Demand Prediction and Placement Optimization for Electric Vehicle Charging Stations

Ragavendran Gopalakrishnan, Arpita Biswas, Alefiya Lightwala, Skanda Vasudevan, Partha Dutta, Abhishek Tripathi
Xerox Research Center India
Ragavendran.Gopalakrishnan@xerox.com, Arpita.Biswas@xerox.com, alefiya.lightwala90@gmail.com, skandavs@cse.iitm.ac.in, Partha.Dutta@xerox.com, abhishektripathi.at@gmail.com

Abstract

Effective placement of charging stations plays a key role in Electric Vehicle (EV) adoption. In the placement problem, given a set of candidate sites, an optimal subset needs to be selected with respect to the concerns of both (a) the charging station service provider, such as the demand at the candidate sites and the budget for deployment, and (b) the EV user, such as charging station reachability and short waiting times at the station. This work addresses these concerns, making the following three novel contributions: (i) a supervised multi-view learning framework using Canonical Correlation Analysis (CCA) for demand prediction at candidate sites, using multiple datasets such as points of interest information, traffic density, and the historical usage at existing charging stations; (ii) a mixed-packing-and-covering optimization framework that models competing concerns of the service provider and EV users; (iii) an iterative heuristic to solve these problems by alternately invoking knapsack and set cover algorithms. The performance of the demand prediction model and the placement optimization heuristic are evaluated using real world data.

1 Introduction

The environmental impact of fossil fuels and the high variability in their prices have led to rising adoption of Electric Vehicles (EVs), which is supported by ambitious government policies for promoting EVs [U. S. Department of Energy, 2012; European Union, 2014]. Despite the technological advances in vehicle efficiency and battery capacity, a key hurdle in EV adoption is that, barring a few pockets of densely populated areas, the distribution of EV charging stations is sparse in most regions.1 Because of this, EV owners and potential buyers frequently worry about whether the vehicle will have sufficient charge to travel to their trip destinations or an intermediate charging station. On the other hand, given the high cost of building a charging station and currently low2 number of EVs, charging station operators would only want to place stations where demands for charging are high. This results in a situation where the charging station density is concentrated near city centres, rapidly decreasing while moving outward.

This work is an attempt to break this deadlock by improving the distribution of new charging stations so as to mitigate the range anxiety of EV users. To be effective, such a solution must simultaneously address the concerns of both EV users and charging station operators. Charging station operators are concerned about (a) demand: charging stations must serve as much demand as possible, and (b) budget: there is a limited budget available for setting up charging stations. EV users are concerned about (c) reachability: there must be a charging station within a short driving distance from most locations, and (d) waiting time: the waiting time to begin charging at charging stations should not be prohibitively large, given that the charging event itself is time-consuming.3

This paper models the above placement problem within a framework that is inspired from the facility location literature [Drezner, 1995]. Given a set of candidate sites for charging stations, e.g., parking lots in a city, the objective is to select an optimal subset of these locations that maximizes the total demand satisfied, subject to the total cost not exceeding the available budget (packing constraint), and given a set of locations of interest, there is at least one charging station within a specified driving range of every location in this set (covering constraint). This mixed-packing-and-covering formulation (Section 3.1) covers concerns (a)-(c). For (d), a queuing model is used to translate the requirement on waiting time to a constraint on the minimum number of charging slots needed at any given location, which is then factored into the setup costs in a pre-processing step.

Before tackling the optimization problem itself, it is necessary to predict the demand for charging at candidate sites. Although demand prediction has been extensively studied for planning transportation infrastructure, the EV charging sta-


2Nevertheless, this number is rapidly growing, e.g., in the U. S., EV market share has risen [Cobb, 2014] from 0.14% in 2011 to 0.72% in 2014.

3There may be several other concerns for charging station operators and EV users, including alternate interpretations of reachability, e.g., [Funke et al., 2015], but in this work, we restrict our attention to modeling just these four.
tion demand prediction problem is difficult because of sparse deployment of such stations, and lack of sufficient and reliable historic data. Thus, in addition to the limited historical demand data for existing charging stations in other locations, we consider two other types of location features which may strongly impact the usage: (i) nearby points of interest (PoI), e.g., shopping malls, institutions, restaurants, hospitals etc., indicating the frequency of visits that could be made by travelers, and (ii) traffic density at nearby road junctions. Using these features, in Section 2, we propose a supervised multi-view learning framework using Canonical Correlation Analysis (CCA) for EV charging station demand prediction.

After estimating the demands, the next step is to solve the mixed packing and covering problem. We propose a family of heuristics\(^4\) in Section 3.2 that seeks to iteratively find the optimal allocation of the available budget between satisfying the packing and the covering constraints, by alternately invoking algorithms that solve reduced knapsack and set-cover subproblems. For example, choosing the well known greedy algorithms for the knapsack and set-cover problems [Vazirani, 2001] yields one instance in this family.\(^5\) In Section 3.3, we present results from an experimental evaluation of this instance for the EV charging station placement problem using charging data from the UK. In most cases, and especially when budget is scarce, our heuristic achieves an improvement of 10-15\% over a “naïve” heuristic (that just solves a set-cover subproblem first and then uses the remaining budget, if any, to solve a residual knapsack subproblem), both in terms of finding feasible solutions and maximizing demand.

The modeling and heuristic frameworks above can be extended to the following alternate scenarios: (i) incremental placement of charging stations when the budget is progressively released over a period of time, and (ii) when there are multiple charging station operators, each with their respective budgets, and a government agency (subject to its own budget) seeks to optimally allocate grants to incentivize these providers to set up charging stations at selected locations (e.g., where the demand is low). We discuss these extensions in the full version [Gopalakrishnan et al., 2016].

1.1 Related Work

With the recent increase in the adoption of EVs, the charging station placement problem has received significant attention. One of the prerequisites of an effective charging station placement is the availability of estimated charging demand at candidate sites. In this context, prior literature has studied the impact of external information on charging station demand. For example, parking demand is combined with facility location problem in [Chen et al., 2013], and the effect of demographic features is studied using regression models

---

4 As we report later in Section 3.3, solving the integer linear program (ILP) exactly using [CPLEX, 2009] takes more than a day on average, which may be prohibitive even for a planning problem, since a designer would likely want to solve numerous instances of the ILP with different parameters during the planning process.

5 Since the primary mixed packing and covering problem, and the associated knapsack and set-cover problems are NP-hard, it is unlikely that any polynomial-time solution, including this instance, would be optimal.
Multi-View Regression Using CCA: We present a multi-view learning based regression framework for the charging demand prediction, which uses CCA in a supervised setting as a multivariate regression tool. CCA is, in fact, closely related to multivariate linear regression; [Borga, 2001; Sun et al., 2011] present least-square formulations. CCA-based approaches model the statistical dependence between two or more data sets and assume a single latent factor to describe information shared between all datasets. In a supervised setting such as multivariate regression task, the output variable is considered as one of the random variables and CCA models the statistical dependence between the output variable (dependent) and a covariate (independent variable). However, when there are more than two datasets (output variable and multiple covariates), modeling shared information may not always be a good choice, since it would discard dataset-specific information which may be crucial to predicting the output variable.

To address this deficiency, a multi-view learning regression framework is proposed when there are \( n > 2 \) covariates, all of which can be used to predict the output variable. The idea is to model the statistical dependence of the output variable with each of the covariates separately and then learn a weighted combination for each such model that maximizes the prediction performance. We call this model Multiple Dependent Regression (MDR), which can be formulated as the ensemble regression function

\[
\hat{y} = \sum_{i} w_i \hat{y}_i,
\]

where \( \sum_i w_i = 1 \). Here, \( \hat{y}_i \) is a regression function learned from covariate \( X_i \) to the output variable \( Y \), and \( w_i \) are the weights denoting the importance of covariate \( X_i \) in the regression task. The regression function \( \hat{y}_i \) for covariate \( X_i \) can be learnt according to a CCA-based regression, given by

\[
\begin{align*}
\theta_{i,j} &= \min_{U_x, U_y} \| X_i U_x - Y U_y \|_2^2, \\
\hat{y}_i &= U_y^T \Theta_i U_x^T, \\
\end{align*}
\]

where \( U_x \) and \( U_y \) are \( k \)-dimensional linear projections for each (covariate, output variable) pair \( (X_i, Y) \). Hence, MDR is a two step framework of multiple regression:

1. For each covariate \( X_i \), apply CCA for \( X_i \) and \( Y \), and compute the prediction error \( \epsilon_i \).

2. For each \( i \), compute \( \epsilon_i = \frac{1}{k-1} \left(1 - \frac{\epsilon_i}{\sum_j \epsilon_j}\right) \).

The values of parameters \( U_x, U_y, w_i \) are learnt from the training data as described in the steps above. The model thus trained can be used to predict the output variable given the input covariates. In our case, the task is to predict the demand of EV charging at a new candidate location given the covariates as explained earlier.

MDR is analogous to ensemble learning approaches where multiple learning models are trained to solve a given problem. In our case, we train multiple CCA-based regression models in the first step and learn a weighted combination of models based on training error in the second step. The weights \( w_i \) explain the importance of each covariate in the prediction task. This procedure is flexible—any CCA-based solution can be used in Step (1). In this work, we use a Bayesian solution with group-wise sparsity proposed in [Klima et al., 2013].

Experimental Results for Demand Prediction: MDR is evaluated on EV charging data obtained from 252 public charging points in North East England through UK’s Plugged-In-Places program [Office for Low Emission Vehicles, 2013]. The location of charging points was obtained from UK’s National Charge Point Registry data [Office for Low Emission Vehicles, 2012]. Pol information is extracted from OpenStreetMap [Haklay and Weber, 2008] API for 11 categories: sustenance, education, transportation, financial, healthcare, entertainment, sports, gardens, places of worship, shops and public buildings. Finally, we use traffic data for each junction-to-junction link on major road networks, provided by [Department for Transport, 2013].

The following data matrices are created: (i) \( Y \in \mathbb{R}^{252 \times 24} \) represents hour-wise charging demand for each of 252 charging points, where \( Y[i,j] \) is the average energy consumed at charging point \( i \) in hour \( j \); (ii) \( X_1 \in \mathbb{R}^{252 \times 11} \) represents PoIs, where \( X_1[i,j] \) is the frequency of Pol category \( j \) within a radius of 500 meters around charging point \( i \); (iii) \( X_2 \in \mathbb{R}^{252 \times 5} \) represents traffic densities at 5 nearest traffic junctions to each charging point; and (iv) \( X_3 \in \mathbb{R}^{252 \times 5} \) represents charging demands at 5 nearest charging points to each charging point. Since charging data is sparse during early mornings and late nights, only data from 07:00 hr. till 23:00 hrs. is used.

MDR is evaluated in a leave-one-out cross-validation manner by training on all but one instance (charging point) and testing on the left-out instance. Prediction performance is measured as the average Root Mean Square Error (RMSE) over all test instances. MDR is compared against (i) Linear Regression (LR) - all covariates are concatenated feature-wise similar to [Wagner et al., 2014], (ii) Multiple Linear Regression (MLR) - weighted sum of multiple LR analogous to MDR, and (iii) Bayesian CCA (bCCA) - group factor analysis [Virtanen et al., 2012]. Figure 1 shows the average error for each hour of the day over all charging stations; MDR clearly outperforms the other three baselines giving an overall reduction in error by 27% compared to simple LR, 21% compared to multiple LR and 18% compared to bCCA.

3 Optimizing Placement of Charging Stations

In this section, we model the charging station placement problem as a mixed packing and covering optimization problem, and present a family of heuristics to solve it.

3.1 Preliminaries

Let \( \mathcal{L} = \{1, \ldots, |\mathcal{L}|\} \) denote the set of all candidate sites for placing a charging station. Let \( r \in \mathbb{R}_+ \) denote the desired reachability radius, that is, the maximum distance to be travelled in order to reach a charging station. Let \( \mathcal{I} \) denote the set
of all locations of interest that are desired to be covered, that is, lie within the reachability radius from at least one charging station. Let $B$ denote the total budget available.

For each candidate site $i \in \mathcal{L}$, $x_i \in \{0, 1\}$, $d_i \in \mathbb{R}_+$, and $c_i \in \mathbb{R}_+$ denote, respectively, whether a charging station is placed, the demand, and the cost of setting up a charging station at $i$. $S'_i \subseteq \mathcal{I}$ is the cover set of $i$, that is, the set of locations of interest that would be covered (within a driving distance of $r$) if a charging station is placed at $i$.

Optimizing for Demand: For a given reachability radius $r$, set of candidate sites $\mathcal{L}$, and set of locations of interest $\mathcal{I}$, the cover sets $\{S'_i\}_{i \in \mathcal{L}}$ can be precomputed. Given predicted demand and charging duration, a queuing model [Kleinrock, 1975] can estimate the number of charging slots required at each candidate site to meet a given constraint on waiting times, which is then used to precompute the costs $\{c_i\}_{i \in \mathcal{L}}$.

Then, the optimization problem to be solved is the Mixed Pack & Cover (MPC) problem shown in Figure 2a.\(^6\)

**Approximation Algorithm for a Transformed Problem:**

MPC (Figure 2a) is NP-hard as it contains as a special case, the set-cover problem [Vazirani, 2001]. For a closely related pure covering problem called Demand & Set-Cover (DSC, Figure 2e), an approximation algorithm is described next.

DSC has as subproblems, two well-known NP-hard problems, Set-Cover (SC) [Vazirani, 2001] and Minimization Knapsack (MinKP) [Csisrik et al., 1991], which are given in Figures 2c and 2d, respectively. Consider, an $f_{sc}$ factor approximation algorithm $A_{sc}$ for set cover and an $f_{kp}$ factor approximation algorithm $A_{kp}$ for minimization knapsack. Let $\mathcal{L}_{sc}$ and $\mathcal{L}_{kp}$ (subsets of $\mathcal{L}$) be the solutions returned by $A_{sc}$ and $A_{kp}$, respectively. Consider algorithm $A_{dsc}$ for DSC that uses $A_{sc}$ and $A_{kp}$ as subroutines, and returns the union of the solutions returned by the two subroutines, that is, $\mathcal{L}_{sc} \cup \mathcal{L}_{kp}$, as a solution for DSC. Lemma 1 (proof deferred to the full version [Gopalakrishnan et al., 2016]) establishes that $A_{dsc}$ is an $(f_{sc} + f_{kp})$ factor approximation algorithm for DSC.

**Lemma 1** $A_{dsc}$ returns a feasible solution for DSC whose cost is at most $(f_{sc} + f_{kp})$ times the optimal cost.

The heuristic framework for MPC presented next is inspired by $A_{dsc}$ and the observation that, similar to DSC, a solution for MPC can be obtained by combining set cover and knapsack algorithms.

### 3.2 Solving the Mixed Pack & Cover Problem

Our goal is to introduce a general methodology for algorithms to solve MPC, which is NP-hard, by breaking it down into the well-known Knapsack (KP) and the Set-Cover (SC) problems (see Figure 2). Let, $\text{KP}(\mathcal{L}, \{d_i\}_{i \in \mathcal{L}}, \{c_i\}_{i \in \mathcal{L}}, B)$ and $\text{SC}(\mathcal{L}, \mathcal{I}, \{c_i\}_{i \in \mathcal{L}}, \{S_i\}_{i \in \mathcal{L}})$ be any pair of algorithms that solve KP and SC, respectively. $\text{KP}(\cdot)$ (respectively, $\text{SC}(\cdot)$) need not be optimal, but it must be nontrivial in the sense that its solution must be maximal (respectively, minimal). That is, it should not be possible to add (remove) an item and still satisfy the packing (covering) constraint(s).

**The Iterative Pack And Cover (IPAC) Framework:** The available budget could be used very differently if the problem had been a pure packing problem (where maximizing the demand is the only concern) or a pure covering problem (where satisfying the covering constraints is the only concern). Thus, a good solution to the mixed packing and covering problem should achieve an appropriate balance by dividing the available budget between these two concerns. The core idea behind the Iterative Pack And Cover (IPAC) framework is to iteratively search for such an optimal split.

In each iteration, the budget $B$ can be thought of as comprising three parts: $B^c$, $B^p$, and some excess (due to the integrality constraint). Here, $B^c$ is the portion of the budget used by a solution to a covering problem, and $B^p$ is the portion of the budget used by a solution to a packing problem when constrained by a reduced budget of $B - B^c$. Starting with $B^c = 0$ (pure packing) and the corresponding solution from $\text{KP}(\cdot)$, during each iteration, $B^c$ is increased until a “covering check” (performed using $\text{SC}(\cdot)$) to determine whether the remaining budget $B - B^p$ is sufficient to satisfy the covering constraints left unsatisfied) passes, at which point the solutions of the packing and covering problems obtained in the last iteration are merged.\(^7\) The resulting solution is guaranteed to be a feasible solution to the MPC problem (Figure 2a). Optionally, $\text{KP}(\cdot)$ can be invoked one final time to use up any remaining portion of the budget. In the worst case, the iterations continue until $B^c = B$, at which point it becomes a pure covering problem, and if the checking check still fails, then IPAC fails to find a feasible solution to MPC. (But if this happens, it cannot necessarily be concluded that MPC is

---

\(^6\)The total demand satisfied by installing charging stations at a subset of candidate sites may be less than the sum of the predicted demands at those sites due to overlapping reachability regions. We ignore this in order to keep the optimization problem simple.

\(^7\)As an extension, note that $B^c$ need not necessarily increase in each iteration, as long as it can be shown that the procedure will terminate in finite time.
Algorithm 1 IPAC

1: \textbf{Input:} $L, I, \{d_i\}_{i\in L}, \{c_i\}_{i\in L}, B, \{S_i\}_{i\in I}, \text{KP}(), \text{SC}(), \text{RANK}()$
2: \textbf{Output:} $\{x_i\}_{i\in L}$
3: $\{x_i\}_{i\in L} \leftarrow \text{KP}(L, \{d_i\}_{i\in L}, \{c_i\}_{i\in L}, B)$;
4: $L^{P} = \{i \in L : x_i = 1\};$ \quad $B^P = \sum_{i \in L} c_i x_i$;
5: $T^P = \bigcup_{i \in L^P} S_i$;
6: $I^P \leftarrow \text{elements of } I \text{ covered by } L^P$;
7: $\{\bar{x}_i\}_{i \in L^P, \{\bar{c}_i\}_{i \in L^P}, \{S_i\}_{i \in L} \in I^P};$
8: $B^P = \sum_{i \in L^P} c_i \bar{x}_i;\quad B^\text{free} \leftarrow B - B^P$;
9: while ($B^P > B^\text{free}$ and $B^P \leq B$) do
10: \quad $\rho \leftarrow \text{RANK}(L^{P}, \{d_i\}_{i \in L^{P}}, \{c_i\}_{i \in L^{P}}, \{S_i\}_{i \in L^{P}})$;
11: \quad $\text{Remove least important items}$;
12: \quad $j \leftarrow 0$;
13: \quad while $B^\text{free} < B^P$ do
14: \quad \quad $j \leftarrow j + 1;\quad x_{j+1} \leftarrow 0;\quad B^\text{free} \leftarrow B^\text{free} + c_{j+1}$;
16: \quad \quad $\text{Recompute quantities}$;
17: \quad $L^{P} = \{i \in L : x_i = 1\};\quad B^P = \sum_{i \in L} c_i x_i$;
18: \quad $\{y_i\}_{i \in L^P} \in I^P \leftarrow \text{SC}(L^{P}, T^P, \{c_i\}_{i \in L^{P}}, \{S_i\}_{i \in L^{P}})$;
19: \quad $B^P = \sum_{i \in L} c_i \bar{x}_i$;
20: \quad if $B^P > B$ then EXIT;
21: \quad \quad $\text{Add required items to knapsack}$;
22: \quad \quad $\text{for } i \in L \text{ do } x_i \leftarrow x_i + \bar{x}_i$;
23: \quad \quad $\text{Add more items if possible}$;
24: \quad $L^{P} = \{i \in L : x_i = 1\};\quad B^P = \sum_{i \in L} c_i x_i$;
25: \quad $\{x_i\}_{i \in L} \in I \leftarrow \text{KP}(L, \{d_i\}_{i \in L}, \{c_i\}_{i \in L}, \{S_i\}_{i \in L}, B - B^P)$;

infeasible, unless $\text{SC}()$ is optimal, which is unlikely in polynomial time since the set-cover problem is NP-hard. See the full version [Gopalakrishnan et al., 2016] for an expanded description of IPAC with an illustration.

The above framework is quite generic and encompasses a large class of algorithms. A particular instantiation of the framework is described in Algorithm 1.

**The RANK() Function:** The effectiveness of IPAC depends on the choice of methods for $\text{KP}()$, $\text{SC}()$, and $\text{RANK}()$. There are several choices in the literature for the first two [Vazirani, 2001], so we briefly discuss one possible choice for the third. A general observation is that the proposed $i \in L$ is more important or desirable if its demand $d_i$ is high, and/or its cost $c_i$ is low, and/or if the number of elements it covers $|S_i|$ is high. Based on this, a viable candidate for $\text{RANK}(L, \{d_i\}_{i \in L}, \{c_i\}_{i \in L}, \{S_i\}_{i \in L})$ would be a method that ranks elements in increasing order according to the value $v_i = \left(\frac{d_i}{c_i} + \frac{1}{|S_i|}\right) / c_i$, where $I = \bigcup_{i \in L} S_i$ denotes the set of all elements covered by items in $L$.

**Computing Charging Station Costs:** We estimate the setup costs $c_i$, based on (i) the number of charging slots necessary at $i$ to satisfy any desired service level agreement (SLA) on waiting times, and (ii) per-slot setup cost, including infrastructure as well as land/licensing costs. Data on the per-unit costs are generally available; so we focus on (i), for which we model candidate charging stations as a multi-server queue.

One possible way to model each candidate charging station at location $i$ in order to estimate its size (and hence cost) is using a multi-server queue that follows an M/MI/Ni discipline, where $N_i \in \mathbb{Z}_+$ is the number of charging slots to be set up. Customers arrive into the queue according to a Poisson process with rate $\lambda_i$ per time unit. The service time for each customer is exponentially distributed with mean $1/\mu_i$ time units. $\lambda_i$ can be derived from the demand $d_i$, whereas $\mu_i$ can be derived from existing data on the average customer service time in nearby existing charging stations. The average time a customer waits before service is then given by $E[W] = \text{Er1C}(N, \lambda_i, \mu_i)$, where $\text{Er1C}(N, \rho) = \left(\frac{e^N N - N \rho}{N - N \rho}\right)$ [Kleinrock, 1975].

Suppose the SLA to be met is given by $E[W] \leq t_i$. Then, because it is well-known [Kleinrock, 1975] that $E[W]$ is a decreasing function of $N_i$, we simply choose the smallest $N_i$ (e.g., using binary search) such that the SLA is satisfied.

### 3.3 Experimental Results

We conduct experiments to evaluate the performance of IPAC using data from North East England. We use parking locations in North East England for both the candidate sites for placing charging stations, as well as the locations of interest that need to be covered. Similarly, the cover set $S_i$ for each candidate site $i$ consists of the candidate sites that are within a distance of $r$ from $i$. The charging demands at candidate sites are predicted using the MDR model trained in Section 2. The costs are computed as $c_i = N_i(L_i + F_i)$; we estimate each of these components as follows:

1. $N_i$, the number of charging spots: We assumed a Level 2 charging rate of 6.4kW and set, for each candidate site, $N_i$ to be the minimum number of Level 2 charging spots necessary (using a queuing model) to ensure that the average peak-demand waiting time (taken as the estimated maximum hourly demand at the candidate site over two years) is less than 5 minutes.

2. $L_i$, the per-spot land cost: We consider realistic land cost, based on the Pol types within 1 km of the candidate site and the minimum per-unit land cost in the region (computed using data obtained from [Valuation Office Agency, 2011]).

3. $F_i$, the per-spot infrastructure cost: For Level 2 charging, we set $F_i = $1852 from Table 6 of [Chang et al., 2012].

The predicted demands and estimated costs of the candidate sites are then used to find the optimal locations using IPAC (Algorithm 1), where, for both $\text{KP}()$ and $\text{SC}()$, we choose the well-known greedy approximation algorithms introduced in [Vazirani, 2001], and for $\text{RANK}()$, we use the function proposed in Section 3.2.

**Performance:** To evaluate the performance of IPAC, we compare it with (i) a naive heuristic that first solves the covering problem using $\text{SC}()$, and then invokes $\text{KP}()$ on the remaining budget to add unselected candidate sites, (ii) the optimal ILP solution of the MPC problem (Figure 2a), and (iii) a solution to the LP relaxation of MPC problem, which gives an upper bound on the actual ILP optimization problem. We use a budget of $B = $80M. The number of pairwise distances between candidate sites is prohibitively large for our experiment; so, we used values of $\rho$ (in units of km), from the set $\{1, \ldots, 40\}$. The results are plotted in Figures 3a-3b.

As we increase the reachability radius $r$, the set of allocating covering constraints steadily gets larger, until, for a large enough radius, any allocation would satisfy the covering constraints, reducing the optimization to a pure
packing problem. Since the feasibility set only gets larger with \( r \), the demand covered should also increase accordingly. The graphs validate this expected behavior. In addition, we observe that the demand captured by IPAC is far closer to that of the ILP solution than that captured by the naive heuristic. In particular, from Figure 3b, we can see that when \( r = 6 \) km, the IPAC heuristic already captures almost 90\% of the LP relaxation demand and its performance is always better than the naive heuristic. Also, the time taken to obtain the ILP solution for a given radius using \([\text{CPLEX, 2009}]\) ranges from 23 – 30 hours for most instances, whereas IPAC finished in less than 0.5 seconds.

Feasibility: Since IPAC is a heuristic, it may falsely deem an instance of the MPC problem (Figure 2a) infeasible, when in fact, feasible solutions exist. An instance of MPC is feasible if and only if the available budget \( B \) is greater than or equal to the minimum required budget determined by solving the corresponding instance of the set-cover subproblem (Figure 2c). In the case of IPAC, it reduces to whether the available budget \( B \) is greater than or equal to the minimum required budget as determined by \( SC() \).

Therefore, given an instance of MPC, we look at the corresponding set-cover subproblem and compare its solutions (the minimum required budgets) as obtained by (i) using the greedy approximation algorithm in [Vazirani, 2001], and (ii) directly solving its relaxed LP. We use different values of \( r \) (in units of km) to generate instances. The results are plotted in Figure 3c. (Since the LP relaxation allows for fractional allocations, the corresponding solution is only a lower bound on the minimum budget required for feasibility.)

**Accounting for Noisy Costs:** Since our model for estimating the costs may not be accurate, we perform experiments for two additional scenarios, where we add zero-mean gaussian noise to the costs — one with a standard deviation of $1000 and another with a standard deviation of $3000. In each case, we compare IPAC and the naive heuristic:

**Performance:** Using a budget of $6M, we calculate the difference in the demands satisfied by the solution obtained using IPAC and the naive heuristic. Figure 3d shows this difference as a percentage, for several values of the reachability radius. IPAC’s advantage is significant (8-18\% in most cases).

**Feasibility:** We calculate the minimum reachability radius for which IPAC and the naive heuristic are able to find a feasible solution to MPC. IPAC is always able to find feasible solutions for smaller radii. Figure 3e shows this difference as a percentage, for several values of the available budget. IPAC’s advantage over the naive heuristic is particularly remarkable (10-15\% in most cases) when the budget is scarce.

**Summary:** Experimental results show that IPAC’s feasibility gap as compared to the LP-Relaxation is not significant, and IPAC’s performance quickly approaches that of the optimal demand. Further, in terms of both feasibility and performance, it can be seen that the advantage of IPAC over the naive heuristic is significant (10-15\% in most cases) when the budget is scarce.

Further analysis of the family of IPAC algorithms would be of interest, in particular, exploring provable performance guarantees of an IPAC algorithm, e.g., Algorithm 1, in terms of those of its constituent knapsack and set-cover algo-
rithms. Another interesting direction would be to integrate the demand prediction model into the placement optimization framework, since the placement of a charging station at one site would likely affect demands at nearby sites.

References


