Electric Cars: Logistics For the Environment

Juan David Echeverri Villada
Juan Sebastián Guerra Hernández
Medellín, 31/05/2021

Data Structures

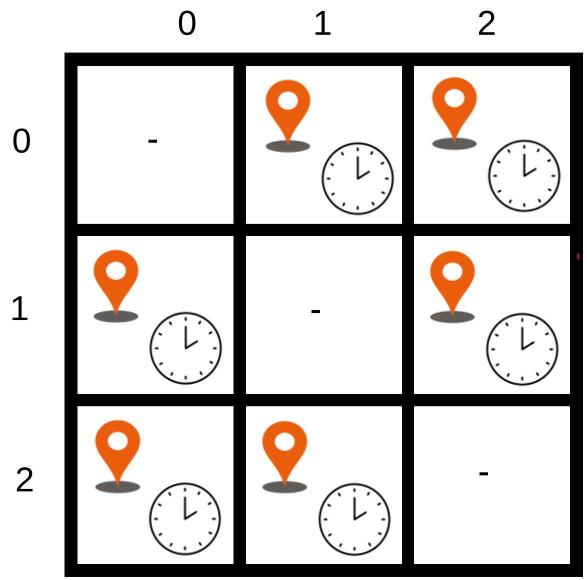

Figure 1: *Adjacency Matrix with weights of pairs Distance - Time. Indexes represent the ID of the respective nodes. The diagonal is empty due both IDs refer to the same node.*

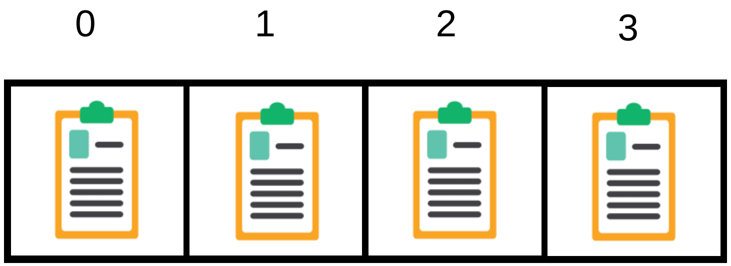

Figure 2: *Array that contains the data of each node [Name of the ubication, x coordinate, y coordinate, type of node (either deposit, station, or client) and if is a station, the type of station (Fast, normal, or slow charging)]. Index represents the ID of the node.*

Variables' dictionary

Variable	*Description*
N	Number of customers in the map. *Equals to W*R*
S	Number of clients that have not been visited.
W	The maximum number of customers that a vehicle can visit.
R	Number of routes.

Table 1: *Dictionary of variables used to express asymptotic complexity.*

Algorithm and Complexity

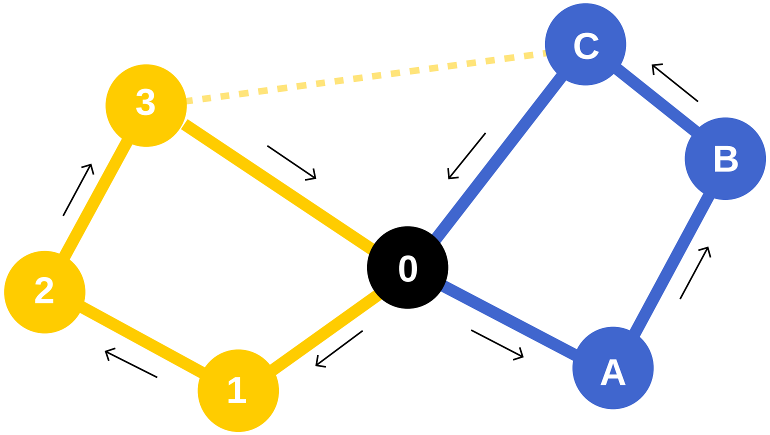

Figure 3: *Procedure of the algorithm to calculate routes.*

Steps	**Asymptotic complexity**
Find the best client	O(S)
Found a route	O(S*W)
Found all the routes	O(S*W*R)
Final algorithm	**O(S*N)**

Algorithm design criteria

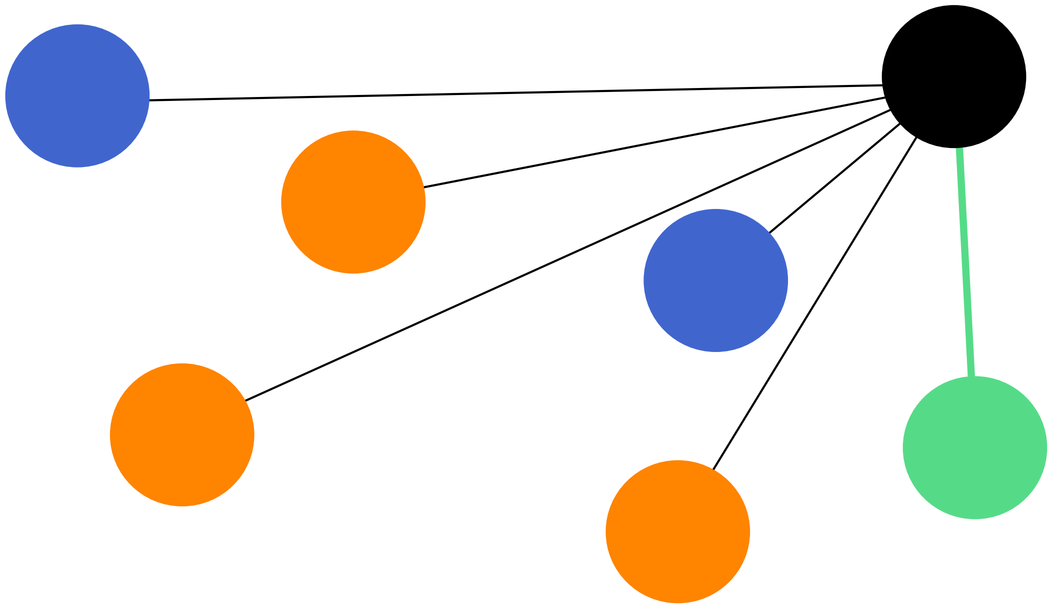

Figure 4: *Procedure of the algorithm to calculate routes.*

After researching of all the different approaches that can be done, we decided to focus in a greedy one, trying constantly to find a random nearest client by the time of calculating a route. This nearest client is selected randomly in a k selected range. This has several optimization benefits since it groups the clients by zones, maximizing the time of the routes and ending with a less use of vehicles thanks to the random search. Asides its good complexity, which is O(S*W) to find each of the different routes, the algorithm also has shown a great performance in the given datasets.

Time and Memory Consumption

Dataset	Best case (sec)	Average case (sec)	Worst case (sec)
1	3.81128	3.87553	3.92024
2	3.09814	3.10244	3.12267
3	2.87452	2.92764	2.993
4	3.68958	3.79476	4.066
5	2.25342	2.3123	2.41406
6	2.67286	2.74125	2.77312
7	3.4899	3.65148	4.17748
8	2.15364	2.23134	2.35968
9	2.59232	2.7165	2.7948
10	3.44148	3.52738	3.69084
11	2.29312	2.37526	2.44696
12	2.50348	2.55964	2.63098

Table 2: *Execution time of the algorithm for different datasets.*

Dataset	Memory consumption (MB)
1	79.63
2	82.10
3	76.23
4	78.09
5	73.91
6	75.68
7	80.32
8	72.82
9	74.98
10	77.75
11	74.21
12	75.96

Table 3: *Memory consumption of the algorithm for different datasets.*

Software prototype

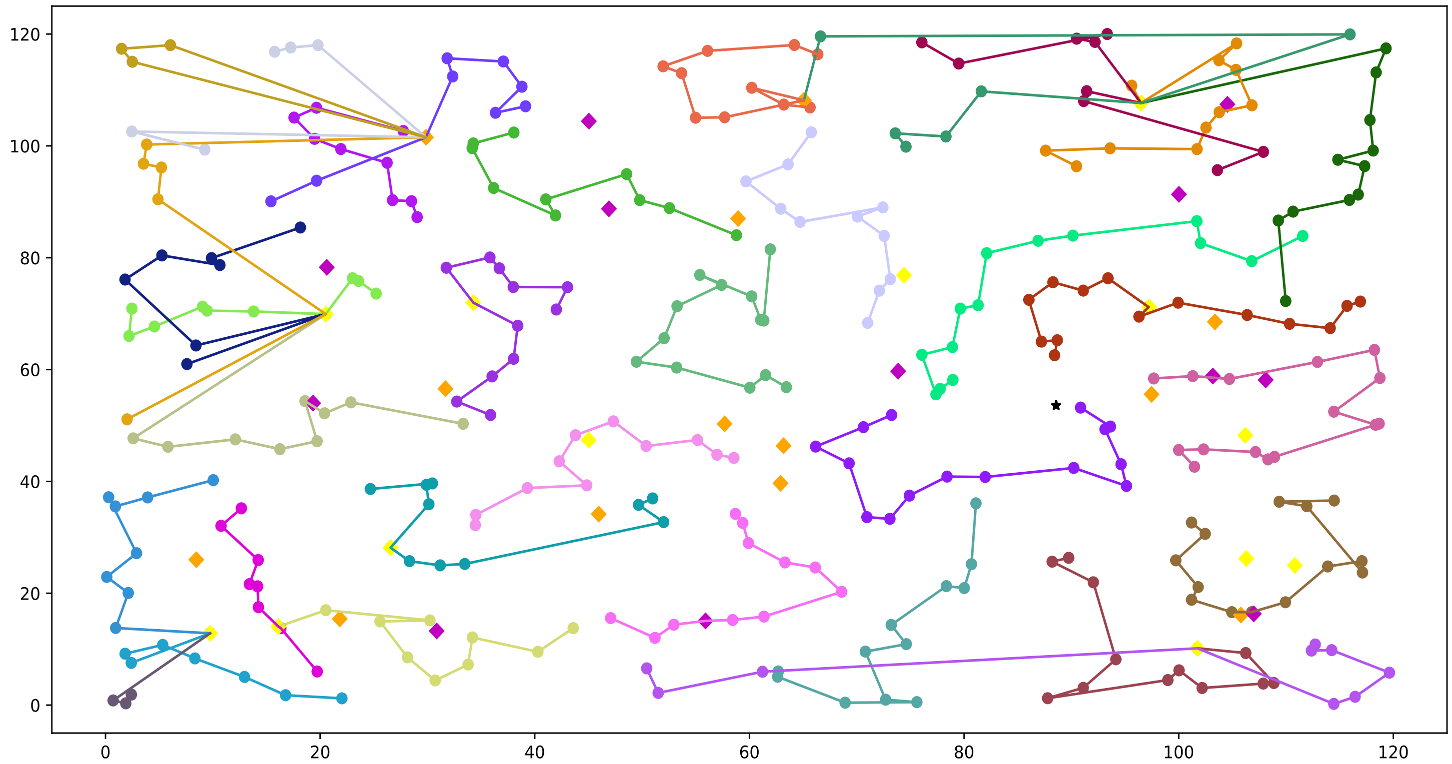

Figure 6: *Illustration designed with matplotlib. It represents each of the routes calculated for one of the datasets. Keep in mind that each color represents a route, and every route starts and ends in the black star, which represents the deposit.*

Software prototype

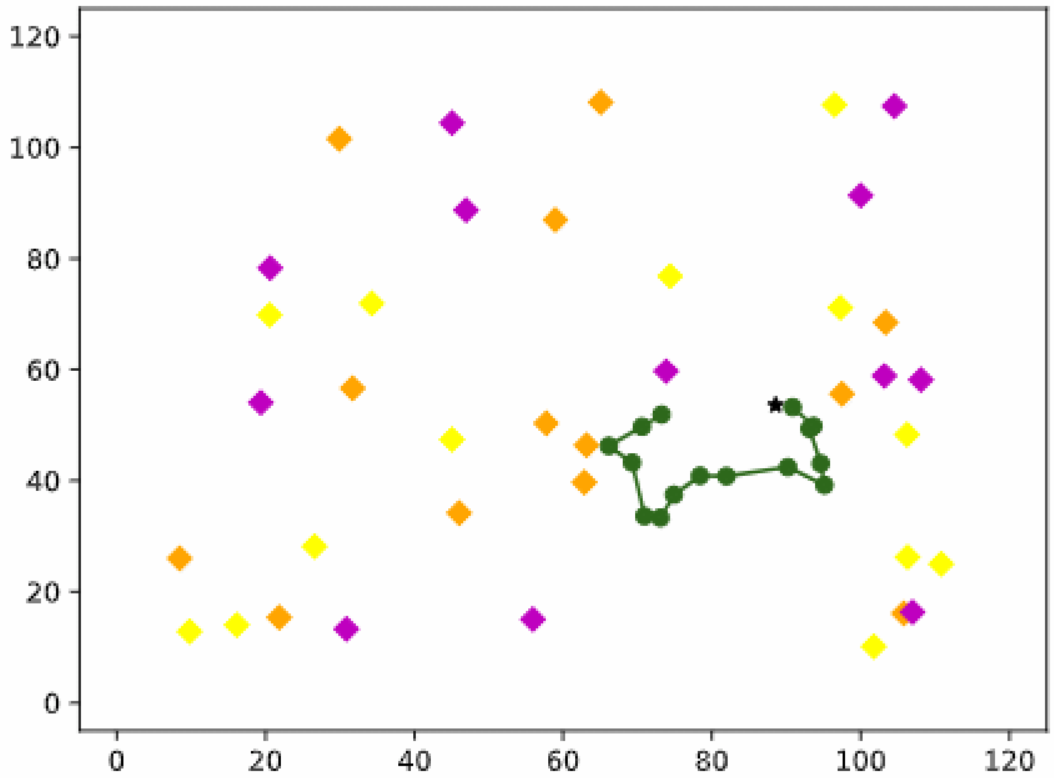

Figure 6: *Illustration designed with matplotlib. It represents each of the routes calculated for one of the datasets. Keep in mind that each color represents a route, and every route starts and ends in the black star, which represents the deposit.*