Optimal placement of electric vehicle slow-charging stations: A continuous facility location problem under uncertainty

H.W. Ljósheim[a], S. Jenkins[a], K.D. Searle[a,*], J.K. Wolff[a]

[a]*School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, James Clerk Maxwell Building, Kings Buildings, Mayfield Road, Edinburgh, EH9 3FD, UK*

Abstract

Electric vehicles (EVs) are becoming a key mechanism to reduce emissions in the transportation industry, and hence contribute to the green transition. In this paper, we present a mathematical programming model which determines the optimal placement of EV charging stations which minimises the annualised infrastructure set up cost and user drive cost. Moreover, in our case we assume that the EV charging demand is uncertain. The model is formulated as a two-stage, continuous location-allocation model in the form of a generalized Weber problem. This formulation is, however, is non-convex and notoriously difficult to solve. We therefore propose a suitable discretization procedure to find high quality solutions in a suitable time. We then perform computational experiments to test the performance of our discretization procedure on solution quality in terms of objection function value and robustness.

A part of this solution procedure was entered into the 15th AIMMS-MOPTA Optimization Modeling Competition.

Keywords: Continuous location, Electric charging, Facility location, Uncertainty

1. Introduction

*Electric Vehicle*s (EVs) are becoming a key mechanism to reduce emissions in the transportation industry, and hence contribute to the green transition on an international 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 Station*s (CSs) [61]. 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 [57, 51].

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 [57, 56]. 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 which should be built at each location to minimize the total cost [49]. In the problem, demand points are stationary, and are not assumed to have additional descriptive data regarding drive-times, nor do vehicles have a time-related charging pattern. The ranges of each EV is stochastic in nature, sampled from a continuous truncated normal distribution, and the probability with which the EV needs to charge is formulaic on their range. The solution space is continuous in the sense that CSs can be built anywhere in the plane, and multiple chargers can be installed at each CS. Each charger is capable of charging multiple EVs. There are costs associated with the building, and with the maintenance of chargers, and with the driving and charging of the EVs. The nature of the charging infrastructure which minimises the total

*Corresponding author

cost is to be determined. The data provided for this problem as part of the 15th AIMMS-MOPTA Optimization Modeling Competition and included the locations of 1 079 EVs.

The remainder of this paper is structured as follows. Relevant models for designing EV charging infrastructure and mathematical frameworks to tackle such facility location problems are briefly reviewed in §2 where we place an emphasis on the contributions of this paper. The optimization modelling approach together with important modelling assumptions is presented in §3. Thereafter, the tailored solution approach for solving the aforementioned model is discussed in §4 which is followed by an ensuing computational analysis illustrating how certain model parameters impact solution performance in terms of cost and robustness. Lastly, the paper comes to a close with some concluding remarks in §7.

2. Literature review

The design of EV CS infrastructure falls into an increasingly important category of operational research pertaining to the transition to net zero emissions and the sustainable development goals. The limited and often uncertain driving range of EVs induces *range anxiety* in many first-time EV buyers, preventing mass adoption of battery-driven vehicles [51]. A "chicken-or-the-egg" paradox naturally arises as stakeholders are only willing to build charging stations according to existing demand but users will be hesitant to invest in EVs without existing infrastructure. Hence demand may be expected to increase with more charging opportunities in the long run [1]. Additionally, both the driving range and the user demand may be uncertain due to EV users' inherently non-uniform charging pattern.

Many optimization models with similar motivations in terms of design and optimization of EV charging networks have been presented in the literature. We follow a similar approach, however, we propose a solution approach which is novel in its application to EV planning. In the remainder of this paper, we restrict the discussion to EV charging infrastructure planning modelled as a continuous facility location problem, considering the choice of objective, the modelling of (stochastic) demand, and the solution procedure. We refer the interested reader to [3, 18, 26, 48, 55] for an overview of facility location problems, and [17, 58] for facility location problems under uncertainty.

2.1. Facility location

The problem of determining the coordinates of facilities in the two-dimensional Euclidean plane which minimize the distance from another set of known locations is widely known in the literature as the Weber problem [13, 19, 22]. In the case exactly p facilities must be placed then it is referred to as the continuous p-median problem [31] which is known to be NP-hard [53]. Hence it can only be solved exactly for small instances, although advances have been made in the last two decades [8, 24, 29]. With the rise of metaheuristics such as tabu search [30], variable neighbourhood search [32], and the popular genetic algorithm [37], larger problem instances of the continuous p-median have been solved [9, 54], with the genetic algorithm producing the best results in [23]. However, as early as the 1960's, American applied mathematician Cooper [14] developed an alternative heuristic for the continuous p-median problem, known as the location-allocation algorithm [15, 16] which reduces the multi-facility Weber problem to a set of smaller single facility location problems. The algorithm begins by initiating a discrete set of candidate facilities and then alternates between (i) allocating demand points to exactly p candidate locations, and (ii) improving the location of the p facilities by solving a one-median problem exactly. After a finite number of iterations or no improved solution can be found, the algorithm terminates. This heuristic remains a widely researched method, with the continuous facility location problem often being referred to as the location-allocation problem [7], and the method was recently improved in [5, 6, 41].

Due to the evidently challenging nature of a large-scale continuous problem, most real-world facility location problems are formulated as fully discrete routing or allocation problems on graphs, in particular for the case of public EV planning (e.g. [1, 10, 20, 28, 45]). In these formulations, facility locations are chosen from a finite set of candidate locations, greatly simplifying the solution procedure. The graph underlying the model, is however, only a fair representation of the original problem if good candidate locations are chosen for the facilities, a decision process which may be highly qualitative and problem-dependent [22]. Furthermore, the number of candidate locations typically quadruples with respect to the size of the planar search region. While only a handful of lots may be available inside e.g. a medium-size neighbourhood, the number becomes intractable for e.g. the size of a state. Nevertheless, the benefit of this approach is the significantly smaller search space in the form of a discrete set of points rather than the whole continuous plane. The location-allocation solution

approach offers an ideal medium between the fully discrete graph formulation and the continuous p-median problem, as it allows for the addition of new candidate locations at the location stage of the algorithm, to then be utilized at the discrete allocation stage. In this paper, we propose an additional filter on the candidate locations such that the discrete problem does not grow to an untractable size.

For the discrete allocation stage, a wide variety of facility location models can be applied, *e.g.* the set covering problem, the *Maximal Covering Location Problem* (MCLP) [12], the fixed-charge problem [27], or the *Flow Capturing Location Problem* (FCLP) [36]. These differ primarily in the manner in which demand is modelled, and in the objective they consider. In a node-based formulation, the demand is modelled at the vertices of a graph, and in a flow-based formulation, the demand is modelled on the edges. These formulations can also be combined to consider weighted paths between vertices of the graph. Secondly, transportation distance or cost is typically minimised in the set covering problem and fixed-charge problem, and demand coverage is maximised in MCLP and FCLP. A balance between differing objectives of the stakeholders may similarly be sought by setting constraints on the cost or coverage of the model, such as in the capacitated fixed charge problem where all demand does not need to be met.

2.2. EV public planning

In an EV charging station planning problem, the "facilities" correspond to CSs, with possibly multiple chargers at each station, and the demand arises from the users of the charging infrastructure, *i.e.* EV owners. The demand can similarly be modelled point-wise, by aggregating registered EVs in a geographical area, or as EV traffic flow. Stakeholders are often incentivized to construct CS ensembles within existing infrastructure, *e.g.* along major national highways with programs like the Alternative Fuel Corridors [61]. For large-scale planning of EV charging stations, the most popular approach is routing-location problems or flow-refueling location problems [33, 34, 38, 40, 42, 43, 44, 45, 50, 52, 62, 63, 64]. In such models, the demand is modelled as flow from an origin to a destination to imitate the travel patterns of EV owners which stop to re-fuel during or at the end of a commute. On the other hand, in node-based models, demand points naturally arise as the centres of clustered housing areas with registered EV owners, parking lots [10, 11], or points of interests such as shops and restaurants. The node-based formulation has been applied to EV charging station planning in [10, 11, 20, 28, 39], and is particularly suited for placement of slow chargers within densely populated areas, while a flow-based model may more accurately capture fast-charging demand arising from inter-city travel [1]. Recently, in [1, 64], both types are incorporated into a hybrid model [35].

2.2.1. Modelling demand

Ensuring accurate, relevant demand data for facility location problems is a research field in its own right, with approaches in EV infrastructure planning ranging from geographical data aggregation [39] to regression analysis [11, 28] and simulation [64]. In some cases, fast-charging demand or the related demand of EV sharing systems may more accurately be captured by a queueing model [33, 38], as owners are expected to be willing to wait 20-30 minutes to charge. Slow charging may take up to eight hours, limiting the relevance of queueing. Introducing multiple time periods also allows for changes in demand over time, balancing the current and future demand of the "chicken-or-the-egg" paradox introduced previously [1, 10, 60]. A third consideration is that the demand may be inherently stochastic such that charging demand is met based on the likelihood of requiring charging [38, 42, 43, 52]. Three main approaches exist in stochastic optimization: Robust optimization (optimizing for multiple scenarios), stochastic optimization (underlying probability distribution is known), chance-constrained (constrain service level to a certain probability of being attained) [46].

With the addition of stochastic demand, the continuous facility location problem becomes significantly more difficult to solve [59]. The intrinsic two-stage decision process of a location-allocation problem makes a *robust* optimization approach particularly suitable, such that in the first stage, a decision is made about allocation based on the expected cost given a set of scenarios, and in the latter stage, a decision is made about allocation. The robust optimization model presented in this paper is based on this approach, as a location-allocation solution is first obtained for a finite set of scenarios, the location is then improved, and the solution is hence tested on a large number of new scenarios in order to obtain a confidence interval for the expected cost. This approach is appropriate since the random vectors we consider are inherently bounded in magnitude, hence a finite set of scenarios may adequately capture the stochasticity of the uncertain parameters.

We assume, for simplicity, that demand is drawn from the same distribution and consider the placement of slow-chargers, hence point-wise aggregation is an appropriate approach.

2.2.2. Our contribution

In terms of the objective function [25], most papers related to the design of EV charging infrastructure consider a (capacitated) maximal covering problem [10, 20, 28], a maximal flow problem [34, 40, 42, 44, 45, 62, 63, 64], or a combination of the two [1, 60]. EV models that minimize cost were considered in [11, 33, 43, 50, 52, 39]. Alternatively, we consider a capacitated *fixed-charge* model where the accumulative *social cost* to users and stakeholders is minimized simultaneously. Assuming a small *price of anarchy*, the collective best solution is ultimately attractive to all parties, allowing for a natural balance between differing objectives. In this utilitarian approach, users are allocated to facilities but a minimum distance is employed as an equity measure in the location improvement stage of the algorithm. Minimal envy between groups of users can be achieved by minimizing the difference in distance for all pairs of demand points [21]. However, this problem has only been solved for a single facility, and a gap remains in the research on equity measures for continuous multi-facility location problems. We refer to [4] for a recent overview of equity measures for discrete and continuous p-facility location problems.

Our work is most similar to [10], which also considers demand points, potential charging locations, and local improvement of charging locations. Nevertheless, they considered a MCLP and parking lots as deterministic demand points, where as we consider more general stochastic demand points. To the best of our knowledge, a novelty presented in this paper is the combination of a node-formulation of stochastic demand, the minimisation of the total cost rather than the maximisation of coverage, and the use of location-allocation algorithm with custom filtering.

3. Optimization model

In this section we describe an EV charging location-allocation problem where charging stations can be positioned anywhere in the geographic region under consideration and where the EVs charge demand is uncertain. We derive an optimisation model suitable for solving this problem. 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 an acceptable computation time.

3.1. Index sets

Let $\mathcal{I}$ denote the finite set of EVs. Each vehicle $i \in \mathcal{I}$ has a location, denoted by $\boldsymbol{a}_i \in \mathbb{R}^2$ and a range, denoted by r_i. In some cases, multiple vehicles can share the same location, modelling that EV owners may posses more than one EV. Let $\mathcal{J}$ denote the set of CSs that need to be located. All elements of $\mathcal{I}$ lie within the bounded subset $[0, X] \times [0, Y] \subset \mathbb{R}^2$, where X and Y are finite values denoting the bounds of the geographical region under consideration which without loss generality we assume has a vertex at the origin. We assume that $|\mathcal{J}|$ is sufficiently large to allow all of the EVs to charge when required.

The problem we consider is inherently stochastic in nature. More specifically, we assume that the range of each EV $i \in \mathcal{I}$ in miles is normally distributed, denoted by $r_i \sim \mathrm{N}(\mu, \sigma^2)$, with finite truncated bounds $r_{\min}$ and $r_{\max}$ with $r_{\min} < r_{\max}$. Next we assume that an EV $i \in \mathcal{I}$ with range r_i will charge with probability $p_i = p(r_i) = \exp(-\lambda^2(r_i - r_{\min})^2)$, such that EVs with the minimum range $r_{\min}$will charge with probability one, and the probability of charging decreases exponentially as the range of the EV increases. This is an appropriate model, since drivers are more likely to charge their vehicles when the range is low in order to complete their next journey. We assume that all EVs have the same range, $r_{\max}$, when fully charged.

3.2. Modelling approach

To account for this uncertainty, we employ a two-stage, robust optimisation framework and consider a finite set of scenarios denoted by $\mathcal{S}$, such that $s \in \mathcal{S}$ represents one scenario. Moreover, in each scenario $s \in \mathcal{S}$ the realisation of the range of EV $i \in \mathcal{I}$, is denoted by r_i^s. A Monte Carlo type simulation is employed to determine which EVs $i \in \mathcal{I}$ charges in scenario $s \in \mathcal{S}$. This is modelled by the Boolean parameters $\delta_i^s \in \{0, 1\}$, where an EV $i \in \mathcal{I}$ requires the use of a charger in a scenario $s \in \mathcal{S}$ if $\delta_i^s = 1$. Therefore, if the probability p_i^s that an EV needs to charge, is greater than or equal to a random uniform number, $\psi_i^s \sim \mathrm{U}(0, 1)$, then the EV $i \in \mathcal{I}$ will require a charger in scenario $s \in \mathcal{S}$. The method is summarised as follows. First, sample r_i^s, compute p_i^s and sample $\psi_i^s \sim \mathrm{U}(0, 1)$ for each $i \in \mathcal{I}$ and $s \in \mathcal{S}$. If $\psi_i^s \leq p_i^s$, then set the Boolean value $\delta_i^s = 1$, otherwise set it to zero. For the sake of convenience, we introduce the set $\mathcal{I}_s = \{i \in \mathcal{I} \ : \ \delta_i^s = 1\}$ of all EVs $i \in \mathcal{I}$, which require

charger in scenario $s \in \mathcal{S}$. An example output for this simulation procedure for a single scenario is shown in Table 1.

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 ψ_i^s	Charging required δ_i^s
1	98	0.42	0.50	0
2	73	0.67	0.13	1
⋮	⋮	⋮	⋮	⋮
$\lvert\mathcal{I}\rvert$	101	0.39	0.48	0

The probability density function of the truncated normal distribution, $\mathbb{P}(r_i)$ is normalized, hence we can consider the law of large numbers to obtain the expected probability that EV $i \in \mathcal{I}$ needs charging

$$\mathbb{E}[p(r_i)] = \lim_{|\mathcal{S}|\to\infty} \sum_{s\in\mathcal{S}} p(r_i^s)\mathbb{P}(r_i^s). \tag{1}$$

Note that since EV $i \in \mathcal{I}$ is arbitrary, this is the same as the proportion of EVs in $\mathcal{I}$ that need charging. Hence we drop the notation r_i in favour of a general range r, such that

$$\mathbb{E}[p(r)] = \int_{-\infty}^{\infty} p(r)\mathbb{P}(r)\,\mathrm{d}r \tag{2}$$

$$= \int_{r_{\min}}^{r_{\max}} e^{-\lambda^2(r-r_{\min})^2} \frac{\phi\left(\frac{r-\mu}{\sigma}\right)}{\Phi\left(\frac{r_{\max}-\mu}{\sigma}\right) - \Phi\left(\frac{r_{\min}-\mu}{\sigma}\right)}\,\mathrm{d}r, \tag{3}$$

where ϕ and Φ are, respectively, the probability density function and the cumulative distribution function of the standard normal distribution with mean μ and standard deviation σ. This expectation can be computed numerically for any given parameter set.

3.3. Decision variables

Let the tuple of decision variables

$$\boldsymbol{x}_j = (x_j^1, x_j^2) \in \mathbb{R}^{2\times|\mathcal{J}|} \tag{4}$$

denote the geographical position of CS $j \in \mathcal{J}$ and let auxiliary variable

$$\tilde{d}_{i,j} = ||\boldsymbol{x}_j - \boldsymbol{a}_i|| \tag{5}$$

denote the distance from an EV $i \in \mathcal{I}$ to a CS $j \in \mathcal{J}$, where the parameter $\boldsymbol{a}_i = (a_i^1, a_i^2) \in \mathbb{R}^2$ is a tuple of spatial coordinates of EV $i \in \mathcal{I}$, and $||\cdot||$ is the standard 2-norm.

Since we will not necessarily build every CS $j \in \mathcal{J}$, we introduce the decision variables $v \in \{0,1\}^{|\mathcal{J}|}$ such that

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

Clearly, the decision to build a CS is independent of the scenarios. If a CS is built, it has to be built across all scenarios. The integer decision variables $w \in \mathbb{Z}_+^{|\mathcal{J}|}$ model the number of chargers to install at each location, such that

$$w_j = \text{ the number of chargers to install at CS } j \in \mathcal{J},$$

which are also fixed across scenarios.

Moreover, an EV $i \in \mathcal{I}$ is allocated to CSs through decision variables, $u \in \{0,1\}^{|\mathcal{I}|\times|\mathcal{J}|\times|\mathcal{S}|}$, such that

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

Allocation in this context has a corresponding 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 the allocation with total lowest cost for all stakeholders overall. In this case, we assume that the allocation variable can change depending on the scenario. This is motivated by the fact that, depending on a range of an EV in each scenario, the user may be able to charge at different CSs. A summary of all model decision variables can be found in Table 2.

Table 2: A summary of the decision variables employed in the derivation of the EV charging location-allocation model.

Decision Variable	Description
$\boldsymbol{x}_j$	The geographical position of CS $j \in \mathcal{J}$
$\tilde{d}_{i,j}$	The distance from an EV $i \in \mathcal{I}$ to a CS $j \in \mathcal{J}$
v_j	Build CS $j \in \mathcal{J}$
w_j	The number of chargers installed at CS $j \in \mathcal{J}$
$u^s_{i,j}$	Assign EV $i \in \mathcal{I}_s$ to CS $j \in \mathcal{J}$ in scenario $s \in \mathcal{S}$

3.4. Constraints

Due to space limitations, the number of chargers that can be built at each CS is bounded from above by some $m \in \mathbb{Z}_+$. Therefore, if a CS is built, it must contain between one and m chargers, otherwise it has to be zero, which we enforce with the constraints

$$v_j \leq w_j \leq m v_j \quad \forall j \in \mathcal{J}. \tag{6}$$

In an attempt to limit the congestion at charging stations, we enforce that the maximum of q EVs can be allocated to each charger. That is, the number of EVs which may be allocated to each CS in each scenario may not exceed q times the number of chargers at that CS. We enforce this by the constraint set,

$$\sum_{i \in \mathcal{I}_s} u^s_{i,j} \leq q w_j \quad \forall j \in \mathcal{J},\ s \in \mathcal{S}. \tag{7}$$

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 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 constraint set,

$$\sum_{j \in \mathcal{J}} u^s_{i,j} \leq 1 \quad \forall i \in \mathcal{I}_s,\ s \in \mathcal{S}. \tag{8}$$

We introduce the notion of satisfying a service level as the fraction of charging demand which must be fulfilled in every scenario. For stakeholders using the model, this gives them influence over the minimal fraction of EVs which need to charge that are allocated to a charger in every scenario. More specifically, let $\alpha \in [0, 1]$ denote the lower bound on service level then we require that

$$\sum_{i \in \mathcal{I}_s} \sum_{j \in \mathcal{J}} u^s_{i,j} \geq \alpha |\mathcal{I}_s| \quad \forall s \in \mathcal{S}. \tag{9}$$

In this model, we assume that an EV $i \in \mathcal{I}_s$ which requires a charger in some scenario $s \in \mathcal{S}$ can only be allocated to a CS if the distance between the between the EV and CS is less than the range of the EV. This is enforced with the constraint set,

$$\tilde{d}_{i,j} u^s_{i,j} \leq r^s_i \quad \forall i \in \mathcal{I},\ j \in \mathcal{J},\ s \in \mathcal{S}. \tag{10}$$

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

$$\boldsymbol{x}_j \in [0, X] \times [0, Y] \subseteq \mathbb{R}^2 \quad \forall j \in \mathcal{J} \tag{11}$$

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

3.5. 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 $r_{\max}$ miles; c_d is the cost of driving to the CS (\$/mile). We assume that the costs are charger and vehicle independent.

The objective function we want to minimize is the overall annualized set up and user cost of the CSs infrastructure in the sense that we seek to minimise

$$f_{\text{NL}} = \frac{365}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s}\left[\sum_{j\in\mathcal{J}}\left[\underbrace{c_d\tilde{d}_{i,j}u^s_{i,j}}_{1.} + \underbrace{c_{\tilde{c}}(r_{\max}+\tilde{d}_{i,j}-r^s_i)u^s_{i,j}}_{2.}\right] + \underbrace{c_{\tilde{c}}(r_{\max}-r^s_i)(1-\sum_{j\in\mathcal{J}}u^s_{i,j})}_{3.}\right] + \sum_{j\in\mathcal{J}}(\underbrace{c_b v_j}_{4.} + \underbrace{c_m w_j}_{5.}) \tag{12}$$

$$= \sum_{j\in\mathcal{J}}\left[c_b v_j + c_m w_j + \frac{365}{|\mathcal{S}|}(c_{\tilde{c}}+c_d)\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}}\tilde{d}_{i,j}u^s_{i,j}\right] + \frac{365}{|\mathcal{S}|}c_{\tilde{c}}\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}}(r_{\max}-r^s_i). \tag{13}$$

The first term in (12) 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 $r_{\max}$ 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 $i^* \in \mathcal{I}_s, j^* \in \mathcal{J}$ 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, the charging of which we have included in our objective function.

The fourth term in (12) 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 incentivise 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 (13), where the constant does not impact the optimization but is included for an accurate total cost. The objective function (13) is nonlinear and non-convex. We have tried to solve the model described in this section with commercial solvers which can solve nonlinear problems, however, even for small problem instances we could not obtain high quality solutions in a reasonable amount of time. Therefore, in the next section we propose a heuristic solution approach.

4. Solution approach

The model can (6)–(13) be transformed into a two-stage location-allocation problem similar to [15]. In this approach we initialize $\mathcal{J}$ with a discrete set of *Potential Charging Locations* (PCLs) such that auxiliary variable $\tilde{d}_{i,j}$ can be replaced by parameters $d_{i,j}$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$. The variable $\boldsymbol{x}$ in the nonlinear model becomes a property of set $\mathcal{J}$ *i.e.* the coordinates of each PCL is known *a priori*. Therefore, in each scenario, if an EV needs to charge, we compute the Euclidean distance from its location to each of the PCLs in the set $\mathcal{J}$ so that $d_{i,j}$ denotes the distance from each EV $i \in \mathcal{I}$ to each PCL $j \in \mathcal{J}$. Now, we note that not all of the allocation decision variables are required to solve the problem for two reasons. Firstly, some vehicles do not need to charge in some scenarios, if $\delta^{s'}_{i'} = 0$ for EV $i' \in \mathcal{I}$, and for $s' \in \mathcal{S}$, then the allocation decision variable $u^{s'}_{i',j}$ can be excluded for all $j \in \mathcal{J}$. Secondly, the range of an EV may be insufficient to reach some PCLs. Specifically, if $d_{i^*,j^*} > r^{s^*}_{i^*}$ for $i^* \in \mathcal{I}, j^* \in \mathcal{J}$, and $s^* \in \mathcal{S}$, then the decision variable $u^{s^*}_{i^*,j^*}$ can be excluded from the model.

Recall that the set of all EVs $i \in \mathcal{I}$, which require charging in scenario $s \in \mathcal{S}$ is denoted by, $\mathcal{I}_s = \{i \in \mathcal{I} \ : \ \delta^s_i = 1\}$. Similarly, let $\boldsymbol{\Omega}_s = \{(i,j) \in (\mathcal{I}_s, \mathcal{J}) \ : \ d_{i,j} \le r^s_i\}$ denote the set of feasible allocations of EVs and CSs in a fixed scenario, $s \in \mathcal{S}$. Therefore, applying the above reasoning, the reduced set of allocation decision variables is $\{u^s_{i,j} \in \{0,1\} \ : \ \forall\ s \in \mathcal{S},\ (i,j) \in \boldsymbol{\Omega}_s\}$. It then follows that the nonlinear objective function (13) becomes the linear objective function

$$f_{\text{L}} = \sum_{j\in\mathcal{J}}(c_b v_j + c_m w_j) + \frac{365}{|S|}(c_{\tilde{c}}+c_d)\sum_{s\in\mathcal{S}}\sum_{(i,j)\in\boldsymbol{\Omega}_s} d_{i,j}u^s_{i,j} + C, \tag{14}$$

where C is the constant from (13) with the constraint set (10) implicitly enforced. This then results in the *Mixed-Integer Linear Programming* (MILP),

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

We have removed the continuous *Charging Point*s (CPs) location decision problem from the model. Instead, the model has only integer decision variables, whether to build chargers at a PCL, and if so how many. In this case, the formulation is linear and the problem becomes a pure MILP model and any solution to (P) is a valid upper bound on the global optimum of the nonlinear problem (6)–(13).

This approach, however, has the drawback that the quality of the upper bound depends on the choice of $\mathcal{J}$. 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 and is illustrated schematically in Figure 1.

Algorithm 1 Location-allocation algorithm

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

Output:$f^*, (u^*, v^*, w^*)$: approximate solution to (6)–(13)

$\mathcal{J}, \boldsymbol{x} \leftarrow$ INITIAL_LOCATIONS(n, '*type*')
$(\hat{u}, \hat{v}, \hat{w}) \leftarrow \emptyset$
Improvement $\leftarrow$ *True*
while Improvement = *True* **do**
    $\hat{f},\ (\hat{u}, \hat{v}, \hat{w}) \leftarrow$ SOLVE_MILP($P, T, \epsilon, (\hat{u}, \hat{v}, \hat{w})$)
    $\mathcal{K},\ \mathcal{J}_R,\ \{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A} \leftarrow$ FIND_IMPROVED_LOCATIONS($\hat{u}, \hat{v}$)
    $\mathcal{K},\ \mathcal{J}_R \leftarrow$ FILTER_LOCATIONS($\mathcal{J}_R, \mathcal{J}, \mathcal{K}, d_{\min}, \kappa$)
    if $|\mathcal{K}| = 0$ **then**
        Improvement $\leftarrow$ *False*
        $f^*, (u^*, v^*, w^*) \leftarrow \hat{f},\ (\hat{u}, \hat{v}, \hat{w})$
    else
        $\mathcal{J} \leftarrow \mathcal{J} \cup \mathcal{K}$
    end if
    $f_{\mathrm{WS}},\ (\hat{u}, \hat{v}, \hat{w}) \leftarrow$ CONSTRUCT_WARMSTART($\hat{u}, \hat{v}, \hat{w}$), $\mathcal{J}_R, \{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}, \mathcal{K}$)
    if $|f_{\mathrm{WS}} - \hat{f}| \leq \epsilon$ **then**
        Improvement $\leftarrow$ *False*
        $f^*, (u^*, v^*, w^*) \leftarrow \hat{f},\ (\hat{u}, \hat{v}, \hat{w})$
    end if
end while

Algorithm 1 takes as input the linearization of the location-allocation problem (P) together with the following parameters. A positive integer, n, which corresponds to the initial cardinality of $\mathcal{J}$. Real numbers, T, and ϵ, which are the maximum allowable time and minimum objective function improvement allowed between successive solutions in the branch-and-cut tree of the MILP solver respectively before termination. Two real number numbers, $d_{\min}$ and κ, which are distance thresholds employed to manage the size of $|\mathcal{J}|$ in a location filtering process. The algorithm returns a solution $(f^*,\ (u^*, v^*, w^*))$ which is an upper bound on the objective value of the problem (6)–(13).

Algorithm 1 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

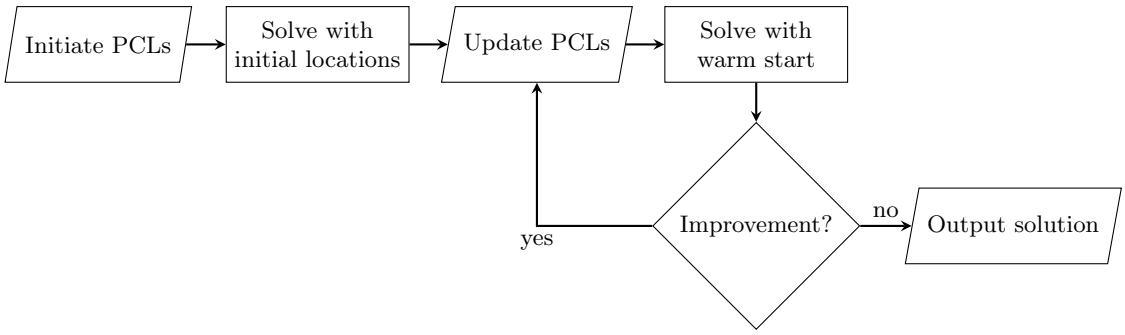


Figure 1: A flow chart of the two-stage solution approach to the location-allocation problem described in Algorithm 1.

process is then repeated until Boolean variable, Improvement, becomes *False*. First, 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 in each scenario, the number of chargers to install at each CS, and the objective function value, denoted by $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w})$. The function FIND_IMPROVED_LOCATIONS takes as input the solution returned by (P) and returns three sets. The first is $\{\mathcal{I}_{A_j}\}_{j \in \mathcal{J}_A}$, which is the set of EVs $i \in \mathcal{I}$ allocated to each built CSs which is denoted by $\mathcal{J}_A = \{j \in \mathcal{J} : \hat{v}_j = 1\}$. The second and third sets, denoted by $\mathcal{K}$ and $\mathcal{J}_R$ can be viewed as a tuple whereby every PCL $k \in \mathcal{K}$ has a corresponding CS $\mathcal{J}_R$ which it improves. 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 cardinality of $\mathcal{J}$ by only adding new locations to $\mathcal{J}$ if the ratio of PCLs to EVs, within a distance threshold, does not exceed a random uniform number, also described in more detail in §4.3. Thereafter, 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) by including the new improved locations, described in detail in §4.4. If the objective value corresponding to the warm start does not result in an improvement that is greater than ϵ, then the Boolean variable Improvement is switched to *False* and the algorithm terminates; otherwise, the iteration is over, and we repeat the process.

Next, we demonstrate the mechanics of Algorithm 1 by means of a simple example in Figure 2. A graphical representation of the solution returned from the function SOLVE_MILP is provided on the far left of Figure 2 where the diamonds represent the CS that are built and the edges illustrate the allocation of vehicles to CS represented by the black dots. Thereafter, the functions FIND_IMPROVED_LOCATIONS and FILTER_LOCATIONS are called and two improved locations are found, 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 the middle of Figure 2. In this Figure, we can see two new PCLs which where added to $\mathcal{J}$, and consequently one of these PCL was used as a CS. Thereafter, another iteration of the algorithm is completed, and a graphical representation of the solution returned from the function SOLVE_MILP is provided in the right of Figure 2. In this case another two PCLs where added to $\mathcal{J}$, however, this time both of the new CSs have EVs allocated to them. This procedure will continue until no further improvements are found and we have obtained a local minimum.

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 n which is the number of initial PCLs and a string '*type*' which specifies which procedure must be employed to determine the initial PCLs. 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, in our code we provide three options for the user to choose from: (i) PCLs generated at random, (ii) obtained with standard k-means clustering, or (iii) k-means constrained clustering, as outlined in Algorithm 2. Option (i) generally prompts an exploratory search with the aim of avoiding premature convergence to local minima. Locations are generated from from a smaller subset of the spatial domain, with corner points defined by the extremal points of the given

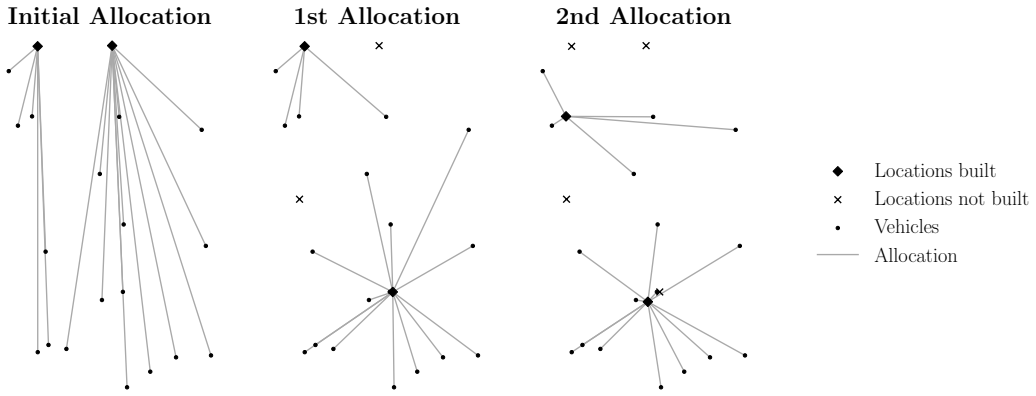


Figure 2: A graphical example of the mechanics of the location-allocation procedure defined by Algorithm 1. In this figure, the diamonds represent the CSs which are build, the crosses represent the PCLs which are not built, the dots represent the EVs and the edges represent the allocation of EVs to CSs.

EVs coordinates, that is, the rectangle $[x_0, x_N] \times [y_0, y_M] \subseteq [0, X] \times [0, Y] \subseteq \mathbb{R}^2$, with

$$x_0 = \min_{i \in \mathcal{I}} \{a_i^1\}, \quad x_N = \max_{i \in \mathcal{I}} \{a_i^1\},$$
$$y_0 = \min_{i \in \mathcal{I}} \{a_i^2\}, \quad y_M = \max_{i \in \mathcal{I}} \{a_i^2\}.$$

The second method for obtaining initial PCLs is k-means clustering, which partitions the set of EVs into n clusters such that a PCL is placed at the centroid 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: n : number of initial locations, '*type*' : random, k-means, or k-means constrained
Output $\mathcal{J}$: initial locations, $\boldsymbol{x}$: the locations of the initial locations

procedure INITIAL_LOCATIONS(n, '*type*')
  if '*type*' = random **then**
    $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n \leftarrow \mathbf{\Pi} \sim (\mathrm{U}(0,1), \mathrm{U}(0,1))$ ▷ Generate 2-dim random uniform tuples
    $\mathcal{J} \leftarrow \{j = 1, \ldots, n\}$
    $\boldsymbol{x}_j \leftarrow \boldsymbol{\pi}_j (x_N - x_0, y_M - y_0) + (x_0, y_0)\ \forall j \in \mathcal{J}$
  else if '*type*' = k-means **then**
    $\mathcal{J} \leftarrow \{j = 1, \ldots, n\}$
    $\boldsymbol{x}_j \leftarrow$ the centre points of k-means clusters
  else
    $\mathcal{J} \leftarrow \{j = 1, \ldots, n\}$
    $\boldsymbol{x}_j \leftarrow$ the centre points of k-means constrained clusters
  end if
  return $\mathcal{J}$, $\boldsymbol{x}$
end procedure

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), and real numbers, T and ϵ which are the maximum allowable time and minimum objective function improvement between successive solutions in the branch-and-cut tree of the MILP solver. The model (P) can be solved to near-optimality for reasonable choices of $\mathcal{J}$ using any commercial solver for MILPs. 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 solve the model until the next incumbent is obtained or the maximal allowable time between successive incumbents has elapsed. If we have find a new incumbent before the time limit has elapsed we then check if there is an improvement in the objective function value of

at least ϵ. If that is the case then we reset the clock and continue the solve, otherwise we end the solve and return the incumbent. The motivation for this is because we do not want to spend a lot of time solving the MILP in each iteration: in preliminary results we found that the greatest improvements in the objective function value are obtained from the FIND_IMPROVED_LOCATIONS heuristic. The pseudo code for the function SOLVE_MILP is outlined 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, ϵ : objective value improvement tolerance, $(\hat{u}, \hat{v}, \hat{w})$: feasible solution to (P),

Output: $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w})$: approximate solution to (P)

procedure SOLVE_MILP($(P), T, \epsilon$, $(\hat{u}, \hat{v}, \hat{w})$)
  $t \leftarrow 0$ ▷ Set clock to zero
  $f_{\text{old}} \leftarrow 0$
  if $(\hat{u}, \hat{v}, \hat{w}) \neq \emptyset$ **then** ▷ No warm start in the first iteration
    f_{WS}, $(\hat{u}, \hat{v}, \hat{w}) \leftarrow$ ADD_WARMSTART$(\hat{u}, \hat{v}, \hat{w})$
    $f_{\text{old}} \leftarrow f_{\text{WS}}$
  end if
  while $t \leq T$ **do**
    $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w}) \leftarrow$ SOLVE(P)
    if incumbent solution found **then**
      if $|f_{\text{old}} - \hat{f}| \leq \epsilon$ **then**
        $t \leftarrow T$ ▷ Insufficient improvement, break
      else
        $f_{\text{old}} \leftarrow \hat{f}$
        $t \leftarrow 0$ ▷ Reset the clock
      end if
    end if
  end while
  return $\hat{f}$, $(\hat{u}, \hat{v}, \hat{w})$
end procedure

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 algorithm 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 *i.e.* $\mathcal{J}_A = \{j \in \mathcal{J} : \hat{v}_j = 1\}$. Thereafter, the set of EVs allocated to each CS in $\mathcal{J}_A$ in every scenario $s \in \mathcal{S}$ which is the set $\mathcal{I}_{A_j} = \{i \in \{\mathcal{I}_s\}_{s\in\mathcal{S}} : \hat{u}^s_{i,j} = 1\}$ is computed. The location of the active CS $j \in \mathcal{J}_A$ can then be improved by the following procedure. If $\boldsymbol{g}$ denotes the coordinates of the geometric median of the EV locations for each $i \in \mathcal{I}_{A_j}$ and the euclidean distance from each $i \in \mathcal{I}_{A_j}$ to $\boldsymbol{g}$ is less than or equal to the range of each EV $i \in \mathcal{I}_{A_j}$, 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, improved location $\boldsymbol{g}$ corresponding to CS $j \in \mathcal{J}_A$. Otherwise, we solve the minimisation problem

$$\left.\begin{array}{lll} \min\limits_{\boldsymbol{x}_j} & f_{\text{dist}} = \sum\limits_{i\in\mathcal{I}_{A_j}} \tilde{d}_{i,j} & \\ \text{subject to} & \tilde{d}_{i,j} \leq r^s_i & \forall i \in \mathcal{I}_{A_j},\ s \in \mathcal{S} \\ & \tilde{d}_{i,j} = ||\boldsymbol{x}_j - \boldsymbol{a_i}|| & \forall i \in \mathcal{I}_{A_j} \end{array}\right\}, \qquad (D_j)$$

where the solution $\boldsymbol{x}^*_j$ is the geographical position of the new PCL that minimizes the distance to each of the EVs in $\mathcal{I}_{A_j}$ with the function called SOLVE_DIST. 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: $\mathcal{J}_R$: old active locations with improvements, $\mathcal{K}$ improved locations, $\{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}$: vehicles allocated to old charging stations

procedure FIND_IMPROVED_LOCATIONS($\hat{u}, \hat{v}$)
  $\mathcal{K} \leftarrow \emptyset$
  $\mathcal{J}_R \leftarrow \emptyset$
  $\mathcal{J}_A \leftarrow \{j \in \mathcal{J} : v_j = 1\}$ ▷ Find active CSs
  for $j \in \mathcal{J}_A$ **do**
    $\mathcal{I}_{A_j} \leftarrow \{i \in \{\mathcal{I}_s\}_{s\in\mathcal{S}} \ : \ \hat{u}^s_{i,j} = 1\}$ ▷ Find EVs allocated to CS j
    $\mathcal{A} \leftarrow \{\boldsymbol{a}_i : i \in \mathcal{I}_{A_j}\}$
    $\boldsymbol{g} \leftarrow$ GEOMETRIC_MEDIAN($\mathcal{A}$)
    if $||\boldsymbol{a}_i - \boldsymbol{g}|| \leq r^s_i$ for all $i \in \mathcal{I}_{A_j}$ and $s \in \mathcal{S}$ **then**
      $\mathcal{K} \leftarrow \mathcal{K} \cup \{|\mathcal{J}| + |\mathcal{K}| + 1\}$ ▷ Add new location
      $\mathcal{J}_R \leftarrow \mathcal{J}_R \cup \{j\}$
      $\boldsymbol{x}_{|\mathcal{J}|+|\mathcal{K}|+1} \leftarrow \boldsymbol{g}$
    else
      $\boldsymbol{x}^*_j \leftarrow$ SOLVE_DIST(D_j)
      $\mathcal{K} \leftarrow \mathcal{K} \cup \{|\mathcal{J}| + |\mathcal{K}| + 1\}$
      $\mathcal{J}_R \leftarrow \mathcal{J}_R \cup \{j\}$
      $\boldsymbol{x}_{|\mathcal{J}|+|\mathcal{K}|+1} \leftarrow \boldsymbol{x}^*_j$
    end if
  end for
  return $\mathcal{K}$, $\mathcal{J}_R$, $\{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}$
end procedure

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 PCLs, $\mathcal{J}$ and newly identified PCLs, $\mathcal{K}$ and if the Euclidean distance between, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$ for some $j \in \mathcal{J}$ and $k \in \mathcal{K}$, is less than a set distance, denoted by $d_{\min}$, then a Monte Carlo method is adopted to randomly remove some new locations. The allocation may be different in the next iteration, and therefore we may not wish to remove all such instances, but still maintain a diverse population of PCLs. For a new PCL $k \in \mathcal{K}$, if there are plenty of PCLs within a radius of κ and few EVs within the same radius expecting to use a CS, then the PCL $k \in \mathcal{K}$ is most likely not needed. Therefore, let C^k_{EV}, C^k_{CS} denote the number of EVs and PCLs within radius κ, respectively of new PCL $k \in \mathcal{K}$. This implies that the ratio,

$$\rho^k = \frac{\mathbb{E}(p)\, C_{\text{EV}}}{qm\, C_{\text{CS}}},$$

will assist in determining if another PCL, $k \in \mathcal{K}$, may be needed nearby. Recall that $\mathbb{E}(p)$ is the expected proportion of EVs needing to charge, and m is the maximum number of chargers at a CS and q is the maximum number of EVs that may be allocated to a CS. If ρ^k is small then we expect that the number of vehicles which require charging is less than the number of available chargers and therefore we expect that the new charger may not be required. Therefore, a random uniform number is generated $\zeta \sim \text{U}(0,1)$, and if $\zeta \geq \rho^k$, 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_WARM-START, for which the pseudo code provided in Algorithm 6, takes as input a feasible solution to (P),

Algorithm 5 Improved location filtering

Input: $\mathcal{J}_R$: old active locations with improvements, $\mathcal{J}$: locations, $\mathcal{K}$: new locations, $d_{\min}$: smallest distance allowed between CSs, κ : radius from CS

Output: $\mathcal{K}$: set of filtered locations, $\mathcal{J}_R$: old active locations with improvements

procedure FILTER_LOCATIONS($\mathcal{J}_R, \mathcal{J}, \mathcal{K}, d_{\min}, \kappa$)
  for $k \in \mathcal{K},\ j \in \mathcal{J}$ **do**
    if $||\boldsymbol{x}_j - \boldsymbol{x}_k|| \leq\ d_{\min}$ **then**
      $C^k_{\text{EV}} \leftarrow |\{i \in \mathcal{I} : d_{i,j} \leq \kappa\ \forall j \in \mathcal{J}\}|$ ▷ Number of EVs within given radius
      $C^k_{\text{CS}} \leftarrow |\{j \in \mathcal{J} : d_{i,j} \leq \kappa\ \forall i \in \mathcal{I}\}|$ ▷ Number of CSs within given radius
      $\rho^k \leftarrow \frac{\mathbb{E}(p)C_{\text{EV}}}{qmC_{\text{CS}}}$ ▷ Ratio of EVs chargers within radius
      Generate $\zeta \sim$ U(0,1)
      if $\zeta \geq \rho^k$ **then**
        $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k\}$ ▷ Controlled random removal of PCLs
        $\mathcal{J}_R \leftarrow \mathcal{J}_R \setminus \{j\}$ ▷ Keep a list of chargers with improvements
      end if
    end if
  end for
  return $\mathcal{K},\ \mathcal{J}_R$
end procedure

the old set of active locations with improvements, $\mathcal{J}_R$, 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. First, the Euclidean distance $d_{i,k}$ from each EV $i \in \mathcal{I}$, to the new set PCLs $k \in \mathcal{K}$, is computed. For each new PCL $k \in \mathcal{K}$ the variable v_k is set to one 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 $j \in \mathcal{J}_R$ 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, $j \in \mathcal{J}_R$, is now allocated to the new 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), $\mathcal{J}_R$: old active locations with improvements, $\{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}$: old set of allocated EVs, $\mathcal{K}$: new locations

Output: $f_{WS}, (\hat{u}, \hat{v}, \hat{w})$: updated warm start solution

procedure CONSTRUCT_WARMSTART($(\hat{u}, \hat{v}, \hat{w}), \mathcal{J}_R, \{\mathcal{I}_{A_j}\}_{j\in\mathcal{J}_A}, \mathcal{K}$)
  for $(k, j) \in (\mathcal{K}, \mathcal{J}_R)$ **do**
    $\hat{v}_k \leftarrow 1$ ▷ New location made active
    $\hat{w}_k \leftarrow \hat{w}_j$ ▷ Assume the same amount of chargers
    $\hat{v}_j \leftarrow 0$ ▷ Old CSs which have improved locations are removed
    $\hat{w}_j \leftarrow 0$
    for $s \in \mathcal{S}$ **do**
      for $i \in \mathcal{I}_{A_j}$ **do**
        $\hat{u}^s_{i,j} \leftarrow 0$ ▷ EVs are no longer allocated CSs which have improved locations
        if $d_{i,k} \leq r^s_i$ **then**
          $\hat{u}^s_{i,k} \leftarrow 1$ ▷ If $d_{i,k} > r^s_i$, raise error, not reachable
        end if
      end for
    end for
  end for
  $f_{\text{WS}} \leftarrow f_L(\hat{u}, \hat{v}, \hat{w})$ ▷ Update objective function value
  return $f_{\text{WS}},\ (\hat{u}, \hat{v}, \hat{w})$
end procedure

5. Computational experiments and results

In this section we perform computational experiments to test the effect of parameters required as input for Algorithm 1 on the trade-off between solution quality, solution time and solution robustness for different problem sizes. More specifically, we run two different experiments. In the first experiment we choose different values of T, the maximum allowable time between two successive incumbent solutions in the branch-and-cut tree in Algorithm 3 for different choices of n, the number of initial starting locations. In the second experiment we test the robustness of the solutions obtained for a different number of samples $|\mathcal{S}|$. For these computational results we draw random subsets from the data provided for the MOPTA 2023 modelling competition. The entire dataset contains contains 1 079 coordinates of EV locations such that $X = 290$ miles, $Y = 150$ miles. The locations of the EVs are illustrated graphically in Figure 3.

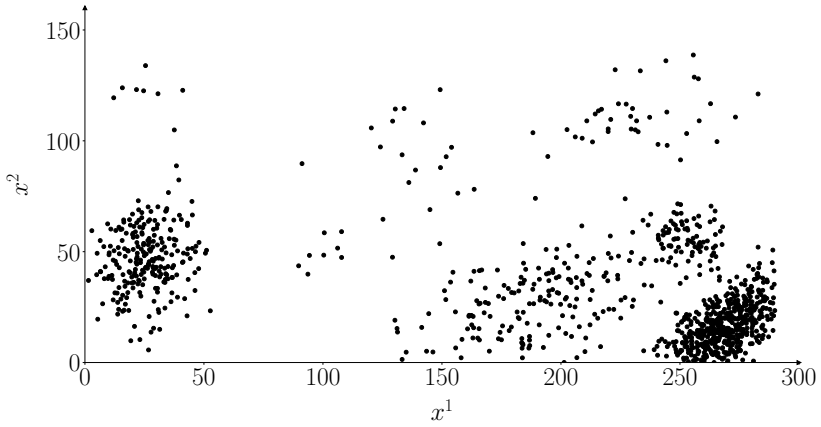


Figure 3: A graphical illustration of the 1 079 EV locations in the $[0, 290] \times [0, 150]$ plane.

In these computational experiments, we let $\alpha = 0.95$ and we use the costs $c_b = 5000, c_m = 500, c_d = 0.041, c_{\tilde{c}} = 0.0388$. We also let $m = 8$ and $q = 16$. That is, at most eight chargers can be built, and at most 16 EVs can be allocated to charging station. We 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 $r_i^s \sim \mathrm{N}(100, 50^2)$, truncated with bounds between $r_{\min} = 20$ and $r_{\max} = 250$ miles. Therefore, the probability that an EV $i \in \mathcal{I}$ requires charging in scenario $s \in \mathcal{S}$ is therefore defined as $p_i^s(r_i^s) = \exp(-\lambda^2(r_i^s - 20)^2)$ where we let $\lambda = 0.012$. Moreover, after running multiple iterations obtained by employing the Monte Carlo Method described in §3.2, the mean probability that an EV is required to charge, and hence the expected proportion of EVs which are required to charge in each scenario, is observed to be approximately 0.42. This follows from numerically evaluating (3) that

$$\begin{aligned}\mathbb{E}[p(r)] &= \int_{20}^{250} e^{-(0.012)^2(r-20)^2} \frac{e^{-\frac{1}{2}\left(\frac{r-100}{50}\right)}}{50\sqrt{2\pi}(\Phi(\frac{250-100}{50}) - (\Phi(\frac{20-100}{50}))} \,\mathrm{d}r \\ &\approx 0.42.\end{aligned}$$

where Φ is the cumulative distribution function of the standard normal distribution. For the remainder of this section we keep all of these parameters the same.

All of the code was written in the Python programming language, we solved the MILP's using CPLEX 20.1.0.0, and the computations where run on a a Windows OS machine with a 11th Gen Intel® Core™ i7-1185G @3.00GHz, with 4 Cores, 8 Logical processors and 32GB RAM.

5.1. Experiment 1: Impact of T and n on Algorithm 1 runtime and solution quality.

In this set of experiments we fix $\epsilon = 100$, $d_{\min} = 0.5$, $\kappa = 10$ and optimise over $|\mathcal{S}| = 5$ scenarios. We solve the problem for three different problem sizes where we randomly select, with a fixed seed, 50%, 75% and 100% of the 1 079 EVs corresponding to $|\mathcal{I}| = \{540, 801, 1\,079\}$ respectively. For each of the above problem sizes perform a full factorial design employing the values of

$$n = \left\lceil \frac{\gamma\,|\mathcal{I}|\,\mathbb{E}[p(r)]}{q} \right\rceil \tag{15}$$

for each $\gamma \in \{2, 3, 4, 10\}$ and $T \in \{60, 300, 900, 1\,800, 3\,600\}$ resulting in a total of 60 experiments. Note that in (15) the numerator is a multiple of the expected number of EVs which will require charging in each scenario. Dividing by the maximum number of EVs which can be allocated to each CS scales this down to an appropriate starting number of PCLs. Concretely, if $\gamma = 1$, then this is a lower bound on the number of CSs required to charge every required EV, assuming all have sufficient range. Moreover, when we populate the locations of the PCLs we also use a fixed seed and therefore for every unique pair $(|\mathcal{I}|, n)$ we have the same set of PCLs $\mathcal{J}$. The results of these experiments are found in Table 3 and a solution for the case where $|\mathcal{I}| = 1\,079$ is illustrated graphically in Figure 4.

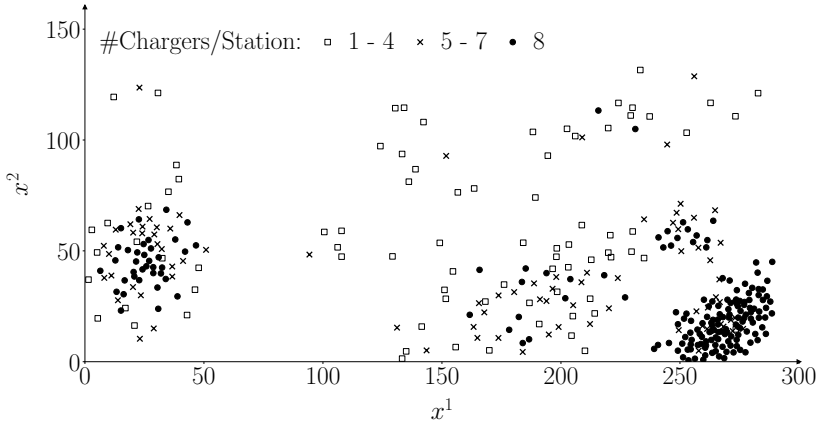


Figure 4: A graphical illustration of the construction plan for the 347 CSs and 2 221 chargers. In this figure, the filled dots correspond to the locations where 8 chargers are built, the crosses correspond to the locations where between 5 and 7 chargers are built and the squares correspond to the locations where 1-4 chargers are built.

In Table 3 we recorded the objective function value, the total solve time, the MIP gap of the final MIP solve, the number of iterations of Algorithm 1 required before convergence and the size of the final set of PCLs, denoted by $|\mathcal{J}|$. For each unique $(|\mathcal{I}|, n)$ pair we indicated the choice of T which resulted in the best objective function value and solve time by writing their values in bold. Moreover, for each choice of $|\mathcal{I}|$ we indicated the best objective function value and solve time by highlighting the associated (T, n) pair row in a dark and light shades of grey, respectively.

Table 3: Computational results for the first experiment.

T	Objective value	Solve time	MIP gap	Iterations	n	$\lvert\mathcal{J}\rvert$
$\lvert\mathcal{I}\rvert = 540$						
60	**745 549**	**692.79**	0.37%	4	29	69
300	745 935	1 967.72	0.39%	4	29	80
900	745 876	7 357.73	0.36%	4	29	77
1 800	745 870	10 659.39	0.36%	4	29	77
3 600	746 820	13 166.84	0.35%	3	29	65
60	745 781	**724.09**	0.38%	3	43	78
300	745 893	3 309.44	0.35%	4	43	80
900	745 861	7 174.83	0.34%	4	43	80
1 800	**745 637**	10 360.53	0.27%	3	43	72
3 600	745 976	14 191.67	0.26%	2	43	67
60	**745 971**	**1 240.60**	0.42%	3	57	93
300	746 565	2 628.41	0.38%	3	57	92
900	746 693	4 309.02	0.40%	2	57	85
1 800	746 346	11 415.00	0.33%	2	57	85
3 600	746 499	18 582.18	0.38%	3	57	92
60	746 636	**639.88**	1.27%	4	142	172
300	746 935	1 845.53	0.57%	2	142	163
900	746 117	11 253.26	0.46%	2	142	166

Continued on next page

Table 3 – continued from previous page

T	Objective value	Solve time	MIP gap	Iterations	n	$\lvert\mathcal{J}\rvert$
1 800	**745 235**	25 401.47	0.36%	3	142	169
3 600	745 235	31 167.06	0.36%	3	142	169
$\lvert\mathcal{I}\rvert = 801$						
60	1 109 915	**366.43**	0.10%	3	43	94
300	1 109 042	2 367.01	0.06%	3	43	94
900	1 109 081	3 264.76	0.06%	3	43	92
1 800	**1 109 030**	6 480.56	0.05%	3	43	94
3 600	1 109 030	12 420.66	0.05%	3	43	94
60	**1 108 891**	**656.41**	0.07%	4	64	134
300	1 109 128	2 912.27	0.06%	3	64	112
900	1 109 128	4 702.48	0.05%	3	64	112
1 800	1 109 166	7 414.03	0.05%	3	64	109
3 600	1 109 166	13 012.46	0.05%	3	64	109
60	1 108 975	**487.43**	0.13%	3	86	132
300	1 108 636	6 489.18	0.08%	3	86	135
900	1 108 783	9 821.22	0.10%	3	86	140
1 800	**1 108 533**	14 520.05	0.06%	3	86	133
3 600	1 108 594	27 382.42	0.07%	3	86	130
60	1 116 743	**404.45**	1.23%	2	213	245
300	1 109 633	2 258.61	0.19%	2	213	246
900	**1 108 304**	13 780.83	0.10%	3	213	255
1 800	1 108 589	32 319.84	0.13%	3	213	256
3 600	1 108 374	51 491.60	0.09%	3	213	260
$\mathcal{I} = 1\,079$						
60	1 480 790	**1 016.12**	0.02%	4	57	139
300	1 480 765	2 318.57	0.01%	4	57	137
900	1 480 779	3 530.93	0.01%	4	57	133
1 800	1 480 765	3 786.50	0.01%	4	57	137
3 600	**1 480 565**	9 389.08	0.01%	6	57	155
60	1 480 476	**1 246.10**	0.02%	3	86	142
300	1 480 245	4 370.68	0.02%	4	86	150
900	1 480 746	7 167.58	0.01%	2	86	127
1 800	**1 479 951**	12 415.80	0.01%	3	86	145
3 600	1 480 243	18 932.00	0.01%	3	86	147
60	1 481 770	**818.59**	0.09%	2	114	154
300	1 480 174	2 072.57	0.02%	3	114	170
900	1 480 085	9 298.53	0.02%	6	114	194
1 800	1 480 677	29 557.89	0.01%	2	114	154
3 600	**1 480 078**	28 977.07	0.01%	3	114	172
60	1 590 290	**351.68**	7.87%	2	284	343
300	1 500 728	4 522.15	1.52%	4	284	351
900	1 487 770	11 223.65	0.64%	6	284	359
1 800	1 487 806	6 013.26	0.58%	2	284	319
3 600	**1 480 271**	32 940.45	0.06%	2	284	312

From these results, we conclude the following. The total solution time is significantly more sensitive to changes in T than to changes in n. In fact, as we would expect, for every unique $(|\mathcal{I}|, n)$ pair, $T = 60$ always resulted in the fastest solution time. This however, comes at the cost of solution quality because for every unique $(|\mathcal{I}|, n)$, except when $(|\mathcal{I}|, n)$ is $(|801|, 64)$ and $(|801|, 213)$, the lowest objective function value was always achieved when $T \geq 1\,800$. In general, larger choices of n are better suited in terms of objective function value when T was large and smaller choices of n better suited in terms of objective function value when T was small. It is also worth noting that when increasing T the improvements to the objective function value are small when compared to the increase in solution time. This is because when we increase T we spend more time in the branch-and-cut algorithm trying

to find improved solutions to each of the subproblems. It seems as if it is more efficient to spend less time in the branch-and-cut solver and rather run the improve location heuristic which takes less time and results in good solutions.

Once the solver has completed, the user may employ the play and pause buttons enable the user to step through the evolution of the algorithm one iteration at a time to observe how the construction plan was improved at each iteration. The metric panels report on the final solution.

5.2. Experiment 2: Impact of $|S|$ on Solution Robustness

After obtaining a good solution, (u^*, v^*, w^*) using Algorithm 1 across multiple scenarios, we want to validate the result by testing the model output on new realisations of sets of EVs. We do this by taking the EV build decisions (v^*, w^*) and checking if a feasible solution exists for unseen scenarios. If a feasible solution exists, we determine the resulting objective function value and if a feasible solution does not exist, what is the best service level that we can obtain from the EV build decisions (v^*, w^*). When performing the Monte Carlo Method described in §3.2, we obtain a collection of Boolean values, $\{\delta_i^s\}^{s\in\mathcal{S},i\in\mathcal{I}}$, describing which EVs need to charge in each scenario $s \in \mathcal{S}$. These are correlated with the EV driving ranges, $\{r_i^s\}^{s\in\mathcal{S},i\in\mathcal{I}}$, generated from a given truncated normal distribution. Adopting a statistical mechanics interpretation, the pairs $\{(\delta_i^s, r_i^s)\}^{s\in\mathcal{S},i\in\mathcal{I}}$ make up an ensemble of all possible configurations of the system of EVs [47]. There are finitely many configurations of the EVs charge requirements as each vehicle either needs to charge or doesn't, but there are infinitely many virtual copies of the ensemble since the driving 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.

To evaluate the robustness of a solution for a given choice of $|\mathcal{S}|$, a re-allocation is performed over a set of $|\mathcal{S}_{\text{v}}|$ unseen realisations of the Monte Carlo-type simulation. That is, having established both where to place CSs v^*, and how many chargers to install at each CS w^*, we test the solution on new realisations of EVs taken from the ensemble $\{(\delta_i^s, r_i^s)\}^{s\in\mathcal{S}_{\text{v}},i\in\mathcal{I}}$. Since we set $v = v^*$ and $w = w^*$ from MILP (P), only the EV allocation variables $u_{i,j}^s$ remain to be solved for each new realisation $s \in \mathcal{S}_{\text{v}}$.

The resulting allocation problem for each realisation $s \in \mathcal{S}_{\text{v}}$ is therefore

$$
\left.
\begin{aligned}
\min \quad & 365(c_{\bar{c}} + c_d) \sum_{i\in\mathcal{I}_s} \sum_{j\in\mathcal{J}_A} d_{i,j} u_{i,j}^s + K && \\
\text{subject to} \quad & \sum_{i\in\mathcal{I}_s} u_{i,j}^s \le q w_j^* && j \in \mathcal{J}_A, \\
& \sum_{j\in\mathcal{J}_A} u_{i,j}^s \le 1 && i \in \mathcal{I}_s, \\
& \sum_{(i,j)\in\Omega_s} u_{i,j}^s \ge \alpha |\mathcal{I}_s| && \\
& u_{i,j}^s \in \{0,1\} && (i,j) \in \Omega_s,
\end{aligned}
\right\} \qquad (Q)
$$

where

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

is a constant corresponding to the building costs from the location-allocation problem (P), $\mathcal{I}_s = \{i \in \mathcal{I} \ : \ \delta_i^s = 1\}$ and $\Omega_s = \{(i,j) \in (\mathcal{I}_s, \mathcal{J}_A) : d_{i,j} \le r_i^s\}$ and $\mathcal{J}_A = \{j \in \mathcal{J} : v_j^* = 1\}$. Lets consider ($Q$), and let $\mathcal{P}_Q$ denote the polytope corresponding to the feasible region of (Q). If $\mathcal{P}_Q \neq \emptyset$ then clearly (Q) admits at least one optimal solution. If $\mathcal{Q} = \emptyset$ then

$$
\mathcal{P}_Q \setminus \sum_{(i,j)\in\Omega_s} u_{i,j}^s \ge \alpha |\mathcal{I}_s| \neq \emptyset
$$

and we have an objective function value of K. This therefore implies that if (Q) is infeasible for any $s \in \mathcal{S}_{\text{v}}$ then for the given charging infrastructure (v^*, w^*) we can not achieve the α service level requirement. In this case the highest attainable service level is computed by finding a maximum bipartite matching, denoted by α_s, between the EVs that require charging $i \in \mathcal{I}_s$ and their reachable CS. We then replace α by α_s in (Q) and find the optimal allocation.

In this set of experiments we fix $\epsilon = 100$, $d_{\min} = 0.5$, $\kappa = 10$, $T = 60$ and $n = 63$. We then consider the choices of $|\mathcal{S}| \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. For each choice of the number of samples, we

employ Algorithm 1 to solve the problem ten times, each time for a new realisation of the choice of $|\mathcal{S}|$. For each of these solutions we create 100 unseen scenarios such that $|\mathcal{S}_{\text{v}}| = 100$ and solve (Q) for each scenario $s \in \mathcal{S}_{\text{v}}$ for the maximum attainable service level α_s. For each of the 100 optimization runs, corresponding to the ten runs for the ten scenario sizes, we recorded the build cost, drive cost and the MIP gap of the final location allocation solve. Then, for each sample size we compute the mean MIP gap, build cost and drive cost, and the standard deviation of the build cost and drive cost across the ten optimization runs. These results are recorded in the optimization columns of Table 4. Moreover, for each of the 10 000 validation runs, corresponding to the 100 scenarios for each of the 100 optimization runs mentioned earlier we recorded the the build cost, drive cost and best attainable service level. Then, for each sample size we compute the mean and standard deviation of the build cost, drive cost and best attainable service level across the 1 000 validation runs. These results are recorded in the validation columns of Table 4. Finally, for each choice of $|\mathcal{S}|$ we recorded the number of validation runs which could obtain the required service level of $\alpha_s = 0.95$ and computed the proportion of the 1 000 validation runs which are feasible, illustrated graphically in Figure 5(e).

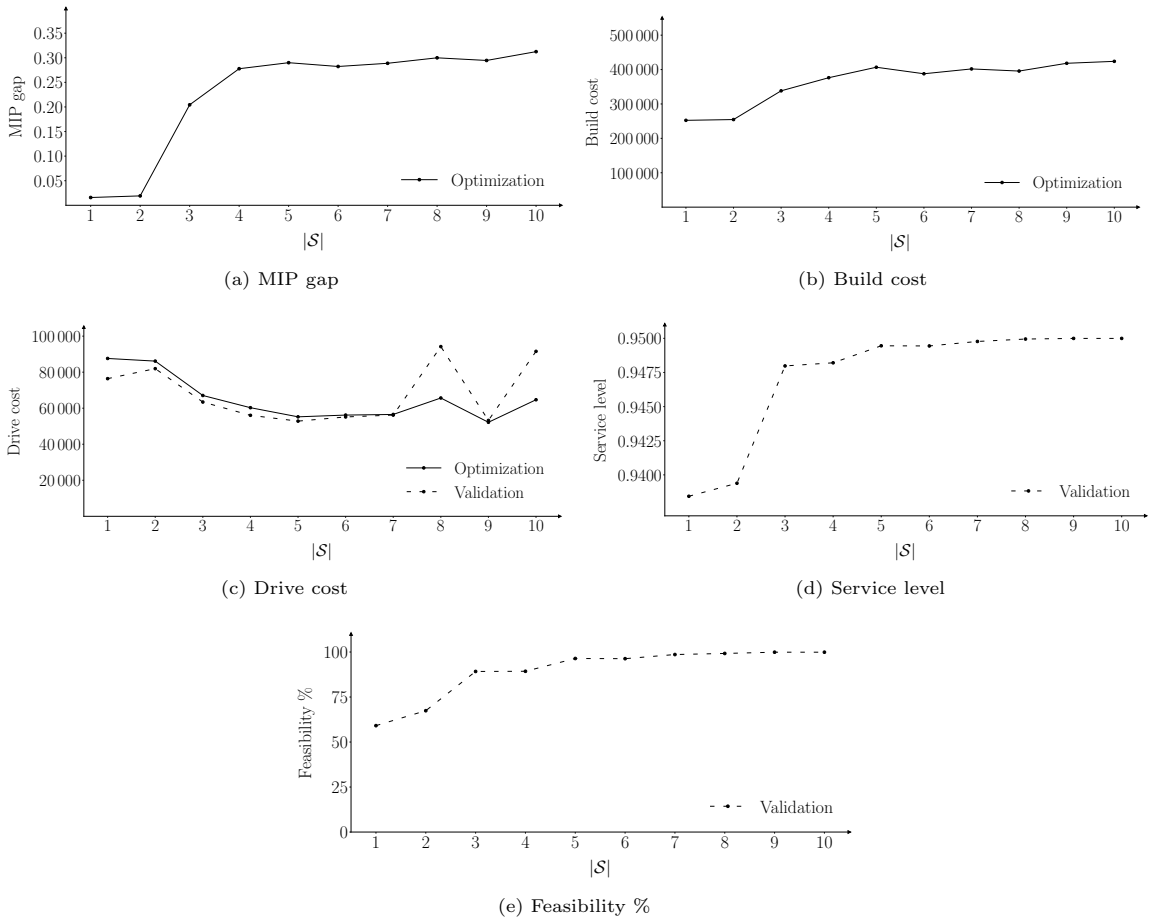
Table 4: A summary of the results obtained from the 10 000 validation runs.

$\lvert\mathcal{S}\rvert$	Optimization						Validation			
	Build cost		Mip gap (%)		Drive cost		Drive cost		Service level	
	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev
1	252 400	10 249	1.58	0.28	76 412	4 368	87 613	12 423	0.9384	0.0187
2	254 650	11 688	1.91	0.73	81 954	3 582	86 145	10 929	0.9394	0.0207
3	338 200	28 416	20.43	8.88	63 428	8 806	67 064	10 311	0.9480	0.0081
4	376 300	27 582	27.78	1.09	56 055	4 292	60 285	9 060	0.9482	0.0073
5	406 750	28 136	29.00	1.08	52 832	3 402	55 219	8 606	0.9495	0.0039
6	387 800	20 842	28.23	0.76	55 153	2 942	56 188	6 503	0.9494	0.0035
7	401 850	36 529	28.88	1.53	56 191	4 883	56 539	7 349	0.9498	0.0035
8	395 650	37 258	29.99	3.92	94 203	119 438	65 669	33 704	0.9499	0.0007
9	418 250	283 55	29.45	1.16	53 234	4 060	52 186	5 804	0.9500	0.0002
10	423 900	25 708	31.25	3.45	91 564	114 047	64 696	41 865	0.9500	0.0002

From these results, we observe the following. The first observation is that (P) becomes increasingly difficult to solve as $|\mathcal{S}|$ increases. This is because as we increase $|\mathcal{S}|$ the average MIP gap also increases, illustrated graphically in Figure 5(a). Moreover, since the standard deviation of the MIP gap is relatively small compared to the mean, then for a given choice of $|\mathcal{S}|$, all of the ten realisations obtained a similar MIP gap. The second observation is that as $|\mathcal{S}|$ increases so too does the solution solution robustness increases. This is demonstrated by the fact that as $|\mathcal{S}|$ increases the average service level and feasibility percentage of the 100 validation runs also increases, illustrated graphically in Figures 5(d) and (e). This increased robustness, however, comes at a cost. That is, as $|\mathcal{S}|$ increases so to does the build cost of charging infrastructure, illustrated graphically in Figure 5(b). Intuitively, this makes sense because as we consider more scenarios we will need charging infrastructure in more locations to meet the demand across each of the scenarios. We also note that a portion of the higher costs can also be attributed to the high MIP gap. The final observation is related to the drive cost for both the optimisation and validation runs. In general, the drive cost for the optimisation validation runs where generally similar as illustrated in Figure 5(c). There are two outliers, $|\mathcal{S}| \in \{8, 10\}$, where the difference is significantly larger. We draw this down to the high standard deviation of the optimisation and validation drive costs relative to the mean. This implies that the average is skewed. Nevertheless, for the other choices of $|\mathcal{S}|$ we observe that as $|\mathcal{S}|$ increases, the drive cost tends to decrease. This intuitively makes sense for exactly the same reason as the improved service levels.

Next, we compute the 95% confidence intervals for the validation results. Let $Z_1^{\text{d}}, Z_2^{\text{d}}, ..., Z_{|\mathcal{S}_{\text{v}}|}^{\text{d}}$ denote the drive cost for each of the $|\mathcal{S}_{\text{v}}|$ validation runs for one choice of $|\mathcal{S}|$. These are independent and identically distributed in terms of the ensemble of demand configurations $\{(\delta_i^s, r_i^s)\}_{s\in\mathcal{S}_{\text{v}}, i\in\mathcal{I}}$, with mean, $\mathbb{E}^{\text{d}}(Z_i)$, and variance, σ_{d}^2. By the central limit theorem, the sample mean, $\bar{Z}^{\text{d}} = \sum_{i=1}^{|\mathcal{S}_{\text{v}}|} Z_i/|\mathcal{S}_{\text{v}}|$, should follow a standard normal distribution $\bar{Z}^{\text{d}} \sim \mathcal{N}(\bar{Z}^{\text{d}}, \sigma_{\text{d}}^2/|\mathcal{S}_{\text{v}}|)$. A confidence interval for the mean cost can then be calculated as

$$\left(\bar{Z}^{\text{d}} - \frac{t_{|\mathcal{S}_{\text{v}}|-1,1-\alpha/2}\,\sigma_{\text{d}}}{\sqrt{|\mathcal{S}_{\text{v}}|}}, \bar{Z}^{\text{d}} + \frac{t_{|\mathcal{S}_{\text{v}}|-1,1-\alpha/2}\,\sigma_{\text{d}}}{\sqrt{|\mathcal{S}_{\text{v}}|}}\right),$$



(a) MIP gap

(b) Build cost

(c) Drive cost

(d) Service level

(e) Feasibility %

Figure 5: A graphical illustration of the results obtained from the second experiments, corresponding to the different choices of $|\mathcal{S}|$.

where t is the test statistic on a $(1-\alpha)\cdot 100\%$ confidence interval with $|\mathcal{S}_{\mathrm{v}}|-1$ degrees of freedom, and σ is the standard deviation of each Z_i^{d}. For the normal distribution, the cumulative distribution gives that $P(-1.96 < \bar{Z}^{\mathrm{d}} < 1.96) = 0.95$, so we set $t = 1.96$ as we use a large number of validation runs. Hence the confidence intervals are calculated as follows

$$\left(\bar{Z}^{\mathrm{d}} - \frac{1.96\,\sigma_{\mathrm{d}}}{\sqrt{|\mathcal{S}_{\mathrm{v}}|}}, \bar{Z}^{\mathrm{d}} + \frac{1.96\,\sigma_{\mathrm{d}}}{\sqrt{|\mathcal{S}_{\mathrm{v}}|}}\right).$$

The one-sided confidence intervals for the service level are obtained in a similar way,

$$\left(\bar{Z}^{\mathrm{s}} - \frac{1.96\,\sigma_{\mathrm{s}}}{\sqrt{|\mathcal{S}_{\mathrm{v}}|}}, 0.95\right),$$

where the mean $\bar{Z}^{\mathrm{s}}$ and standard deviation σ_{s} now correspond to the service level obtained from the validation runs, with a maximal attainable service level of 0.95. The resulting confidence intervals are recorded in Table 5, using $|\mathcal{S}_{\mathrm{v}}| = 1\,000$ validation runs. From the right column, we see that the one-sided confidence interval for the service gets significantly narrower for the many-sample problem. This corresponds to the increase in solution robustness. Similarly, the confidence intervals become narrower for the drive costs with larger sample sizes, apart from the two outliers $|\mathcal{S}| \in \{8, 10\}$.

6. Graphical user interface

To make this model and solution approach more accessible to the user, we embedded the model in a Graphical User Interface. The interface was developed using streamlit.io an open source *Graphical*

Table 5: The confidence intervals for validation results corresponding to each choice of $|\mathcal{S}|$

$\lvert\mathcal{S}\rvert$	Drive cost	Service level
1	(86 843, 88 383)	(0.9373, 0.9500)
2	(85 468, 86 827)	(0.9381, 0.9500)
3	(66 425, 67 703)	(0.9475, 0.9500)
4	(59 723, 60 847)	(0.9478, 0.9500)
5	(54 685, 55 752)	(0.9492, 0.9500)
6	(55 785, 56 591)	(0.9492, 0.9500)
7	(56 084, 56 995)	(0.9496, 0.9500)
8	(63 580, 67 758)	(0.9499, 0.9500)
9	(51 826, 52 546)	(0.9500, 0.9500)
10	(62 101, 67 291)	(0.9500, 0.9500)

User Interface (GUI) development suite. The GUI will allow the user to use any of the datasets employed in the computational experiments in §5 or upload their own new dataset. Moreover, the GUI also allows the user to make changes to the model parameters of §3 and parameters required as input for Algorithm 1. The user will then be able to run the solution algorithm. An example of the GUI out is demonstrated in Figures 6 and 7, where the vehicle locations together with the locations of the chargers are illustrated in the prior and some summary statistics are illustrated in the latter.

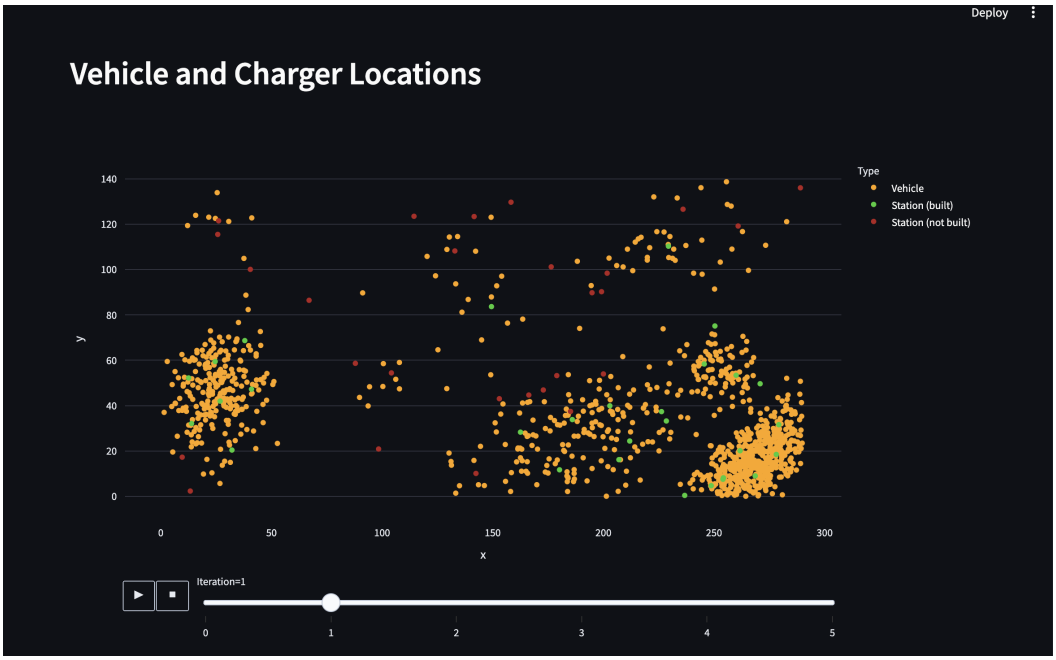


Figure 6: A graphical illustration of the solution representation returned in the GUI.

Once the solver has completed, the user may employ the play and pause buttons to step through the progression of the algorithm one iteration at a time to observe how the construction plan was improved at each iteration. Thereafter, the user is able to directly run the validation procedure described in §5.2. An example output of the validation process is illustrated graphically in Figure 8. The interface, along with our code and some set up instructions can be found in the following github repository https://github.com/justuskilianwolff/ev-station-solver.

7. Conclusion

In this paper, we proposed a novel model formulation and accompanying heuristic solution approach for solving a continuous facility location problem demand under uncertainty in the context of EV charging infrastructure. The paper opened in §2 with a brief serve of the literature in an

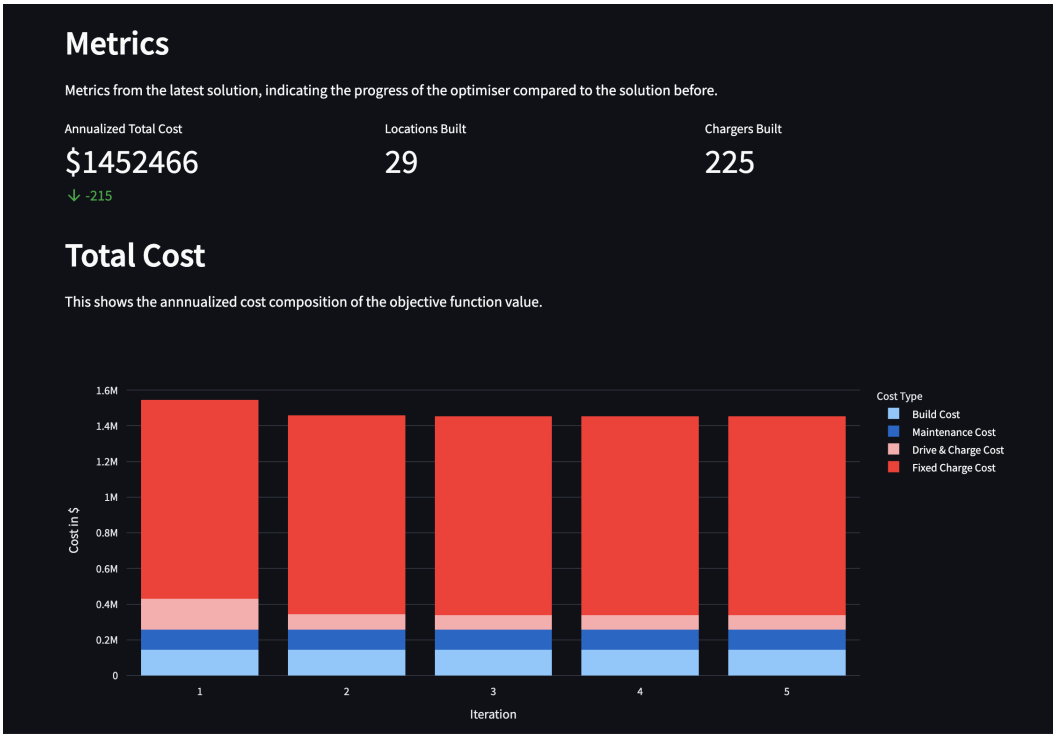


Figure 7: A graphical illustration of the summary statistics returned in the GUI.

attempt to provide context to the contributions of this paper. The model, derived in §3 takes the form of a nonlinear, two-stage robust location-allocation optimisation model where for a given set of customers, with uncertain demand, EV CS must be placed anywhere in the geographic region under consideration such that the location and allocation costs are minimised. The location cost takes the form of an annualised build cost and a yearly maintenance cost and the allocation cost is the cost associated with the EV users driving to their allocated CS. The model proposed is notorious difficult to solve, especially for problem instances considered in this paper and therefore we proposed a heuristic solution approach in §4. The solution approach was inspired by the famous Cooper location allocation algorithm [14] whereby we initialised a set of PCLs in which case our model becomes a MILP. Once we obtained a solution to this model, we then identified new PCLs which may improve the objective function value. In an attempt to reduce the number of PCLs we introduced a novel filtering heuristic. Thereafter, in §5, we performed two computational experiments. In the first, we tested a parameter which sets the size of the initial PCLs and a parameter which sets the maximal allowable time between incumbent solutions in the branch-and-cut tree. From this experiment we found that if solution quality is important, then the user should use a large number of initial PCLs and allow a large period of time between successive solutions in the branch-and-cut tree. We also observed that in this case the improvement in objective function value is relatively small when compared the parameter choice where the number of initial PCLs and time between successive solutions in the branch-and-cut tree is small. In the second experiment we tested the robustness of solutions returned from a different number of scenarios employed in the robust optimisation framework. We found that as the number of scenarios increase, the solution becomes more robust to unseen scenarios. This, however, comes at an additional cost of building more CSs.

References

[1] Anjos MF, Gendron B & Joyce-Moniz M, 2020, *Increasing electric vehicle adoption through the optimal deployment of fast-charging stations for local and long-distance travel*, European Journal of Operational Research, **285(1)**, pp. 263–278, Available from https://www.sciencedirect.com/science/article/pii/S0377221720300928.

[2] Aragón FJ, Goberna MA, López MA & Rodríguez MML, 2019, *Unconstrained Optimization Algorithms*, pp. 183–252 in Aragón FJ, Goberna MA, López MA & Rodríguez MM

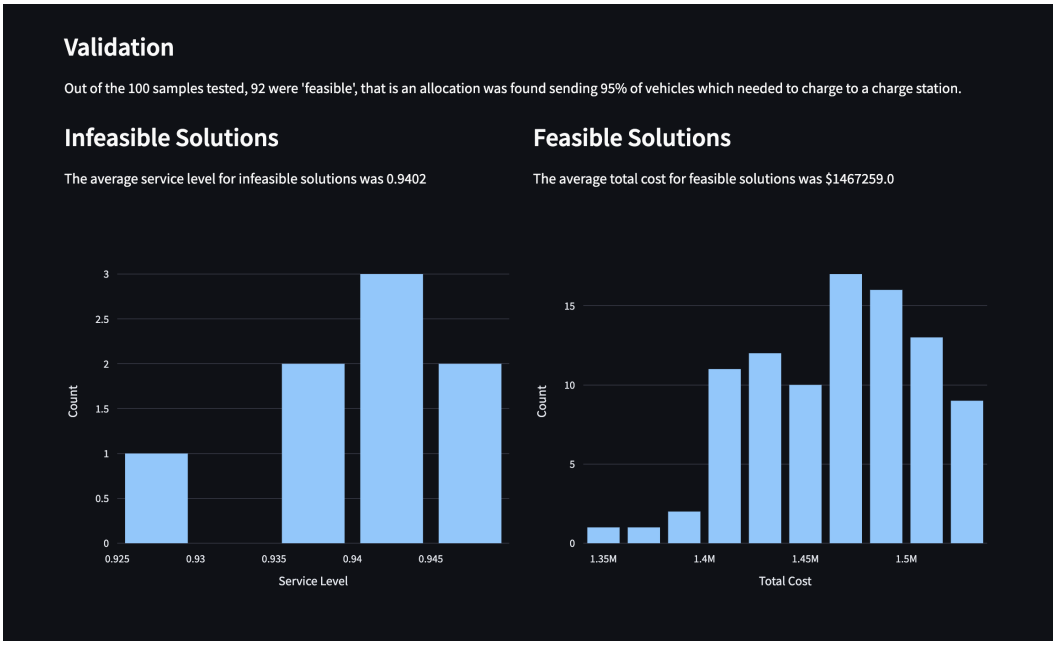


Figure 8: A graphical illustration of the validation results returned in the GUI.

(Eds), *Nonlinear Optimization*, Springer Undergraduate Texts in Mathematics and Technology, pp. 183–252. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-030-11184-7_5.

[3] Blanco V & Gázquez R, 2021, *Continuous maximal covering location problems with interconnected facilities*, Computers & Operations Research, **132**, p. 105310, Available from https://linkinghub.elsevier.com/retrieve/pii/S0305054821000988.

[4] Blanco V, Marín A & Puerto J, 2024, *Intra-facility equity in discrete and continuous p-facility location problems*, Computers & Operations Research, **162**, p. 106487, Available from https://www.sciencedirect.com/science/article/pii/S0305054823003519.

[5] Brimberg J & Drezner Z, 2013, *A new heuristic for solving the p-median problem in the plane*, Computers & Operations Research, **40(1)**, pp. 427–437, Available from https://www.sciencedirect.com/science/article/pii/S0305054812001578.

[6] Brimberg J, Drezner Z, Mladenovic N & Salhi S, 2017, *Using injection points in reformulation local search for solving continuous location problems*, Yugoslav Journal of Operations Research, **27(3)**, pp. 291–300, Available from https://doiserbia.nb.rs/Article.aspx?ID=0354-02431600018B.

[7] Brimberg J, Hansen P & Mladenovi N, 2008, *A Survey of Solution Methods for the Continuous Location-Allocation Problem*, **5(1)**.

[8] Brimberg J, Hansen P & Mladenović N, 2006, *Decomposition strategies for large-scale continuous location–allocation problems*, IMA Journal of Management Mathematics, **17(4)**, pp. 307–316, Available from https://ieeexplore.ieee.org/document/8132342, conference Name: IMA Journal of Management Mathematics.

[9] Brimberg J, Hansen P, Mladenović N & Taillard ED, 2000, *Improvements and Comparison of Heuristics for Solving the Uncapacitated Multisource Weber Problem*, Operations Research, **48(3)**, pp. 444–460, Available from https://pubsonline.informs.org/doi/10.1287/opre.48.3.444.12431, publisher: INFORMS.

[10] Cavadas J, Homem de Almeida Correia G & Gouveia J, 2015, *A MIP model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours*, Transportation Research Part E: Logistics and Transportation Review, **75**, pp. 188–201, Available from https://www.sciencedirect.com/science/article/pii/S136655451400194X.

[11] Chen TD, Kockelman KM & Khan M, 2013, *Locating Electric Vehicle Charging Stations: Parking-Based Assignment Method for Seattle, Washington*, Transportation Research Record, **2385(1)**, pp. 28–36, Available from https://doi.org/10.3141/2385-04, publisher: SAGE Publications Inc.

[12] Church R & ReVelle C, 1974, *The maximal covering location problem*, Papers of the Regional Science Association, **32(1)**, pp. 101–118, Available from https://doi.org/10.1007/BF01942293.

[13] Church RL, 2019, *Understanding the Weber Location Paradigm*, International Series in Operations Research & Management Science, pp. 69–88, Available from https://ideas.repec.org//h/spr/isochp/978-3-030-19111-5_2.html, publisher: Springer.

[14] Cooper L, 1963, *Location-Allocation Problems*, Operations Research, **11(3)**, pp. 331–343, Available from https://www.jstor.org/stable/168022, publisher: INFORMS.

[15] Cooper L, 1964, *Heuristic Methods for Location-Allocation Problems*, SIAM Review, **6(1)**, pp. 37–53, Available from https://www.jstor.org/stable/2027512, publisher: Society for Industrial and Applied Mathematics.

[16] Cooper L, 1968, *An Extension of the Generalized Weber Problem†*, Journal of Regional Science, **8(2)**, pp. 181–197, Available from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9787.1968.tb01323.x, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9787.1968.tb01323.x.

[17] Correia I & Saldanha-da Gama F, 2019, *Facility Location Under Uncertainty*, pp. 185–213 in Laporte G, Nickel S & Saldanha da Gama F (Eds), *Location Science*, pp. 185–213. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-030-32177-2_8.

[18] Daskin M, 2013, *Network and Discrete Location: Models, Algorithms and Applications*, second edition Edition, John Wiley & Sons, Ltd, Available from https://onlinelibrary.wiley.com/doi/book/10.1002/9781118537015.

[19] Daskin MS & Maass KL, 2015, *The p-Median Problem*, pp. 21–45 in Laporte G, Nickel S & Saldanha da Gama F (Eds), *Location Science*, pp. 21–45. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-319-13111-5_2.

[20] Dong G, Ma J, Wei R & Haycox J, 2019, *Electric vehicle charging point placement optimisation by exploiting spatial statistics and maximal coverage location models*, Transportation Research Part D: Transport and Environment, **67**, pp. 77–88, Available from https://www.sciencedirect.com/science/article/pii/S1361920918303729.

[21] Drezner T & Drezner Z, 2011, *A note on equity across groups in facility location*, Naval Research Logistics (NRL), **58(7)**, pp. 705–711, Available from https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20476, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20476.

[22] Drezner Z, 2022, *Continuous Facility Location Problems*, pp. 269–306 in Salhi S & Boylan J (Eds), *The Palgrave Handbook of Operations Research*, pp. 269–306. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-030-96935-6_9.

[23] Drezner Z & Drezner TD, 2020, *Biologically Inspired Parent Selection in Genetic Algorithms*, Annals of Operations Research, **287(1)**, pp. 161–183, Available from https://doi.org/10.1007/s10479-019-03343-7.

[24] Drezner Z & Salhi S, 2017, *Incorporating neighborhood reduction for the solution of the planar p-median problem*, Annals of Operations Research, **258(2)**, pp. 639–654, Available from https://doi.org/10.1007/s10479-015-1961-y.

[25] Eiselt H & Laporte G, 1995, *Objectives in location problems*, pp. 151–180 in *Facility Location : A Survey of Applications and Methods*, pp. 151–180. Springer-Verlag, Berlin, publisher: Springer.

[26] Farahani RZ, Asgari N, Heidari N, Hosseininia M & Goh M, 2012, *Covering problems in facility location: A review*, Computers & Industrial Engineering, **62(1)**, pp. 368–407, Available from https://www.sciencedirect.com/science/article/pii/S036083521100249X.

[27] Fernández E & Landete M, 2019, *Fixed-Charge Facility Location Problems*, pp. 67–98 in Laporte G, Nickel S & Saldanha da Gama F (Eds), *Location Science*, pp. 67–98. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-030-32177-2_4.

[28] Frade I, Ribeiro A, Gonçalves G & Antunes AP, 2011, *Optimal Location of Charging Stations for Electric Vehicles in a Neighborhood in Lisbon, Portugal*, Transportation Research Record, **2252(1)**, pp. 91–98, Available from https://doi.org/10.3141/2252-12, publisher: SAGE Publications Inc.

[29] García S, Labbé M & Marín A, 2011, *Solving Large p-Median Problems with a Radius Formulation*, INFORMS Journal on Computing, **23(4)**, pp. 546–556, Available from https://pubsonline.informs.org/doi/abs/10.1287/ijoc.1100.0418, publisher: INFORMS.

[30] Glover F & Laguna M, 1998, *Tabu Search*, pp. 2093–2229 in Du DZ & Pardalos PM (Eds), *Handbook of Combinatorial Optimization: Volume1–3*, pp. 2093–2229. Springer US, Boston, MA, Available from https://doi.org/10.1007/978-1-4613-0303-9_33.

[31] Hakimi SL, 1964, *Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph*, Operations Research, **12(3)**, pp. 450–459, Available from https://www.jstor.org/stable/168125, publisher: INFORMS.

[32] Hansen P, Mladenović N, Brimberg J & Pérez JAM, 2019, *Variable Neighborhood Search*, pp. 57–97 in Gendreau M & Potvin JY (Eds), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, pp. 57–97. Springer International Publishing, Cham, Available from https://doi.org/10.1007/978-3-319-91086-4_3.

[33] He L, Ma G, Qi W & Wang X, 2019, *Charging an Electric Vehicle Sharing Fleet*, Available from https://papers.ssrn.com/abstract=3223735.

[34] He L, Mak HY, Rong Y & Shen ZJM, 2017, *Service Region Design for Urban Electric Vehicle Sharing Systems*, Manufacturing & Service Operations Management, Available from https://pubsonline.informs.org/doi/abs/10.1287/msom.2016.0611, publisher: INFORMS.

[35] Hodgson M & Rosing K, 1992, *A network location-allocation model trading off flow capturing and p-median objectives*, Annals of Operations Research, **40(1)**, pp. 247–260.

[36] Hodgson MJ, 1990, *A Flow-Capturing Location-Allocation Model*, Geographical Analysis, **22(3)**, pp. 270–279, Available from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1990.tb00210.x, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1538-4632.1990.tb00210.x.

[37] Holland JH, 1992, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA.

[38] Hung YC, PakHai Lok H & Michailidis G, 2022, *Optimal routing for electric vehicle charging systems with stochastic demand: A heavy traffic approximation approach*, European Journal of Operational Research, **299(2)**, pp. 526–541, Available from https://www.sciencedirect.com/science/article/pii/S0377221721005890.

[39] Ip A, Fong S & Liu E, 2010, *Optimization for allocating BEV recharging stations in urban areas by using hierarchical clustering*, Proceedings of the 2010 6th International Conference on Advanced Information Management and Service (IMS), pp. 460–465.

[40] Kadri AA, Perrouault R, Boujelben MK & Gicquel C, 2020, *A multi-stage stochastic integer programming approach for locating electric vehicle charging stations*, Computers & Operations Research, **117**, p. 104888, Available from https://www.sciencedirect.com/science/article/pii/S0305054820300058.

[41] Kalczynski P, Brimberg J & Drezner Z, 2022, *Less is more: discrete starting solutions in the planar p-median problem*, TOP, **30(1)**, pp. 34–59, Available from https://doi.org/10.1007/s11750-021-00599-w.

[42] Kchaou Boujelben M & Gicquel C, 2019, *Efficient solution approaches for locating electric vehicle fast charging stations under driving range uncertainty*, Computers & Operations Research, **109**, pp. 288–299, Available from https://www.sciencedirect.com/science/article/pii/S030505481930125X.

[43] Kim S, Rasouli S, Timmermans HJP & Yang D, 2022, *A scenario-based stochastic programming approach for the public charging station location problem*, Transportmetrica B: Transport Dynamics, **10(1)**, pp. 340–367, Available from https://doi.org/10.1080/21680566.2021.1997672, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/21680566.2021.1997672.

[44] Kuby M & Lim S, 2005, *The flow-refueling location problem for alternative-fuel vehicles*, Socio-Economic Planning Sciences, **39(2)**, pp. 125–145, Available from https://www.sciencedirect.com/science/article/pii/S0038012104000175.

[45] Kinay & Alumur SA, 2021, *Full cover charging station location problem with routing*, Transportation Research Part B: Methodological, **144**, pp. 1–22, Available from https://www.sciencedirect.com/science/article/pii/S0191261520304434.

[46] Kinay & Kara BY, 2019, *On multi-criteria chance-constrained capacitated single-source discrete facility location problems*, Omega, **83**, pp. 107–122, Available from https://www.sciencedirect.com/science/article/pii/S0305048317305133.

[47] Landau LD & Lifshitz EM, 1980, *CHAPTER I - THE FUNDAMENTAL PRINCIPLES OF STATISTICAL PHYSICS*, pp. 1–33 in Landau LD & Lifshitz EM (Eds), *Statistical Physics (Third Edition)*, pp. 1–33. Butterworth-Heinemann, Oxford, Available from https://www.sciencedirect.com/science/article/pii/B9780080570464500087.

[48] Laporte G, Nickel S & Saldanha da Gama F (Eds) 2015, *Location Science*, Springer International Publishing, Cham, Available from https://link.springer.com/10.1007/978-3-319-13111-5.

[49] Lehigh University Dept of Industrial and Systems Engineering, 2023, *15th AIMMS-MOPTA Optimization Modeling Competition*, Available from https://coral.ise.lehigh.edu/~mopta2023/competition.

[50] Li S, Huang Y & Mason SJ, 2016, *A multi-period optimization model for the deployment of public electric vehicle charging stations on network*, Transportation Research Part C: Emerging Technologies, **65**, pp. 128–143, Available from https://www.sciencedirect.com/science/article/pii/S0968090X16000267.

[51] Lim MK, Mak HY & Rong Y, 2014, *Toward Mass Adoption of Electric Vehicles: Impact of the Range and Resale Anxieties*, Manufacturing & Service Operations Management, Available from https://pubsonline.informs.org/doi/abs/10.1287/msom.2014.0504, publisher: INFORMS.

[52] Mak HY, Rong Y & Shen ZJM, 2013, *Infrastructure Planning for Electric Vehicles with Battery Swapping*, Management Science, Available from https://pubsonline.informs.org/doi/abs/10.1287/mnsc.1120.1672, publisher: INFORMS.

[53] Megiddo N & Supowit KJ, 1984, *On the Complexity of Some Common Geometric Location Problems*, SIAM Journal on Computing, **13(1)**, pp. 182–196, Available from https://epubs.siam.org/doi/10.1137/0213014, publisher: Society for Industrial and Applied Mathematics.

[54] Mladenović N, Brimberg J, Hansen P & Moreno-Pérez JA, 2007, *The p-median problem: A survey of metaheuristic approaches*, European Journal of Operational Research, **179(3)**, pp. 927–939, Available from https://www.sciencedirect.com/science/article/pii/S0377221706000750.

[55] Owen SH & Daskin MS, 1998, *Strategic facility location: A review*, European Journal of Operational Research, **111(3)**, pp. 423–447, Available from https://www.sciencedirect.com/science/article/pii/S0377221798001866.

[56] Pennsylvania Department of Environmental Protection, 2023, *Driving PA Forward*, Available from https://storymaps.arcgis.com/stories/6f5db16b8399488a8ef2567e1affa1e2.

[57] Pennsylvania Government - Department of Environmental Protection, 2024, *Electric Vehicles in PA*, Available from https://www.dep.pa.gov:443/Business/Energy/OfficeofPollutionPrevention/ElectricVehicles/Pages/default.aspx.

[58] Snyder LV, 2006, *Facility location under uncertainty: a review*, IIE Transactions, **38(7)**, pp. 547–564, Available from https://doi.org/10.1080/07408170500216480, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07408170500216480.

[59] Snyder LV & Daskin MS, 2006, *Stochastic p-robust location problems*, IIE Transactions, **38(11)**, pp. 971–985, Available from https://doi.org/10.1080/07408170500469113, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07408170500469113.

[60] Tu W, Li Q, Fang Z, Shaw Sl, Zhou B & Chang X, 2016, *Optimizing the locations of electric taxi charging stations: A spatial–temporal demand coverage approach*, Transportation Research Part C: Emerging Technologies, **65**, pp. 172–189, Available from https://www.sciencedirect.com/science/article/pii/S0968090X15003538.

[61] US Department of Transportation - Federal Highway Administration, 2023, *Alternative Fuel Corridors*, Available from https://www.fhwa.dot.gov/environment/alternative_fuel_corridors/.

[62] Wang YW & Lin CC, 2013, *Locating multiple types of recharging stations for battery-powered electric vehicle transport*, Transportation Research Part E: Logistics and Transportation Review, **58**, pp. 76–87, Available from https://www.sciencedirect.com/science/article/pii/S1366554513001403.

[63] Xi X, Sioshansi R & Marano V, 2013, *Simulation–optimization model for location of a public electric vehicle charging infrastructure*, Transportation Research Part D: Transport and Environment, **22**, pp. 60–69, Available from https://www.sciencedirect.com/science/article/pii/S1361920913000345.

[64] Yi Z, Chen B, Liu XC, Wei R, Chen J & Chen Z, 2023, *An agent-based modeling approach for public charging demand estimation and charging station location optimization at urban scale*, Computers, Environment and Urban Systems, **101**, p. 101949, Available from https://www.sciencedirect.com/science/article/pii/S0198971523000121.