_i X_{ij} C_{ij} \tag{1}$$

$$\text{s.t.:} \sum_{g_j \in G} X_{ij} = 1, \qquad \forall g_i \in G \tag{2}$$

$$\sum_{g_j \in G} y_j \leq K + L, \tag{3}$$

$$X_{ij} \leq y_j, \qquad \forall g_i, g_j \in G \tag{4}$$

$$X_{ij}, y_j = \{0, 1\}, \qquad \forall g_i, g_j \in G \tag{5}$$

$$y_j = 1, \qquad \forall g_j \in G_L \tag{6}$$

G_L in constraint eq.(6) represents the set of L grids with existing charging stations, $G_L \subseteq G$, which can also be empty, indicating that initially no charging station is deployed. The objective function in eq.(1) captures the average travel time for each EV to find a charging station. The first constraint (in eq.(2)) states that each EV has to be served by a charging station. The constraint in eq.(3) presents the budget of charging stations, namely, in total no more than $K + L$ charging stations can be deployed, including the existing L stations. The constraint in eq.(4) guarantees the validity of the assignment, that is, if grid g_i is assigned to g_j for charging with $X_{ij} = 1$, g_j should has a charging station, i.e., $y_j = 1$. The constraint in eq.(5) states that each X_{ij} or y_j has to be either 0 or 1,

where the constraint in eq.(6) specifies the existing charging stations by setting those y_j's to be 1.

Approximate Solution with LP-Rounding. The above integer linear programming (ILP) problem is a generalized uncapacitated k-median problem [31] except that there exist L given medians. A k-median problem is proven to be NP-hard [31], thus the above ILP problem is NP-hard, and approximating this problem is as hard as approximating a set cover problem, where there is no polynomial-time algorithm that is guaranteed to find the optimal solution for all instances, unless $P = NP$.

In the literature, there are a variety of approximation algorithms that employ LP-rounding method to solve the k-median problem, which contains two stages, namely, LP-relaxation and rounding the optimal fractional LP solution. For example, the LP-rounding algorithm used in [11], [35] achieves a $6\frac{2}{3}$ approximation for k-median problem. [22] gives a 3-approximation algorithm using primal-dual schema. [19] utilizes randomization to improve the approximation guarantee to 2.408, which is further improved in [13] to $(1 + 2/e)$. Note that these approximate algorithms for solving k-median problem with different error bounds and computational complexities. In this study, we adopt and extend the approximation solution algorithm proposed in [28] based on LP-rounding. Other algorithms can be chosen, depending on the specific requirements on the error bound and complexity. Our approximation solution algorithm consists of two stages.

Stage 1: LP relaxation. We first relax the ILP (in eq.(1)-(6)) to LP by allowing $0 \leq X_{ij}, y_j \leq 1$ to be fractional values. The resulting LP can thus be solved in polynomial time with fractional X_{ij} and y_j obtained.

Stage 2: Rounding LP solution. The idea behind the rounding mechanism is as follows. Suppose the optimal solution to ILP problem has cost (i.e., the objective function value) B^{opt}_{ILP}. Since this solution is feasible for the linear programming, the optimal LP solution has some cost $B^{opt}_{LP} \leq B^{opt}_{ILP}$. Following [28], we apply the rounding algorithm below which leads to up to $2K + L$ centers and has cost at most $4B^{opt}_{LP}$.

In the LP solution, we denote $C_i = \sum_{g_j \in G} X_{ij} C_{ij}$ the seeking time for an EV in grid g_i to find a charging station. The total cost for all seeking events is thus $B_{opt,LP} = \sum_i W_i C_i$. We will find a set of $2K + L$ center grids such that each grid g_i is within distance at most $4C_i$ to its nearest center grid. Then, the total cost of seeking for charging stations will be at most $4B_{opt,LP}$.

Algorithm 1 outlines the approximation algorithm. The charging station grid set F is empty initially (Line 2). We start from those grids with the smallest values of C_i. We thus pick the grid with the smallest C_i and include g_i as a center grid (Line 4–6). We denote the set of grids whose distance from g_i is at most $2C_i$ as $P(g_i, 2C_i)$. We use g_i to cover any grid $g_{i'}$ such that $P(g_i, 2C_i) \vee P(g_{i'}, 2C_{i'}) \neq \phi$ (Line 7–11). To see this, notice that since the two grid sets overlap, they have some grid g_ℓ in common, and thus $C_{ii'} \leq C_{i\ell} + C_{\ell i'} \leq 2C_i + 2C_{i'} \leq 4C_{i'}$. Therefore, define the extended neighborhood of grid g_i as $\bar{V}_i = \{g_{i'} \in G | P(g_i, 2C_i) \vee P(g_{i'}, 2C_{i'}) \neq \phi\}$. Now, we can state the algorithm simply. Theorem 1 below provides the approximation bound of Algorithm 1, which can be proven by extending the proof in [28].

Algorithm 1 Approximate Charging Station Sitting Algorithm

1: Solve the LP and compute the values C_i;
2: $F \leftarrow \{\}$;
3: **while** $G \neq \{\}$: **do**
4: pick the $g_i \in G$ with the smallest C_i;
5: $F \leftarrow F \vee g_i$;
6: $y_i = 1$;
7: **for** $g_j \in \bar{V}_i$ **do**
8: $y_j = 0$ and $X_{ji} = 1$;
9: **for** $g_j \in G \backslash \bar{V}_i$ **do**
10: $X_{ji} = 0$;
11: $G \leftarrow G \backslash \bar{V}_i$;

Theorem 1. *Algorithm 1 selects $|F| \leq 2K + L$ grids, with the cost function in eq.(1) $Cost(F) \leq 4B_{opt,LP}$.*

Proof: (*Proof sketch.*) [28] proves that when there is no existing charging stations ($L = 0$), namely, all y_i's are unknown initially, Algorithm 1 selects $|F| \leq 2K$ grids, with the cost function in eq.(1) $Cost(F) \leq 4B_{opt,LP}$.

In our ILP, we have L initial $y_i = 1$, where in the relaxed LP solution, those grids should have the smallest C_i's, since all seeking events from those grids will find the charging stations in the same grid, namely, corresponding $X_{ii} = 1$. Hence, L initial stations will be extracted first, and the left K stations follow the error bounds in [28]. Overall, Algorithm 1 selects $|F| \leq 2K + L$ grids, with the cost function in eq.(1) $Cost(F) \leq 4B_{opt,LP}$. ∎

Practical issue. When we apply Algorithm 1 to deploy K new charging stations, it yields up to $2K$ new charging stations. Thus we use $K/2$ as the input to the OCSD framework to compensate the difference.

2) Optimal Charging Point Assignment (OCPA): Given the K charging stations deployed by the above placement solution, we are now in a position to address the problem of assigning $M \geq K$ charging points to those stations, with minimized average portion of time for each charging point being occupied.

Based on the station placement solution y_j (with $1 \leq j \leq n$), there are in total $K + L$ charging stations, and each grid $g_i \in G$ without a charging station is assigned to one nearby charging station (specified by X_{ij}). Hence, the entire city is divided into $K + L$ clusters of grids, each of which centers at a charging station grid, and all EVs that start to seek charging station from g_i will go to its cluster center grid for charging.

We denote $\hat{S} = [\hat{S}_\ell]$ with $1 \leq \ell \leq K + L$ as the number of existing charging points in each station ℓ (before new deployment), and $S = [S_\ell]$ as the number of new charging points deployed in station ℓ. Obviously, $\sum_\ell S_\ell \leq M$ holds. We consider the arrival pattern of EVs at each station follows Poisson distribution, thus the arrival process at each station ℓ

can be formulated as an $M/M/(S_\ell + \hat{S}_\ell)$ queue by queueing theory[3] [33]. The *arrival rate* (denoted as λ_ℓ) of station ℓ can be extracted from the trajectory data as the average number of per hour EVs that seek for charging. The *service rate* (denoted as μ_ℓ) of each charging point can be empirically obtained from the trajectory data, as the average number of per hour EVs being served at a charging point. Based on the queueing theory [33], The $M/M/(S_\ell + \hat{S}_\ell)$ model is a type of birthdeath process. The charging point utilization, captured by $\rho_\ell = \lambda_\ell/((S_\ell + \hat{S}_\ell)\mu_\ell)$ requires $\rho_\ell < 1$ for the queue to be stable (namely, otherwise, the queue length would go to infinity). ρ_ℓ represents the average proportion of time each charging point is occupied. Hence, our charging point assignment goal is to minimize the average charging point utilization in all charging stations, which is formally formulated as follows.

$$\min_S : \sum_{\ell=1}^{K+L} \frac{\lambda_\ell}{(S_\ell + \hat{S}_\ell)\mu_\ell} \quad \text{s.t.:} \quad \sum_{\ell=1}^{K+L} S_\ell = M. \tag{7}$$

Theorem 2 below states the optimal assignment solution to the OCPA problem.

Theorem 2. *Let $r = \sum_{\ell=1}^{L+K} \lambda_\ell/\mu_\ell$ and $r_\ell = \lambda_\ell/(\mu_\ell r)$. The optimal solution of S_ℓ's to the optimal charging point assignment (OCPA) problem is*

$$S_\ell = (M + \hat{M})r_\ell - \hat{S}_\ell, \quad \textit{with } \hat{M} = \sum_{\ell=1}^{K+L} \hat{S}_\ell.$$

.

Proof: The objective in eq.(7) can be rewritten as

$$\min_S : \quad \sum_{\ell=1}^{K+L} \frac{1}{(S_\ell + \hat{S}_\ell)\mu_\ell/\lambda_\ell} \tag{8}$$

Arithmetic-Harmonic Means Inequality [10] in eq.(9) holds

$$\frac{a_1 + \cdots + a_n}{n} \geq \frac{n}{\frac{1}{a_1} + \cdots + \frac{1}{a_n}}, \tag{9}$$

where the equality holds if and only if $a_1 = \cdots = a_n$. We apply the inequality eq.(9) to eq.(8), and obtain

$$\sum_{\ell=1}^{K+L} \frac{1}{(S_\ell + \hat{S}_\ell)\mu_\ell/\lambda_\ell} \geq \frac{(K+L)^2}{\sum_{\ell=1}^{K+L}(S_\ell + \hat{S}_\ell)\mu_\ell/\lambda_\ell},$$

where the equality is attained with $S_\ell = (M + \hat{M})r_\ell - \hat{S}_\ell$. ■

IV. Evaluations

To evaluate the performance of our optimal charging station deployment (OCSD) framework, we conduct comprehensive experiments using a large scale EV taxi trajectory dataset. OCSD framework consists of two components, i.e., optimal charging station placement (**OCSP**) and optimal charging point assignment (**OCPA**). By comparing with baseline algorithms, the experimental results demonstrate that OCSD can achieve a 26%-94% reduction on the average seeking time to find a charging station, and up to two orders of magnitude reduction on the waiting time. Moreover, when considering the total *idle time*, namely, the sum of the travel and waiting time, our OCSD framework outperforms all other methods with significant idle time reduction. Below, we present the baseline algorithms, experiment settings and results.

A. Baseline Algorithms

Baselines for charging station placement. Given a road map grid structure $G = \{g_j\}$, we denote G_L as the subset of L grids with existing charging stations. Hence, the goal of charging station placement is to find up to K grids from G/G_L to deploy new charging stations.

(1) *Random station placement* (Rand-SP): This baseline algorithm uniformly at random chooses K grids from G/G_L to deploy new charging stations.

(2) *Top seeking events* (Top): This baseline method chooses those grids from G/G_L with the top number of seeking events in the EV trajectory data.

The output of each baseline method is an indicator vector $y = [y_j]$ for grids. $y_j = 1$ indicates that a charging station is built in grid g_j, and $y_j = 0$, otherwise. Note that the total number of non-zero entries in y (i.e., the number of grids with charging stations) is $K + L$, with K newly deployed plus L existing stations. Then, we assume that a seeking event in a grid g_i without a charging station always goes to the closest charging station for charging, namely, the grid g_j with the shortest path travel time. This assignment is expressed with a *grid assignment matrix* $X = [X_{ij}]$, with $X_{ij} = 1$ representing that seeking events from grid g_i will go to grid g_j for charging, and with $X_{ij} = 0$, otherwise. The state-of-the-art methods for hydrogen and gas station sitting [14], [21], [9] cannot be used for comparison, since they require the source-destination of each trip, which is hard to obtain.

Baselines for charging point assignment. Given the indicator vector $y = [y_i]$, listing $K + L$ grids with charging stations, we have the following two baseline algorithms to assign $M \geq K + L$ charging points to those $K + L$ stations.

(1) *Random point assignment* (Rand-PA): This baseline randomly assigns M charging points to $K + L$ charging stations.

(2) *Average charging point assignment* (Aver.): This baseline method equally assigns M charging points to $K + L$ charging stations. To be precise, firstly, each station is assigned $d = \lfloor M/(K + L) \rfloor$ charging points. The rest $r = M - d(K + L)$ charging points are assigned to $r = M - d(K + L)$ randomly selected grids from G_{K+L}, i.e., the set of grids deployed with charging stations.

The output of each baseline method is a vector $S = [S_\ell]$ $(1 \leq \ell \leq K + L)$ for those grids with charging stations, where S_ℓ indicates the number of new charging points assigned to the station in grid ℓ. By combining with the vector of existing charging points numbers, i.e., $\hat{S} = [\hat{S}_\ell]$, we obtain a vector

[3] M stands for Markovian; $M/M/c$ means that the system has a Poisson arrival process, an exponential service time distribution, and c servers.

$\bar{S} = [\bar{S}_\ell]$, with $\bar{S}_\ell = \hat{S}_\ell + S_\ell$, indicating the total number of charging points in grid ℓ after new deployment.

B. Experiment Settings

From the sub-trajectory extraction, we obtain in total $N = 44,159$ seeking sub-trajectories from all 490 EV taxis during November 2013. We run 3-fold cross validation as follows. We divide the one month data into three parts, namely, (1) 11/01/2013–11/10/2013, (2) 11/11/2013–11/20/2013, and (3) 11/21/2013–11/30/2013. We take seeking events from each dataset to deploy charging stations and points using our OCSP and OCPA methods and different baseline algorithms, respectively. Then, we use seeking events from the other two datasets to calculate the average seeking, waiting time, and their sum (idle time). The overall results are the average of the results obtained by three different folds.

• *Seeking time.* The charging station placement method determines the seeking time of seeking events. The starting grid g_i of a seeking sub-trajectory indicates where the seeking sub-trajectory starts from. The grid assignment matrix X, i.e., the output of a charging station placement method, answers which grid g_j the seeking event should go to. The corresponding shortest path travel time C_{ij} (from grid g_i to g_j) is thus the seeking time for the seeking event.

• *Waiting time.* Each seeking sub-trajectory is followed by a charging sub-trajectory, thus we know the actual charging time of each seeking event. Then, for a seeking event, we know (1) its *starting time*, *starting grid*, and *charging time* from the trajectory data, and (2) its *charging grid*, *seeking time* from the output of the charging station placement method. By taking into consideration of S, i.e., the output of a charging point assignment method, we can thus mimic the EV taxi charging processes and estimate the waiting time based on the availability of charging points when each taxi arrives. Figure 6 illustrates how the waiting time is computed. There is only one charging point, where EV 1 arrives when the charging point is idle, thus it takes no waiting time; EV 2 arrives when EV 1 is charging, it waits until EV 1 finishes charging. The EV 3 arrives while EV 1 is charging and EV 2 is waiting, and it has to wait until EV 2 finishes charging.

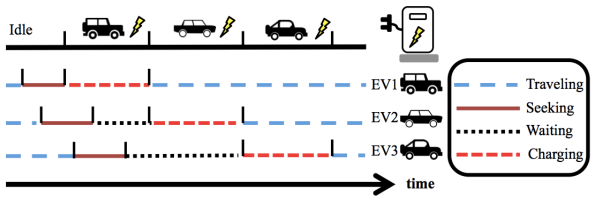

Fig. 6. Evaluation process

Table III lists configurations used in our evaluation.

TABLE III
EVALUATION CONFIGURATIONS

Side length (s)	$\{0.005, 0.01, 0.02, 0.03\}$
New Charging stations (K)	$\{5, 10, \cdots, 45, 50\}$
New Charging points (M)	$\{100, 150, \cdots, 500, 550\}$
Charging station placement methods	OCSP, Rand-SP, & Top
Charging point assignment methods	OCPA, Rand-PA, & Aver.

C. Seeking Time Evaluation

Figures 7–9 show the comparison on average seeking time when applying our OCSP and the baseline algorithms (i.e., Rand-SP and Top). The results with a grid side length of 0.005 are presented in Figure 7. As the number of new charging stations (K) increases, the average seeking time, namely, on average each EV spends to find a charging station, decreases for all the methods. Our OCSP requires about 98s to 159s with 5 to 50 new charging stations deployed, where Rand-SP and Top methods require 139s to 235s and 178s to 246s to reach the nearest charging station, respectively. Denote $T_s^{OCSP}(K)$, $T_s^{Top}(K)$, and $T_s^{Rand-SP}(K)$ as the seeking time needed with OCSP, Top, Rand-SP methods, when the number of newly deployed charging stations is K. The seeking time reduction rate $\Delta T_s^*(K)$ is defined as the relative reduced seeking time by our OCSP method from the baseline algorithm:

$$\Delta T_s^*(K) := (T_s^*(K) - T_s^{OCSP}(K))/T_s^{OCSP}(K).$$

where * represents Rand-SP or Top method. Hence, from the result, in Figure 7 (side length $s = 0.005$), OCSP achieves a seeking time reduction rate from 26% (with $K = 30$) to 48% (with $K = 5$) over Top, and 54.7% (with $K = 5$) to 94.4% (with $K = 20$) over Rand-SP. Figures 8–9 show consistent results with the grid side length set to 0.01 and 0.02, respectively.

D. Waiting Time Evaluation

Figures 10– 12 present the comparison results on the waiting time between our OCPA and the baseline algorithms (i.e., Rand-PA and Aver.). To eliminate the effect of the station placement stage, we fix the station distribution generated by OCSP, Top, or Rand-SP, and compare their average waiting time. Due to the limited space, we in this section only present the results with OCSP for placing the stations, and vary the number of new stations K and new charging points M. Figure 10 shows that when $K = 5$ new stations are deployed (given 25 existing stations), our OCPA leads to the lowest average waiting time. As the number of allowed charging points M increases, the average waiting time decreases drastically, say, from 495s (with $M = 100$) to 20s (with $M \geq 300$). On the other hand, Aver. yields roughly doubled average waiting time over OCPA. Rand-PA performs even worse, with the average waiting time from 1300s (20 minutes) to 3100s (50 minutes). This happens because Aver. treats each station equally, namely, evenly distribute the number of charging points to every station, without considering the underlying uneven distribution of seeking events.

Figures 11–12 present similar results for $K = 25$ and $K = 50$, respectively. From the results, we observe consistent patterns as that in Figure 10. One thing to note here is that given the same number of charging points M to assign, the average waiting time (for more charging stations, larger K) is slightly higher than that of less stations (smaller K). This happens because the more stations, the less number of charging events each station may cover; thus, the less chance a charging

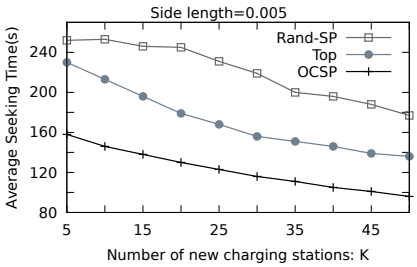

Fig. 7. Seeking time: side length: 0.005

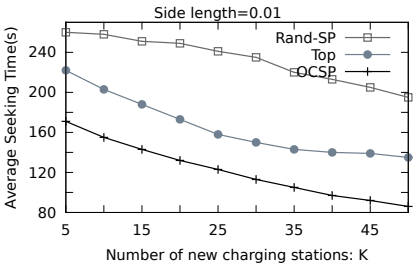

Fig. 8. Seeking time: side length: 0.01

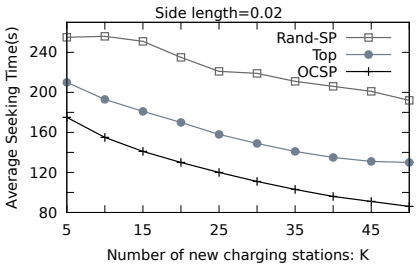

Fig. 9. Seeking time: side length: 0.02

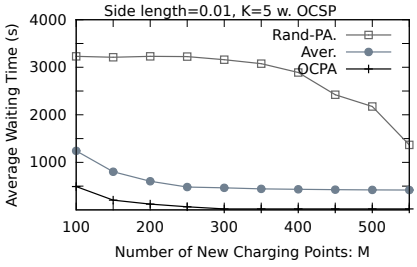

Fig. 10. Waiting time: K=5, side length: 0.01

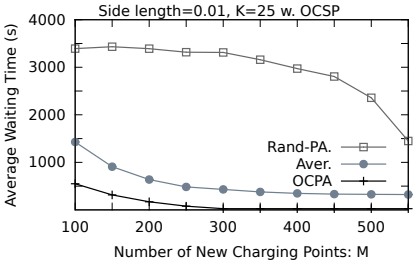

Fig. 11. Waiting time: K=25, side length: 0.01

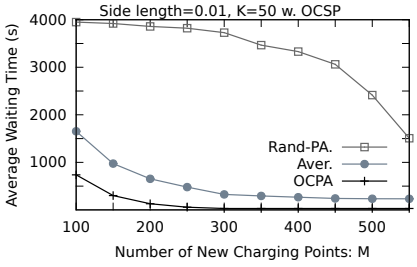

Fig. 12. Waiting time: K=50, side length: 0.01

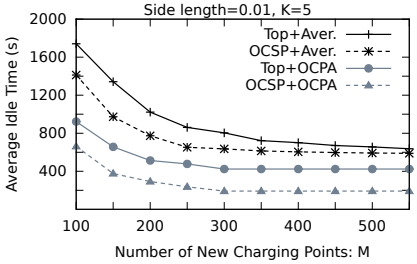

Fig. 13. Idle time: K=5, side length: 0.01

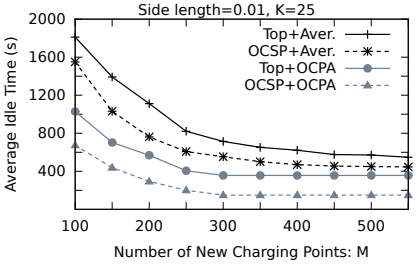

Fig. 14. Idle time: K=25, side length: 0.01

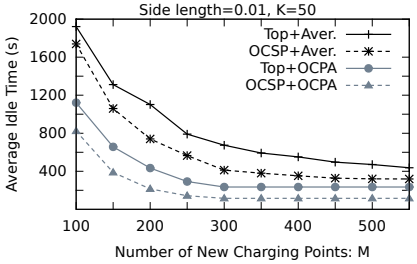

Fig. 15. Idle time: K=50, side length: 0.01

point is going to be reused. Consider two extreme cases: given 500 charging points, in one scenario, we build only one charging station, with all 500 points assigned to it; in another scenario, we build 500 charging station, where each has only one charging point. Obviously, for the seeking time (time needed to go to the station), 500 stations are far better than one station. However, for waiting time, one super station (with 500 points) will provide highest utilization for all charging points. This reveals an interesting trade-off that the charging station sitting problem is different from gas/hydrogen station sittings: to build more super stations or small stations? We will investigate this question by considering the sum of both seeking and waiting time in the next subsection, and provide an empirical answer using our dataset.

E. Idle Time Evaluation

Now, we consider the average idle time, namely, the sum of average seeking and waiting time. Given three charging station placement methods, OCSP, Top, and Rand-SP, and three charging point assignment methods, OCPA, Aver., and Rand-PA, we evaluate the idle time for all nine possible combinations. We observe that any combination with Rand-SP or Rand-PA will yield very large average idle time, say, larger than $3,000$s for most of the cases. Hence, we omit those results for brevity, and only present results for our framework OCSP+OCPA, and baseline combinations including OCSP+Aver., Top+OCPA, and Top+Aver. The results for $K = 5$ and $s = 0.01$ are presented in Figure 13, where we observe that our OCSP+OCPA always performs the best for different numbers of charging points, M, over other the baseline combinations. Moreover, as the number of charging points M increases, the average idle time decreases quickly, and reaches a convergence state when M is large enough. Figures 14–15 present consistent results for the configurations of $\{K = 25, s = 0.01\}$ and $\{K = 50, s = 0.01\}$. We observe that for the cases with smaller M, i.e., less charging points, more stations lead to a longer idle time; on the other hand, for the cases with larger M, i.e., more charging points, more stations yield a shorter idle time. This answers the trade-off question, "Super or small stations?": When a sufficient number of charging points are allowed, more smaller stations are better; when the number of charging points is insufficient, supper stations are preferred. This happens simply because of the trade-off between the seeking time and the waiting time. When M is small, the waiting time is much larger than the seeking time, thus super stations are better. On the other hand, when M is large, the waiting time decreases significantly, say, even lower than the seeking time (e.g., around 20s for $M \geq 300$ in Figure 10–12), more small stations become better.

The evaluation results (on seeking, waiting, idle time) from three folds are quite similar, beacause the geo-distributions of seeking events in the three periods are almost identical. There are 527 grids with non-zero seeking events. Figure 16 show the geo-distributions of seeking events in each ten-day period, where we can see that the geo-distributions are all the same.

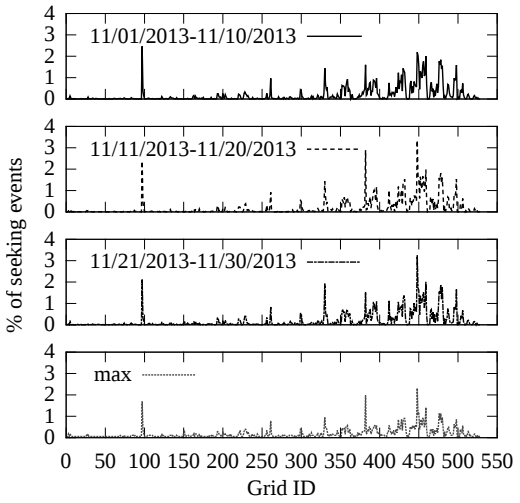

Fig. 16. Geo-distributions of seeking events in three periods. The last subfigure presents the geo-distribution of rush-hour arrival rates in Nov. 2013.

V. Discussions

Now, we present two extensions to OCSD framework, including charging point assignment using rush-hour seeking demands, and adaptation to time-varying seeking policy.

A. Charging Point Assignment using Rush-Hour Demands

OCPA aims to assign charging points to minimize the *average portion of time* for each charging point being occupied. However, in reality, the peak demands, i.e., maximum hourly number of seeking events, usually occur during rush hours and become the bottle-neck that leads to high waiting time. Consider a time interval T, e.g., $T = 4$ hours. Denote the seeking event arrival rate for station ℓ during the t-th interval of length T, as $\lambda_\ell^{(t)}$, with $1 \le t \le t_{max}$. Each station ℓ is thus associated with a sequence of seeking event arrival rates. We denote $\lambda_\ell^{max} := \max_t \lambda_\ell^{(t)}$ as the maximum arrival rate of station ℓ among all intervals from 1 to t_{max}, which captures the rush hour seeking demands of station ℓ. To account for such rush hour effect, the stations with larger demand variation should be assigned with more charging points, so they can perform well even during the rush hours. For those stations with relatively flat temporal distribution of seeking demands, we can assign relatively less charging points to allow a bit higher utilization ratio. To achieve this goal, the objective function in OCPA stage should be tuned to capture the rush hour station utilizations rather than the average utilizations, which can be done by substituting the average arrival rate λ_ℓ, with the maximum arrival rate λ_ℓ^{max} as follows. μ_ℓ remains the same, since the serving rate is in general stable over time.

$$\min_S : \sum_{\ell=1}^{K+L} \frac{\lambda_\ell^{max}}{(S_\ell + \hat{S}_\ell)\mu_\ell} \quad \text{s.t.:} \quad \sum_{\ell=1}^{K+L} S_\ell = M. \tag{10}$$

The optimal solution can be obtained by following Theorem 2 as $S_\ell = (M + \hat{M}) r_\ell^{max} - \hat{S}_\ell$, where $r_\ell^{max} = \lambda_\ell^{max}/(\mu_\ell r^{max})$ and $r^{max} = \sum_{\ell=1}^{L+K} \lambda_\ell^{max}/\mu_\ell$.

Comparing OCPA using average arrival rates vs rush hour arrival rates, the results hinge on the difference between the geo-distributions of the two sets of arrival rates. The last subfigure in Figure 16 presents the geo-distribution of the rush hour arrival rates, which is almost identical to the distribution of average arrival rates, shown in the first three subfigures. Thus, the solution and system performance in terms of seeking, waiting, and idle time do not show difference when using the average arrival rates vs rush hour arrival rates. Here, we omit the evaluation results for brevity.

B. Time-Varying Seeking Policy

Throughout the design of OCSD, we assume the seeking policy as: EV taxis always go to the nearest charging station for charging. Going beyond this basic assumption, OCSD framework can be extended to a (more general) time-varying seeking policy, namely, an EV taxi chooses a charging station depending on both the seeking time (i.e., travel distance) and how busy the station is at that time. Inspired by many studies where multiple types of objectives exist (e.g., [23], [27], [26]), the formulation can be written below as a time-dependent linear combination of OCSP and OCPA with a trade-off parameter[4] $\theta \ge 0$. Then, an EV taxi will choose a charging station that is relatively nearby and less busy.

$$\min : \sum_{t=1}^{t_{max}} \sum_{g_j \in G} \left(\sum_{g_i \in G} \frac{W_i^{(t)}}{W} X_{ij}^{(t)} C_{ij} + \frac{\theta \lambda_j^{(t)}}{(S_j + \hat{S}_j)\mu_j^{(t)}} \right) \tag{11}$$

$$\text{s.t.:} \sum_{g_j \in G} X_{ij}^{(t)} = 1, \qquad \forall g_i \in G, t \in [1, t_{max}] \tag{12}$$

$$\sum_{g_j \in G} y_j \le K + L, \tag{13}$$

$$X_{ij}^{(t)} \le y_j, \qquad \forall g_i, g_j \in G, t \in [1, t_{max}] \tag{14}$$

$$X_{ij}^{(t)}, y_j = \{0, 1\}, \qquad \forall g_i, g_j \in G, t \in [1, t_{max}] \tag{15}$$

$$y_j = 1, \qquad \forall g_j \in G_L \tag{16}$$

$$\sum_{g_j \in G} S_j = M, \tag{17}$$

$\lambda_j^{(t)} = \sum_{g_i \in G} W_i^{(t)} X_{ij}^{(t)}/T$, $\mu_j^{(t)} = \sum_{g_i \in G} M_i^{(t)} X_{ij}^{(t)}/T$, with $M_i^{(t)}$ as the number of EV taxis (from grid g_i) being served during the t-th interval, and $*^{(t)}$ represents the variable $*$ during the t-th interval of length T. In the above joint optimization, the boolean variable $X_{ij}^{(t)}$ is time-varying, which means that at different time intervals, seeking events occuring at grid g_i may go to different stations for charging.

The joint optimization problem eq.(11)-(17) is an integer nonlinear programming problem, which can be solved by optimization solver toolboxes such as BARON [37], [34] (employing branch-and-bound method [15], [24]), AIMMS Outer Approximation (AOA) [1] (utilizing standard outer approximation algorithm [17]), etc.

[4] θ captures the weight between seeking time and charging point utilization in deploying charing stations and points, which is determined empirically.

Result Analysis. In this work, we can safely assume the nearest charging policy, where the solutions of y and S, as well as the evaluation results, do not have much difference from assuming the time-varying seeking policy. This is because in the dataset, we observe that for more than 90% of seeking events, EVs went to the nearest station for charging. We explain this phenomena by the fact that the current number charging stations is still far from sufficient, and they are distributed in the city in general far away from each other (comparing to the distribution of gas-stations). Hence, when a taxi driver wants to charge the vehicle, the primary concern in mind is still the travel distance, given other choices of stations would be roughly equally busy.

VI. Related work

To the best of our knowledge, we are the first to utilize large-scale electric vehicle (EV) trajectory data to facilitate urban deployment of charging stations and charging points. In this section, we discuss three topics that are closely related to our work, including (1) urban computing, and (2) facility location, and (3) EV charging.

Urban Computing, as an emerging research area, integrates urban sensing, data management, data analytic, and service providing together as a unified process for an unobtrusive and continuous improvement of peoples lives, city operation systems, and the environment [43]. The goal is to solve a variety of emerging problems in urban areas, such as traffic congestion, energy consumption, and pollution, based on the data of traffic flow, human mobility, and geographical data, etc. In [44], they inferred the real-time and fine-grained air quality information throughout a city, based on the air quality data reported by existing monitoring stations and a variety of data sources observed in the city. In [30], they tried to identify the hot spots of moving vehicles in an urban area via a novel, non-density-based approach, called mobility-based clustering. In [39], they proposed a framework, called DRoF, to discover regions of different functions in a city using both human mobility among regions and points of interests (POIs) located in a region. In [41], the authors tried to sense the refueling behavior and citywide petrol consumption in real-time, based on the trajectories of vehicles. In [42] and [36], they tried to discover the traveling companions and gathering patterns of vehicles, respectively. In [40], [38], authors exploit the phone user mobility data collected from cell towers to perform Point-of-Interest prediction and outdoor advertising. As a classic urban computing problem, we in this paper aim to design a framework to strategically deploy *charging stations* and *points* in a city for EVs.

Facility location has been studied extensively in the literature, primarily in operations research. It concerns with the optimal placement of facilities to minimize transportation costs while considering various factors and constraints, such as avoiding placing hazardous materials near housing and competitors' facilities. In particular, there are a variety of works investigating the station sitting problem for deploying gas stations and hydrogen filling stations [32], [14], [21], [8], [20], [29]. The problem is primarily formulated as facility location in road networks [14], [21], [8], [20] or as a subset of the existing gasoline station network [32]. However, these facility location models cannot be applied for charging station sitting, because of the following two reasons. (1) Differing from the gasoline cars, the charging durations of EVs are long, i.e., around a few hours, which yields long waiting time for incoming EVs (when charging points are all occupied). Existing models do not capture such (long) charging time and waiting time. (2) The existing facility location models all require trip origin-destination data as an input, which is in general hard to obtain. To address these challenges, we in this work make the first attempt to study how to strategically deploy *charging stations* and assign *charging points* for EVs.

EV charging. The surge of EVs imposes a significant load on the distribution network: with AC Level 2 charging, EVs can be charged at up to 80A at 240V [5], a load of 19.2kW, whereas a typical North American home has an average load of only 1kW. Therefore, a single EV being charged at the peak Level 2 rate could impose an instantaneous load as large as that imposed by nearly twenty average homes. There are a few works addressing how to control the EV charging load. Ardakania *et al.* [7] propose a distributed control algorithm that adapts the charging rate of EVs to the available capacity of the network ensuring that network resources are used efficiently and each EV charger receives a fair share of these resources. They obtain sufficient conditions for stability of this control algorithm in a static network, and demonstrate through simulation in a test distribution network that their algorithm quickly converges to the optimal rate allocation. Gerding *et al.* [18] design an online auction protocol for this problem, where EV owners use agents to bid for power and state time windows in which an EV is available for charging. All above works consider how to balance between the electricity distribution and EV charging load, given an existing charging station infrastructure, where they do not address our charging station deployment and charging point assignment problems, which are the focus of this paper.

VII. Conclusion

In this paper, we study the problem of how to strategically deploy charging stations and charging points throughout a city so as to minimize the average time each electric vehicle needs to spend for finding an available charging point for charging. We develop a data-driven optimal charging station deployment (OCSD) framework that takes a variety of data sources as inputs, including EV taxi trajectory data, city road map data, and existing charging station information, and performs optimal charging station placement (OCSP), and optimal charging point assignment (OCPA). These two optimization components are designed to minimize the average time to travel to the nearest charging station, and the average waiting time for an available charging point, respectively. To evaluate the performance of our OCSD framework, we conduct extensive experiments using one-month EV taxi trajectory data. The evaluation results demonstrate that our OCSD framework

achieves a 26%–94% reduction rate on the average seeking time to find a charging station, and six times reduction on the waiting time before charging, over the baseline methods.

Our results also answer an interesting question: "Super or small stations?": When a sufficiently larger number of charging points can be deployed, more small stations are preferred; when relatively less charging points can be deployed, super stations is a wiser choice. This observation motivates us to further investigate the inconsistency of the charging station deployment for different K, and tackle the issue by designing a roll-out strategy for charging station deployment process. We leave this problem for our future work.

VIII. Acknowledgement

We would like to thank the anonymous reviewers and our shepherd, Dr. Divesh Srivastava, for their helpful feedbacks on this paper. This work was partially funded by National Natural Science Foundation of China (Grant No. 61100220 & 11271351), Shenzhen New Industry Development Fund under grant No. JCYJ20120617120716224, and 973 Program No. 2014CB340304. C.-Y. Chow and K.-L. Chan were partially supported by Guangdong Natural Science Foundation of China under Grant S2013010012363 and research grants (CityU Project No. 9231131 & 9680117).

References

[1] AIMMS Outer Approximation (AOA). http://www.aimms.com/aimms/solvers/aoa/.

[2] Electric car market growth soars in 2013. http://www.energyandcapital.com/articles/electric-car-market-growth-soars-in-2013/4173.

[3] Google Geo-coding API. https://developers.google.com/maps/documentation/geocoding/.

[4] Openstreetmap. http://www.openstreetmap.org/.

[5] SAE J1772 Standard. http://standards.sae.org/j1772_201210.

[6] Why we need electric cars. http://www.motherearthnews.com/green-transportation/electric-cars-zmaz06onzraw.aspx.

[7] O. Ardakanian, C. Rosenberg, and S. Keshav. Distributed control of electric vehicle charging. In *ACM e-Energy*, 2013.

[8] R. Bapna, L. S. Thakur, and S. K. Nair. Infrastructure development for conversion to environmentally friendly fuel. *European Journal of Operational Research*, 142(3):480–496, 2002.

[9] O. Berman, R. C. Larson, and N. Fouska. Optimal location of discretionary service facilities. *Transportation Science*, 26(3):201–211, 1992.

[10] P. S. Bullen, D. S. Mitrinovic, and P. M. Vasic. *Means and their Inequalities*. D. Reidel Dordrecht, 1988.

[11] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *ACM STOC*, pages 1–10, 1999.

[12] S. Chawla, Y. Zheng, and J. Hu. Inferring the root cause in road traffic anomalies. In *ICDM*, pages 141–150, 2012.

[13] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.

[14] R. Church and C. R. Velle. The maximal covering location problem. *Papers in regional science*, 32(1):101–118, 1974.

[15] J. Clausen. Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, 1999.

[16] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Giant strongly connected component of directed networks. *Physical Review E*, 64(2):025101, 2001.

[17] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.

[18] E. H. Gerding, V. Robu, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings. Online mechanism design for electric vehicle charging. In *AAMAS*, 2011.

[19] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *SODA*, pages 649–657. SIAM, 1998.

[20] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.

[21] J. Hooker, R. Garfinkel, and C. Chen. Finite dominating sets for network location problems. *Operations Research*, 39(1):100–118, 1991.

[22] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *FOCS*, pages 2–13. IEEE, 1999.

[23] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. Cooperative content distribution and traffic engineering in an isp network. In *ACM SIGMETRICS*, pages 239–250, 2009.

[24] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[25] Y. Li, M. Steiner, J. Bao, L. Wang, and T. Zhu. Region sampling and estimation of geosocial data with dynamic range calibration. In *ICDE*, pages 1096–1107. IEEE, 2014.

[26] Y. Li, Z.-L. Zhang, and D. Boley. The routing continuum from shortest-path to all-path: A unifying theory. In *IEEE ICDCS*, pages 847–856, 2011.

[27] Y. Li, Z.-L. Zhang, and D. Boley. From shortest-path to all-path: The routing continuum theory and its applications. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1745–1755, 2014.

[28] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992.

[29] Z. Lin, J. Ogden, Y. Fan, and C.-W. Chen. The fuel-travel-back approach to hydrogen station siting. *International journal of hydrogen energy*, 33(12):3096–3101, 2008.

[30] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li. Towards mobility-based clustering. In *ACM KDD*, pages 919–928, 2010.

[31] A. Meyerson, L. O'Callaghan, and S. Plotkin. A k-median algorithm with running time independent of data size. *Machine Learning*, 56(1-3):61–87, 2004.

[32] M. A. Nicholas, S. L. Handy, and D. Sperling. Using geographic information systems to evaluate siting and networks of hydrogen stations. *TRR Journal*, 1880(1):126–134, 2004.

[33] T. L. Saaty. *Elements of queueing theory*, volume 423. McGraw-Hill New York, 1961.

[34] N. V. Sahinidis. *BARON 12.1.0: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2013.

[35] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *ACM STOC*, 1997.

[36] L.-A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C.-C. Hung, and W.-C. Peng. On discovery of traveling companions from streaming trajectories. In *IEEE ICDE*, 2012.

[37] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[38] R. Wang, C.-Y. Chow, S. Nutanong, Y. Lyu, Y. Li, M. Yuan, and V. C. Lee. Exploring cell tower data dumps for supervised learning-based point-of-interest prediction. *ACM SIGSPATIAL GIS*, 2014.

[39] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and POIs. In *ACM KDD*, pages 186–194, 2012.

[40] M. Yuan, K. Deng, J. Zeng, Y. Li, B. Ni, X. He, F. Wang, W. Dai, and Q. Yang. OceanST: A distributed analytic system for large-scale spatiotemporal mobile broadband data. *VLDB Endowment*, 7(13):1–4, 2014.

[41] F. Zhang, D. Wilkie, Y. Zheng, and X. Xie. Sensing the pulse of urban refueling behavior. In *ACM UbiComp*, 2013.

[42] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. On discovery of gathering patterns from trajectories. In *IEEE ICDE*, 2013.

[43] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 2014.

[44] Y. Zheng, F. Liu, and H.-P. Hsieh. U-Air: When urban air quality inference meets big data. In *ACM KDD*, 2013.

[45] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, and E. Chang. Diagnosing New York City's Noises with Ubiquitous Data. In *Ubicomp*, 2014.