

Electric Cars: Logistics For the Environment

Juan David Echeverri Villada

Juan Sebastián Guerra Hernández

Medellín, ??/??/????

Data Structures













	0	1	2
0	-	 	 
1	 	-	 
2	 	 	-

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.



0	1	2	3
			

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

<i>Variable</i>	<i>Description</i>
N	Number of points of interest in the map. <i>Equals to $W \cdot R$</i>
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

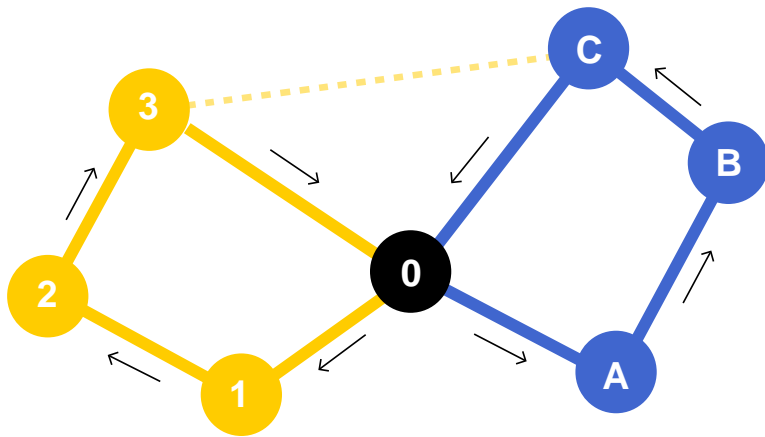


Figure 3: Procedure of the algorithm to calculate routes.

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

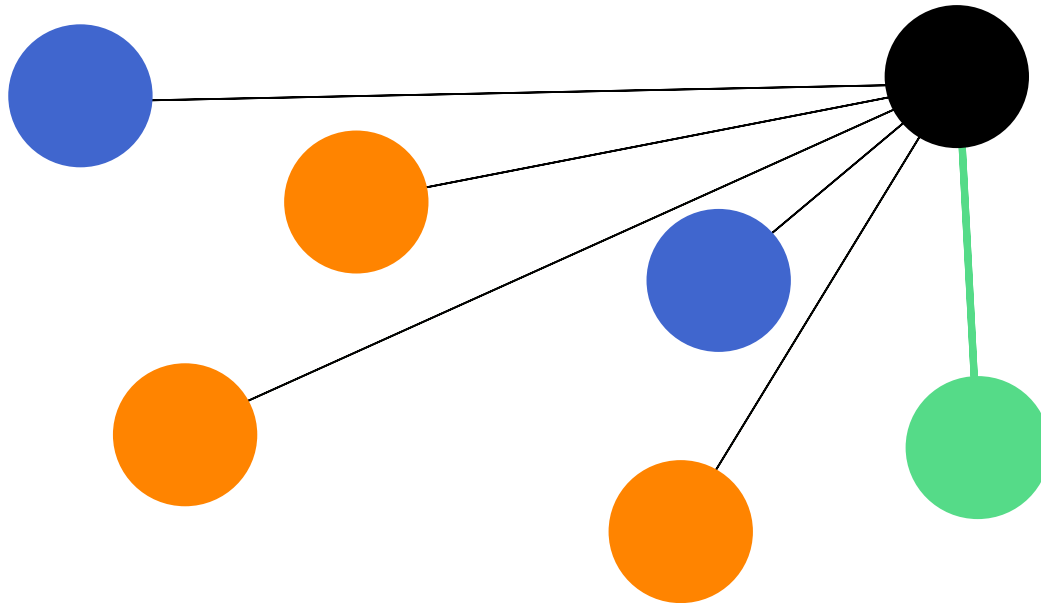


Figure 4:
*Procedure of the
algorithm to
calculate routes.*

After researching of all the different approaches that can be done, we decided to focus in a greedy one, trying constantly to find the nearest client by the time of calculate a route. 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. Besides its good complexity, which is $O(S*W)$ to find each of the different routes, the algorithm has shown a great yet improvable performance in the used dataset.

Time and Memory Consumption

Dataset	Best case (sec)	Average case (sec)	Worst case (sec)
1	0.16804	0.1973	0.26572
2	0.14065	0.15778	0.19604
3	0.14917	0.17525	0.23807
4	0.1563	0.18258	0.25008
5	0.14067	0.16158	0.2274
6	0.15626	0.16626	0.18904
7	0.16359	0.18758	0.21882
8	0.15628	0.16878	0.21387
9	0.09352	0.15085	0.24463
10	0.1094	0.1191	0.1563
11	0.09378	0.10596	0.13384
12	0.09377	0.10833	0.12629

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

Dataset	Memory consumption (MB)
1	1.0567754
2	1.2037474
3	1.1925678
4	1.0484738
5	1.2387344
6	1.1849829
7	1.0487832
8	1.2284026
9	1.1799384
10	1.0418389
11	1.2579394
12	1.2038384

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

Software prototype

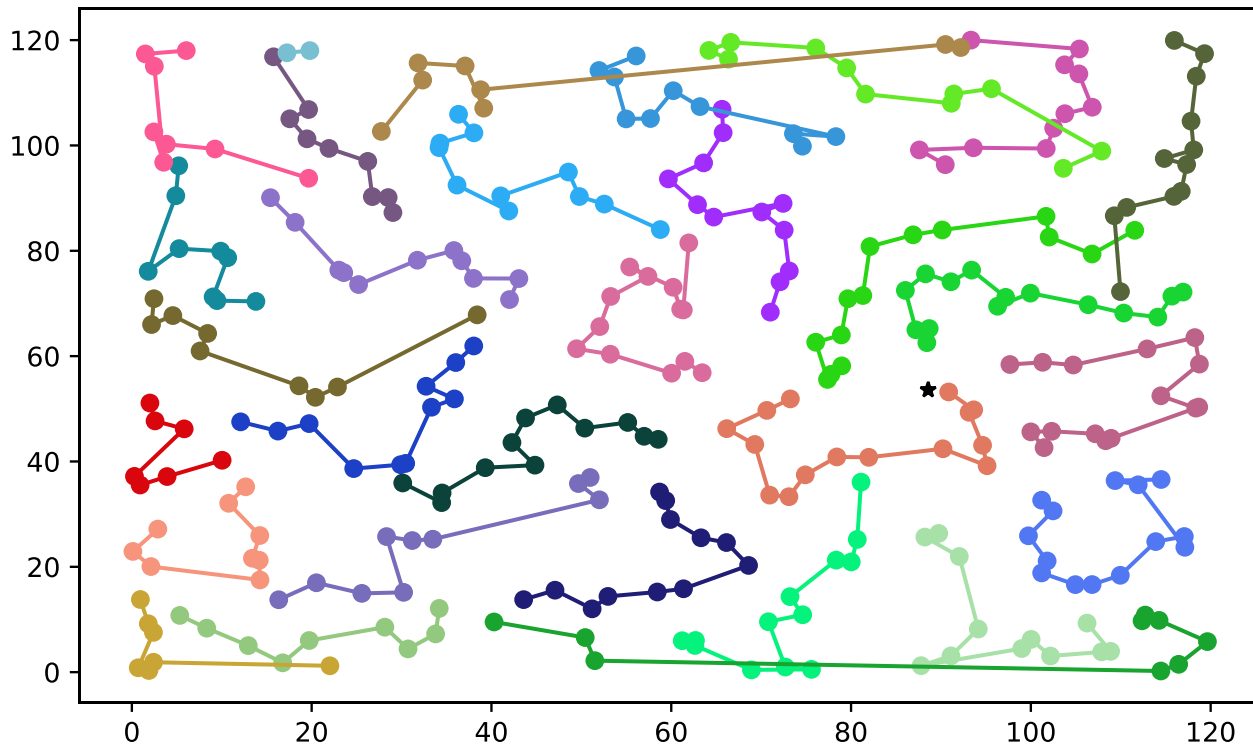


Figure 6: Illustration designed with matplotlib. It represents each of the routes calculated for one of the datasets. Keep in mind that each color represents a route, and every route starts and ends in the black star, which represents the deposit.