Demand-Aware Charger Planning for Electric Vehicle Sharing

Bowen Du
BDBC and SKLSDE Lab,
Beihang University
Beijing, China
dubowen@buaa.edu.cn

Yongxin Tong*
BDBC and SKLSDE Lab,
Beihang University
Beijing, China
yxtong@buaa.edu.cn

Zimu Zhou
TIK,
ETH Zurich
Zurich, Switzerland
zzhou@tik.ee.ethz.ch

Qian Tao
BDBC and SKLSDE Lab,
Beihang University
Beijing, China
qiantao@buaa.edu.cn

Wenjun Zhou
Department of BAS,
University of Tennessee
Knoxville, USA
wzhou4@utk.edu

ABSTRACT

Cars of the future have been predicted as shared and electric. There has been a rapid growth in electric vehicle (EV) sharing services worldwide in recent years. For EV-sharing platforms to excel, it is essential for them to offer private charging infrastructure for exclusive use that meets the charging demand of their clients. Particularly, they need to plan not only the places to build charging stations, but also the amounts of chargers per station, to maximally satisfy the requirements on global charging coverage and local charging demand. Existing research efforts are either inapplicable for their different problem formulations or are at a coarse granularity. In this paper, we formulate the Electric Vehicle Charger Planning (EVCP) problem especially for EV-sharing. We prove that the EVCP problem is NP-hard, and design an approximation algorithm to solve the problem with a theoretical bound of $1 - \frac{1}{e}$. We also devise some optimization techniques to speed up the solution. Extensive experiments on real-world datasets validate the effectiveness and the efficiency of our proposed solutions.

CCS CONCEPTS

• **Mathematics of computing → Combinatorial optimization**;
• **Theory of computation → Facility location and clustering**;

KEYWORDS

Electric Vehicles; Location Selection; Submodularity

ACM Reference Format:
Bowen Du, Yongxin Tong, Zimu Zhou, Qian Tao, and Wenjun Zhou. 2018. Demand-Aware Charger Planning for Electric Vehicle Sharing. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3220032

*Corresponding Author

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220032

1 INTRODUCTION

Electric vehicles (EVs) are fast expanding in the fleets of car-sharing programs, due to the limit on greenhouse gas emissions, drop in battery cost, and more EV models on the market. Autolib'[1], a pioneer EV-sharing company, runs 3, 980 EVs and more than 126, 900 registered subscribers in Paris [21]. Leading car-sharing platforms such as Uber[2] and Lyft[3] are also adding more EVs and promoting zero-emission rides. Morgan Stanley predicts the integration of car-sharing and EVs as the next frontier and expects EVs to account for 50 − 60% of global light vehicle sales by 2040 [15].

The practicability of EV-sharing relies on not only the population of EVs, but also the infrastructure that meets the charging demand [22]. Autolib' offers 1, 084 charging stations in Paris [21], while Car2Go[4] is reported to switch EVs to gas-fueled cars due to the lack of charging stations in San Diego. Thus it is crucial for EV-sharing platforms to plan and deploy proper charging infrastructure to ensure user experience of clients.

It is non-trivial to plan charging infrastructure for EV-sharing. Ideal charging infrastructure planning should *(i)* ensure pervasive coverage so that EV-drivers can reach as many points of interest (POIs) as possible and *(ii)* offer sufficient numbers of chargers in each charging station since the local charging demand may vary dramatically at different POIs. Therefore, one challenge that EV-sharing platforms confront is *how to select locations for charging stations and deploy proper amount of chargers to meet as much charging demand as possible given a limited budget.*

Previous research on gas station planning is not directly applicable because EV charging takes much longer time (more than 8 hours to fill up the battery [1]) and can easily saturate a gas station [2][5][9]. There is an increasing research interest in EV charging infrastructure deployment [12][13][10] [23][8][11]. Most works [12][13][10][23][8] target at urban planning for governments, where the infrastructure is expected to fulfill all the charging demand of citizens. These schemes are impractical for EV-sharing companies. On the one hand, a company may not have enough resource and budget to meet all the charging demand. On the other hand, the plan that yields the maximum profit may not necessarily

[1] https://www.autolib.eu/
[2] http://www.uber.com/
[3] http://www.lyft.com/
[4] https://www.car2go.com/

cover all the charging demand. Other works focus on maximizing the demand satisfaction at a coarse granularity of regions [11].

In this paper, we study the Electric Vehicle Charger Planning (EVCP) problem for EV-sharing platforms. The problem accounts for both coverage of charging infrastructure and local charging demand per charging station. It aims to jointly select the locations of stations and the number of chargers to maximize the charging demand under a budget constraint. We design solutions with theoretical guarantees and also propose techniques to speed up the solution. Evaluations on real datasets validate the effectiveness and efficiency of the proposed solution.

The main contributions of this paper are summarized as follows.

- We formulate a novel Electric Vehicle Charger Planning (EVCP) problem for EV-sharing. It optimizes the distribution of EV chargers to maximize a balance function between coverage of POIs and local charging demand. We prove the EVCP problem is NP-hard.
- We analyze the submodularity of the EVCP problem and design an effective and efficient greedy approximation algorithm with a competitive ratio $1 - \frac{1}{e}$ and a time complexity of $O(m^2 n)$, where m is the number of charging stations, and n is the number of POIs.
- Extensive experiments on real datasets show that our algorithms outperform traditional optimization methods in terms of effectiveness and efficiency.

In the rest of this paper, we formulate the EVCP problem and prove its hardness in Sec. 2, propose the solution and speedup techniques in Sec. 3, and present the evaluations in Sec. 4. Finally we review related work in Sec. 5 and conclude in Sec. 6.

2 PROBLEM FORMULATION

This section formally defines the Electric Vehicle Charger Planning problem and proves its NP-hardness.

2.1 Preliminaries

We first define the important notations to formulate the Electric Vehicle Charger Planning (EVCP) problem.

Definition 2.1 (Road Network). A road network is an undirected graph $G = (V, E)$. Each node in V is either a POI $v_i \in \{v_1, v_2, \cdots, v_n\}$ or a candidate location $w_j \in W = \{w_1, w_2, \cdots, w_m\}$, where n and m are the number of POIs and candidate locations, respectively. Each edge adjacent two nodes is associated with a weight, representing the distance between these two nodes. The distance $dis(.,.)$, is the hops of the shortest path between the two nodes in G.

Note that we distinguish the nodes of POIs and candidate locations in a road network. This is because usually EV-sharing platforms are only allowed to deploy chargers in their stores (*i.e.* candidate locations) rather than any POI.

Definition 2.2 (Candidate Station). A candidate station is a tuple $c_j = (w_j, d_j, r_j)$, where w_j is a candidate location node in V, representing the location of c_j, and d_j is the number of local charging demand of c_j. POIs with a distance no greater than radius r_j can be covered by c_j.

We assume the set of candidate stations is predetermined. Given two candidate stations c_{j_1} and c_{j_2}, $d_{j_1} = d_{j_2}$ and $r_{j_1} = r_{j_2}$ always hold if their locations coincide, *i.e.*, $w_{j_1} = w_{j_2}$. For simplicity, we use the notation $c_{j_1} = c_{j_2}$ to represent $w_{j_1} = w_{j_2}$. Note that it is not the main focus to estimate the local charging demand d_j for a candidate station c_j. One simple and practical approach is to use the number of EVs returned to a candidate station as a rough estimate. Therefore we assume d_j is a non-negative integer in our paper.

Definition 2.3 (Selection). A selection is a tuple $s_j = (c_j, n_j)$, which means n_j chargers are planned to be deployed in candidate station c_j, where n_j is an integer no less than 0. Two selections s_{j_1} and s_{j_2} are said to be *coincident* if their corresponding candidate stations coincide, *i.e.* $c_{j_1} = c_{j_2}$.

Definition 2.4 (EV Charger Plan). Given a set of candidate stations C, an EV charger plan (plan for short) is a set of selections S, where each candidate station c_j appears in exactly one selection in S. That is, $S = \{s_j | s_j = (c_j, n_j), n_j \in \mathbb{Z}^*, j = 1, 2, \cdots, m\}$. Further define the *size* of S as total number of EV chargers of S, *i.e.* $size(S) = \sum_{j=1}^{m} n_j$.

2.2 Reward of Meeting Charging Demand

As discussed in Sec. 1, an optimal plan should account for both coverage of charging stations and local charging demand per charging station. This subsection defines the reward a plan can derive from these two aspects of charging demand.

2.2.1 Reward of POI Coverage. A plan is expected to cover as many POIs as possible in a road network so that EV-drivers can reach a wide spectrum of places. We model this requirement on charging demand by the total number of POIs covered by the charging stations in a plan.

A POI is said to be covered by a plan if it is located in the radius of a selection with at least one charger. Denote $P(s_j)$ as the set of POI nodes covered by a selection s_j, and $P(S)$ as the set of POI nodes covered by the EV charger plan S. Formally,

$$P(s_j) = \begin{cases} \emptyset, & \text{if } n_j = 0 \\ \{v | v \in V - W, dis(v, w_j) \le r_j\}, & \text{if } n_j \ge 1, \end{cases} \quad (1)$$

$$P(S) = \cup_{j=1}^{m} P(s_j). \quad (2)$$

We further define a reward of a plan S derived from POI coverage as the total number of POI nodes covered by S:

$$\mathcal{R}_c(S) = |P(S)|. \quad (3)$$

2.2.2 Reward of Local Charging Demand. Another aspect of a good plan is to provide sufficient amounts of chargers to satisfy the local charging demand at each charging station.

Denote by u the rate of local charging demand that a single charger can satisfy over a period of a time (*e.g.*, a week). Then the number of satisfied local charging demand by a selection s_j is:

$$\mathcal{R}_d(s_j) = min(d_j, u \cdot n_j). \quad (4)$$

Accordingly, we define the reward that a plan S can derive from the local charging demand of all selected charging stations as:

$$\mathcal{R}_d(S) = \sum_{j=1}^{m} \mathcal{R}_d(s_j). \quad (5)$$

Table 1: Parameters of candidate stations in Example 2.6.

Candidate Station	Demand	Radius	S^1	S^2	S^3
w_1	9	6	3	2	3
w_2	0	5	0	1	1
w_3	1	6	1	1	0

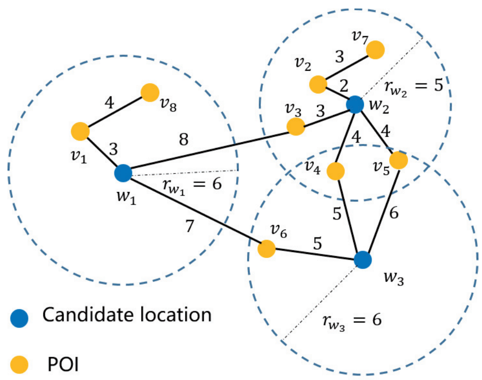


Figure 1: The road network information for Example 2.6.

2.3 Problem Statement

Given the definitions of the rewards for POI coverage and local charging demand, the Electric Vehicle Charger Planning problem can be formulated as follows.

Definition 2.5 (EVCP Problem). Given a road network $G = (V, E)$, a set of candidate stations C, the rate of local charging demand satisfied by one charger u, an adjustable parameter α, and a budget B on the total number of chargers, the EVCP problem finds a plan S of size no greater than B, such that the total reward $\mathcal{R}(S)$,

$$\mathcal{R}(S) = \alpha \cdot \mathcal{R}_c(S) + (1 - \alpha) \cdot \mathcal{R}_d(S), \tag{6}$$

is maximized.

Unlike traditional facility location problems [3], which decide whether to build the station in each candidate location, our problem further considers the number of chargers in each station as well as the satisfaction of local charging demand.

We illustrate the EVCP problem via the following example.

Example 2.6. Suppose there are 3 candidate locations $w_1 - w_3$ and 8 POIs $v_1 - v_8$, whose local demand and radius are shown in Table 1. The underlying road network is shown in Fig. 1. The number on each edge represents the distance between its adjacent nodes. Note that v_8 cannot be covered by w_1 because $dis(w_1, v_8) = 7 > r_1$. We set B, u, and α as 4, 3 and 0.5, respectively.

If we build 3 chargers in w_1 and 1 charger in w_3, we satisfy 10 local charging demands, and 4 POIs ($\{v_1, v_4, v_5, v_6\}$) are covered. The total reward is $\mathcal{R}(S^1) = 7$. If we build 1 charger in each station, and the surplus charger in w_1, there are 7 local charging demands satisfied (6 in w_1, and 1 in w_3), and 7 POIs ($v_1 - v_7$) are covered. The total reward is $\mathcal{R}(S^2) = 7$. However, the optimal selection is to build 3 chargers in w_1 and 1 charger in w_2. In this plan 9 local charging demands are satisfied and 6 POIs are covered. The total reward is $\mathcal{R}(S^3) = 7.5$.

2.4 NP-Hardness of EVCP Problem

In this subsection we prove the NP-hardness of the EVCP problem.

Theorem 2.7. *The EVCP problem is NP-hard.*

Proof. We prove the NP-hardness of the EVCP problem by reducing the maximum coverage problem [7] with unit weights to a special case of the EVCP problem where $\alpha = 1$.

The decision version of maximum coverage problem is illustrated as follows. Given a finite universal set $X = \{x_1, x_2, ..., x_n\}$, a set of X's subsets $\mathcal{T} = \{T_1, T_2, ..., T_m\}$, an integer k, and a threshold t, the decision maximum coverage problem is to decide whether there exists a set $\mathcal{T}' \subseteq \mathcal{T}$ with its size no more than k, such that the number of elements in X covered by $\mathcal{T}'$, *i.e.* $|\cup_{T \in \mathcal{T}'} T|$, is no less than the threshold t.

The decision version of the special EVCP problem is illustrated as follows. Given a road network $G = (V, E)$, a set of candidate stations C, a budget B, the rate of local charging demand satisfied by a unit charger u, and a threshold t', the objective is to decide whether there exists a plan S with its size no greater than the budget B, such that the reward, $\mathcal{R}(S) = \mathcal{R}_c(S)$ is no smaller than t'.

We map an instance of the maximum coverage problem, $I = (X, \mathcal{T}, k, t)$, to the instance of the special EVCP problem, denoted by $I' = (G, C, B, u, t')$ as follows. The element $x_i \in X$ is mapped to a POI node $v_i \in V$ for $i = 1, 2, \cdots, n$, and T_j is mapped to a candidate station node $w_j \in V$ for $j = 1, 2, \cdots, m$. Then V is expressed as $\{v_1, v_2, \cdots, v_n, w_1, w_2, \cdots, w_m\}$. There is an edge with weight 1 between two nodes v_i and w_j if (and only if) $x_i \in T_j$. The candidate stations are then located in the nodes $\{w_1, w_2, ..., w_m\}$ with their local demands $d_j = 1$ and radii $r_j = 1$. Formally, $C = \{c_j | c_j = (w_j, 1, 1), j = 1, 2, \cdots, m\}$. B and t' are mapped from the maximum size k and threshold t in I, respectively, and u is set to be 1. Note that the values of d_j and u have no effect on the objective function. This way, the shortest distance between any two nodes in V is 1 only if the two nodes are v_i and w_j, and $x_i \in T_j$.

On the one hand, if there is a certification of maximum coverage problem $\mathcal{T}' = \{T_{q_1}, T_{q_2}, ..., T_{q_l}\}$, we choose the selection s_j for the instance I' as

$$s_j = \begin{cases} (c_j, 1), & \text{if } c_j \in \{q_1, \cdots, q_l\} \\ (c_j, 0), & otherwise. \end{cases} \tag{7}$$

Then $S = \{s_j | 1 \le j \le m\}$ is a certification of the special EVCP problem.

On the other hand, if there is a certification of the special EVCP problem $S = \{(c_1, n_1), (c_2, n_2), \cdots, (c_m, n_m)\}$, then $\mathcal{T}' = \{T_j | n_j \ge 1\}$ is a certification of the maximum coverage problem. Note that the total number of chargers is always no fewer than the total number of candidate locations with at least one charger:

$$|\mathcal{T}'| = \sum_{j=1}^{m} \mathbb{I}(n_j \ge 1) \le \sum_{j=1}^{m} n_j = B, \tag{8}$$

where $\mathbb{I}(.)$ is the indicator function.

Thus as long as there is a certification for the decision maximum coverage problem, there is a certification for the decision version of the special EVCP problem, and vice versa. Then the decision maximum coverage problem can be reduced to the decision version

Algorithm 1: Charger-based Greedy (CG) Algorithm

input : A road network G, a set of candidate stations C, total budget of chargers B
output: The EV charger plan S

1 $S \leftarrow \{s_j | s_j = (c_j, 0), j = 1, 2, \ldots, m\}$;
2 **while** *the number of deployed chargers does not exceed* B **do**
3 $\quad j' \leftarrow argmax_{1 \le j \le m}\{\alpha\Delta\mathcal{R}_c^j(S) + (1-\alpha)\Delta\mathcal{R}_d^j(S)\}$;
4 $\quad$ **if** $\alpha\Delta\mathcal{R}_c^{j'}(S) + (1-\alpha)\Delta\mathcal{R}_d^{j'}(S) = 0$ **then**
5 $\quad\quad$ **break**;
6 $\quad S \leftarrow S_+^{j'}$;
7 **return** S;

of the special EVCP problem. Since the decision maximum coverage problem is NP-complete, the general EVCP problem is NP-hard. ◻

3 METHODS

Due to the NP-hardness of the EVCP problem, we design an approximation algorithm (Sec. 3.1) and prove its theoretical guarantees (Sec. 3.2). We also propose speedup techniques to improve the efficiency of the proposed algorithm (Sec. 3.3).

3.1 Charger-based Greedy Algorithm

In this subsection we illustrate the details of our approximation algorithm and analyze its time complexity.

Before describing the algorithm in more detail, we first introduce some useful notations. Denote by s_j^+ the selection that adds one more charger than s_j in the same candidate station.

$$s_j^+ = (c_j, n_j + 1). \tag{9}$$

Denote by S_+^j the EV charger plan where there is one more charger built in c_j than plan S, while the numbers of chargers built in other candidate stations are the same as that in S.

$$S_+^j = (S - \{s_j\}) \cup \{s_j^+\}. \tag{10}$$

Denote by $\Delta\mathcal{R}^j(S)$ the increased reward after we deploy one more charger to the candidate station c_j of plan S.

$$\Delta\mathcal{R}^j(S) = \mathcal{R}(S_+^j) - \mathcal{R}(S). \tag{11}$$

Similarly, define the increased reward of POI coverage as $\Delta\mathcal{R}_c^j(S)$ and the increased reward of local charging demand as $\Delta\mathcal{R}_d^j(S)$:

$$\Delta\mathcal{R}_c^j(S) = \mathcal{R}_c(S_+^j) - \mathcal{R}_c(S) \tag{12}$$

$$\Delta\mathcal{R}_d^j(S) = \mathcal{R}_d(S_+^j) - \mathcal{R}_d(S) \tag{13}$$

Then we propose a Charger-based Greedy (CG) algorithm, which adds one charger to the EV charger plan greedily.

Alg. 1 shows the pseudocode of the CG algorithm. In line 1, an empty EV charger plan S is initialized. Then in each iteration, the CG algorithm greedily chooses the candidate station with the maximum increased reward (line 3), and updates the plan (line 6). Note that when the maximum increased reward decreases to 0 ("yes" judgement in line 4), the total reward will no long change. As a result the algorithm can break (line 5).

Table 2: Values of $\mathcal{R}_+^j(S)$ in each iteration.

Iteration	$\Delta\mathcal{R}^1(S)$	$\Delta\mathcal{R}^2(S)$	$\Delta\mathcal{R}^3(S)$
1^{st}	2	**2.5**	2
2^{nd}	**2**	0	1
3^{rd}	**1.5**	0	1
4^{th}	**1.5**	0	1

Example 3.1. Consider the example in Example. 2.6. The values of $\Delta\mathcal{R}^1(S)$, $\Delta\mathcal{R}^2(S)$, and $\Delta\mathcal{R}^3(S)$ in every iteration are shown in Table 2. The maximum increased rewards of every iteration are marked in bold. In the 1^{st} iteration, c_2 is selected because it can cover a large number of POIs. Then in the 2^{nd}, 3^{rd} and 4^{th} iterations, c_1 is selected because it can satisfy more demands. The total reward gained from this selection is $2.5 + 2 + 1.5 + 1.5 = 7.5$.

3.2 Algorithm Analysis

Despite the simplicity of the CG algorithm, we prove its theoretic guarantee via submodularity. We also analyze its time complexity in this subsection.

We first define a new set function upon the set of all chargers. It operates the same as the objective function on the selection set S, but has a characteristic of submodularity.

Specifically, for each candidate station c_j, make B duplicates of the EV chargers, denoted by $e_j^{(1)}, e_j^{(2)}, \ldots, e_j^{(B)}$, and obtain a set $E_j = \{e_j^{(1)}, e_j^{(2)}, \ldots, e_j^{(B)}\}$. Further define a universal set $U = \cup_{j=1}^m E_j$. Given a subset $E \in U$, define a plan *derived from* E, denoted by S_E, as $S_E = \{s_j | s_j = (c_j, |E_j \cap U|), j = 1, 2, ..m\}$. This means that the number of chargers in station c_j in S_E is exactly the size of $E_j \cap U$, for $j = 1, 2, \ldots, m$. Finally, define the objective function of a subset $E \in U$, as the reward of E's corresponding derived plan. Without loss of generation, denote the objective function by $\mathcal{R}(E)$. We have $\mathcal{R}(E) = \mathcal{R}(S_E)$. In the same way define $\mathcal{R}_c(E)$ and $\mathcal{R}_d(E)$. Then we have $\mathcal{R}(E \cup \{e_j^{(k)}\}) = \mathcal{R}(S_+^j)$ if $e_j^{(k)} \notin E$. This means that if we obtain a subset E of U by using a greedy algorithm with its size $|E| \le B$, the plan derived from E, say S_E, is exactly the plan generated by our CG algorithm, and the same as for the optimal objective values.

Theorem 3.2. *The competitive ratio of Charger-based Greedy (CG) algorithm is* $1 - \frac{1}{e}$.

Proof. $\mathcal{R}_c(E)$ is nondecreasing submodular referring to [7] (when adding one charger to the station c_j with $n_j = 0$) and Lemma 3.3 (when adding one more charger to the station c_j with $n_j \ge 1$).

The nondecreasing submodularity of $\mathcal{R}_d(E)$ can be derived from Eq.17, since $min(d_j, n_j \cdot u) = d_j + min(0, n_j \cdot u - d_j)$. We omit the details due to the space limit.

According to Proposition 2.7 in [14], the positive linear combination of nondecreasing submodular functions, $\mathcal{R}(E) = \alpha\mathcal{R}_c(E) + (1-\alpha)\mathcal{R}_d(E)$, is also nondecreasing submodular. Hence we have the competitive ratio of the greedy algorithm on E with a budget B is $1 - \frac{1}{e}$, due to [14]. This consequently leads to the competitive ratio of our CG algorithm. ◻

Time Complexity. Denote by n, m and B the number of POIs, charging stations, and EV chargers, respectively. Assume that the sets of POIs are implemented by an unordered set. Then the time complexities of $\Delta\mathcal{R}_c^j(S)$ and $\Delta\mathcal{R}_d^j(S)$ are $O(n)$ and $O(1)$, respectively. Hence each time line 3 is executed, it consumes $O(m(1+n)) = O(mn)$ time totally. The time complexity of lines 4-5 is $O(n)$, and that of line 6 is $O(1)$. Since lines 2-6 are executed at most $O(B)$ times, their total time complexity is $O(mnB)$.

3.3 Speedup Techniques

Although the CG algorithm has a performance guarantee, it has a time complexity of $O(mnB)$. This subsection introduces techniques to accelerate the algorithm.

3.3.1 Observations. The speedup techniques are inspired by two observations.

First, if we deploy one more charger in a charging station c_j, the total number of covered POIs increases only when there is no charger in c_j before. This observation leads to the following lemma.

Lemma 3.3. *After adding one more charger to a station c_j with $n_j \geq 1$ upon the plan S, the increased reward for POI coverage of any station stay invariable. Formally, if $n_j \geq 1$ in a selection S, then*

$$\Delta\mathcal{R}_c^{j'}(S_+^j) = \Delta\mathcal{R}_c^{j'}(S), 1 \leq j' \leq m. \tag{14}$$

Proof. From the definition of the reward of POI coverage, if adding one more charger to a station c_j with $n_j \geq 1$, the POIs covered by S_j^+ is the same as that covered by S, *i.e.* $P(S_j^+) = P(S)$ because $P(s_j^+) = P(s_j)$. The increased reward for POI coverage of S_+^j, *i.e.* $\Delta\mathcal{R}_c^{j'}(S_+^j)$, can be derived as follows:

$$\Delta\mathcal{R}_c^{j'}(S_+^j) = \begin{cases} 0, & \text{if } n_{j'} \geq 1 \\ \mathcal{R}((S_+^j)_+^{j'}) - \mathcal{R}_c(S_+^j), & \text{otherwise.} \end{cases} \tag{15}$$

If $j' = j$, then $n_j + 1 \geq 1$. $\Delta\mathcal{R}_c^j(S_+^j) = 0$, which is exactly the value of $\Delta\mathcal{R}_c^j(S)$. Otherwise if $j' \neq j$, since $P(S_j^+) = P(S)$, $\Delta\mathcal{R}_c^{j'}(S_+^j) = \Delta\mathcal{R}_c^{j'}(S)$ also holds. □

The second observation is, when adding one more charger to the charging station c_j, the reward of local charging demand generated from other stations does not change. Hence the reward of local charging demand of S_+^j, *i.e.* $\mathcal{R}_d(S_+^j)$, can be formulated by

$$\mathcal{R}_d(S_+^j) = \sum_{\substack{1 \leq j' \leq m \\ j' \neq j}} \mathcal{R}_d(s_{j'}) + \mathcal{R}_d(s_j^+). \tag{16}$$

The increased reward of local charging demand can consequently be derived from the following equation.

$$\begin{aligned} \Delta\mathcal{R}_d^j(S) &= \mathcal{R}_d(S_+^j) - \mathcal{R}_d(S) \\ &= \sum_{\substack{1 \leq j' \leq m \\ j' \neq j}} \mathcal{R}_d(s_{j'}) + \mathcal{R}_d(s_j^+) - \sum_{j'=1}^{m} \mathcal{R}_d(s_{j'}) \\ &= \mathcal{R}_d(s_j^+) - \mathcal{R}_d(s_j) \\ &= min(d_j, (n_j+1) \cdot u) - min(d_j, n_j \cdot u) \end{aligned} \tag{17}$$

Eq.17 directly leads to the following lemma.

Lemma 3.4. *After adding one more charger to a charging station c_j upon the plan S, the increased reward of local charging demand of any other stations stay invariable. Formally,*

$$\Delta\mathcal{R}_d^{j'}(S_+^j) = \Delta\mathcal{R}_d^{j'}(S), 1 \leq j' \leq m, j' \neq j. \tag{18}$$

Proof. The lemma derives from the fact that $\Delta\mathcal{R}_d^{j'}(S)$ can only be affected by the selection $s_{j'} = (c_{j'}, n_{j'})$, and $s_{j'}$ does not change when adding one more charger to a different station c_j. □

Lemma 3.3 and Lemma 3.4 indicate that $\Delta\mathcal{R}_c^j(S)$ and $\Delta\mathcal{R}_d^j(S)$ are possibly invariable. Hence we can record these increased rewards, and update them when their values change. In this way we can decrease the number of calculations of $\Delta\mathcal{R}_c^j(S)$ and $\Delta\mathcal{R}_d^j(S)$. However, in the worst case the algorithm still needs to operate B iterations.

Finally, consider the following lemma. The lemma inspires us to reduce the number of iterations to $O(m)$.

Lemma 3.5. *If we add one more charger to station c_j upon the plan S, and n_j satisfies $1 \leq n_j \leq \lfloor \frac{d_j}{u} \rfloor - 2$, then*

$$\Delta\mathcal{R}^{j'}(S_+^j) = \Delta\mathcal{R}^{j'}(S), j' = 1, 2, ..., m \tag{19}$$

Proof. On the one hand, since $n_j \geq 1$, $\Delta\mathcal{R}_c^{j'}(S_+^j) = \Delta\mathcal{R}_c^{j'}(S) = 0$ according to Lemma 3.3. On the other hand, we have obtained that $\Delta\mathcal{R}_d^{j'}(S_+^j) = \Delta\mathcal{R}_d^{j'}(S)$ for $1 \leq j' \leq m, j' \neq j$. Now consider the value of $\Delta\mathcal{R}_d^j(S_+^j)$ and $\Delta\mathcal{R}_d^j(S)$. Since $n_j < n_j + 1 < n_j + 2 \leq \lfloor \frac{d_j}{u} \rfloor \leq \frac{d_j}{u}$, we get $n_j \cdot u < (n_j+1) \cdot u < (n_j+2) \cdot u \leq d_j$. Substituting the inequation into Eq.17, we get,

$$\begin{aligned} \Delta\mathcal{R}_d^j(S_+^j) &= min(d_j, (n_j+2) \cdot u) - min(d_j, (n_j+1) \cdot u) \\ &= (n_j+2) \cdot u - (n_j+1) \cdot u \\ &= u = (n_j+1) \cdot u - n_j \cdot u = \Delta\mathcal{R}_d^j(S) \end{aligned} \tag{20}$$

□

Based on Lemma 3.5, when a charger is deployed in a station with the maximum increased reward, say c_j, and there are at least one charger deployed in c_j, we can deploy multiple chargers in c_j until it reaches $\lfloor \frac{d_j}{u} \rfloor$ chargers.

3.3.2 Fast-CG Algorithm. Based on Lemma 3.3, Lemma 3.4, and Lemma 3.5, we propose the Fast-CG algorithm. It has the same theoretical guarantee as the CG algorithm, but is more efficient in terms of time complexity.

Alg. 2 shows the pseudocode of the Fast-CG algorithm. In line 1, an empty EV charger plan S is initialized. Lines 2-3 initialize arrays $r_c[1..m]$, $r_d[1..m]$, which record the values of $\Delta\mathcal{R}_c^j(S)$ and $\Delta\mathcal{R}_d^j(S)$ in the current iteration, respectively. Then in each iteration, the algorithm first greedily chooses the candidate station with the maximum increased reward (line 5). Note that the same as Alg. 1, the Fast-CG algorithm stops when the maximum increased reward reduces to 0 (lines 6-7). According to Lemma 3.3, the increased rewards of POI coverage $r_c[1..m]$ are updated only when there is no charger deployed to the chosen station before, as lines 8-10 show. According to Lemma 3.4, only the increased reward of local charging demand of the chosen station, $r_d[j']$, is updated in each iteration, as line 15 shows. Otherwise if the number of current deployed

Algorithm 2: Fast-CG Algorithm

input : A road network G, a set of candidate stations C, total budget of chargers B
output: The EV charger plan S
1 $S \leftarrow \{s_j | s_j = (c_j, 0), j = 1, 2, ..., m\}$;
2 Define arrays $r_c[1..m]$,$r_d[1..m]$;
3 $r_c[j] \leftarrow \Delta\mathcal{R}_c^j(S)$, $r_d[j] \leftarrow \Delta\mathcal{R}_d^j(S)$ for $j = 1, 2, ..., m$;
4 **while** *number of deployed chargers* $n_b < B$ **do**
5 $\quad j' \leftarrow argmax_{1 \le j \le m}\{\alpha \cdot r_c[j] + (1 - \alpha) \cdot r_d[j]\}$;
6 $\quad$ **if** $\alpha \cdot r_c[j'] + (1 - \alpha) \cdot r_d[j'] = 0$ **then**
7 $\quad\quad$ **break**;
8 $\quad$ **if** $n_{j'} = 0$ **then**
9 $\quad\quad S \leftarrow S_+^{j'}$;
10 $\quad\quad r_c[j] \leftarrow \Delta\mathcal{R}_c^j(S)$ for $j = 1, 2, ..., m$;
11 $\quad$ **else if** $n_{j'} = 1$ *and* $n_{j'} < \lfloor \frac{d_j}{u} \rfloor$ **then**
12 $\quad\quad$ Deploy $min(\lfloor \frac{d_j}{u} \rfloor - n_{j'}, B - n_b)$ chargers to $c_{j'}$ and update S;
13 $\quad$ **else if** $n_{j'} = \lfloor \frac{d_j}{u} \rfloor$ **then**
14 $\quad\quad S \leftarrow S_+^{j'}$;
15 $\quad r_d[j'] \leftarrow \Delta\mathcal{R}_d^{j'}(S)$;
16 **return** S;

chargers $n_{j'}$ is smaller than the integer $\lfloor \frac{d_{j'}}{u} \rfloor$, we straightly deploy chargers until there are $\lfloor \frac{d_{j'}}{u} \rfloor$ chargers in $c_{j'}$, as shown in lines 11-12. Finally, in lines 13-14, if there are already $\lfloor \frac{d_{j'}}{u} \rfloor$ chargers in $c_{j'}$, we can only deploy one more charger to $c_{j'}$. This is because when $n_{j'} \ge \lfloor \frac{d_{j'}}{u} \rfloor + 1$, $\Delta\mathcal{R}^{j'}(S) = 0$ according to Eq.15 and Eq.17. This means the algorithm will break in line 7.

Table 3: Values of $\mathcal{R}_+^j(S)$ in each iteration.

Iteration	$\Delta\mathcal{R}^1(S)$	$\Delta\mathcal{R}^2(S)$	$\Delta\mathcal{R}^3(S)$	Chargers
1^{st}	2	**2.5**	2	1
2^{nd}	**2**	0	1	1
3^{rd}	**1.5**	0	1	2

Example 3.6. Back to our running example in Example. 2.6. The values of $\Delta\mathcal{R}^1(S)$, $\Delta\mathcal{R}^2(S)$ and $\Delta\mathcal{R}^3(S)$, and numbers of deployed chargers in every iteration are shown in Table 3. The maximum increased rewards of every iteration are marked in bold. The 1^{st} iteration and the 2^{nd} iteration are the same as those in Example. 3.1.

Note that after choosing w_1 in the 3^{rd} iteration, $\Delta\mathcal{R}^2(S)$ and $\Delta\mathcal{R}^3(S)$ do not need to be updated because of Lemma 3.3 and Lemma 3.4. In the 3^{rd} iteration, $1 = n_1 < \lfloor \frac{d_1}{u} \rfloor = 3$, hence $\lfloor \frac{d_1}{u} \rfloor - n_1 = 2$ chargers are deployed in c_1, and the Fast-CG algorithm returns the final EV charger plan.

Time Complexity. Similarly to Sec. 3.2, denote by n,m and B the number of POIs, charging stations, and EV chargers, respectively. For any $1 \le j \le m$, c_j is chosen by line 5 at most 3 times: the first time n_j from 0 to 1, the second time n_j from 1 to $\lfloor \frac{d_j}{u} \rfloor$ (if possible), and the third time from $\lfloor \frac{d_j}{u} \rfloor$ to $\lfloor \frac{d_j}{u} \rfloor + 1$ (if possible). This means that lines 5-15 are executed at most $O(m)$ times. Based on this conclusion, the total time complexity of line 5 is $O(m^2)$, that of line 10 is $O(m^2 n)$, and that of any other lines is $O(m)$. Total time complexity of this Fast-CG algorithm is $O(m^2 n)$.

4 EXPERIMENTAL EVALUATION

This section presents the evaluations of our proposed algorithms.

4.1 Experimental Settings

Datasets. Our dataset is generated from a real electric car-sharing platform in Beijing, China. Totally there are 55, 715, 107 real-time GPS data of 5, 216 vehicles, and their renting and returning records from June 2017 to September 2017. 1, 684 stations and 699, 142 POIs scatter on the map of Beijing, as shown in Fig. 2 and Fig. 3.

Table 4 lists the important constants in our experiments. Only those stations and POIs with longitudes between [116.42, 116.52] and latitudes between [39.86, 39.96] are considered. For simplicity, we reshape the area to be a 1000×1000 square. There are 139 stations and 11, 757 POIs located in this square. The number of local charging demand in each station is generated from the summation of charging demands of this station in a week in June 2017.

Table 5 shows the major parameters in our experiments. We assume that all candidate stations have the same coverage radius r. Since 1km on the real map is transformed to about 10 on the square, r varies from 10 to 50 with a step 10. As for u, since it takes $8 - 12$ hours to fill up an EV [1], the number of local charging demand satisfied by a single charger u varies from 6 to 14, with a step 2. Finally, since the running time of our algorithms is related to m, n and B, we change the values of m and B in the experiments of scalability (n is ignored because it is large enough). Note that in the experiments of scalability, since there are not enough stations in reality, the locations are synthetically generated.

Baseline. Most of existing works model their problems as integer programming problems, and use the optimization methods to

Table 4: Experimental constants.

Constant	Values
Longitude	$[116.42, 116.52] \times 10k$
Latitude	$[39.86, 39.96] \times 10k$
Number of Stations	139
Number of POIs	11, 757

Table 5: Experimental variable settings.

Variable	Values
r	10, 20, **30**, 40, 50
α	0.1, 0.3, **0.5**, 0.7, 0.9
B	200, 400, **600**, 800, 1000
u	6, 8, **10**, 12, 14
Scalability($m \times B$)	$200\times2k$, $400\times4k$, $600\times6k$, $800\times8k$, $1k \times 10k$

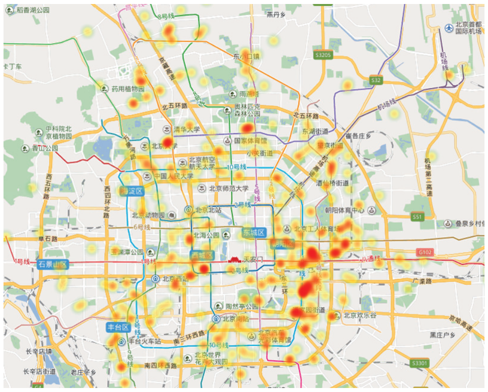


Figure 2: Distribution of stations.

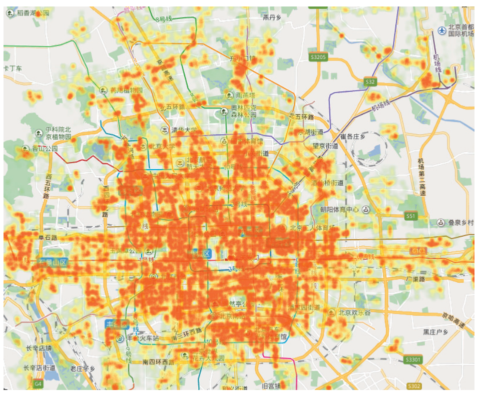


Figure 3: Distribution of POIs.

obtain the result [13][10][23][8][4][6][11]. Thus for the baseline, we formulate the EVCP problem as an integer programming problem, and then use the methods of simplex and branch and bound [16] to approximately achieve the solution.

Implementation. All the algorithms are implemented in C++, and the experiments were performed on a machine with 4 Intel(R) Core(TM) i5-5200H CPU and 8G memory.

Metrics. We compare the performances of the algorithms in terms of the output rewards, as well as time and memory costs.

4.2 Experiment Results

Effect of r. The first column of Fig. 4 shows the results of varying r. The total reward output by all the three algorithms increases when r increases, and our algorithms outperform the baseline. This is because when r increases, the number of covered POIs increases as well. Fast-CG consumes least time among the three algorithms. With the increase of r, CG and Fast-CG consume more time, but the time Fast-CG varies far less than that of CG. This might be because when r increases, it takes more time to make set union, but the union times of Fast-CG are less than CG. CG and Fast-CG also outperform the baseline in terms of the running time. As for memory, CG and Fast-CG consume a little more space as r increases, but are more efficient than the baseline. This is because when r increases, these two algorithms need more space to store the set of POIs covered by each candidate station.

Effect of α. The second column of Fig. 4 shows the results of varying α. The first observation is that when α increases, the rewards of the three algorithms all decrease linearly. This might be because the number of satisfied demands lessens than the number of covered POIs as α decreases (since a charger leads to a coverage of all POIs in range, but only satisfies at most u demands). Besides, CG and Fast-CG outperform the baseline, and Fast-CG performs the best in terms of running time. The memory costs of the three algorithms stay stable as α increases. CG and Fast-CG consume less space than the baseline.

Effect of B. The third column of Fig. 4 shows the results of varying B. We can observe that the rewards of three algorithms increase when B increases. This is reasonable because more chargers mean more satisfied local charging demand and covered POIs. The running time of Fast-CG is stable, but that of CG and the baseline increases as B increases. This is because CG takes B iterations to get the results, but Fast-CG does not. The memory costs of three algorithms are similar to that when varying α.

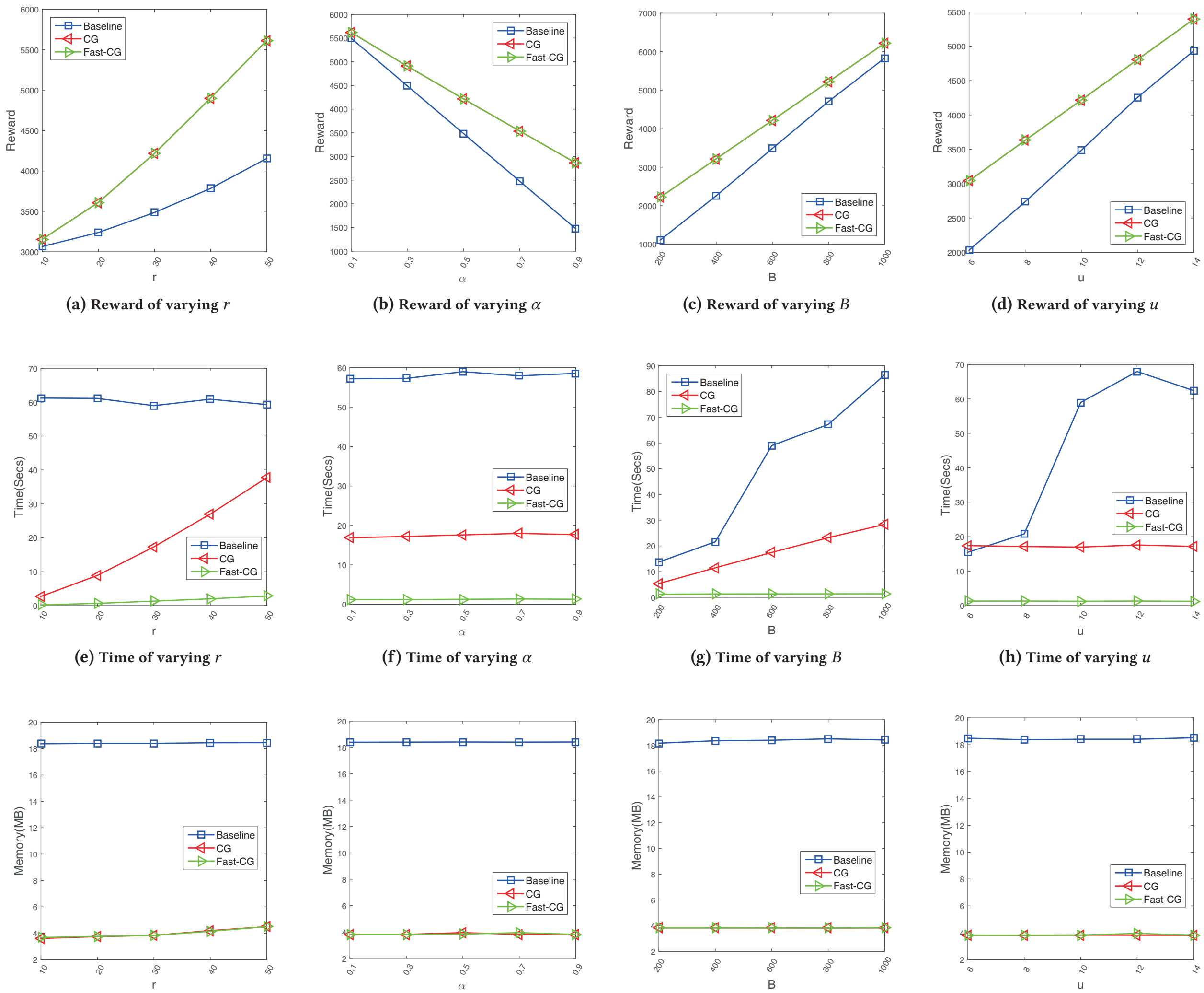
Effect of u. The forth column of Fig. 4 shows the results of varying u. The rewards of all the three algorithms increase as u increases, and CG and Fast-CG outperform the baseline. As for running time, CG and Fast-CG take stable time and outperform the baseline. Fast-CG performs the best in terms of running time. Still, CG and Fast-CG consume less memory than the baseline, which is similar to the trends in Fig. 4j, and Fig. 4k.

Scalability. The experimental results on scalability are shown in Fig. 5. CG and Fast-CG still outperform the baseline in terms of reward, and we omit the trend due to the limited space. The running time of CG becomes extremely large when $m \times B$ increases, due to redundant iterations in CG. The running time of Fast-CG also increases as $m \times B$ increases, but is still acceptable (307.97 seconds when $m \times B = 10^7$). The memory costs of the three algorithms increase as $m \times B$ increases, but CG and Fast-CG consume less memory than the baseline.

Summary of Results. CG and Fast-CG always perform the same in terms of reward, since our speedup techniques do not change the results of the greedy algorithm. For real dataset, CG and Fast-CG outperform the baseline in terms of the total reward, time and memory. Besides, comparing with CG, Fast-CG has the same reward and similar consuming memory, but consumes less time. CG consumes most time, and is unfit for large-scale dataset (*e.g.* 17398.6 seconds when $m \times B = 10^7$). Fast-CG performs the best on the real datasets, and has an acceptable running time even on large-scale datasets. The experiment results validate the effectiveness and efficiency of our algorithm and the speedup techniques.

5 RELATED WORK

With the rapid development of spatial crowdsourcing and car-sharing platforms, various issues in car-sharing have attracted extensive research attention [18][17][20][19]. EV-sharing, as an environment-friendly car-sharing service, is also gaining growing research interests. A particularly active topic is planning charging



(a) Reward of varying r (b) Reward of varying α (c) Reward of varying B (d) Reward of varying u

(e) Time of varying r (f) Time of varying α (g) Time of varying B (h) Time of varying u

(i) Memory of varying r (j) Memory of varying α (k) Memory of varying B (l) Memory of varying u

Figure 4: Results on varying r, α, B, and u.

infrastructure for EVs under different constraints and with diverse objectives. This section reviews the most representative works that involve charging demand either as constraints or objectives.

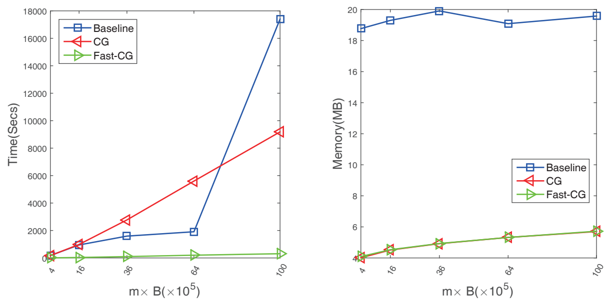
5.1 Charging Demand as Constraints

This thread of research requires all charging demand to be fulfilled when planning the placement of EV charging infrastructure. These schemes are mainly designed to facilitate governments for urban planning and policy making. Liu *et al.* [13] minimize the total cost of a charging transformer plan under the constraints of voltage and current of grids. Li *et al.* [10] minimize the average seeking and waiting time of all charging demands based on large scale taxi trajectory data. Xiong *et al.* [23] model the movements of charging demands among regions as the integer variables, and apply the Nash equilibrium to minimize the expectation of total charging time. Jia *et al.* [8] consider the construction cost and charging time at the same time. The objective is a summation of the construction cost, affected by the number of chargers in stations, and the operation cost, derived from the movement of consumers among regions.

Our work is different from these studies in two aspects. *(i)* We aim to provide an EV charger plan for EV-sharing platforms rather than for urban planning. The target is to maximize the satisfied charging demand with a limited budget to make profit. *(ii)* Most of these studies optimize not only the deployment of chargers but also the movement of charging demands. Our work has no assumptions or requirements on the movement of EV-drivers.

5.2 Charging Demand as Objectives

A more related thread of research focuses on maximizing the coverage or revenue of charging demand under different constraints.



(a) Time of scalability **(b) Memory of scalability**

Figure 5: Results on scalability.

Frade *et al.* [4] formulate the problem as a traditional maximum coverage problem. He *et al.* [6] deploy chargers with the equilibrium of transportation and electric power flow. Lam *et al.* [9] propose to maximize the revenue and minimize the cost at the same time. A bilevel optimization is formulated to solve the problem.

Our work is different from [4] by considering both coverage and the local demand at each charging station. Compared with [6] and [9], which optimize charger deployment for different *regions*, our work is finer-grained by planning charger deployment for each location. We also provide solutions that have theoretical guarantees.

6 CONCLUSION

In this paper, we formulate the Electric Vehicle Charger Planning (EVCP) problem, which deploys charging infrastructure (locations of stations and numbers of chargers per station) that maximizes the satisfied charging demand (POI coverage and local charging demand). We show the EVCP problem is NP-hard, and design a charger-based greedy solution with theoretical guarantees. We further propose some speedup techniques. Extensive experiments on real-world datasets from EV-sharing platforms validate the effectiveness and efficiency of our solutions. We envision our work as a practical reference for EV-sharing platforms to optimize their planning on their private charging infrastructure.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive comments on this work. This work is partially supported by the National Science Foundation of China (NSFC) under Grant No. 61502021 and 71531001, National Grand Fundamental Research 973 Program of China under Grant 2014CB340300, the Base Construction and Training Program Foundation for the Talents of Beijing under Grant No. Z171100003217092, and the Science and Technology Major Project of Beijing under Grant No. Z171100005117001.

REFERENCES

[1] Charles Botsford and Adam Szczepanek. 2009. Fast Charging vs. Slow Charging: Pros and cons for the New Age of Electric Vehicles. In *International Battery Hybrid Fuel Cell Electric Vehicle Symposium*.
[2] T Donna Chen, Kara M Kockelman, William J Murray, and Moby Khan. 2013. The electric vehicle charging station location problem: a parking-based assignment method for Seattle. In *Transportation Research Board 92nd Annual Meeting*, Vol. 340. 13–1254.
[3] Reza Zanjirani Farahani and Masoud Hekmatfar. 2009. *Facility location: concepts, models, algorithms and case studies.* Springer.
[4] Inês Frade, Anabela Ribeiro, Gonçalo Gonçalves, and António Antunes. 2011. Optimal location of charging stations for electric vehicles in a neighborhood in Lisbon, Portugal. *Transportation research record: journal of the transportation research board* 2252 (2011), 91–98.
[5] Yajing Gao and Yandong Guo. 2013. Optimal Planning of Charging Station for Phased Electric Vehicle. *Energy & Power Engineering* 05, 4 (2013), 1393–1397.
[6] Fang He, Di Wu, Yafeng Yin, and Yongpei Guan. 2013. Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological* 47 (2013), 87–101.
[7] Dorit S Hochbaum. 1996. *Approximation algorithms for NP-hard problems.* PWS Publishing Company.
[8] Long Jia, Zechun Hu, Yonghua Song, and Zhuowei Luo. 2012. Optimal siting and sizing of electric vehicle charging stations. In *IEEE International Electric Vehicle Conference, IEVC 2012.* IEEE, 1–6.
[9] Albert Y. S. Lam, Yiu-Wing Leung, and Xiaowen Chu. 2014. Electric Vehicle Charging Station Placement: Formulation, Complexity, and Solutions. *IEEE Transactions on Smart Grid* 5, 6 (2014), 2846–2856.
[10] 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 *31st IEEE International Conference on Data Engineering, ICDE 2015.* 1376–1387.
[11] Chen Liu, Ke Deng, Chaojie Li, Jianxin Li, Yanhua Li, and Jun Luo. 2016. The Optimal Distribution of Electric-Vehicle Chargers across a City. In *16th IEEE International Conference on Data Mining, ICDM 2016.* 261–270.
[12] Zhipeng Liu, Fushuan Wen, and Gerard Ledwich. 2012. Optimal Planning of Electric-Vehicle Charging Stations in Distribution Systems. *IEEE Transactions on Power Delivery* 28, 1 (2012), 102–110.
[13] Zi Fa Liu, Wei Zhang, Ji Xing, and Ke Li. 2012. Optimal Planning of charging station for electric vehicle based on particle swarm optimization. In *IEEE Innovative Smart Grid Technologies-Asia, ISGT Asia 2012.* 1–5.
[14] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming* 14, 1 (1978), 265–294.
[15] Morgan Stanley Blue Papers. 2016. Shared Mobility on the Road of the Future. https://goo.gl/onaZn1. (2016).
[16] Alexander Schrijver. 1998. *Theory of linear and integer programming.* John Wiley & Sons.
[17] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands based on Large-Scale Online Platforms. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2017.* 1653–1662.
[18] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. 2016. Online mobile Micro-Task Allocation in spatial crowdsourcing. In *32nd IEEE International Conference on Data Engineering, ICDE 2016.* 49–60.
[19] Yongxin Tong, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. 2018. Flexible Online Task Assignment in Real-Time Spatial Data. In *Proceedings of the 37th ACM SIGMOD International Conference on Management of Data, SIGMOD 2018.*
[20] Yongxin Tong, Libin Wang, Zimu Zhou, Bolin Ding, Lei Chen, Jieping Ye, and Ke Xu. 2017. Flexible Online Task Assignment in Real-Time Spatial Data. *PVLDB* 10, 11 (2017), 1334–1345.
[21] Wikipedia. 2017. Autolib'. https://en.wikipedia.org/wiki/Autolib%27. (2017).
[22] James H Williams, Andrew DeBenedictis, Rebecca Ghanadan, Amber Mahone, Jack Moore, William R Morrow, Snuller Price, and Margaret S Torn. 2012. The technology path to deep greenhouse gas emissions cuts by 2050: the pivotal role of electricity. *Science* 335, 6064 (2012), 53–59.
[23] Yanhai Xiong, Jiarui Gan, Bo An, Chunyan Miao, and Ana L. C. Bazzan. 2015. Optimal Electric Vehicle Charging Station Placement. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015.* 2662–2668.