Electric Cars: Logistics For the Environment

Juan David Echeverri Villada
Universidad Eafit
Colombia
jdecheverv@eafit.edu.co

Juan Sebastian Guerra Hernandez
University Eafit
Colombia
jsguerrah@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

ABSTRACT
Finding a way to combat climate change is one of the most important priorities in the last few years, that is why reducing the use of fossil fuels is a must for all the people. On other hand, the delivering companies have been growing exponentially and its services are more and more demanded with the time, occasioning high emissions of CO2 due to transportations matter. The electronic vehicles are clearly one of the best alternatives to solve this problem, but due to its characteristics, implementing them represents a complex logistic challenge. Looking to the precedents, the traveler salesman and its variant are problems that have a similar approach and the given solutions to them are a fundamental way to tackle the issue.

ACM CLASSIFICATION Keywords.
•Applied computing → Operations research → Transportation.

•Applied computing → Operations research → Decision analysis → Multi-criterion optimization and decision-making.

•Mathematics of computing → Discrete mathematics → Graph theory → Paths and connectivity problems.

•Theory of computation → Design and analysis of algorithms → Graph algorithms analysis → Shortest paths.

•Theory of computation → Design and analysis of algorithms → Streaming, sublinear and near linear time algorithms → Nearest neighbor algorithms.

1. INTRODUCTION
Due to the fact of the logistics needed to coordinate all the solicitudes and complete them with efficiency, delivering packages has always been a complex topic. Every time that a new technology can be implemented in this ambit, the logistics need to be recalculated, improving the efficiency but giving a hard time to the coordinators. That is the case of one of the most recent technologies developed, the electric cars, whose implementation could significantly improve the use of fossil fuels thus benefiting the environment. Nevertheless, implementing this technology has a certain price.

2. PROBLEM
To explain the problem, it is necessary to know what does implementing electric cars implies since they have certain limitations that must be considered by the time of calculating the logistics needed. Some of them are its limited conduction range and its long charging time, that represent a certain problem for the drivers, because they have a work schedule that cannot be exceeded. From here comes the necessity to adapt the planning and optimize the logistic process. developing a tool that evaluates the routes of each vehicle in a fleet and finds the most proper ones will allow the drivers to administrate efficiently the energy and the time, making easier the implementation of electric cars in this ambit and thus reducing the contamination of the environment.

.3. RELATED WORK

3.1 The Traveler Salesman Problem

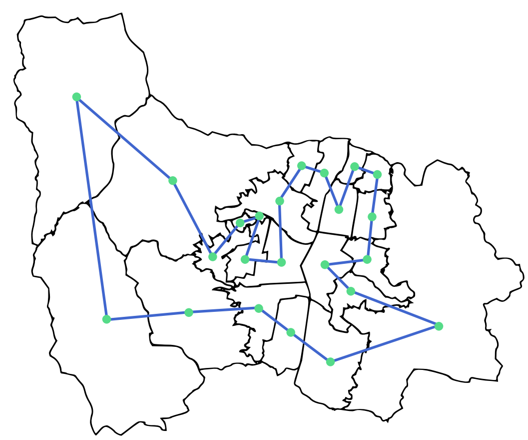

Figure 1: Illustration that represents a generalization of the TSP problem on Medellín.

Proposed by William Hamilton and Thomas Kirkman, consists in searching for the best path that can be carried out in a set of n nodes, passing only once through each node until returning to the first one. Various solutions to this optimization problem have been proposed throughout the years. One of them is the Christofides' algorithm, which consists in choosing and replacing edges to keep less distance. It is a heuristic algorithm since its operation looks for an approximate route with the weights that are raised in the data [1].

3.2 Vehicle Routing Problem (VRP)

The VRP is the distribution of the Traveler Salesman Problem to different vehicles. It "decides which vehicle handles which request in which sequence so that all vehicle routes can be feasibly executed" [2]. In this case, a viable solution is the Clark and Wright savings algorithm, which consists in combining two possible routes, taking the savings that are produced from each one. Thus, in an approximate way, it is possible to attribute the best path for each vehicle.

3.3 Electric Vehicle Routing Problem (E-VRP)

“As corporations are becoming more conscious about the environment and the associated externality costs, electric commercial vehicles are gaining tractions in firms that deliver products" [3]. Under this premise, a variant of the vehicle routing problem was created, the E-VRP, which focuses on finding an optimal routing strategy with minimal travel time cost. energy cost and electric vehicles dispatched. Doing a heuristic approach as random search inspired algorithm or a dynamic programming solution such as Held Karp (with resources constraints) could be the best way to tackle the EVRP.

3.4 Sequential Ordering Problem (SOP)

It is a variant of the TSP where there are precedence conditions. That means that there are certain nodes that must be visited before another ones, therefore, the most optimal route may not be correct. Carina Vega (2014) used the algorithm SOP 3 Exchange to solve this problem [4]. This algorithm has 2 components. The first one oversees checking the infraction of a precedence while the other ones change axis. When changing axis while preserving roads you get in a heuristic way a path that has all the requisites.

4. Greedy Pathfinding applied to VRP

In what follows we explain the data structure and the algorithm.

4.1 Data Structure:

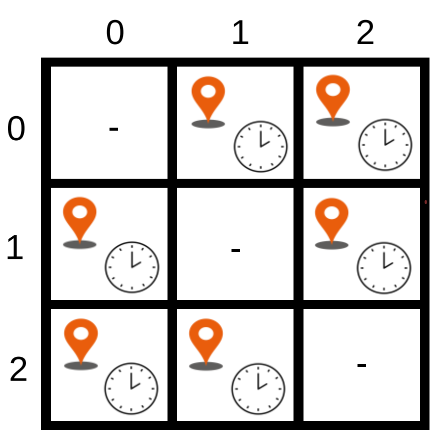

Figure 1: Adjacency Matrix with weights of pairs Distance - Time. Indexes represent the ID of the respective nodes. The diagonal is empty due both IDs refer to the same node.

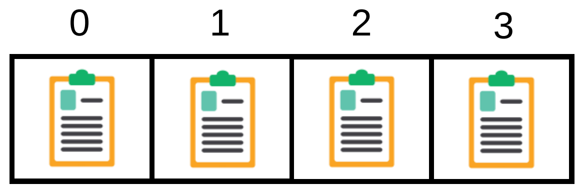

Figure 2: Array that contains the data of each node [Name of the ubication, x coordinate, y coordinate, type of node (either deposit, station, or client) and if is a station, the type of station (Fast, normal, or slow charging)]. Index represents the ID of the node.

4.2 Operations of the data structure

4.2.1 Getting edges:

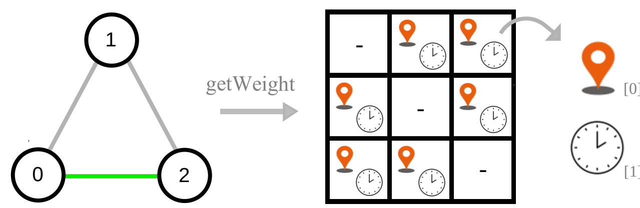

Figure 3: Given two IDs of references, we access to a specific position of the adjacency matrix and extract its data.

4.2.2 Find the successors:

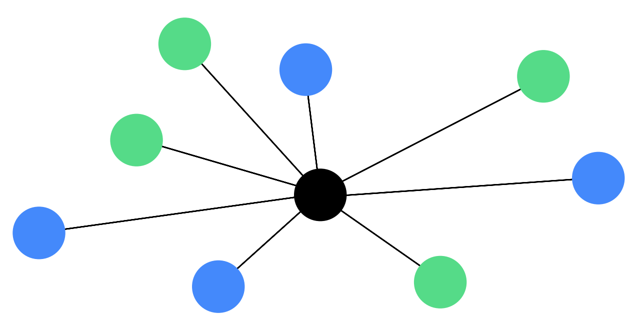

Figure 4: From a start node (black), find all the adjacent nodes (green), discarding those that have been visited (blue).

4.2.3 Find the nearest client:

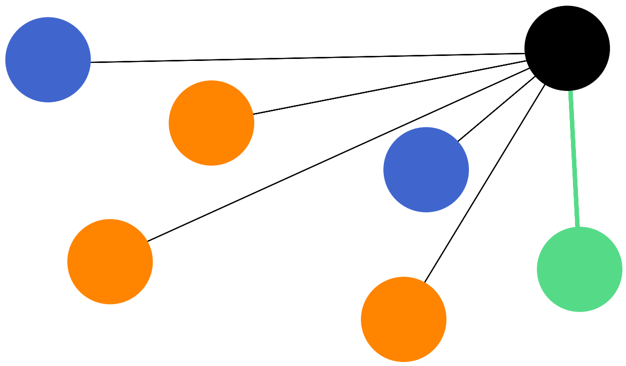

Figure 5: From a start node (black) we search in our adjacency matrix the closest node (green), discarding those that have been visited (blue) and temporarily the other ones (orange).

4.2.4 Show the graph:

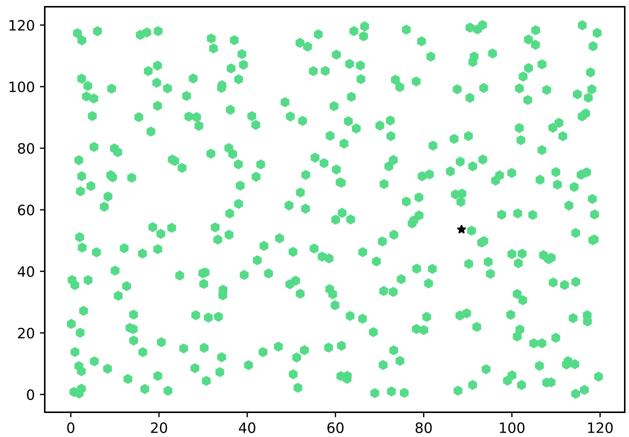

Figure 6: Plot of the graph using *matplotlib.pyplot* library and information saved in the array that has all the information of the nodes. The black star represents the depot, the other ones are clients.

4.3 Design criteria of the data structure

Analyzing the problem, is notorious that every node has a connection with all the other ones, making a map that can be interpreted as a complete graph. Now, knowing the bases of the different types of graphs implementations, it is seen that adjacency list implementation is not optimum at all in complete graph (since it will consume the same memory that adjacency matrix, it only makes worst the access) and for that reason, we choose adjacency matrix. This decision is clearer by the time of searching data in the graph, where adjacency matrix allow us to get it in O(1) unlike adjacency list, who takes O(n) to search.

4.4 Complexity analysis

Firstly, it is necessary the variables that are going to be used to define complexity in each section.

Variable	***Description***
N	Number of points of interest 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.

Operation	***Asymptotic complexity***
Create the graph	$O(N^2)$
Access to an edge	O(1)
Find successors	O(N)
Find the nearest client	O(S)
Show the graph	O(N)

Table 2: Table to report complexity analysis.

4.5 Execution time

Data set	Best case (sec)	Average case (sec)	Worst case (sec)
1	1.03	1.13	2.19
2	1.03	1.12	1.34
3	1.04	1.13	1.27
4	1.04	1.14	1.29
5	1.08	1.15	1.27
6	1.06	1.13	1.29
7	1.14	1.23	1.34
8	1.14	1.24	1.34
9	1.18	1.24	1.37
10	1.13	1.24	1.35
11	1.13	1.21	1.32
12	1.14	1.21	1.3

Table 3: Execution time during the creation of the data structure for each dataset.

4.6 Memory consumption

Data set	Memory consumption (MB)
1	8.5039
2	8.4960
3	6.6992
4	6.9570
5	6.2382
6	6.4414
7	6.4414
8	6.6992
9	5.9257
10	6.4414
11	5.4101
12	5.9257

Table 4: Memory used during the creation of the data structure for each dataset.

4.7 Algorithm

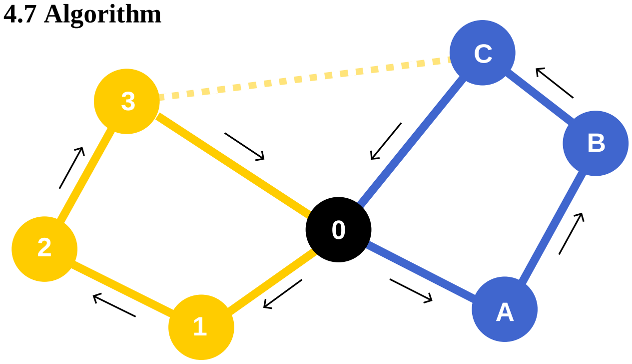

Figure 7:

Step 1: The nearest client is found, and the algorithm travel there.

Step 2: As seen in the figure, step one is repeated until it is not possible to go to another node due to the lack of time (In the figure, the algorithm tries to go from '3' to 'c' but since it is not possible, 'c' is removed from the path and we return to '3', were the algorithm come back to '0' and saves the path).

Step 3: Step 1 and 2 are repeated until all the clients are visited.

4.8 Complexity analysis of the algorithm

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)**

Table 5: Complexity of each step in the algorithm. Dictionary of variables available in Table 1.

4.9 Design criteria of the algorithm

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.

4.10 Execution times

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 6: Execution time of the algorithm for different datasets.

4.11 Memory consumption

Data set	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 7: Memory consumption of the algorithm for different datasets.

4.12 Analysis of the results

Data sets	Executions' time	Memory used	Total vehicles	Routes' time	Total clients
1	1.3273 s	9.5606 MB	34	322 h	320
2	1.2777 s	9.6997 MB	28	265 h	320
3	1.3052 s	7.8917 MB	30	283 h	320
4	1.3225 s	8.0054 MB	34	322 h	320
5	1.3115 s	7.4769 MB	28	265 h	320
6	1.2962 s	7.6263 MB	30	283 h	320
7	1.4175 s	7.4901 MB	34	322 h	320
8	1.4087 s	7.9276 MB	28	265 h	320
9	1.3908 s	7.1056 MB	30	283 h	320
10	1.3591 s	7.4832 MB	34	322 h	320
11	1.3159 s	6.6680 MB	28	265 h	320
12	1.3183 s	7.1295 MB	30	283 h	320

Table 8: Analysis of the results obtained from the creation of the data structure and algorithm execution.

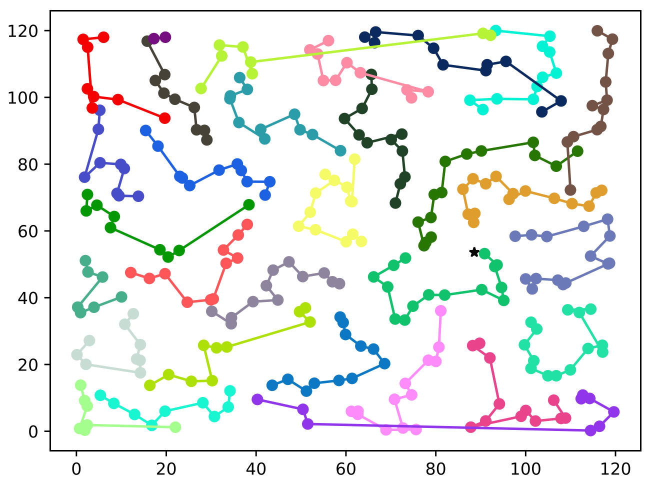

Figure 8: Illustration designed with *matplotlib*. It represents each of the routes calculated for dataset 9. Keep in mind that each color represents a route, and every route starts and ends in the black star, which represents the deposit.

REFERENCES

[1] Bernal, J., Hontoria, E. and Alekvsovski, D., 2015. El problema del viajante de comercio: Búsqueda de soluciones y herramientas asequibles. ASEPUMA, 16(2), pp.117-133.

[2] Irnich, S., Toth, P., and Vigo, D., 2014. Vehicle Routing Problems, Methods, and Applications. Philadelphia: Society for Industrial and Applied Mathematics.

[3] Jane Lin, Wei Zhou, Ouri Wolfson, Electric Vehicle Routing Problem, Transportation Research Procedia, Volume 12, 2016, 508-521, https://doi.org/10.1016/j.trpro.2016.02.007.

[4] Vega, C., 2014. *BRKGA para el Problema de Ordenamiento Secuencial*. [online] Buenos Aires. Available at: <http://dc.sigedep.exactas.uba.ar/media/academic/grade/thesis/vega.pdf> [Accessed 21 February 2021].