15th AIMMS-MOPTA

Optimization Modeling Competition

Team ChargED

# Optimal placement of electric vehicle charging stations in PA

*Author:*

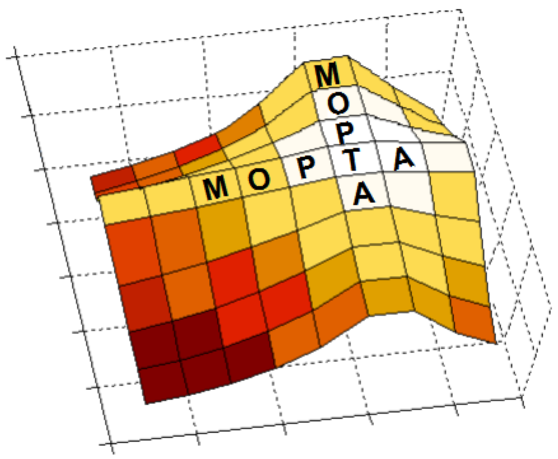
Heidi Wolles L.

Kilian Wolff

Scott Jenkins

*Advisor:*

Dr. Kit Daniel Searle



June 1, 2023

# Contents

## 1 Introduction

Electric vehicles (EVs) are becoming a key incentive to reduce emissions in the transportation industry, and hence contribute to the green transition on a national scale. The Biden administration's efforts to cut emissions and tackle climate change include a shift towards greener transportation options. The Federal Highway Administration Designated Alternative Fuel Corridor is currently the largest U.S. investment into the EV charging sector with a national network of half a million charging stations (CSs) [1]. Encouraging more people to switch to electric vehicles will, in the long run, reduce pollution and minimize the impact of fluctuating fuel prices. Although most EVs are charged at home, having a reliable network of public CSs will diversify the use of EVs and minimize car owners' chances of being out of range of a charger, hence relieving first-time buyers' anxiety over choosing an EV [8, 19].

The Pennsylvania Department of Transportation and the Department of Environmental protection are also partaking in the systemic change through initiatives like the Alternative Fuels Incentive Grant and Driving PA Forward [8, 21]. The latter specifically prioritizes projects which (i) consider strategic locations, (ii) contribute to existing or planned fueling networks, (iii) provide expected usage levels, and (iv) are cost-effective. Along with uncertain driving ranges, user behaviours and a large number of potential locations, these conditions highlight the necessity of a mathematical framework for EV CS system design and operations.

The problem proposed for the 15th AIMMS-MOPTA Optimization Modeling Competition is to identify CS locations in Pennsylvania and determine the number of CSs at each location to minimize total cost. The EV charging demand is stochastic in nature, and is given as 1 079 locations, called demand points, each with 10 EVs. These demand points are stationary and are not assumed to have additional descriptive data regarding typical length of journeys nor to have a time-related charging pattern. The challenge is divided into two phases; firstly, the location of 600 CSs must be determined to obtain the minimum cost charging infrastructure. Secondly, the *number* of CSs and corresponding locations which minimize the cost of establishing the charging infrastructure are to be determined.

The remainder of this report is structured as follows: Relevant models for EV charging infrastructure and mathematical frameworks to tackle such facility location problems are briefly reviewed in §2. The optimization modelling approach adopted by Team ChargED together with important modelling assumptions is presented in §3. Thereafter, the tailored solution approach for solving the aforementioned model is discussed in §4 which, in §5, is followed by an ensuing analysis of the performance measures obtained from the solution. A sensitivity analysis of model results are discussed in §6, illustrating how certain model parameters impact solution performance. Lastly, we tie this back into our model assumptions and discuss possible further extensions to the model in §6.1.

## 2 Literature review

The design of EV CS infrastructure falls into an increasingly important category of research into the green transition and sustainable operations research. Many optimization models with similar motivations in terms of design and optimization of EV charging networks have been presented in the literature. Typical approaches range from standard facility location models; p-median problems and maximal covering problems [10]; to simulation-optimization models [14, 24], including agent-based modelling [25], and data-driven multi-criteria methods [11]. Three main considerations are the choice of objective (and distance measure), the standard trade-off between a discrete vs. continuous formulation, and the more difficult case of time-/space-varying, potentially stochastic, demand [18].

Public EV service region design (districting) is closely related to optimization of EV sharing systems and battery swapping. He et al. [12, 13] incorporated user decisions based on EV energy levels and formulated a nonlinear program for a queuing-location model for sharing systems which can be approximated as a mixed-integer second order cone program. Mak et al. [20] proposed a robust optimization model for the case of infrastructure accommodating EV battery swapping. The transportational aspect of planning EV CSs also highlights a close connection to routing problems which are becoming increasingly popular as a primary part of the optimization model [15] or as a hybrid [16]. Stakeholders are often incentivized to construct CS ensembles within existing infrastructure, e.g. along major national highways with programs like the Alternative Fuel Corridors [1].

We refer the interested reader to [3, 7, 9, 18] for the more general case of continuous facility location problems, and [22, 23] for facility location problems under uncertainty. Sections §3-4 further elaborate on the relevancy of the modelling approach adopted by TeamED with regards to the previously established literature.

## 3 Optimization model

The problem of determining the coordinates of facilities in the two-dimensional Euclidean plane which minimize the distance (or some other cost-metric) from another set of known locations (or demand points) is widely known in the literature as the Weber problem [4, 5, 6]. In this section we formulate a suitable continuous location-allocation model in an attempt to solve the problem described in §1. This takes the form of a generalized Weber problem which is non-convex and notoriously difficult to solve. Therefore, in §4 we propose a suitable discretization procedure to find high quality solutions in a suitable time at the sacrifice of a proof of optimality.

Let $\mathcal{I}$ denote the set of EVs and let $\mathcal{J}$ denote the set of potential charging locations (PCLs) for CSs in the $290 \times 150$ mile area in Pennsylvania. For now we assume that $\mathcal{J}$ is sufficiently large. Given the stochastic nature of the problem, we employ a robust optimisation framework and consider a finite set $S$ of scenarios, such that $s \in \mathcal{S}$ represents one scenario.

### 3.1 Decision variables

Let the tuple of decision variables $\boldsymbol{x} = (x, y) \in \mathbb{R}^{2\times\mathcal{J}}$ denote the geographical position of CSs, such that $\boldsymbol{x}_j$ is the spatial coordinates of a CS $j \in \mathcal{J}$. Hence let auxiliary variable $\tilde{d}_{i,j} = ||\boldsymbol{x}_j - \boldsymbol{a}_i||$ denote the distance from a EV $i \in \mathcal{I}$ to a PCL $j \in \mathcal{J}$, where the parameter $\boldsymbol{a}_i \in \mathbb{R}^2$ is a tuple of spatial coordinates of EV $i \in \mathcal{I}$, and $||\cdot||$ is the standard 2-norm[1] in $\mathbb{R}^2$ (Euclidean distance).

Moreover, EVs are allocated to CSs through decision variables, $u \in \{0,1\}^{\mathcal{I}\times\mathcal{J}\times\mathcal{S}}$, such that

$$u^s_{i,j} = \begin{cases} 1 & \text{if EV } i \in \mathcal{I} \text{ is assigned to CS } j \in \mathcal{J} \text{ in } s \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$$

Allocation in this case corresponds to the assumption that each EV $i \in \mathcal{I}$ charges at its assigned CS, *i.e.* that there is some "authority" assigning EVs to CSs. Hence EV $i \in \mathcal{I}$ can be "assigned" to CS $j \in \mathcal{J}$, which will be in the setup with overall lowest cost for all stakeholders.

Since we will not build CSs at every PCL, we introduce the decision variables $v \in \{0,1\}^{\mathcal{J}}$ such that

$$v_j = \begin{cases} 1 & \text{if a CS is built at PCL } j \in \mathcal{J}, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the integer decision variables $w \in \mathbb{Z}^{\mathcal{J}}_+$ to model the number of chargers to install at each location, such that $w_j$ is the number of chargers to install at CS $j \in \mathcal{J}$.

### 3.2 Constraints

The number of chargers that can be built at each CS is bounded from above by some $m \in \mathbb{Z}_+$. It is given in the problem description that $m = 8$. Therefore, if a CS is built, it must contain between 1 and 8 chargers, which we enforce with the constraints:

$$v_j \leq w_j \leq m v_j \quad \forall j \in \mathcal{J}. \tag{1}$$

Also from the problem description, it is given that at most one EV can queue for each charger, so that the number of EVs which are allocated to each CS in each scenario may not exceed two times the number of chargers at that CS. We enforce this with the constraints,

$$\sum_{i\in\mathcal{I}} u^s_{i,j} \leq 2w_j \quad \forall j \in \mathcal{J},\ s \in \mathcal{S}. \tag{2}$$

However, we would rarely expect all EV owners to visit the same allocated CS at the same time. This constraint assumes a worst-case scenario where all assigned EVs need to use the CS at any given time. Realistically, more EVs can be assigned to a CS than we expect to see in queue for that CS simultaneously. Note that this constraint set also ensures that for any scenario $s \in \mathcal{S}$ an EVs cannot be assigned to CSs which are not built, i.e. if there exists

[1]The 2-norm in in $\mathbb{R}^2$ is $\tilde{d}_{i,j} = \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$, where $\mathbf{a}_i = (a_i, b_i)$.

an $j^* \in \mathcal{J}$ such that $v_{j^*} = w_{j^*} = 0$ then $u^s_{i,j^*} = 0 \; \forall i \in \mathcal{I}, s \in \mathcal{S}$. Similarly, each EV can be assigned to at most one CS in any scenario by the constraints,

$$\sum_{j \in \mathcal{J}} u^s_{i,j} \leq 1 \quad \forall i \in \mathcal{I}, \; s \in \mathcal{S}. \tag{3}$$

We introduce the notion of service level from queuing theory to signify a level of charging demand which must be fulfilled. This allows stakeholders the option to choose how important it is that all EVs which need to charge are granted service, and e.g. only plan for the chargers which fulfill 50% of the demand. The concept is implemented by setting a relative lower bound, $\alpha \in [0,1] \subseteq \mathbb{R}$, on the total number of allocated EVs,

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} u^s_{i,j} \geq \alpha L \quad \forall s \in \mathcal{S}, \tag{4}$$

where $L \in \mathbb{R}$ denotes the total number of EVs which are able to reach any CS. By default, we choose $\alpha = 0.95$. Note that this constraint heavily affects the "feasibility" of the model.

Let $r^s_i$ denote the range of EV $i \in \mathcal{I}$ in scenario $s \in \mathcal{S}$. If, in any scenario $s \in \mathcal{S}$, EV $i \in \mathcal{I}$ is out of reach of a PCL, $j \in \mathcal{J}$, that is, there exist $i \in \mathcal{I}, j \in \mathcal{J}, s \in \mathcal{S}$ such that $d_{i,j} > r^s_i$, then EV $i \in \mathcal{I}$ cannot be allocated to that CS $j \in \mathcal{J}$ in that scenario $s \in \mathcal{S}$. This is enforced with the constraint,

$$\tilde{d}_{i,j} u^s_{i,j} \leq r^s_i \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, s \in \mathcal{S}. \tag{5}$$

Finally, we have the domain constraint on the CS coordinates,

$$\boldsymbol{x}_j \in [0, 290] \times [0, 150] \subseteq \mathbb{R}^2 \quad \forall j \in \mathcal{J} \tag{6}$$

to ensure that each CS is only built in the region under consideration.

## 3.3 Objective function

Costs associated with the problem description are given by $c_b, c_m, c_d, c_{\tilde{c}} \in \mathbb{R}$, where $c_b$ is the annualized construction investment of a CS (build cost \$/station); $c_m$ is the maintenance cost of a charger (\$/charger); $c_{\tilde{c}}$ is the cost of charging (\$/mile) which is constant up to the full range of 250 miles; $c_d$ is the cost of driving to the CS (\$/mile). In particular, we are given in the problem description that $c_b = 5000, c_m = 500, c_d = 0.041, c_{\tilde{c}} = 0.0388$.

The objective function we want to minimize is the overall annualized cost of setting up CSs in Pennsylvania,

$$\begin{aligned} f_{\text{NL}} &= \frac{365}{|\mathcal{S}|} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{s \in \mathcal{S}} \left[ \underbrace{c_d \tilde{d}_{i,j} u^s_{i,j}}_{1.} + \underbrace{c_{\tilde{c}}(250 + \tilde{d}_{i,j} - r^s_i) u^s_{i,j}}_{2.} + \underbrace{c_{\tilde{c}}(250 - r^s_i)(1 - u^s_{i,j})}_{3.} \right] \\ &\quad + \sum_{j \in \mathcal{J}} (\underbrace{c_b v_j}_{4.} + \underbrace{c_m w_j}_{5.}) \qquad (7) \\ &= \sum_{j \in \mathcal{J}} \left[ c_b v_j + c_m w_j + \frac{365}{|\mathcal{S}|}(c_{\tilde{c}} + c_d) \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} \tilde{d}_{i,j} u^s_{i,j} \right] + \frac{365}{|\mathcal{S}|} |\mathcal{J}| c_{\tilde{c}} \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} (250 - r^s_i), \qquad (8) \end{aligned}$$

where $|\mathcal{S}|$ is the number of scenarios. The first term in (7) models the driving costs of EVs to their assigned CS. The second term is the cost of charging to full capacity, that is, the range subtracted from the total of 250 miles and the driving distance. However, cars which are not assigned to a CS still need to charge to full capacity by some other means. Hence the third term adds the cost of charging to full capacity whenever an EV is not assigned, i.e. if there exist $i^* \in \mathcal{I}, j^* \in \mathcal{J}$ such that $u^s_{i^*,j^*} = 0$ in any scenario $s \in \mathcal{S}$. In this case, we assume the EV owner cannot make use of the CSs in our model and will need to use private chargers or have the EV towed at an extra cost.

The fourth term in (7) accounts for the construction cost of each CS, and the fifth for the maintenance cost of each charger at a CS. Since the charger maintenance cost is lower than the CS build cost, these terms incentivize utilising existing CSs over opening new ones. These costs incur on an annual basis, while driving and charging costs incur discretely for each scenario. Hence the first three terms (which sum over the scenarios) are annualized by multiplying by 365 days and dividing by the number of scenarios.

The terms have been simplified in (8), where the constant does not impact the optimization but is included for an accurate total cost. The objective function (8) is nonlinear and nonconvex. Whilst the continuous-allocation problem can be solved to optimality in a reasonable time for a small number of vehicles, the problem for 10790 vehicles poses a significant challenge to any state-of-the-art solver (AIMMS, Gurobi, Xpress, etc.). In section 4, we discuss our heuristic solution approach.

## 3.4 Data and simulation

In the given case study, the file `MOPTA2023_car_locations.csv`, contains 1079 coordinates in Pennsylvania, and we are told that 10 EVs are located at each. Hence, we begin by producing `full_car_locations.csv` by duplicating each row entry 10 times to obtain a list of 10 790 locations, one for each EV. More generally, our model can solve any dataset of vehicle locations, and does not necessitate on uniform numbers of EVs across locations. In our model interface, we allow the user to upload their own dataset of vehicle locations.

Suppose that the range $r_i^s$ of EV $i \in \mathcal{I}$ in scenario $s \in \mathcal{S}$ is normally distributed such that $r_i^s$ is a realization of $R \sim \mathrm{N}(100, 50^2)$, truncated with bounds between 20 and 250 miles. Then the probability that an EV $i \in \mathcal{I}$ requires charging in scenario $s \in \mathcal{S}$ is determined by the map, $p_i^s : [20, 250] \subseteq \mathbb{R} \mapsto [0, 1] \subseteq \mathbb{R}$, defined by $p_i^s(r_i^s) = \exp(-\lambda^2(r_i^s - 20)^2)$, where $\lambda$ is a scaling parameter. In this case, we let $\lambda = 0.012$.

A Monte Carlo type simulation is employed to determine which EVs $i \in \mathcal{I}$ require charging at a CS $j \in \mathcal{J}$ for each scenario $s \in \mathcal{S}$. To this end, for every scenario $s \in \mathcal{S}$, an EV $i \in \mathcal{I}$ requires charging, modelled by the Boolean parameter $\delta_i^s \in \{\mathit{True}, \mathit{False}\}$, if the probability of the EV needing to charge, $p_i^s$, is greater than or equal to a random uniform number, $\psi_i^s \sim \mathrm{U}(0, 1)$. The method is constructed as follows: Firstly, generate $\psi_i^s \sim \mathrm{U}(0, 1)$ for each $i \in \mathcal{I}$ and $s \in \mathcal{S}$. If $\psi_i^s \leq p_i^s$, then set the Boolean value $\delta_i^s$ to *True*, otherwise set it to *False*. Hence, each scenario $s \in \mathcal{S}$ constitutes 10 790 EV ranges and Boolean values describing whether EVs needs to charge or not. An example output for this simulation procedure is shown in Table 1. This modelling approach allows us to reduce the size of the problem, by removing the decision variables and constraints corresponding EVs which do not need to charge in a scenario $s \in \mathcal{S}$.

Table 1: Table of generated values in one outcome $s \in \mathcal{S}$ of a Monte Carlo simulation.

| EV $i$ | Range $r_i^s$ | Probability of charge $p_i^s$ | Uniform random number $\psi_i^s$ | Charging required $\delta_i^s$ |
|---|---|---|---|---|
| 1 | 98 | 0.42 | 0.50 | *False* |
| 2 | 73 | 0.67 | 0.13 | *True* |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| 10 790 | 101 | 0.39 | 0.48 | *False* |

With 10 790 samples and multiple replications, the mean probability that an EV is required to charge is observed to be approximately 0.42. Hence the expected proportion of EVs which are required to charge in each replication is 0.42. We argue that this is the weighted sum of the density function of the truncated normal distribution, denoted $\mathbb{P}(r)$, and the given probability-of-charge function, $p(\delta; r) = \exp(-\lambda^2(r - 20)^2)$, which is not normalized and hence not a density function. In the large-system limit (i.e. after a large number of replications), the sum becomes a standard Riemann integral over the continuous domain of possible ranges,

$$\begin{aligned}
\mathbb{E}[p(\delta; r)] &= \lim_{i,s \to \infty} \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} p(\delta; R = r_i^s)\mathbb{P}(R = r_i^s) &\quad (9)\\
&= \int_{20}^{250} p(\delta; r)\mathbb{P}(r)dr &\quad (10)\\
&= \int_{20}^{250} e^{-(0.012)^2(r-20)^2} \frac{e^{-\frac{1}{2}\left(\frac{r-100}{50}\right)}}{50\sqrt{2\pi}(\Phi(\frac{250-100}{50}) - (\Phi(\frac{20-100}{50}))} dr &\quad (11)\\
&\approx 0.42, &\quad (12)
\end{aligned}$$

where $\Phi$ is the cumulative distribution function of the standard normal distribution.

## 4 Solution approach

The nonlinear continuous model (1)–(8) proved difficult to solve for the full vehicle set in reasonable time, so we propose a heuristic approach, accepting that we can not guarantee a global optimum is reached.

The model can be transformed into a two-stage location-allocation problem similar to [5]. In this approach we initialize $\mathcal{J}$ with a discrete set of PCLs such that auxiliary variable $\tilde{d}_{i,j}$ becomes a parameter which can be replaced by parameters $d_{i,j}$ for all $i \in \mathcal{I}, j \in \mathcal{J}$. The variable $\boldsymbol{x}$ in the nonlinear model becomes a property of set $\mathcal{J}$. Clearly, in this case, the objective value is linear and the problem becomes a Mixed-Integer Linear Programming (MILP) model which can be solved with commercial software and any solution to this problem will be an upper bound on the global optimum of the nonlinear problem.

This approach obviously has the drawback that the quality of the upper bound depends on the choice of $\mathcal{J}$ and so we can iteratively improve the global objective function by including additional PCL which we know will result in an improved solution. The pseudo code for this procedure is outlined in Algorithm 1.

**Algorithm 1** Location-allocation algorithm

**Input:** $(P)$ : mixed integer linear program (MILP), $n$ : number of initial locations, '*type*' : random, $k$-means or $k$-means constrained, $T$ : time limit in seconds, $\epsilon$ : objective value improvement tolerance, $d_{\min}$ : smallest distance allowed between CSs, $\kappa$ : radius from CS

**Output:** $f^*, (u^*, v^*, w^*)$ : solution to (1)–(8)

```
 1: $\mathcal{J} \leftarrow$ INITIAL_LOCATIONS($n$, '$type$')
 2: $(\hat{u}, \hat{v}, \hat{w}) \leftarrow \emptyset$
 3: Improvement $\leftarrow$ True
 4: while Improvement = True do
 5:     $\hat{f},\ (\hat{u}, \hat{v}, \hat{w}) \leftarrow$ SOLVE_MILP$(P, T, \epsilon, (\hat{u}, \hat{v}, \hat{w}))$
 6:     $\mathcal{K} \leftarrow$ FIND_IMPROVED_LOCATIONS$(\hat{u}, \hat{v})$
 7:     $\mathcal{K},\ \mathcal{J}_A,\ \{\mathcal{I}_{A_j}\}_{j \in \mathcal{J}_A} \leftarrow$ FILTER_LOCATIONS$(\mathcal{J}, \mathcal{K}, d_{\min}, \kappa)$
 8:     if $|\mathcal{K}| = 0$ then
 9:         Improvement $\leftarrow$ False
10:     else
11:         $\mathcal{J} \leftarrow \mathcal{J} \cup \mathcal{K}$
12:     $(\hat{u}, \hat{v}, \hat{w}) \leftarrow$ CONSTRUCT_WARMSTART$(\mathcal{J}_A, \{\mathcal{I}_{A_j}\}_{j \in \mathcal{J}_A}, \mathcal{K}, \hat{f}, (\hat{u}, \hat{v}, \hat{w}))$
13:     $f_{\mathrm{WS}},\ (\hat{u}, \hat{v}, \hat{w}) \leftarrow$ ADD_WARMSTART$(\hat{u}, \hat{v}, \hat{w})$
14:     if $|f_{\mathrm{WS}} - \hat{f}| \leq \epsilon$ then
15:         Improvement $\leftarrow$ False
```

The algorithm takes as input the linearization of the location-allocation problem, $(P)$; a positive integer, $n$, which corresponds to the initial size of $\mathcal{J}$; real numbers, $T$, and $\epsilon$, which are the maximum allowable time and minimum objective function improvement between successive solutions in the branch-and-cut tree of the MILP solver; a real number, $d_{\min}$, denoting a soft threshold distance between CSs (greater than which new PCLs are always added, and otherwise with a certain probability); and finally a real number $\kappa$ which is a reasonable radius from a CS to other CSs and EVs (within which alternative allocations are considered). The algorithm returns a solution $(f^*,\ (u^*, v^*, w^*))$ which is an upper bound on the objective value of the problem (1)–(8).

The algorithm begins by initiating the set of PCLs, $\mathcal{J}$, by calling the function INITIAL_LOCATIONS, described in detail in §4.1, and initiating the Boolean variable Improvement as *True*. The following process is then repeated until Boolean variable Improvement becomes *False*. The function SOLVE_MILP is called, in which the linearized location-allocation problem $(P)$ is solved, described in detail in §4.2. It returns a solution, in the form of a set of locations which are used as CSs, the allocation of EVs to CSs, the number of chargers to install at each CS, and the objective function value. The function FIND_IMPROVED_LOCATIONS takes as input the locations where CSs are built and the corresponding allocations of EVs to CSs, for every scenario, and returns a set of additional PCLs, denoted by $\mathcal{K}$, which minimize the total distance between a CS and the EVs which are allocated to it across all scenarios. This procedure is described in detail in §4.3. The algorithm then calls a helper function called FILTER_LOCATIONS to remove some of the improved locations. This function is employed in an attempt to reduce the size of $\mathcal{J}$ by only adding new locations to $\mathcal{J}$ if the ratio of PCLs to EVs does not exceed a random uniform number, also described in more detail in §4.3. If there are no improved locations then the Boolean variable Improvement is switched to

*False*, and the algorithm terminates. Otherwise, the set of PCLs, $\mathcal{J}$, is updated to also include the filtered subset of $\mathcal{K}$. The function CONSTRUCT_WARMSTART is then called to build a warm start to the MILP ($P$), described in detail in §4.4. If the objective value corresponding to the warm start is not sufficient, then the Boolean variable Improvement is switched to *False* and the algorithm terminates; otherwise, the iteration is over, and we repeat the process. Algorithm 1 is illustrated schematically by means of a flow chart in Figure 1.

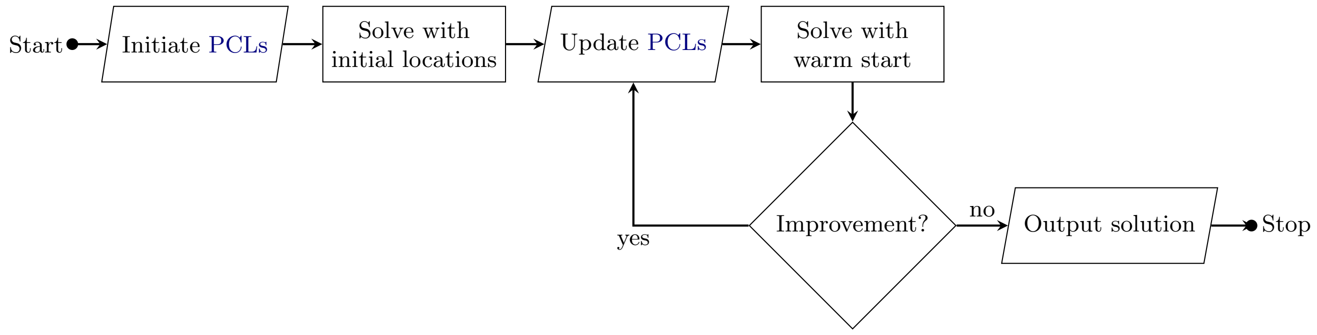
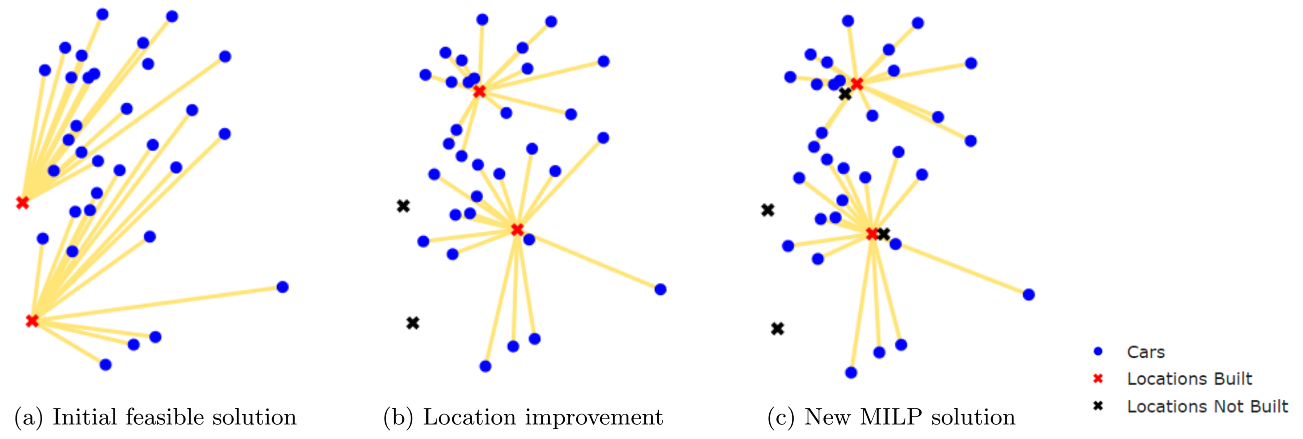


Figure 1: Flow chart of the two-stage solution to the location-allocation problem.

Next, we demonstrate the mechanics of Algorithm 1 by means of a simple example. A graphical representation of the solution returned from the function SOLVE_MILP is provided in Figure 2(a). Thereafter, the functions FIND_IMPROVED_LOCATIONS and FILTER_LOCATIONS are called and two improved locations are found an a warm start for ($P$) is constructed and, another iteration of the algorithm is initiated. The graphical representation of the solution returned from the function SOLVE_MILP is provided in Figure 2(b). In this Figure, we can see two new PCLs which where added to $\mathcal{J}$, and consequently all of the EVs were allocated to these CSs. Thereafter, another iteration of the algorithm is completed, and graphical representation of the solution returned from the function SOLVE_MILP is provided in Figure 2(c). In this case another two PCLs where added to $\mathcal{J}$, however, this time only one of the new CSs is allocated to EVs. At this point, there are no more improved locations and therefore the algorithm is terminated.



(a) Initial feasible solution (b) Location improvement (c) New MILP solution

Figure 2: Two-stage location-allocation solution approach.

## 4.1 Initial locations

In this section we provide more details on the first function in Algorithm 1, INITIAL_LOCATIONS This function takes as input a positive integer $\tilde{k}$ which is the number of initial PCLs and a string '*type*' which specifies which procedure must be employed to determine the initial PCLs. By choosing a set of good initial PCLs, we can potentially reduce the number of iterations required for Algorithm 1 to converge to a good solution. By default, we provide three options for the user to choose from (i) PCLs generated at random, or obtained with (ii) standard $k$-means clustering or (iii) $k$-means constrained clustering, as outlined in Algorithm 2. The first option generally prompts an exploratory search with the aim of avoiding premature convergence to local minima. Locations are generated from from a smaller subset the spatial domain, with corner points defined by the extremal points of the given EVs coordinates, $\boldsymbol{x}$, that is, the rectangle $[x_0, x_N] \times [y_0, y_M] \subseteq [0, 290] \times [0, 150] \subseteq \mathbb{R}^2$, with

$$x_0 = \min_{i \in \mathcal{I}}\{x_i\}, \quad x_N = \max_{i \in \mathcal{I}}\{x_i\},$$
$$y_0 = \min_{i \in \mathcal{I}}\{y_i\}, \quad y_M = \max_{i \in \mathcal{I}}\{y_i\}.$$

The second method for obtaining initial PCLs is $k$-means clustering, which partitions the set of EVs into $\tilde{k}$ clusters such that each PCL is placed at the geographical centre of each cluster. In constrained clustering, an additional constraint is set on the minimal/maximal number of vehicles assigned to each cluster.

**Algorithm 2** Initial locations

**Input:** $\tilde{k}$ : number of initial locations, '*type*' : random, $k$-means or $k$-means constrained
**Output:** $\mathcal{J}$ : initial locations

```
1: procedure INITIAL_LOCATIONS($\tilde{k}$, '$type$')
2:     if '$type$' = random then
3:         $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n \leftarrow \boldsymbol{\Pi} \sim (\mathrm{U}(0,1), \mathrm{U}(0,1))$        ▷ Generate 2-dim random uniform tuples
4:         $\mathcal{J} \leftarrow \{j = 1, \ldots, \tilde{k} : \boldsymbol{x}_j = \boldsymbol{\pi}_j(x_N - x_0, y_M - y_0) + (x_0, y_0)\}$
5:     else if '$type$' = $k$-means then
6:         $\mathcal{J} \leftarrow \{j = 1, \ldots, \tilde{k} : \boldsymbol{x}_j$ are centre points of $k$-means clusters$\}$
7:     else
8:         $\mathcal{J} \leftarrow \{j = 1, \ldots, \tilde{k} : \boldsymbol{x}_j$ are centre points of $k$-means constrained clusters$\}$
```

## 4.2 Solve MILP

In this section we explain the function SOLVE_MILP. This function takes as input the linearization of the location-allocation problem, $(P)$, real numbers, $T$, and $\epsilon$, which are the maximum allowable time and minimum objective function improvement between successive solutions in the branch-and-cut tree of the MILP solver. We begin by explaining the linearization of $(P)$. For each scenario, if an EV needs to charge, we compute the Euclidean distance from its location to each of the PCLs in the set $\mathcal{J}$, denoted by $d_{i,j}$, from EV each EV $i \in \mathcal{I}$ to each CS $j \in \mathcal{J}$. At this point, we note that many of the allocation decision variables are actually not required to solve the problem. Suppose there exist $i^* \in \mathcal{I}, j^* \in \mathcal{J}, s^* \in \mathcal{S}$ such that $d_{i^*,j^*} > r_{i^*}^{s^*}$, i.e. the EV $i^* \in \mathcal{I}$ cannot reach CS $j^* \in \mathcal{J}$ in that scenario $s^* \in \mathcal{S}$, then we set decision variable $u_{i^*,j^*}^{s^*} = 0$ and exclude it from the model. Moreover, if EV $i' \in \mathcal{I}$, in scenario $s' \in \mathcal{S}$, does not require charging, i.e. $\delta_{i'}^{s'} = \mathit{False}$, then the allocation decision variables $u_{i',j}^{s'}$ can be excluded for all $j \in \mathcal{J}$.

For the sake of convenience, we introduce the set of all EVs $i \in \mathcal{I}$, which are set to require charging in scenario $s \in \mathcal{S}$, that is, $\mathcal{I}_s = \{i \in \mathcal{I} \ : \ \delta_i^s = \mathit{True}\}$, where $s \in \mathcal{S}$ is fixed. Similarly, let $\boldsymbol{\Omega}_s = \{(i,j) \in (\mathcal{I}_s, \mathcal{J}) \ : \ d_{i,j} \leq r_i^s\}$ denote the set of feasible configurations of EVs and CSs in a fixed scenario, $s \in \mathcal{S}$. Therefore, the set of allocation decisions to be made is $\{u_{i,j}^s \in \{0,1\} \ : \ \forall \ s \in \mathcal{S}, \ (i,j) \in \boldsymbol{\Omega}_s\}$. In this case, the nonlinear objective function (8) is simplified to the linear objective function

$$f_{\mathrm{L}} = \sum_{j \in \mathcal{J}} (c_b v_j + c_m w_j) + \frac{365}{|S|}(c_{\tilde{c}} + c_d) \sum_{(i,j) \in \boldsymbol{\Omega}_s} \sum_{s \in \mathcal{S}} d_{i,j} u_{i,j}^s + C, \tag{13}$$

where $C$ is the constant from (8) with the constraint set (5) implicitly enforced. This then results in the MILP,

$$\left.\begin{array}{rlll}
\min & f_{\mathrm{L}} & & \\
\text{subject to} & v_j - w_j \le 0 & & \forall j \in \mathcal{J} \\
& w_j - m v_j \le 0 & & \forall j \in \mathcal{J} \\
& \displaystyle\sum_{(i,\cdot)\in\boldsymbol{\Omega}_s} u^s_{i,j} - 2w_j \le 0 & & \forall s \in \mathcal{S},\ j \in \mathcal{J} \\
& \displaystyle\sum_{(\cdot,j)\in\boldsymbol{\Omega}_s} u^s_{i,j} \le 1 & & \forall s \in \mathcal{S},\ i \in \mathcal{I}_s \\
& \displaystyle\sum_{(i,j)\in\boldsymbol{\Omega}_s} u^s_{i,j} \ge \alpha L & & \forall s \in \mathcal{S} \\
& u^s_{i,j} \in \{0,1\}, \quad v_j \in \{0,1\}, \quad w_j \in \mathbb{Z} & & \forall s \in \mathcal{S}, (i,j) \in \boldsymbol{\Omega}_s.
\end{array}\right\} \qquad (P)$$

This model can be solved to near-optimality for reasonable choices of $\mathcal{J}$ using any commercial solver for mixed integer programs. In our case, however, we may be required to solve a potentially difficult problem many times and therefore in each iteration of Algorithm 1 we will not always solve ($P$) to optimality. Instead, we employ an incumbent call-back for two reasons. The first is to check if there is an improvement in the objective function of at least $\epsilon$, and the second is to reset the time that is allowed between each incumbent solution. The pseudo code for the function SOLVE_MILP is given in Algorithm 3. In this algorithm, the function SOLVE denotes the call of the commercial branch-and-cut solver we utilize, and ADD_WARMSTART is a built-in function which adds a warm start to the solver, and consequently a upper bound to the optimal value of ($P$).

**Algorithm 3** Solve MILP

**Input:** ($P$) : MILP, $T$ : time limit in seconds, $\epsilon$ : objective value improvement tolerance
**Output:** $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w})$ : intermediate solution to ($P$)

```
 1: procedure SOLVE_MILP(($P$), $T$, $\epsilon$)
 2:     $f_{\mathrm{old}} \leftarrow 0$
 3:     if $(\hat{u}, \hat{v}, \hat{w}) \neq \emptyset$ then                ▷ No warm start in the first iteration
 4:         $f_{\mathrm{WS}}$, $(\hat{u}, \hat{v}, \hat{w}) \leftarrow$ ADD_WARMSTART($\hat{u}, \hat{v}, \hat{w}$)
 5:         $f_{\mathrm{old}} \leftarrow f_{\mathrm{WS}}$
 6:     while $t \le T$ do
 7:         $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w}) \leftarrow$ SOLVE($P$)
 8:         if incumbent solution found then
 9:             if $|f_{\mathrm{old}} - \hat{f}| \le \epsilon$ then
10:             else
11:                 $f_{\mathrm{old}} \leftarrow \hat{f}$
12:                 $t \leftarrow 0$
13:     return $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w})$
```

## 4.3 Find improved locations

Having solved the linear program ($P$) for a given set $\mathcal{J}$, we can further improve the locations of all the CSs in continuous space with a local search heuristic outlined in Algorithm 4. The algorithim takes as input components of the solution returned from SOLVE_MILP, namely $(\hat{u}, \hat{v})$, and returns a set of improved PCLs, denoted by $\mathcal{K}$. The algorithm begins by populating the set $\mathcal{J}_A$, which is the set of PCLs where CSs are built. Thereafter, the set of EVs allocated to each CS in $\mathcal{J}_A$ in every scenario $s \in \mathcal{S}$ which is the set $\mathcal{I}_{A_j} = \{i \in \mathcal{I}_s \ \forall s \in \mathcal{S} : u^s_{i,j} = 1\}$. The location of the active CS $j \in \mathcal{J}_A$ can then be improved by one of two ways. If the distance from $i \in \mathcal{I}_{A_j}$ is less than or equal to the geometric median of the coordinate tuples in $\mathcal{I}_{A_j}$, then we simply add the geometric median as a new coordinate PCL and avoid further computations. Note that the standard geometric median of a set of points, $\mathcal{A} = \{\boldsymbol{a}_i : i \in \mathcal{I}_{A_j}\}$ is defined as GEOMETRIC_MEDIAN($\mathcal{A}$) $= \arg\min_{\boldsymbol{g}\in\mathbb{R}^2} \sum_{i\in\mathcal{I}_{A_j}} ||\boldsymbol{a}_i - \boldsymbol{g}||$, where $||\cdot||$ is the standard 2-norm in $\mathbb{R}^2$. The function GEOMETRIC_MEDIAN returns the coordinates of the new, improved location

$\boldsymbol{g}$ corresponding to CS $j \in \mathcal{J}_A$. Otherwise, we solve the minsum problem,

$$\begin{aligned} \min_{\boldsymbol{x}_j} f_{\text{dist}} = & \sum_{i \in \mathcal{I}_{A_j}} \tilde{d}_{i,j} \\ \text{subject to} \quad & 0 \le \tilde{d}_{i,j} \le r_i^s \quad i \in \mathcal{I}_{A_j},\ s \in \mathcal{S} \\ & \tilde{d}_{i,j} = ||\boldsymbol{x}_j - \boldsymbol{a_i}||, \quad i \in \mathcal{I}_{A_j}, \end{aligned} \tag{$D_j$}$$

where the solution $\boldsymbol{x}_j^*$ is the geographical position of the new PCL that minimizes the distance to each of the EVs in $\mathcal{I}_{A_j}$. This problem can be solved using a trust region method to obtain the optimal placement of a CS when considering the sum of distances to its allocated EVs [2]. By repeating this process for every active CS, $j \in \mathcal{J}_A$, we obtain a full set of improved locations, $\mathcal{K}$, each with a corresponding geographical position.

**Algorithm 4** Location improvement heuristic

```
Input: $(\hat{u}, \hat{v})$ : solution to $(P)$ (locations and allocations)
Output: $\mathcal{K}$ : set of improved locations, $\mathcal{J}_A$ : old active locations, $\{\mathcal{I}_{A_j}\}_{j \in \mathcal{J}_A}$ : old set of allocated EVs
1: procedure FIND_IMPROVED_LOCATIONS$(\hat{u}, \hat{v})$
2:     $\mathcal{K} \leftarrow \emptyset$
3:     $\mathcal{J}_A \leftarrow \{j \in \mathcal{J} : v_j = 1\}$                         ▷ Find active CSs
4:     for $j \in \mathcal{J}_A$ do
5:         $\mathcal{I}_{A_j} \leftarrow \emptyset$
6:         $\mathcal{I}_{A_j} \leftarrow \{i \in \{\mathcal{I}_s\}_{s \in \mathcal{S}} \ : \ u_{i,j}^s = 1\}$   ▷ Find EVs allocated to CS $j$, and their ranges
7:         $\mathcal{A} \leftarrow \{\boldsymbol{a}_i : i \in \mathcal{I}_{A_j}\}$
8:         $\boldsymbol{g} \leftarrow$ GEOMETRIC_MEDIAN$(\mathcal{A})$
9:         if $||\boldsymbol{a}_i - \boldsymbol{g}|| \le r_i^s$ for all $i \in \mathcal{I}_{A_j}$ and $s \in \mathcal{S}$ then
10:            $\mathcal{K} \leftarrow \mathcal{K} \cup \{|\mathcal{J}_A| + |\mathcal{K}| + 1\}$              ▷ Add new location
11:            $\boldsymbol{x}_{|\mathcal{J}_A|+|\mathcal{K}|+1} \leftarrow \boldsymbol{g}$
12:        else
13:            $f_{\text{dist}}, \boldsymbol{x}_j^* \leftarrow$ SOLVE_DIST$(D_j)$
14:            $\mathcal{K} \leftarrow \mathcal{K} \cup \{|\mathcal{J}_A| + |\mathcal{K}| + 1\}$
15:            $\boldsymbol{x}_{|\mathcal{J}_A|+|\mathcal{K}|+1} \leftarrow \boldsymbol{x}_j^*$
```

### 4.3.1 Filtering potential charging locations

Since some CSs may already be in their best possible location for a given allocation, adding another location may not always be necessary or desirable. If the set of PCLs $\mathcal{J}$ becomes too large, then the MILP $P$ will become increasingly difficult to solve. To avoid this issue we introduce the function FILTER_LOCATIONS with pseudo code in Algorithm 5. In essence, Algorithm 5 filters through existing (old) locations, $\mathcal{J}$, and newly identified PCLs, $\mathcal{K}$. If the Euclidean distance between each location, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$, is less than a set distance $d_{\min}$, then a Monte Carlo method is adopted to randomly remove some, but not all, new locations. The allocation may be different in the next solution, and hence we may not wish to remove all such instances, but still maintain a diverse population of PCLs. If there are plenty of chargers within a radius $\kappa$ of CS $j \in \mathcal{J}$, and few EVs within the same radius expecting to use CS $j \in \mathcal{J}$, then the PCL $k \in \mathcal{K}$ is most likely not needed. Let $C_{\text{EV}}$, $C_{\text{CS}}$ denote the number of EVs and PCLs within radius $\kappa$, respectively. Hence the ratio,

$$\rho := \frac{\mathbb{E}(p)\, C_{\text{EV}}}{2m\, C_{\text{CS}}},$$

measures the "probability" that another PCL, $k \in \mathcal{K}$, will be needed nearby. Note that $\mathbb{E}(p)$, as previously defined, is the expected proportion of EVs needing to charge, and $m$ is the maximum number of chargers at a CS. Next, a random uniform number is generated, $\zeta \sim$ U(0,1). If it is less than $\rho$, then PCL $k$ is excluded from set $\mathcal{K}$.

## 4.4 Warm start

A warm start can be constructed from the improved locations returned from the functions FIND_IMPROVED_LOCATIONS and FILTER_LOCATIONS described in §4.3. The function CONSTRUCT_WARMSTART, for which the pseudo code pro-

**Algorithm 5** Improved location filtering

```
Input: $\mathcal{J}$ : old locations, $\mathcal{K}$ : new locations, $d_{\min}$ : smallest distance allowed between CSs, $\kappa$ : radius from CS
Output: $\mathcal{J}$ : old (filtered) locations, $\mathcal{K}$ new (filtered) locations
1: procedure FILTER_LOCATIONS($\mathcal{J}, \mathcal{K}, d_{\min}, \kappa$)
2:     for $j \in \mathcal{J}, k \in \mathcal{K}$ do
3:         if $||\boldsymbol{x}_j - \boldsymbol{x}_k|| \leq d_{\min}$ then
4:             $C_{\text{EV}} \leftarrow |\{i \in \mathcal{I} : d_{i,j} \leq \kappa, j \in \mathcal{J}\}|$    ▷ Number of EVs within given radius
5:             $C_{\text{CS}} \leftarrow |\{j \in \mathcal{J} : d_{i,j} \leq \kappa, i \in \mathcal{I}\}|$    ▷ Number of CSs within given radius
6:             $\rho \leftarrow \frac{\mathbb{E}(p)C_{\text{EV}}}{2mC_{\text{CS}}}$    ▷ Ratio of EVs needing charge and available chargers within radius
7:             Generate $\zeta \sim$ U(0,1)
8:             if $\zeta < \rho$ then
9:                 $\mathcal{K} \leftarrow \mathcal{K} \setminus k$    ▷ Controlled random removal of PCLs
```

vided in Algorithm 6, takes as input a feasible solution to ($P$), the old set of active locations, $\mathcal{J}_A$, the old set of allocated EVs, $\{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}$, and the new locations, $\mathcal{K}$, and returns $(\hat{u}, \hat{v}, \hat{w})$, a solution constructed from the old solution and the new, improved locations. Firstly, the Euclidean distance, $d_{i,k}$, from each EV, $i \in \mathcal{I}$, to the new set of points, $k \in \mathcal{K}$, is computed. For each of the new locations, $k \in \mathcal{K}$, the variable $v_k$ is set to 1 to indicate that the new CS is now active, and $w_k$ is set to the same number of chargers as in the location it is replacing. For each CS which was active in the previous solution, we set the variables $v_j$ and $w_j$ for all $j \in \mathcal{J}_A$ to zero to revert the decision on the built locations. Next, the new allocations are made in every scenario $s \in \mathcal{S}$ whereby every EV, $i \in \mathcal{I}_s$, which was allocated to the old CS, $j \in \mathcal{J}_A$, is now allocated to the new, improved CS. The constraints enforced for the FIND_IMPROVED_LOCATIONS heuristic, will ensure that the new allocation will always be feasible, however, we raise an exception if the allocation is ever no longer feasible.

**Algorithm 6** Warm start construction

```
Input: $(\hat{u}, \hat{v}, \hat{w})$ : solution to ($P$), $\mathcal{J}_A$ : old active locations, $\{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}$ : old set of allocated EVs, $\mathcal{K}$ : new locations
Output: $(\hat{u}, \hat{v}, \hat{w})$ : updated warm start solution
1: procedure CONSTRUCT_WARMSTART($(\hat{u}, \hat{v}, \hat{w}), \mathcal{J}_A, \{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}, \mathcal{K}$)
2:     for $k \in \mathcal{K}$ do
3:         $v_k \leftarrow 1$    ▷ New location made active
4:         $w_k \leftarrow w_{k-|\mathcal{J}_A|}$    ▷ Assume the same amount of chargers
5:         $v_{k-|\mathcal{J}_A|} \leftarrow 0$    ▷ Old CSs which have improved locations are removed
6:         $w_{k-|\mathcal{J}_A|} \leftarrow 0$
7:         for $s \in \mathcal{S}$ do
8:             for $i \in \mathcal{I}_{A_j}$ do
9:                 $u^s_{i,k-|\mathcal{J}|_A} \leftarrow 0$    ▷ EVs are no longer allocated CSs which have improved locations
10:                if $d_{i,k} \leq r^s_i$ then
11:                    $u^s_{i,k} \leftarrow 1$    ▷ If $d_{i,k} > r^s_i$, raise error, not reachable
```

## 5 Output analysis

After obtaining a good solution, $(u^*, v^*, w^*)$, across multiple scenarios using Algorithm 1, it may be the case that the solution and associated cost is exceptionally optimistic, e.g. if the randomly generated samples are all very similar, or too few scenarios were considered. When performing the Monte Carlo-type simulation described in §3.4, we obtain a collection of Boolean values, $\{\delta^s\}^{s\in\mathcal{S}}$, describing which EVs need to charge in each scenario $s \in \mathcal{S}$. These are correlated with the EV driving ranges, $\{r^s\}^{s\in\mathcal{S}}$, generated from a given truncated normal distribution. Adopting a statistical mechanics interpretation, the pair, $\{(\delta^s, r^s)\}^{s\in\mathcal{S}}$, makes up an ensemble of all possible configurations of the system of EVs [17]. There are only finitely many configurations of EVs needing to charge or not, but there are nevertheless infinitely many virtual copies of the ensemble since the ranges are drawn from a continuous distribution.

Similarly, the spatial search space of the problem is continuous so we allow infinitely many possible placements for CSs. Hence there is no guarantee of optimality in the two-stage location-allocation problem which chooses from only a finite set of PCLs. However, to evaluate the statistical validity of the solution, a re-allocation is performed over another (larger) set of replications, to establish the total cost within a 95% confidence interval of its true value from §4.

## 5.1 Model validation

Having established where to place CSs, and how many to install at each, we test the system on new replications to obtain allocations which depend on the random state of the ensemble $\{(\delta^s, r^s)\}^{s\in\mathcal{S}}$. Setting $v = v^*$, $w = w^*$, in MILP ($P$), only the variable $u_{i,j}$ remains to be evaluated. The resulting allocation problem is

$$\begin{aligned}
\min \quad & 365(c_{\tilde{c}} + c_d)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} d_{i,j}u_{i,j} + K && && (Q)\\
\text{subject to} \quad & \sum_{i\in\mathcal{I}} u_{i,j} \le 2w_j^* && j\in\mathcal{J}, d_{i,j}\le r_i\\
& \sum_{j\in\mathcal{J}} u_{i,j} \le 1 && i\in\mathcal{I}, d_{i,j}\le r_i\\
& \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} u_{i,j} \ge \alpha L && d_{i,j}\le r_i\\
& u_{i,j}\in\{0,1\} && i\in\mathcal{I}, j\in\mathcal{J},
\end{aligned}$$

where

$$K = \sum_{j\in\mathcal{J}}(c_b v_j^* + c_m w_j^*) + \frac{365}{|\mathcal{S}|}|\mathcal{J}|c_{\tilde{c}}\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}}(250 - r_i^s)$$

is a constant from the location-allocation problem ($P$).

Let $n$ denote the number of replications in this validation stage, i.e. the number of times ($Q$) is re-solved. In each replication, we obtain the optimal values $Z_1, Z_2, ..., Z_n$. These are independent and identically distributed in terms of the ensemble of demand configurations $\{(\delta^s, r^s)\}^{s\in\mathcal{S}}$, with mean, $\mathbb{E}(Z_i)$, and variance, $\sigma^2$. By the central limit theorem, the sample mean, $\bar{Z} = \sum_{i=1}^n Z_i/n$, should follow a standard normal distribution $\bar{Z} \sim \mathcal{N}(\bar{Z}, \sigma^2/n)$. A confidence interval for the mean cost can then be calculated as

$$\left(\bar{Z} - \frac{t_{n-1,1-\alpha/2}\,\sigma}{\sqrt{n}}, \bar{Z} + \frac{t_{n-1,1-\alpha/2}\,\sigma}{\sqrt{n}}\right),$$

where $t$ is the test statistic on a $(1-\alpha)100\%$ confidence interval with $n-1$ degrees of freedom, and $\sigma$ is the standard deviation of each $Z_i$. For the normal distribution, the cumulative distribution gives that $P(-1.96 < \bar{Z} < 1.96) = 0.95$, so we set $t = 1.96$ for simplicity. The number of replications needed to obtain an interval of relative width $\epsilon\bar{Z}$, is

$$n = \left(\frac{t_{n-1,1-\alpha/2}\,\sigma}{\epsilon\bar{Z}}\right)^2$$

This can be shown by plugging this expression in for $n$ in the confidence interval.

A solution to ($P$) with $|\mathcal{S}| = 3$ scenarios considered, is one with a total cost of $f^* = 14\,014\,957\$$. By performing $n = 20$ replications, i.e. solving ($Q$) 20 times, we obtain the mean, and variance, respectively, for the total cost,

$$\bar{Z} = \sum_{i=1}^{20}\frac{Z_i}{20} = 13\,753\,884, \qquad \sigma^2 = \sum_{i=1}^{20}\frac{(Z_i - \bar{Z})^2}{19} = (96\,924)^2.$$

The 95% confidence interval for the total cost is hence

$$\left(\bar{Z} - \frac{1.96\,\sigma}{\sqrt{n}}, \bar{Z} + \frac{1.96\,\sigma}{\sqrt{n}}\right) = \big(\bar{Z} - 42\,479, \bar{Z} + 42\,479\big) = (13\,711\,405, 13\,796\,363),$$

which is also very small confidence interval relative to the mean cost after $n = 20$ replications. By calculating the number of iterations needed for relative accuracy $\epsilon = 0.05$, we find that only one iteration is needed. Note that the original objective value, $f^*$, is not within the 95% confidence interval, highlighting the importance of validating the solution. Nevertheless, $f^*$, deviates from the mean $\bar{Z}$ by only 2%.

## 5.2 Sensitivity analysis

To further test the statistical validity of the model and its solution output, we perform a sensitivity analysis on (i) the number of scenarios considered, $|\mathcal{S}|$, and (ii) the number of initial locations $|\mathcal{J}|$, and investigate how these relate to the constraint on the service level required. All experiments and results were run the same machine for a fair comparison (a Windows OS machine with a 11th Gen Intel® Core™ i7-1185G @3.00GHz, with 4 Cores, 8 Logical processors and 32GB RAM).

By solving ($P$) using algorithm 1 with 1, 2, 3, and 4 scenarios, respectively, we find that including more scenarios improves the robustness of the solution found but, as expected, the model takes longer to run. For each of the 4 solutions, we perform 20 replications, representing 20 unseen EV ranges and probabilities of charging which need to be allocated to the built CSs. As shown in Figure 3, increasing the number of scenarios clearly improves feasibility in the secondary validation stage. With one scenario, $|\mathcal{S}| = 1$, only 15% of the 20 solutions obtained are feasible. With two scenarios, this percentage rises to 80%, and with 3 scenarios, all of the solutions are feasible. However, the percentage of feasible solutions decreases to 95% with 4 scenarios, indicating that 3 scenarios may be enough to maximize feasibility in the validation stage.

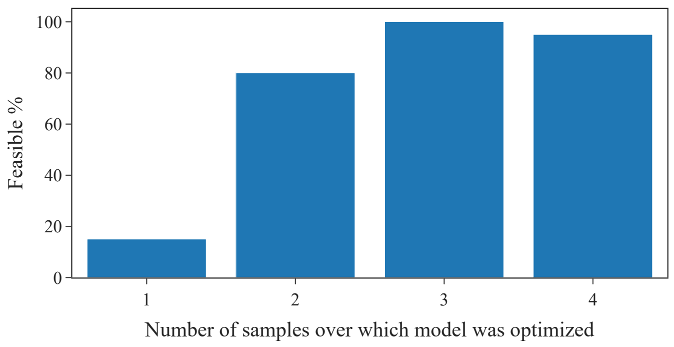


Figure 3: The proportion of feasible solutions in the validation stage, with varying number of scenarios in the location-allocation problem.

Nevertheless, feasible solutions in the validation stage generally have a higher total cost. Figure 4 shows the total cost in bins of width 0.1M, and the number of feasible solutions with that objective value. The bell curve is centred around a total cost of \$13.8 M which is strikingly close to the mean $\bar{Z} = 13.8$ obtained on a different set of scenarios, $\mathcal{S}$, and a different set of replications in the validation stage, in §5.1. A general pattern emerges that fewer scenarios give a lower cost, and more scenarios give a higher cost. Hence stakeholders may want to consider a reasonable amount of scenarios which keep the total cost low and fulfill most of the stochastic demand, hence obtaining mostly feasible solutions at the validation stage.

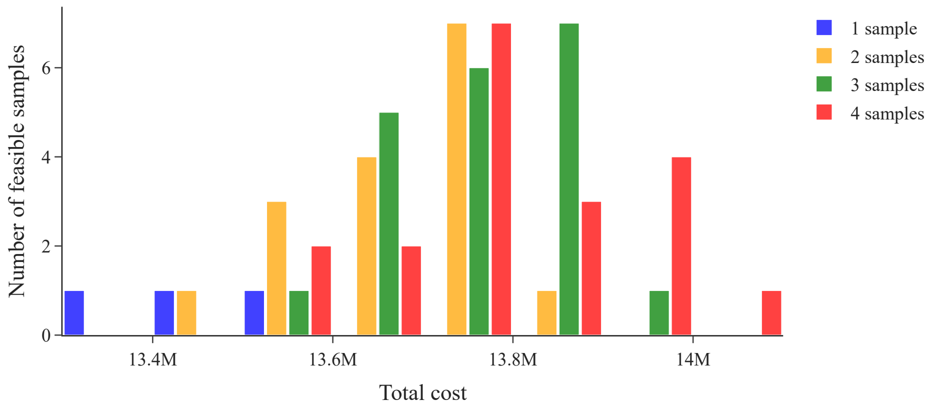


Figure 4: Distribution of total cost for feasible solutions, with different numbers of scenarios.

For the infeasible solutions, we investigate which constraints are violated and find that feasibility is generally affected by the service level. The highest attainable service level is computed by finding a maximum bipartite matching between the EVs that require charging and are within reach of a CS, and all active CSs. This maximal service level and the corresponding objective value is plotted in Figure 11 in Appendix A.

### 5.2.1 Number of initial locations

Since the heuristic approach is initialized with a set of PCLs, we test how the number of initial locations, $\tilde{k}$, affects the objective value obtained at the end of the optimization procedure 1. Figure 5 shows that when the solver is fed too few initial locations, the problem is infeasible. This is because we enforce the 95% service level constraint from the very first iteration. For between 300 and 550 initial locations, the final objective value is stable. All trials were performed with a 300 second time limit. We see that initializing the model with a large number of locations results in a worse final objective value; the model would require more time to search this larger solution space.

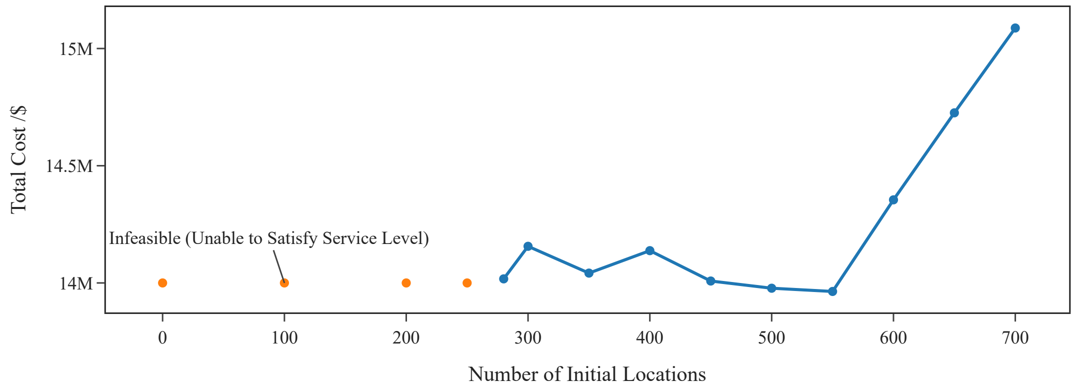


Figure 5: Objective value for different numbers of initial locations using $k$-means clustering.

## 5.3 Graphical User Interface

The sensitivity of different parameter settings, model configurations, and scenarios, can be further explored in an interactive front-end we have built in addition to our solver, using the Python library Streamlit. On opening the application, the user is presented with a map of the EVs locations. The "Optimize" button calls our solver, and after each iteration of Algorithm 1, the map, along with metric boxes and additional plots, is updated. In the left-hand side-bar, the user can select different solver parameters, fix the number of CSs to be built, or provide a new data-set of EV locations. Once the optimization run is complete, the "Validate" button tests the solution for robustness on a specified number of new scenarios, and plots the results. We include screenshots of our interface in Appendix A.

## 6 Final recommendations

We lastly provide a construction plan for 600 CSs. The total cost is \$ 15.42M, and 2 174 CSs are built. The total solve time for this solution was 1 hour, 8 minutes (4153 seconds), and the solver 1 took the following parameters; time limit $T = 300$, $\epsilon = 10\,000$, with 600 initial locations obtained with $k$-means. In Figure 6 we see a strong alignment between the EV locations and the 600 station construction plan. Since each CS can meet the demand of $2m = 16$ vehicles, each EV location can be fully satisfied by a single CS, and at most 1 079 CSs would ever be built.

When we relax the condition to build a fixed number of CSs, a solution with lower total cost can be obtained, as seen in Figure 6. In total, 347 CSs are built, with a total of 2221 chargers. The total cost is \$ 14.42M, a \$1M saving on the 600 station solution. The total solve time for this solution was 1 hour, 6 minutes (4017 seconds), and the solver took the following parameters: timelimit=300, epsilon=10000, number of initial locations = 400.

In table 2 we see that by building fewer CSs, the drive and charge cost is slightly increased, but that the savings in infrastructure cost make it more than worth it. Both solutions build a similar number of chargers, but Figure 7 shows how differently these are distributed: when 600 CSs are built, the number of chargers at each CS is lower, with a mode of 2. The bottom plot in Figure 6 shows how in the optimal construction plan, the majority of CSs have the maximum allowed number of chargers.

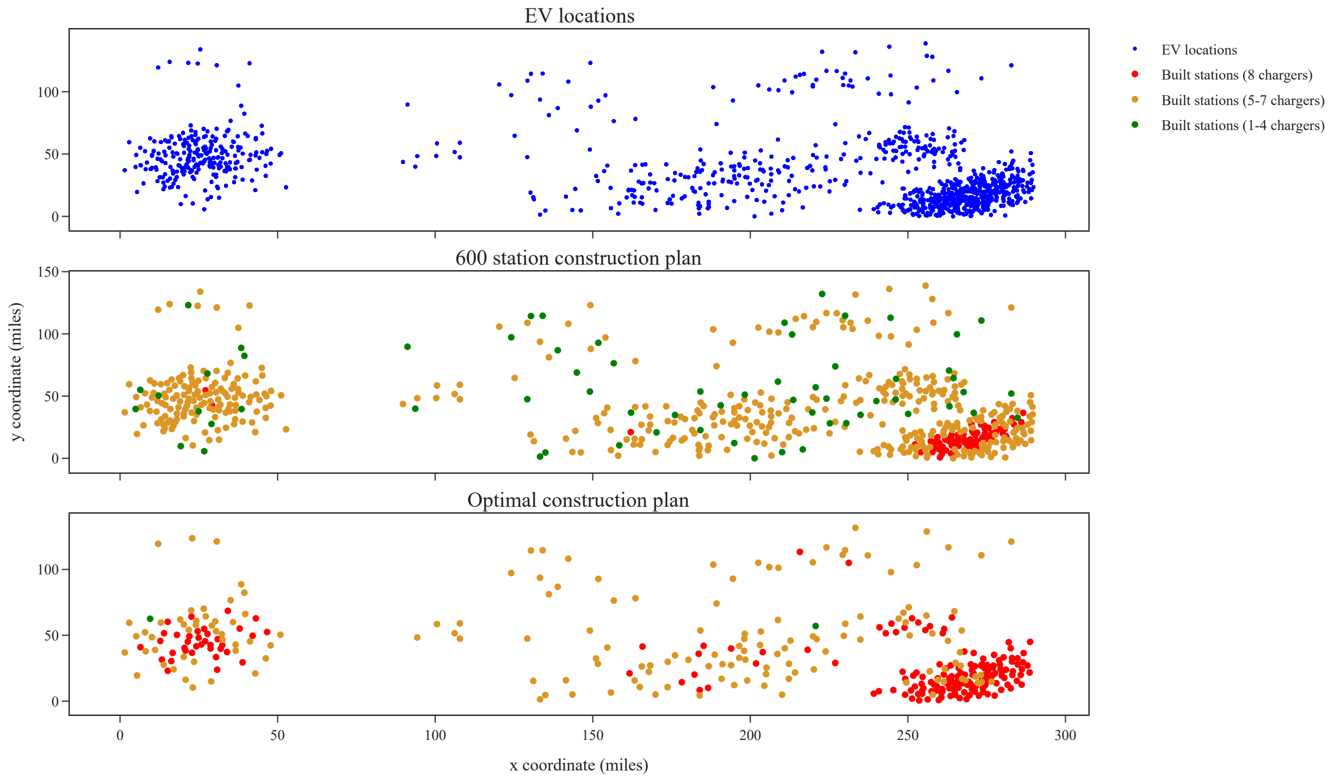


Figure 6: Map of PA showing EV locations (top), the construction plan for 600 CSs (middle), and the unconstrained construction plan (bottom).

All models were implemented in Python, and DOcplex, a Python API for CPLEX, was employed to solve the MILP.

## 6.1 Possible Extensions

In our model, the allocation of vehicles is made to maximise the number of vehicles able to charge. In reality, EV owners have agency, they can choose which charger to visit without the steering hand of a central authority. Given a solution, it would be interesting to compare a scenario in which all EV owners charge at their closest CS, in particular, the queue lengths. Additionally, our model does not consider the time of day at which EVs are needing to charge. Over the course of a day, many more than 2 EVs may make use of a charger, and a queuing simulation would loosen the "single vehicle queue constraint" enforced by our model. Similarly, CSs may be interconnected, e.g. where the price of charging may vary by location, incentivizing user behaviour. This would further complicate the nonlinear model, and has been shown to be NP-hard [3], for the case of an interconnected energy grid.

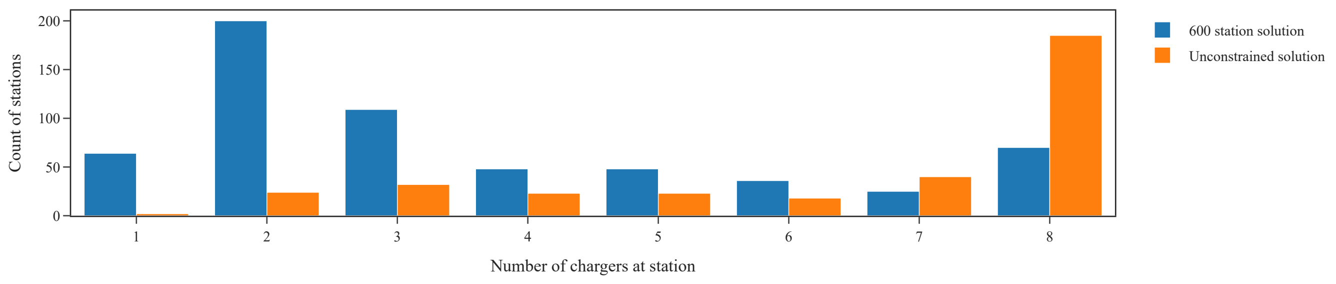


Figure 7: Distribution of chargers at each CS in the 600 station solution, and the unconstrained solution.

Table 2: Comparison of construction plans for a fixed number of CSs, 600, and unconstrained.

| Metric | Unconstrained | 600-station |
|---|---|---|
| # CSs | 347 | 600 |
| # Chargers | 2221 | 2174 |
| Infrastructure cost, $M | 2.85 | 4.09 |
| Drive & charge cost, $M | 11.57 | 11.33 |
| Total cost, $M | 14.42 | 15.42 |

We assumed in our model that EVs with a range out of reach from all CS in a scenario would need not make use of CSs in our model. This was under the assumption that this is a rare scenario. In reality drivers rarely let the range fall below a reasonable mileage as most systems provide multiple low-charge warnings, or enter battery-saving mode below a given threshold [19]. An EV which is critically low on charge would arguably benefit more from a nearby battery-swapping station rather than a standard CS [20].

Sensitivity analysis over other model parameters would be a good exercise. For example, increasing the upper bound on the permitted number of chargers at any given CS. In Figure 7, we see that the optimal solution consists mostly of 8-charger CSs. If more than 8 could be built at each, then perhaps the total cost would come down further.

## References

[1] *Alternative Fuel Corridors - Environment - FHWA*. URL: https://www.fhwa.dot.gov/environment/alternative_fuel_corridors/ (visited on 05/29/2023).

[2] F. J. Aragón et al. "Unconstrained Optimization Algorithms". en. In: *Nonlinear Optimization*. Ed. by F. J. Aragón et al. Springer Undergraduate Texts in Mathematics and Technology. Cham: Springer International Publishing, 2019, pp. 183–252. ISBN: 978-3-030-11184-7. DOI: 10.1007/978-3-030-11184-7_5.

[3] V. Blanco and R. Gázquez. "Continuous maximal covering location problems with interconnected facilities". en. In: *Computers & Operations Research* 132 (Aug. 2021), p. 105310. ISSN: 03050548. DOI: 10.1016/j.cor.2021.105310.

[4] L. Cooper. "An Extension of the Generalized Weber Problem†". en. In: *Journal of Regional Science* 8.2 (1968). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9787.1968.tb01323.x, pp. 181–197. ISSN: 1467-9787. DOI: 10.1111/j.1467-9787.1968.tb01323.x.

[5] L. Cooper. "Heuristic Methods for Location-Allocation Problems". In: *SIAM Review* 6.1 (1964). Publisher: Society for Industrial and Applied Mathematics, pp. 37–53. ISSN: 0036-1445.

[6] L. Cooper. "Location-Allocation Problems". In: *Operations Research* 11.3 (1963). Publisher: INFORMS, pp. 331–343. ISSN: 0030-364X.

[7] M. Daskin. *Network and Discrete Location: Models, Algorithms and Applications*. Second Edition. John Wiley & Sons, Ltd, June 2013.

[8] *Electric Vehicles in PA*. en-US. URL: https://www.dep.pa.gov:443/Business/Energy/OfficeofPollutionPrevention/ElectricVehicles/Pages/default.aspx (visited on 05/07/2023).

[9] R. Z. Farahani et al. "Covering problems in facility location: A review". en. In: *Computers & Industrial Engineering* 62.1 (Feb. 2012), pp. 368–407. ISSN: 0360-8352. DOI: 10.1016/j.cie.2011.08.020.

[10] I. Frade et al. "Optimal Location of Charging Stations for Electric Vehicles in a Neighborhood in Lisbon, Portugal". In: *Transportation Research Record* 2252.1 (Jan. 2011). Publisher: SAGE Publications Inc, pp. 91–98. ISSN: 0361-1981. DOI: 10.3141/2252-12.

[11] D. Guler and T. Yomralioglu. "Suitable location selection for the electric vehicle fast charging station with AHP and fuzzy AHP methods using GIS". In: *Annals of GIS* 26.2 (Apr. 2020). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/19475683.2020.1737226, pp. 169–189. ISSN: 1947-5683. DOI: 10.1080/19475683.2020.1737226.

[12] L. He et al. *Charging an Electric Vehicle Sharing Fleet*. en. SSRN Scholarly Paper. Rochester, NY, Sept. 2019. DOI: 10.2139/ssrn.3223735.

[13] L. He et al. "Service Region Design for Urban Electric Vehicle Sharing Systems". en. In: *Manufacturing & Service Operations Management* (Apr. 2017). Publisher: INFORMS. DOI: 10.1287/msom.2016.0611.

[14] K. Huang et al. "Analysis of Optimal Operation of Charging Stations Based on Dynamic Target Tracking of Electric Vehicles". en. In: *Electronics* 11.19 (Jan. 2022). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, p. 3175. ISSN: 2079-9292. DOI: 10.3390/electronics11193175.

[15] Y.-C. Hung, H. PakHai Lok, and G. Michailidis. "Optimal routing for electric vehicle charging systems with stochastic demand: A heavy traffic approximation approach". en. In: *European Journal of Operational Research* 299.2 (June 2022), pp. 526–541. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2021.06.058.

[16] Ö. B. Kınay, F. Gzara, and S. A. Alumur. "Full cover charging station location problem with routing". en. In: *Transportation Research Part B: Methodological* 144 (Feb. 2021), pp. 1–22. ISSN: 0191-2615. DOI: 10.1016/j.trb.2020.12.001.

[17] L. D. Landau and E. M. Lifshitz. "CHAPTER I - THE FUNDAMENTAL PRINCIPLES OF STATISTICAL PHYSICS". en. In: *Statistical Physics (Third Edition)*. Ed. by L. D. Landau and E. M. Lifshitz. Oxford: Butterworth-Heinemann, Jan. 1980, pp. 1–33. ISBN: 978-0-08-057046-4. DOI: 10.1016/B978-0-08-057046-4.50008-7.

[18] G. Laporte, S. Nickel, and F. Saldanha da Gama, eds. *Location Science*. en. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-13110-8 978-3-319-13111-5. DOI: 10.1007/978-3-319-13111-5.

[19] M. K. Lim, H.-Y. Mak, and Y. Rong. “Toward Mass Adoption of Electric Vehicles: Impact of the Range and Resale Anxieties”. en. In: *Manufacturing & Service Operations Management* (Dec. 2014). Publisher: INFORMS. DOI: 10.1287/msom.2014.0504.

[20] H.-Y. Mak, Y. Rong, and Z.-J. M. Shen. “Infrastructure Planning for Electric Vehicles with Battery Swapping”. en. In: *Management Science* (Apr. 2013). Publisher: INFORMS. DOI: 10.1287/mnsc.1120.1672.

[21] P. D. o. E. Protection. *Driving PA Forward*. da. Mar. 2023. URL: https://storymaps.arcgis.com/stories/6f5db16b8399488a8ef2567e1affa1e2 (visited on 05/07/2023).

[22] L. V. Snyder. “Facility location under uncertainty: a review”. In: *IIE Transactions* 38.7 (June 2006). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07408170500216480, pp. 547–564. ISSN: 0740-817X. DOI: 10.1080/07408170500216480.

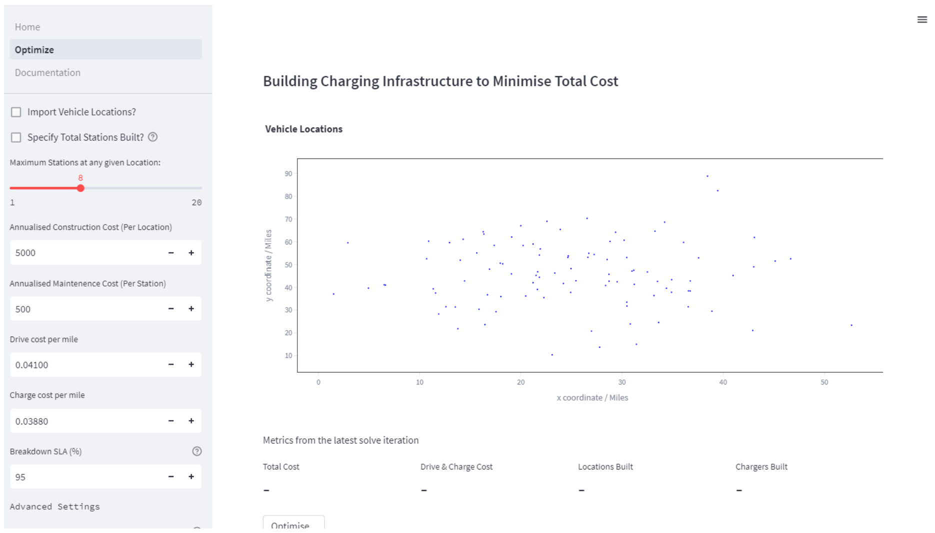
[23] L. V. Snyder and M. S. Daskin. “Stochastic p-robust location problems”. In: *IIE Transactions* 38.11 (Nov. 2006). Publisher: Taylor & Francis, pp. 971–985. ISSN: 0740-817X. DOI: 10.1080/07408170500469113.

[24] X. Xi, R. Sioshansi, and V. Marano. “Simulation–optimization model for location of a public electric vehicle charging infrastructure”. en. In: *Transportation Research Part D: Transport and Environment* 22 (July 2013), pp. 60–69. ISSN: 1361-9209. DOI: 10.1016/j.trd.2013.02.014.

[25] Z. Yi et al. “An agent-based modeling approach for public charging demand estimation and charging station location optimization at urban scale”. en. In: *Computers, Environment and Urban Systems* 101 (Apr. 2023), p. 101949. ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2023.101949.

## A Streamlit interface walkthrough

Here, we share screenshots of the interface at different stages of the solve. In figure 8, the streamlit interface has been opened to the optimization page, to a small dataset of vehicles which we have used for testing. The sidebar is open so that default parameters can be altered, or a different set of vehicle data can be uploaded. The map shows the vehicle locations, and the optimization button is yet to be clicked. The metric panels are empty, since the solver has not yet run.



Building Charging Infrastructure to Minimise Total Cost

☐ Import Vehicle Locations?

☐ Specify Total Stations Built?

Maximum Stations at any given Location: 8 (1–20)

Annualised Construction Cost (Per Location): 5000

Annualised Maintenence Cost (Per Station): 500

Drive cost per mile: 0.04100

Charge cost per mile: 0.03880

Breakdown SLA (%): 95

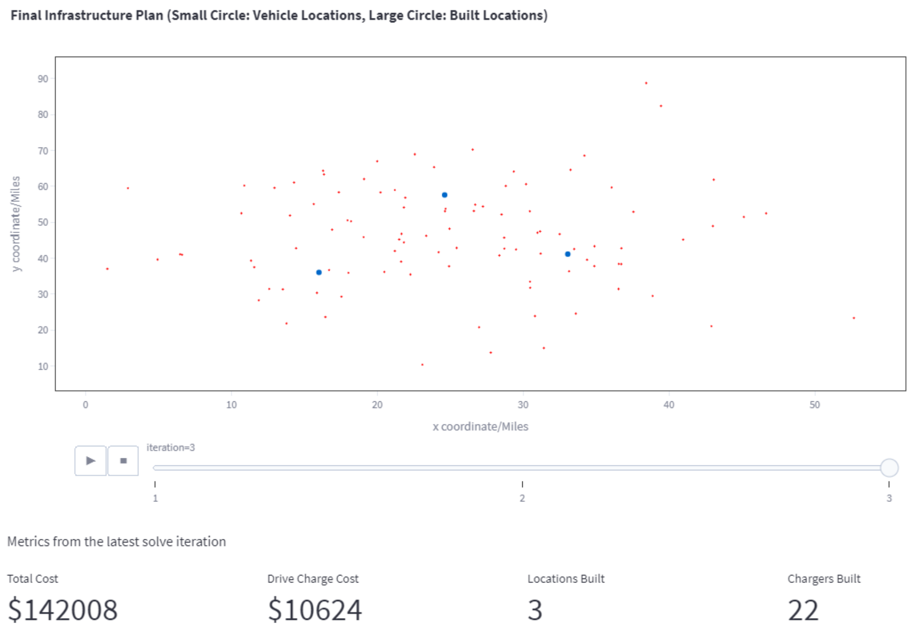
Advanced Settings

Metrics from the latest solve iteration

| Total Cost | Drive & Charge Cost | Locations Built | Chargers Built |
|---|---|---|---|
| - | - | - | - |

Figure 8: The optimize page of the model interface

Once the solver has completed, figure 9 shows how the solution will be presented. Play and pause buttons enable the user to step through the evolution of the solve to see how the construction plan was improved at each iteration. The metric panels report on the final solution.

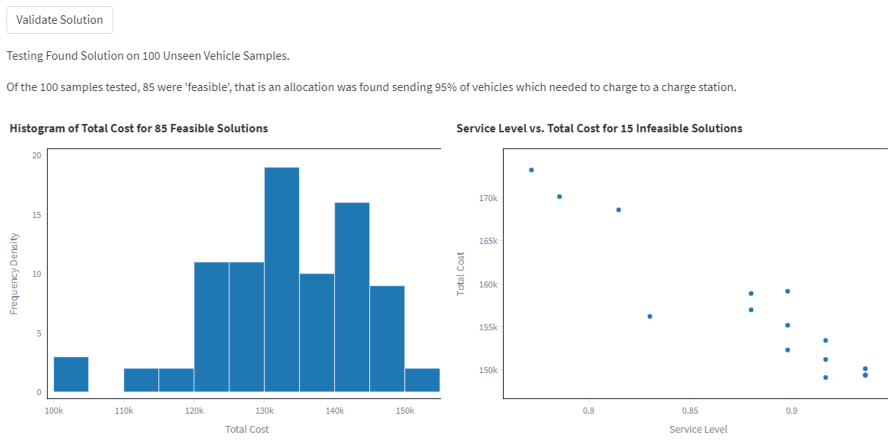

Final Infrastructure Plan (Small Circle: Vehicle Locations, Large Circle: Built Locations)
y coordinate/Miles
x coordinate/Miles
iteration=3

Metrics from the latest solve iteration

| Total Cost | Drive Charge Cost | Locations Built | Chargers Built |
|---|---|---|---|
| $142008 | $10624 | 3 | 22 |

Figure 9

In figure 10, validation of the solution has taken place: 100 new samples were tested against the solution found previously. A histogram of the total costs for the feasible solutions indicates the robustness of the solution, and the scatter chart of the infeasible solutions shows how close to the desired service level the solution was.


Testing Found Solution on 100 Unseen Vehicle Samples.

Of the 100 samples tested, 85 were 'feasible', that is an allocation was found sending 95% of vehicles which needed to charge to a charge station.

Figure 10: The validation stage output

## A Maximal service level and infeasibility

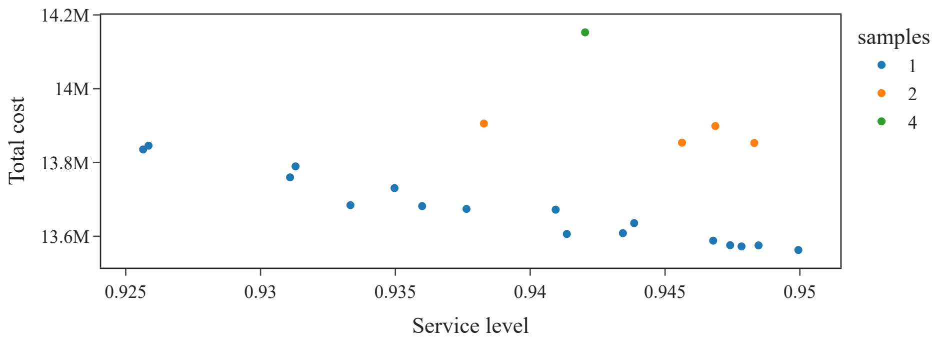


Figure 11: Maximal service level achieved, and corresponding total cost of infeasible scenarios (95% service level not attainable)