Shell.ai Hackathon

for Sustainable and Affordable Energy

EV Charging Network Challenge
Problem Statement

powered by Microsoft UDACITY

shell.com/hackathon

Shell.ai Hackathon for Sustainable and Affordable Energy

EV Charging Network Challenge: Problem Statement

Introduction

The mobility sector accounts for around 18% of the carbon dioxide emissions from burning fuel[1]. A considerable market share of electric vehicles (EVs) running on clean power is key to mobility decarbonisation. As the number of EVs on our roads continues to increase, the demand for charging is following suit. Today around 10% of drivers are choosing EVs when buying a new car, and this is expected to grow rapidly[2]. **We need accelerated technology and infrastructure development to support EV market growth**.

While EV drivers are looking for a charging experience that is as fast and comfortable as possible, **the main obstacle to mobility decarbonisation is inconvenience caused by charging infrastructure and charging time**[3] Proper placement of charging points can alleviate this problem. It is a subject of current research and business interest simultaneously and has immense implications for the adoption of EVs, including:

1. lowering range anxiety among EV owners,
2. optimal utilisation of EV charging points,
3. minimal travel time and waiting time for EV owners.

In addition, the challenges around this problem are evolving as EVs penetrate different geographies. For example, the EV charging placement in a U.S. city has different nuances than in a suburban town in India.
The demand for solving this problem at different geographies and scales will increase exponentially in the next decades as EVs spread from cities and urban areas to villages.

---

[1] https://www.iea.org/topics/transport
[2] https://www.iea.org/data-and-statistics/data-product/global-ev-outlook-2022
[3] https://www.shell.com/promos/motorists/decarbonising-mobility-together-we-drive-change/_jcr_content.stream/1644314678955/bce5d350bceac29f652046eb352c3514edf76b52/decarbonising-mobility-report-december-2021.pdf

Problem Statement

In Shell AI Hackathon 2022, we challenge you to optimally place EV charging stations so that the configuration remains robust to demographic changes.

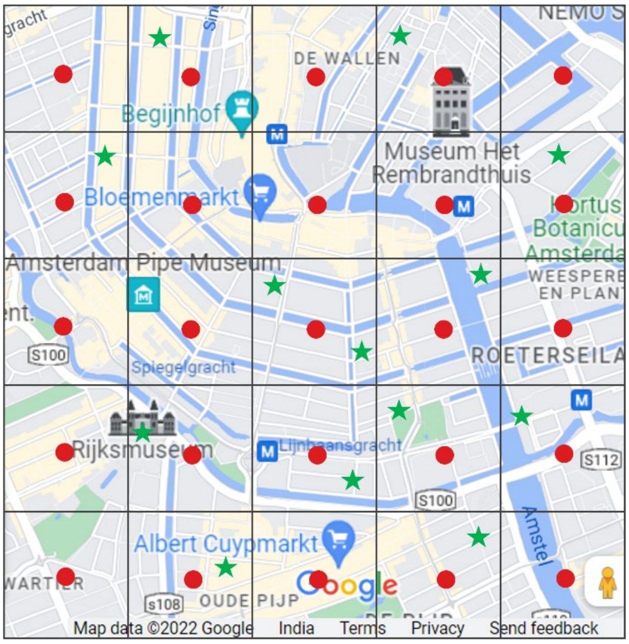

Figure 1: Problem Description. This map is used only for describing the problem. The challenge data set is not related to this location.

Let's try to understand forecasting of EV charging demand and EV infrastructure optimisation using Fig. 1. Typically for such problems, a geographic region is divided into equal size blocks. Total EV charging demand of each block is represented at the centre of the block. Let's call these centre points as *demand points,* which are represented as red circles in Fig. 1. All demand points of a geographic region collectively create a *demand map* over the region of interest. For each demand point, forecasting can be done using the historical demand maps.

EV charging stations are typically installed at public parking locations so that EVs can be charged during the idle parking time. These parking locations are predefined based on how real estate has been developed in the region. These are called *supply points* and are represented as green stars in Fig. 1. Each parking location has a fixed number of parking slots i.e. potential places to install EV charging stations. Typically, two types of charging station are installed based on their supply capacity: i) slow charging station (SCS) and ii) fast charging station (FCS). Based on how many SCS and FCS are installed at a parking location and their respective charging capacities, we can calculate the maximum supply that can be given by each *supply point*. All supply points of a geographic region collectively create a *supply map* over a region.

Using the demand map, supply map, demand-supply constraints and objective, we can optimally choose to place the EV charging stations so that the designed EV infrastructure is best suited to cater the forecasted demand. More details on this are provided in the subsequent sections.

Data

We are providing the following datasets:

1. Demand_History.csv: A time-series of EV charging demand over a region. We have considered demand map of 64x64 equi-spaced points. For ease of use, we have flattened the demand map and provided the index, x-coordinate and y-coordinate of each demand point. Demand history is provided from year 2010 to 2018.
2. Existing_EV_infrastructure_2018.csv: Details of the existing EV infrastructure of year 2018. This dataset comprises of index, x-coordinate and y-coordinate of each of the 100 parking locations (supply points). It also provides the maximum number of parking slots available at each parking location along with the number of SCS and FCS already in place as of 2018.

Using the above data, you are required to forecast demand and optimize EV infrastructure for year 2019 and 2020 under certain practical constraints.

Note: All quantities/values provided in the above datasets are dimensionless. The demand and supply data are not selected from any real location. However, the data set has all the characteristics of a practical use case.

Notations and Constants

i = Index of the demand point. i varies from 0 to 4095

j = Index of the supply point (parking location). j varies from 0 to 99

D_i = EV charging demand at i^{th} demand point.

SCS_j = Number of slow charging stations at j^{th} supply point

FCS_j = Number of fast charging stations at j^{th} supply point

PS_j = Total parking slots available at j^{th} supply point

Cap_{SCS} = Charging capacity of a slow charging station = 200

Cap_{FCS} = Charging capacity of a fast charging station = 400
$Smax_j$ = Maximum supply that can be given from j^{th} supply point = $(Cap_{SCS} \times SCS_j) + (Cap_{FCS} \times FCS_j)$
$Dist_{ij}$ = Distance between i^{th} demand point and j^{th} supply point (Distance matrix)
DS_{ij} = How much demand of the i^{th} demand point is satisfied by the j^{th} supply point (Demand-Supply matrix)

Objective

The objective of the EV infrastructure optimisation problem is to minimize the overall cost which comprises of total 3 components.

1. Cost of Customer Dissatisfaction ($Cost_{CD}$): This cost depends on how far customers have to travel to satisfy their EV charging demand.
$$Cost_{CD} = \sum_{i,j} Dist_{ij} \times DS_{ij}$$
2. Cost of Demand Mismatch ($Cost_{DM}$): This cost depends on the demand mismatch arising due to incorrect demand forecast as compared to the true values.
$$Cost_{DM} = \sum_{i} abs(D_{forecast,i} - D_{true,i})$$
3. Cost of infrastructure ($Cost_{IF}$): This cost depends on the operational, maintenance and amortised capital cost of owning EV infrastructure.
$$Cost_{IF} = \sum_{j} (SCS_j + r \times FCS_j)$$
Where, $r = 1.5$ is the ratio of the cost of fast charging station to that of slow charging station.

So, overall cost ($Cost$) = $a \times Cost_{CD} + \ b \times Cost_{DM} + \ c \times Cost_{DM}$. Here, $a = 1$, $b = 25$ and $c = 600$ are constants.

Constraints

You are required to optimise the EV infrastructure under the following practical constraints.

1. All values of the demand-supply matrix (DS_{ij}) must be non-negative.

2. All values of the number of slow (SCS_j) and fast charging stations (FCS_j) must be a non-negative integer.
3. Sum of slow (SCS_j) and fast charging stations (FCS_j) must be less than or equal to the total parking slots (PS_j) available at each j^{th} supply point.
4. You can only build incremental EV infrastructure on top of the 2018 infrastructure. That means, SCS_j and FCS_j must increase or stay constant year-on-year at each j^{th} supply point.
5. (Sum of fractional) Demand satisfied by each j^{th} supply point must be less than or equal to the maximum supply available.

$$\sum_i DS_{ij} \leq Smax_j$$

6. (Sum of fractional) Forecasted demand at each i^{th} demand point must exactly be satisfied.

$$\sum_j DS_{ij} = D_{forecast,i}$$

Evaluation

You are required to submit one file (solution.csv) on HackerEarth portal for evaluation. This file will comprise of your solution (matrix DS_{ij}, SCS_j and FCS_j) with following columns in .csv format,

Column names	Valid Entries
year	2019, 2020
data_type	DS, SCS, FCS
demand_point_index	0 to 4095
supply_point_index	0 to 99
value	Any value following problem constraints

<u>Notes:</u>
- We have provided a sample solution.csv file for your reference. Your submission must follow the same format.
- If you don't provide values for all valid indices, default i.e. zero will be considered as the value for those indices. This may result into constraint violation.
- For *data_type* SCS and FCS, entries under *demand_point_index* column are not required and will be disregarded. You may keep these entries empty/blank.

- Your solution will be eligible for ranking only if it satisfies all the constraints for both 2019 and 2020.
- We will keep first year (2019) of your solution for the public leader board. You can test your solution any time and see how it ranks.
- We will keep second year (2020) of your solution for the private leader board and it will be used to determine the finalists.

<u>Scoring:</u>

If your solution satisfies all constraints, we will first calculate your solution's cost using the overall cost function. Your cost (lower the better) will then be converted to score (higher the better) between 10 to 100 using following transformation function for the leaderboard ranking:

$$Leaderboard\ score\ =\ \max\left[10,\left(100-\frac{90\times cost}{15{,}000{,}000}\right)\right]$$

Scores between 0 to 8 are reserved for the error codes detailed below.

<u>Error Codes:</u>

If your solution doesn't satisfy any of the constraints or submission format, you will get following error codes on the leaderboard:

0 → Format error: solution.csv not following the correct format. Check sample solution file for reference.

1 → Constraint 1 violated

2 → Constraint 2 violated

3 → Constraint 3 violated

4 → Constraint 4 violated

5 → Constraint 5 violated

6 → Constraint 6 violated

7 → Index error: Supply point index must be an integer value between 0 to 99

8 → Index error: Demand point index must be an integer value between 0 to 4095