Planning EV Charging Stations

VUltures team: Fergus Hathorn, Iker Olarra Maldonado, Alessandro Zocca

August 2023

1 Introduction

In recent years, the popularity of electric vehicles (hereafter *EVs*) in the United States has increased dramatically. Although EVs still make up only a small portion of roadworthy vehicles in the US, the number of EVs sold per year has been growing strongly [5]. Many EV owners conduct charging on their properties, but the availability of publicly accessible chargers is known to support the continued growth of the EV market, helping to alleviate a phenomenon known as *range anxiety* [1] from potential EV owners. Constructing EV charging stations (hereafter *stations*) is a capital-intensive activity, thus it is important that each station is positioned effectively. Some considerations are driving distances between EV owners and nearby stations, and also the number of chargers that each station contains. Similar problems have been tackled in literature, where it has become evident that inappropriately positioned or sized EV charging stations could have a significant influence on the development and uptake of EVs [6]. In [2], the authors found that evolutionary algorithms (including the genetic algorithm) took longer to find optimal solutions compared with other optimization algorithms, but on a strategic planning level this computation time was inconsequential because the results were often "*superior*". One such evolutionary approach was investigated in [7], showing that a Particle Swarm Optimization (hereafter *PSO*) approach was well suited to the problem of positioning EV charging stations in an optimal configuration. A core reason behind the motivation for using algorithms such as PSO is that they can handle the non-convex and non-linear properties of objective functions that are often associated with such problems.

This research aims to address the challenge of positioning and sizing EV charging stations within a local region, and furthermore to contribute to the widespread electrification of transportation. Two scenarios are evaluated in this research: (1) evaluating and solving the problem of (near) optimally positioning and sizing exactly 600 charging stations to service EV drivers in a region; and (2) positioning and sizing a (near) optimal number of EVs in a region, such that charging is convenient for drivers and the cost of the infrastructure is minimized. It should be noted that a strong emphasis has been placed on solving scenario 2. Thus, much of the report is focused on the methods, experiments, and analysis of the proposed evolutionary algorithm.

First, in Section 2 the methods for both scenarios are described. Experiments and results from the scenarios are discussed in Section 3. Additionally, we report the sensitivity analysis we conducted to better understand the results before leading into a discussion on the results. Lastly, in Section 4 we give a glimpse of the dashboard we developed and its functionalities.

2 Methods

In this section, we describe our proposed strategy for each scenario. We describe the strategy for scenario 1 as a K-centroid allocation method, and for scenario 2 we define the proposed *Particle-Swarm-k-Nearest-Neighbour (PS-k-NN)* algorithm. As the name suggests, we combine together a Particle-Swarm optimization step (henceforth denoted by PSO) to position EV stations in the continuous search space given by a rectangle with dimensions 290×150, and a k-Nearest-Neighbour procedure to assign EVs to stations. The global objective of the PS-k-NN is to minimize the total cost of constructing and maintaining charging stations, as well as the cost of driving to, and charging at these stations while maintaining an appropriately high service level for visiting EVs. In this context, the service level refers to the probability of being able to use (or wait for) a charger at a station, given that the queue length for a charger may not exceed one waiting EV and one EV being serviced.

2.1 Scenario 1: K-centroid allocation

This model was largely based on the *KMeans* algorithm. It is a two-stage approach, consisting of allocation ignoring the constraints, followed by a greedy reallocation of vehicles that exceeded the station's capacity. For the reallocation stage, the cost of a station after reassigning a vehicle was computed and the vehicle was assigned to the station which added the least to the total cost. Algorithm 1 summarizes the process.

Algorithm 2 K-centroid algorithm for station assignment.

Require: EV array x_{ij}, number of stations K, maximum number of chargers per station c

Use the *KMeans* algorithm with K centroids to determine the station locations.;
Assign each EV x_{ij} to their closest centroid.;
for k in $\{1 \dots K\}$ **do**
  if chargers in station$_k > c$ **then**
    Reassign the EVs x_{ij} to the station where it adds the least additional cost.
  end if
end for
Return: Station assignment;

2.2 Scenario 2: Particle Swarm Optimization

Even if the PSO component is primarily responsible for the positioning of charging stations, the overarching structure of the PSO-k-NN is inherited from traditional PSO algorithms. This structure consists of a *population*, *individual representation*, *variation operator*, and an *evaluation function*.
A basic particle swarm optimization typically involves setting candidate solutions as *particles*, and then "flying" them through a search space, adjusting the position of each particle based on the knowledge of the search space that is gained and shared through the entire population [4].
This section describes the three key aspects of this evolutionary algorithm: (1) the population and individual representation; (2) the variation operator; and (3) the evaluation function. Aspect (1) is focused on defining how a *solution* is represented by the algorithm, and how these solutions are organized into a *population*. The second aspect is focused on how the solutions are updated, with the aim of evolving into better solutions. Aspect (3) addresses how *better* is defined, *i.e.* a method of evaluating a solution.

2.2.1 Population and Individual Representation

The population consists of N individual solutions (hereafter *individuals*), where for every $i = 1, \dots, N$, each individual x_i is an $n \times 2$ array of coordinates describing station locations (hereafter *particles*). More specifically,

$$x_i = (x_{ij})_{j=1}^{n} \in \mathbb{R}^{2n},$$

where $x_{ij} = (\mathrm{x}_j, \mathrm{y}_j) \in \mathbb{R}^2$ is the location of station j in individual x_i. The value of n is not larger than 600 (as the number of stations may not exceed 600) and is equal for each individual in the population.

It is important that each particle x_{ij} is initialized in the same neighborhood for all i. This makes the comparison between particles from different solutions useful, as they are located in approximately the same region of the search space, avoiding situations where a particle on one side of the search space is being directly compared with a particle on the opposite side. These neighborhoods were generated by creating *anchor points* by sampling n EV locations and using these points as neighborhoods where particles are initialized. That is, for each sampled EV position $\nu \in \mathbb{R}^{2n}$, a particle x_{ij} is randomly initialized uniformly within $[\nu - 10, \nu + 10]$. This means that, for each j, the particles x_{ij} are initialized in the same neighborhood across all i.

Upon initialization of individual i, each particle x_{ij}, $j = 1, \ldots, n$, is given a velocity $v_{ij} \in \mathbb{R}^2$ by sampling $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$, where $\mu_v = 1$, and $\sigma_v^2 = 0.2$. The mean starting velocity μ_v was initialized with a relatively low value to preserve a neighborhood particle structure. When particles leave their initialization neighborhoods there may be an observed reduction in the quality of a particle's convergence to a (near) optimal position. Similarly, σ_v^2 was chosen to reduce the probability for extreme realizations to disrupt the neighborhood structure. Each sample of the v_{ij} velocity vector is multiplied by $-1 + 2H$, where $H \sim \mathrm{Bernoulli}(1/2)$, so that when $H = 0$ the velocity vector element is negative, and when $H = 1$ the vector element is positive, and velocities have equal probability of being positive or negative.

The best overall solution from the population is stored as g, and the best configuration per individual is stored as b_i. The quantities g and $\mathbf{b}$ are initialized using random points in the search space.

2.2.2 Variation Operator

In PSO, the position of each particle is updated according to the particle velocity v_{ij} in each generation $t = 1, \ldots, T$. A generation can be understood as a single iteration of the particle swarm, in which the position of each particle has been adjusted. Specifically, the new velocity for each particle in generation t is calculated as

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1,j}(t)[b_i - x_{ij}(t)] + c_2 r_{2,j}[g - x_{ij}(t)], \tag{1}$$

where $r_{1,j}, r_{2,j} \sim \textit{Uniform}(0,1)$ are random variables that take a value between 0 and 1, ω is the *inertia coefficient*, c_1 is the *cognitive coefficient*, and c_2 is the *social coefficient*. The position of each particle is then updated as follows:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}. \tag{2}$$

The tree terms in (1) can be understood as follows:

- the inertia term $\omega v_{ij}(t)$ controls how much influence the particle's previous velocity has on the current movement of the particle;
- the cognitive term $c_1 r_{1,j}(t)[b_i - x_{ij}(t)]$ compares the performance of x_{ij} to its previous best performance, hence drawing particles back to their previous best positions;
- the social term $c_2 r_{2,j}[g - x_{ij}(t)]$ compares the performance of x_{ij} with the corresponding particles of the best overall performance in the population, drawing particles towards the global best positions.

The magnitude of the ω, c_1, c_2 coefficients controls the exploration/exploitation behavior of the swarm. Specifically, higher inertia promotes exploration, while higher c_1, c_2 promotes exploitation.

2.2.3 Evaluation Function

The given objective is minimizing the cost of constructing and maintaining charging stations, along with the estimated cost of driving to a station and recharging. Measuring these costs requires EVs to be assigned to a charging station. It was assumed that for each EV location k, where $k = 1, \ldots, 1079$, the 10 associated EV drivers would follow similar behavior when selecting a station to charge their EV. That is, they would first check the Euclidean distance between EV and station positions before determining if the closest station has charger availability; if the station has availability then the EV driver would use that station for charging. In the event that there was no availability at a station, the driver would progress to the next closest station, and so on. The assignment subroutine will be covered in more detail in Section 2.3.

To evaluate a solution x_i, EVs should be assigned to stations x_{ij}, and then the total cost of the solution should be calculated. The goal is to minimize the following objective function (as per the problem description):

$$C_i^{\text{prac}}(t) = \alpha_c C^{\text{c}} \sum_{j=1}^{n} X_{ij}(t) + \alpha_m C^{\text{m}} \sum_{j=1}^{n} K_{ij}(t) + \alpha_d C^{\text{d}} \sum_{k=1}^{1079} d_k(t) + \alpha_e C^{\text{e}} \sum_{k=1}^{1079} e_k(t), \tag{3}$$

where C^{c} is the construction cost for a station, $X_{ij} = 1$ if particle x_{ij} has at least one EV assigned to it, and equals 0 otherwise, C^{m} is the maintenance cost for a station, K_{ij} is the number of chargers at station j in solution i, C^{d} is the cost of driving one unit, d_k is the distance that EVs at location k are required to drive in order to reach their assigned charging station, C^{e} is the charging cost per unit for an EV, and e_k is the charging units required by EV k. Additionally, the α coefficient for each cost component represents a weighting. At each generation t, the population is evaluated, and the global and local best positions (and scores) and updated if the current generation has individuals that achieve a lower total cost. It may be that EVs are not assigned to a station (because the stations are full, or because no station is within range), in this case, the practical objective proposed in (3) would not be a true reflection on the performance of the current generation. To better evaluate the population and ensure that solutions with all EVs allocated to stations are preferred, a penalty ρ can be introduced to penalize unassigned EVs. It should be noted that this penalty does not form part of the original problem formulation. Thus, (3) becomes:

$$C_i(t) = C_i^{\text{prac}}(t) + \rho \sum_{k=1}^{1079} u_k(t), \tag{4}$$

where u_k is the number of unassigned EVs at EV location k. Note that the $\alpha_d = 1e4$ was used to further discourage larger travel distances between chargers and EVs.

2.3 Scenario 2: *k-NN* subroutine

The methods described in Section 2.2 are aimed at defining the geographical position of each station. This section aims to use these station coordinates and a proposed heuristic to assign EVs to the nearest available station.

The mechanism of assigning EVs to stations is initiated by calculating Euclidean distances from each EV to each station. The closest K stations are considered to be feasible options for EVs to visit. The value of K should be selected such that an appropriately high service level can be met, as any group of EVs that is unassigned after more than K assignment attempts will not be assigned to a station. As K becomes larger, there is a higher chance of EVs being assigned to stations that are inconveniently (or prohibitively) far away. For each EV location, the expected number of EVs that will visit a charger is given by

$$\mathbb{E}V_r = \lceil q e^{-\lambda^2 (r-20)^2} \rceil, \tag{5}$$

where q is the number of EVs at a EV location, $\lambda = 0.012$, and r is the range of the EV. The mean range of an EV is approximately 100 miles and the standard deviation is 50 miles, although sampling from a truncated normal distribution with a minimum of 20 and maximum of 250 yields an expected range of $r \approx 105.7$ miles. This approximation can then be used to estimate $\mathbb{E}V_r = \lceil 4.2 \rceil = 5$ EVs per EV location.

In the first iteration of the algorithm, each EV is assigned to the nearest active station. If all stations are inactive then the nearest station is selected. The order of assignment is based on the probability of visiting, *e.g.* EV locations with a higher $\mathbb{E}V_r$ are assigned first. All EVs at a location are assigned to the same station.

The assignment process is altered in subsequent generations. That is, the movement of an EV's assigned station is inspected at each generation. If the inspection reveals that the station has moved by a magnitude of more than M units, then the assignment process for that EV location is reinitiated. Otherwise, the EVs remain assigned to the station, as it was determined that the position of the station did not change significantly. It may be that even a small adjustment in the station position *could* warrant a reassignment, indeed, many small adjustments in position could easily exceed M. To address these challenges, a mutation operator was added to the $k-NN$ algorithm. Each EV location was reassigned if its station moved by more than M in a single generation, and with probability p_{mutation} or if the distance between the EV and the station exceeds D_{max}. By adding the mutation, more of the search space is considered, as a particularly good solution could be very similar to an incumbent member of the population in geographical positioning but differ only in assignment decisions. It was observed that repeating the assignment decision for each EV in every generation resulted in oscillatory behavior from the population, due to the cost effects of stations switching between being active and inactive. Therefore, retaining assignment decisions between generations can be preferable, especially if stations do not move significantly. The k-NN approach is outlined in Algorithm 3.

Algorithm 3 K-NN EV Assignment Algorithm

Require: EV array x_{ij}, K
Ensure: Assigned EVs, list of unassigned EVs
for EV in $k = 1, \ldots, 1079$ **do**
  if Previously assigned station has moved more than δ, distance between station and charger is $\geq D_{max}$, or it is the first generation **then**
    Find the closest K stations
    Order the K nearest stations, prioritizing stations with chargers available
    for each station in the ordered list of K nearest stations **do**
      if there is remaining capacity for $\mathbb{E}V_r$ EVs at the station **then**
        Assign the EV to the station
      else
        Continue to the next station
      end if
    end for
    if EV is not assigned **then**
      Store the EV index
    end if
  end if
end for

2.4 Termination Conditions

The PS-k-NN algorithm can be run for a set number of generations, or until the population converges to a solution. In the latter context, converging to a solution requires that the difference in population fitness from one generation to the next does not change by a specified tolerance. In this study, termination was handled by setting a limit on the number of generations and observing the behavior of the population within the specified number of generations. It may be the case that the global best solution is not the same as the solution that the population converges to, this can be due to various factors such as velocity explosion (when particles separate from their neighborhoods and thus the difference between their position and the position of a local or global best grows larger, which can accelerate particles even further from their neighborhood), or premature convergence (when particles slow down too quickly and are unable to properly explore the search space). The convergence process is heavily dependent on several factors:

(i) the choice of the three parameters ω, c_1, c_2;

(ii) the population size N;

(iii) the initial magnitude of the velocity;

(iv) the choice of K.

In particular, item (i) should be tuned, as [4] suggests they can be problem dependent. Item (ii) can be tuned by experimenting with different population sizes, although it is generally a trade-off between solution quality and computational burden; large N would generally result in a more thorough exploration of the search space and smoother particle trajectories. A brief experiment that considers convergence speed would benefit this decision. Item (iii) is influenced by item (i), as a phenomenon called *velocity explosion* can occur when particles leave their neighborhoods in the initial search phase. Finally, a smaller value of K could lead to an increase in the number of EVs that remain unassigned, which could reduce the effectiveness of the search process of the algorithm.

3 Results

In this section, we outline the results achieved from applying the proposed methods to solving scenario 1 and scenario 2. As previously mentioned, a specific emphasis is placed on the methods and results for scenario 2. First, the results for scenario 1 are briefly outlined. Next, we perform hyper-parameter tuning for the PSO-k-NN approach and then discuss the resulting performance of the algorithm. Finally, we perform a sensitivity analysis on the scenario 2 method.

3.1 Scenario 1

To solve the first problem, a **K-centroid** approach was used. The *KMeans* algorithm was used to determine the location of the stations. Then, the EV locations were assigned to their closest station. To prevent stations from having more than 8 chargers, EVs were reassigned to smaller stations. The resulting assignment is the final solution. The costs obtained using this approach are summarized in table 1.

Category	**Cost**
Charging cost	\$59,371.05
Driving cost	\$71,778.02
Construction cost	\$3,000,000.00
Maintenance cost	\$1,566,500.00
Total cost	\$4,697,649.07

Table 1: Results obtained using the K-centroid approach

3.2 Scenario 2

The results for scenario 2 are discussed in this section. First, hyper-parameter tuning is performed (Section 3.2.1) to ensure that the algorithm is operated using optimal hyper-parameters. The results using the parameters from Section 3.2.1 are outlined in Section 3.2.2.

3.2.1 Hyper-Parameter Tuning

The three parameters ω, c_1, c_2 were tuned using *generate-and-test* principles, that is, establishing the utility of a generated parameter vectors [3] by systematically testing vector combinations on a subset of the full problem. A grid search was performed for values of $\omega \in [0.81, 0.83, \ldots, 0.99]$, $c_1, c_2 \in [0.05, 0.06, \ldots, 0.2]$. Table 2 shows the grid search results for the top-10-performing combinations of parameters (in terms of mean population objective value). Each row objective represents the mean of 10 runs for the combination of hyper-parameters. The grid search was performed on a subset of the full problem that contained only the EVs in the bottom left quadrant of the search space. It was assumed that the behavior of this smaller problem is a reasonable representation of the full problem.

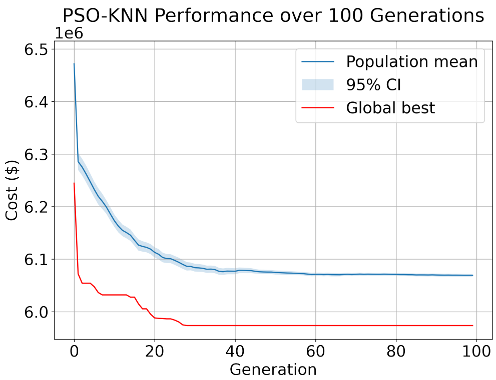
c_1	c_2	ω	Objective
0.05	0.14	0.83	193927.27
0.05	0.12	0.89	194256.47
0.04	0.20	0.84	194278.71
0.09	0.12	0.89	194427.71
0.05	0.12	0.83	194446.26
0.11	0.13	0.95	194448.91
0.04	0.16	0.84	194570.25
0.12	0.14	0.83	194615.93
0.05	0.11	0.87	194651.76
0.06	0.12	0.85	194671.78

Table 2: Grid search results for the best ten solutions

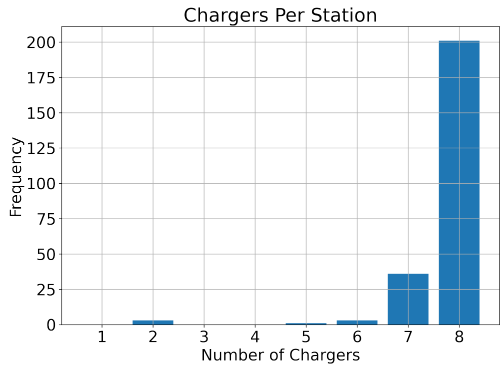
Based on the grid-search results reported in Table 2, we decided that the hyper-parameters should be set at $\omega = 0.83$, $c_1 = 0.05$, and $c_2 = 0.14$. For each parameter $s \in \{\omega, c_1, c_2\}$, a linear regression model of the form $Y = \alpha + \theta_s X_s + \epsilon_s$ was constructed, where Y is the objective value, θ_s is the coefficient for parameter s, X_s is the value of parameter s, α is the intercept, and ϵ_s is the error for parameter s. The value of θ_s was significant under a confidence level of 0.05 for $s \in \{c_1, c_2\}$, but not for ω. The coefficient θ_{c_1} was positive, while the coefficient c_2 was negative, indicating that lower values of c_1 and higher values of c_2 would generally be associated with better solutions. This is consistent with Table 2, where it is clear that the best solutions have $c_1 < c_2$.

3.2.2 Scenario 2: Tuned Results

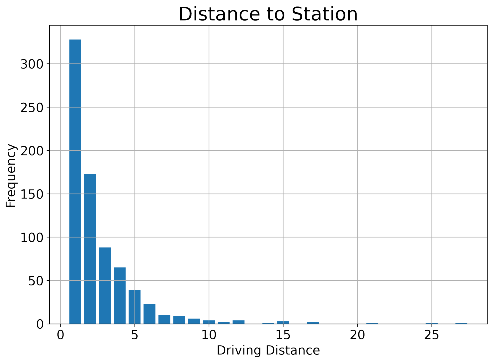
The PS-k-NN algorithm was run ten times for 100 generations with a population of 100 individuals. Statistical properties of the batch of results were estimated and are shown in Table 3. Figure 1a shows a progression of the population fitness over each generation and demonstrates how the global best solution is improved upon over time. Figure 1d shows the distribution of stations within the search space. It can be observed that, as expected, regions with a high density of EVs also contain a high density of stations, conversely sparsely populated regions have far fewer stations. Figure 1c shows a histogram of the distance from the EV to the assigned station, where it is clear that all EVs are within range of at least one station. The hyper-parameters associated with the lowest population mean from Section 3.2.1 were used for this experiment. To form a fairly robust solution, a slightly inflated EV demand $\mathbb{E}V_r$ was used, where the 20th percentile of sampled ranges was provided as input for charging probability calculations. Within the k-NN subroutine, $\delta = 1$ was used, and $D_{max} = 30$. That is, EVs were reassigned when the station moved more than 1 unit of magnitude between generations, or if the distance between an EV and station exceeded $D_{max} = 30$ units.



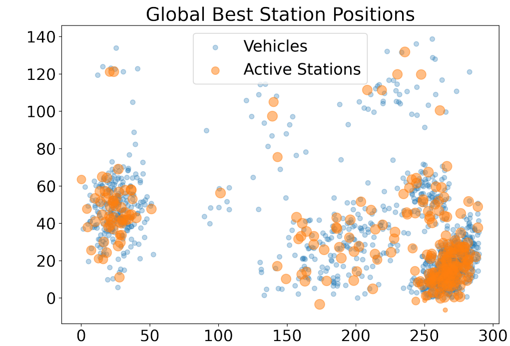
(a) Performance over generations.



(b) Number of chargers per station.



(c) Distance from EV to station.



(d) Spatial distribution of stations.

Figure 1: Performance results of PS-k-NN.

These values can be considered sensible, as it is unlikely that EV owners would find it convenient (or possible) to travel more than 30 miles to a station. Likewise, if a station moves by more than 1 mile, they may reconsider their local options. The α_d coefficient from Eq. (3) was set at $1e4$, as the low cost of driving could slow down the PSO from learning to position stations nearer to EVs. The remaining α values were set to 1.

Table 3: Results of the PS-k-NN.

Population Mean	95% CI	Global Best	95% CI
4649469.1	±3483.4	4648458.9	±3212.5

3.3 Sensitivity Analysis

A sensitivity analysis is useful for assessing the stability, reliability, and robustness of the findings. In this case, several critical parameters can be varied in order to observe the impact on results. The critical parameters are:

(i) the cost: the cost of constructing a station, maintaining a charger, driving, and charging, this can be evaluated by adjusting a cost and observing how the best solution changes;

(ii) the EV range and probability of charging: this can be evaluated by increasing and decreasing the probability of an EV going to a charger;

(iii) the spatial positioning of the best solution: this can be evaluated by making small adjustments to the best solution. A solution should be robust to small changes as it may not always be possible to place a station in the location suggested by the PS-k-NN.

3.3.1 Cost Sensitivity

The fluctuations in the objective value were recorded for adjustments to each cost parameter (construction, maintenance, driving, charging). The adjustments to each cost parameter consisted of increasing and decreasing the cost by 10% and 20%. Table 4 shows the resulting total costs. In each row all costs were held fixed as their default values, and the parameter in the first column was adjusted. The values in Table 4 represent the [lower, upper] bound of a 95% CI on the percentage increase or decrease compared to the default value. It is clear from this experiment that the construction cost has a linear impact on the solution, while the charging and driving costs have a lesser impact. Maintenance costs exhibit a similar (but reduced) influence on the objective compared to construction costs.

Cost	-20%	-10%	$+10\%$	$+20\%$
Construction	$[-11.23, -10.96]$	$[-5.63, -5.45]$	$[5.45, 5.65]$	$[11.06, 11.30]$
Driving	$[-0.09, 0.04]$	$[-0.03, 0.17]$	$[-0.13, 0.05]$	$[-0.07, 0.06]$
Charging	$[-0.26, -0.03]$	$[-0.07, 0.10]$	$[-0.06, 0.06]$	$[0.03, 0.33]$
Maintenance	$[-8.82, -8.66]$	$[-4.43, 4.32]$	$[4.23, 4.41]$	$[8.52, 8.72]$

Table 4: 95% CI [lower, upper] percentage difference in total cost with adjusted cost parameters compared to total cost with default cost parameters.

3.3.2 EV Charging Probability

It is expected that increasing or decreasing the probability of an EV requiring charging will influence the solution given by the PS-k-NN algorithm. At any EV location, there are 10 EVs that each have a probability of visiting a charger, and this probability is dependent on the range of the EV. Sensitivity to the expected number of EVs that require charging $\mathbb{E}V_r$ can be tested by sampling EV ranges from each station and taking different percentiles of these samples. For example, it is possible to observe the influence of increased demand by taking the 10th percentile of EV ranges r_{10} at a position, and using this to calculate charging probabilities. In this case, the charging probability value would be inflated, thus, $\mathbb{E}V_{r_{10}}$ would be higher. The PS-k-NN was run in batches of 10 for different sample range percentiles, allowing for statistical analysis of the solutions. Table 5 shows the relation between the percentile of EV range samples, probability of charging (p_c) and $\mathbb{E}V$, objective function value, and associated 95% confidence intervals.

Percentile	Range	p_c	$\mathbb{E}V$	Objective	Lower CI	Upper CI
0.01	65.88	0.74	8	6403675.70	6394256.84	6413094.56
0.1	68.09	0.72	8	6386215.10	6370346.73	6402083.46
1	75.89	0.64	7	5906155.91	5897969.57	5914342.25
5	84.13	0.55	6	4581748.10	4573233.32	4590262.88
10	87.75	0.52	6	4547870.42	4542399.19	4553341.66
20	93.7	0.46	5	3569065.95	3560083.43	3578048.48
30	98.3	0.41	5	3549481.79	3547517.11	3551446.46
40	102.3	0.38	4	3548735.18	3545658.43	3551811.92
50	106.5	0.34	4	2746600.25	2738282.72	2754917.78

Table 5: Results from sensitivity analysis on the EV charging demand.

A linear regression model of the form $\hat{y}_i = \alpha + \theta_i \hat{x}_i + \epsilon_i$ was used to evaluate how the objective changes for different p_c, where $\hat{y}_i$ is the objective value for percentile $i \in \{0.01, 0.1, 1, 5, 10, 20, 30, 40, 50\}$, $\alpha \in \mathbb{R}$ is a constant, $\hat{x}_i$ is the probability of charging i, θ_i is the coefficient linearly relating $\hat{x}_i$ to $\hat{y}_i$, and ϵ is a normally distributed error term. The resulting model indicated the p_c parameter was significant under a confidence level of 0.05, and that for each 0.1 increase in the probability of charging, the objective value would increase by \$919,726.60. In terms of range, this result indicates that increasing the mean sampled range by one unit results in a cost *reduction* of \$91,353.45.

3.3.3 Location Sensitivity

The solution of the PS-k-NN algorithm assumes that stations may be placed anywhere in the search space. In reality, this may not always be possible, and station positions may need to be adjusted in order to allow for feasible construction. Moving stations, even a short distance, could influence the total cost of the solution, so it is a worthwhile study to examine the cost sensitivity of a solution to minor perturbations in the stations' positions.

The sensitivity analysis was conducted by taking a global best solution $x_* \in \mathbb{R}^{2n}$ and adding a noise vector $z \in \mathbb{R}^{2n}$, whose entries are i.i.d. normal variables $z_i \sim \mathcal{N}(0, \sigma^2)$, such that a perturbation of the solution is given by $\hat{x}_* = x_* + z$. The k-NN subroutine can then be repeated for different σ and the total cost be re-estimated. Figure 2 shows the distribution of total cost for each value of σ. Samples of size 100 were used to generate this analysis. The figure indicates that all tested perturbations with $\sigma < 5$ did not yield a total cost with statistically significant differences. It can be concluded that adjusting the station positions slightly is unlikely to cause any significant unexpected costs.

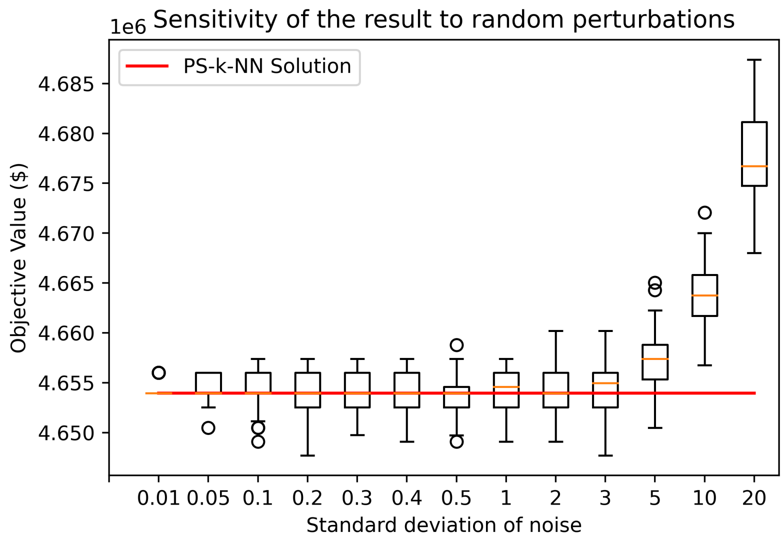


Figure 2: Box plot showing the distribution of total cost with respect to random perturbations in station position.

4 User interface

In this section, we describe the features and capabilities of the user interface. It was developed to enable a user to explore the presented solution and asses its quality but also allows them to obtain a new solution after having modified the parameters. Figure 3 gives a general overview of the interface, which has the following main components:

- Control panel
- Key indicators
- Cost breakdown
- Histograms
- Map

Each of them will be further explained in the following subsections.

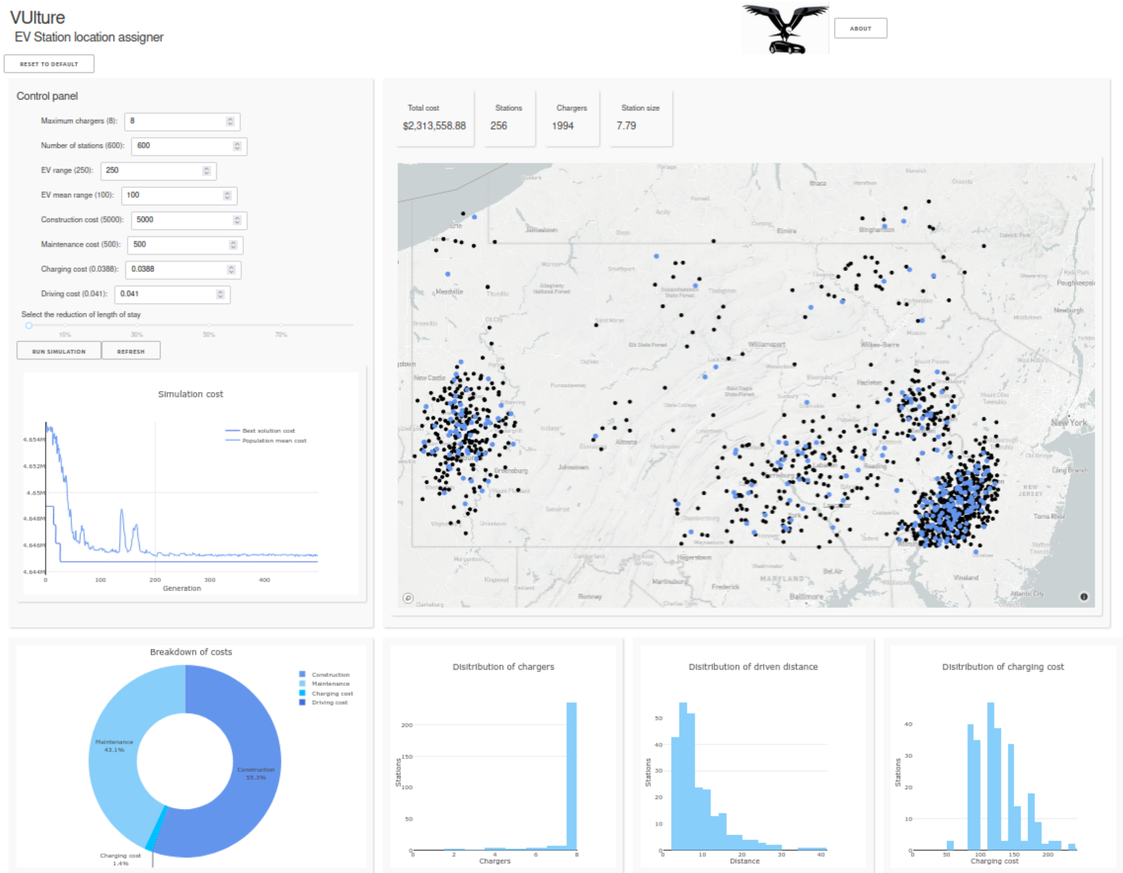


Figure 3: General view of the user interface.

4.1 Map

The map visualizes the solution by showing the location of the stations as well as the electric vehicle locations. Hovering over a station will allow the user to know details about the station such as the number of chargers, location, and cost. The map is shown in Fig. 4 below. Note that, even if no geographical position was provided for the EV locations, we chose to display them as if they were scattered in the state of Pennsylvania.

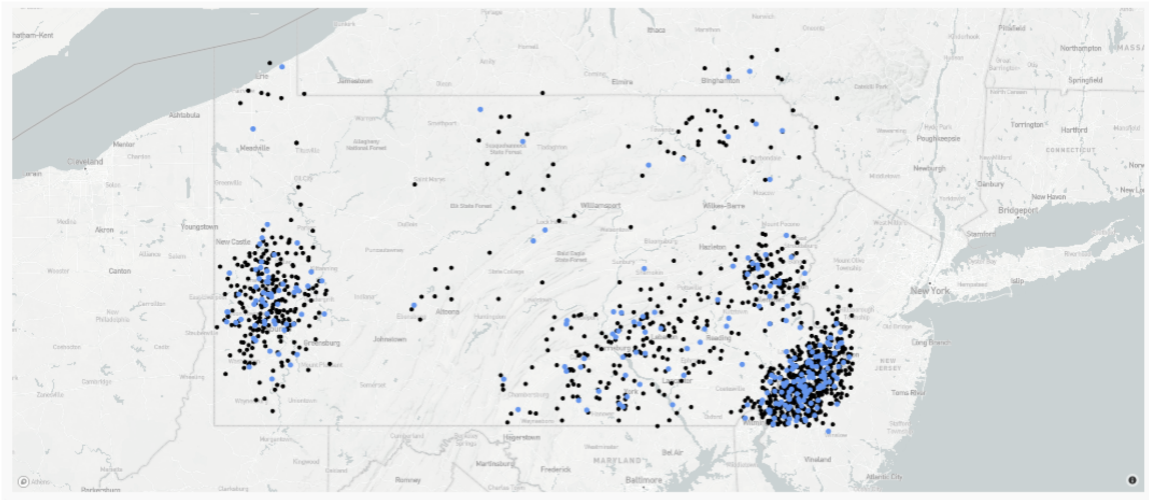


Figure 4: Map displaying the solution.

4.2 Key Indicators

The purpose of the indicators shown in Fig. 5 is to quickly display important information that summarizes the main features of the solution, namely the total cost, the number of chargers, the number of stations, and the number of chargers per station.

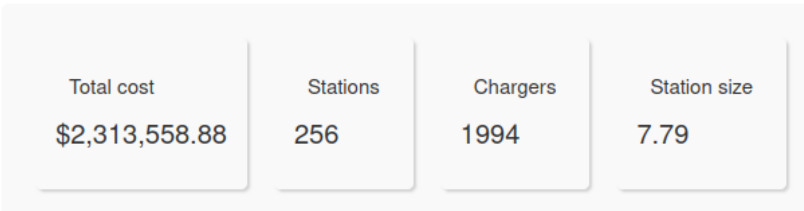


Figure 5: Key indicators of the solution.

4.3 Cost Breakdown

The chart in Fig. 6 displays the proportion of the total cost that is attributed to each type of cost.

4.4 Control Panel

The control panel enables the user to interact with the algorithm to produce new solutions by modifying the default values of the parameters. After selecting the desired parameters, the interface will execute the algorithm to approximate the new optimal solution. The progress can be monitored by the cost graph included in the control panel, which dynamically displays the best individual and the average of all individuals from the PS-k-NN algorithm as the generations increase. Once the execution is completed, the rest of the plots will be updated.

4.5 Histograms

The histograms in Fig. 8 display the distribution of relevant characteristics of the solution, including the driven distance, charging costs, and number of chargers per station. These allow the user to evaluate the quality of the solution and the expected behavior of the customers.

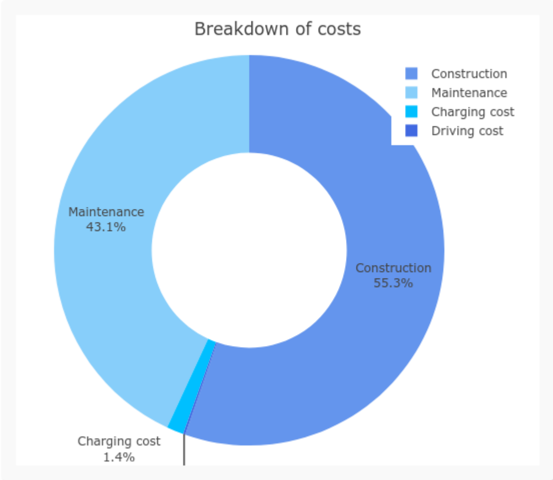


Figure 6: Cost breakdown chart.

5 Repository

The resources used for this implementation are available in `github.com/ikerforce/mopta/`. The repository has 5 main sections:

- `src` Contains the code for the algorithms.
- `data` Has the data used for the implementation.
- `results` Includes the files for the main results of the algorithms.
- `ui` Contains the source code and resources needed for the user interface.

6 Discussion and Conclusions

The proposed PS-k-NN algorithm aimed to address the challenge of positioning and sizing EV charging stations within a pre-defined region. The PSO component of the proposed approach was tuned using a grid search, which revealed some underlying relationships between parameters and the objective value. Specifically, the social and cognitive components of the PSO had a statistically significant influence on the objective, and it was found that $c2 > c1$ for best performance. In Problem 1, a K-centroid method was used to solve the scenario when 600 stations should be placed optimally in the search space. However, it may be that it is not optimal to place 600 stations in the search space. The PS-k-NN approach in Problem 2 was restricted to a maximum of 600 stations, although (even for rare events such as in the 0.01 percentile of range samples) 600 stations were never used. The results in Section 3 show that the PS-k-NN was able to converge to a solution very close to the global optimal. It was also observed that for most individuals more than 90% of the EVs were assigned a station within 5 miles. A sensitivity analysis was conducted to determine the influence of some of the assumptions, such as costs, probability of charging, and perturbations to the final solution. The cost sensitivity analysis revealed that the solution was influenced by construction and maintenance costs while driving and charging costs did not have a strong impact. The probability of charging was found to have a proportional and positive statistically significant relationship with the objective function, meaning that as the probability of an EV charging increased, there was a measurable increase in the objective. Finally, the location sensitivity analysis revealed that minor perturbations did not result in a significant effect on the objective, which means that the PS-k-NN solution in continuous space can indeed be adjusted by small variations in order to correct for impractically placed stations.

A UI was proposed that enables users to explore a solution, modify parameters, and give a cost breakdown. The map allows users to inspect the positions of stations in relation to landmarks,

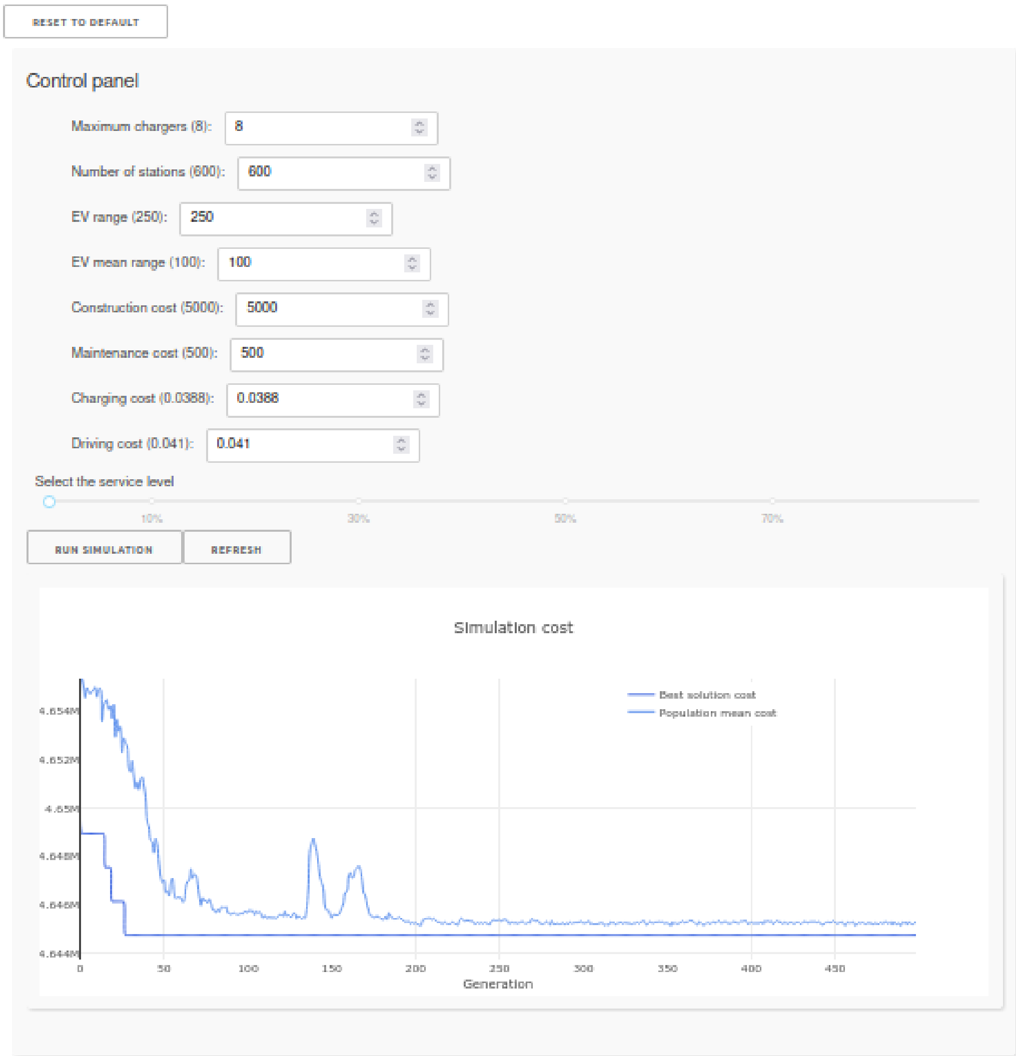


Figure 7: Control panel of the interface.

which can assist with identifying stations that are placed in impractical locations (*e.g.* Lake Erie). The proposed approach does have drawbacks related to:

- The PSO aspect of PS-k-NN, whereby the algorithm is known to be affected by phenomena such as premature convergence and velocity explosion, which have an effect on the quality of the solution. However, when examining Figure 1a, the oscillatory behavior associated with velocity explosion is not present, indicating that the hyper-parameter tuning was effective.
- The k-NN subroutine, it may be that allocations are not always made in an optimal way - some heuristics could be useful to further improve this subroutine.
- The problem formulation, adjusting the sampled ranges and probability of charging has a significant influence on the objective. It may be that incorporating a more stochastic approach to the PS-k-NN could be a valuable exercise. That is, in each generation a new sample of vehicle ranges is drawn and are not aggregated on any level. It should be noted that convergence under stochastic conditions can be a challenge with PSO. A brief experiment is shown in Section 7, where stochastic conditions were imposed on the problem with promising results.

Despite the drawbacks discussed, the PS-k-NN did provide a foundation for optimally positioning and sizing EV charging stations in a continuous search space. Analysis of the solution indicates that it could be useful as a planning tool, and could provide stakeholders with actionable information, and it is recommended that additional research is conducted to further improve the effectiveness of

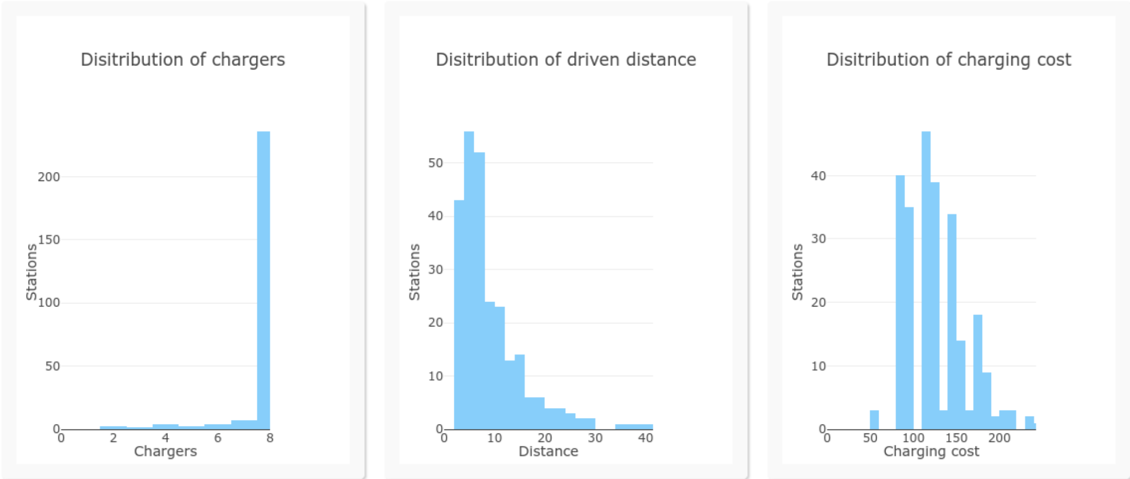


Figure 8: Histograms of the interface.

the proposed PS-k-NN.

References

[1] H. A. Bonges and A. C. Lusk. Addressing electric vehicle (ev) sales and range anxiety through parking layout, policy and regulation. *Transportation Research Part A: Policy and Practice*, 83:63–73, 2016.

[2] D. Efthymiou, K. Chrysostomou, M. Morfoulaki, et al. Electric vehicles charging infrastructure location: a genetic algorithm approach. *Eur. Transp. Res. Rev.*, 9(27), 2017.

[3] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, 2nd. edition, 2015.

[4] A. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley Publishing, 2007.

[5] D. Hawley. What percent of us car sales are electric? `https://www.jdpower.com/cars/shopping-guides/what-percent-of-us-car-sales-are-electric`, April 03 2023.

[6] Z. Liu, F. Wen, and G. Ledwich. Optimal planning of electric-vehicle charging stations in distribution systems. *IEEE Transactions on Power Delivery*, 28(1):102–110, 2013.

[7] Z.-f. Liu, W. Zhang, X. Ji, and K. Li. Optimal planning of charging station for electric vehicle based on particle swarm optimization. In *IEEE PES Innovative Smart Grid Technologies*, pages 1–5, 2012.

7 Appendix: Stochastic EV Charger Demand

An additional experiment was conducted to determine how the PSO-k-NN handles stochastic conditions. That is, EV ranges are resampled in each generation instead of being held at their expected value. It was expected that the ability for the PSO-k-NN to converge would be hindered by the changing demand of EVs. However, the algorithm was able to adapt and converge with stochastic EV charging demand rather effectively, as evidenced in Fig. 9. It should be noted that Fig. 9 generally results in a more expensive solution than Fig. 1a, and more variance can be observed in the population mean (particularly in later generations, where the deterministic version becomes very stable). Additionally, it can be seen that the curve is initially less steep in Fig. 9 compared to Fig. 1a, indicating that the learning rate may be lower under stochastic conditions.

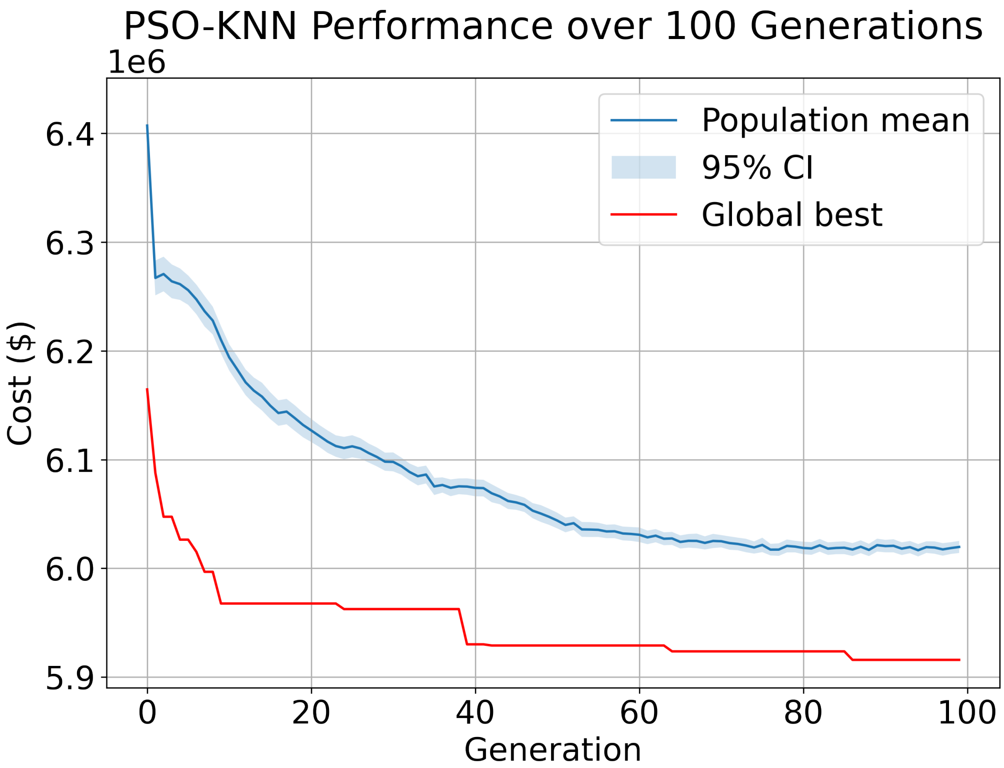


Figure 9: Convergence process of PSO-k-NN with stochastic EV charging demand.