# **ALGORITHM USED TO FIND OPTIMAL ROUTES FOR ELECTRIC VEHICLES**

| Maria Paulina Ocampo Duque  University Eafit  Colombia  mpocampod@eafit.edu.co | Jose Manuel Fonseca Palacio  University Eafit  Colombia  jmfonsecap@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |
| --- | --- | --- |

# **ABSTRACT**

The objective of this project is to design an algorithm that finds optimal routes to electric vehicles, in this specific case to make more efficient the way of distribute merchandise. This in order to solve the problem which is presented in the batteries of the cars for having a limited durability range and a very long loading time.

# **1. INTRODUCTION**

In the following report it is intended to solve the problematic that companies have when delivering products with electrical cargo trucks, for them to be able to deliver them in the most efficient way, taking into account working hours, number of cars and the lifespan of the car battery. Solving this problem would be able to improve deliveries and assure that companies will use electric cars, even with the need of charging them, allowing a reduction in the carbon emissions released by this cargo trucks.

# **2. PROBLEM**

The problem is based on the low efficiency that electric car batteries have to distribute merchandise, because they take a long time to make the routes due to the battery is constantly unloaded. This has a high impact on society because if it is not solved, transporters would be forced to use other vehicles more harmful to the planet, that is why to prevent that from happening it is necessary to solve this problem.

## **3. RELATED WORK**

## **3.1 Dijkstra´s algorithm**

Shortest path problem is a way to find the shortest possible distances between two vertices on a graph such that the sum of the weights of its constituent edges is minimized, for that reason Dijkstra´s algorithm can help us because is an algorithm that we can use to find the shortest distances or minimum costs depending on what is represented in a graph. Through following steps:  
-Start at the ending vertex by marking it with a distance of 0

-Identify all of the vertices that are connected to the current vertex with an edge

-Label the current vertex as visited by putting an X over it

-Of the vertices you must marked, find the one with the smallest mark, and make it your current vertex.

-Once you’ve labeled the beginning vertex as visited, stop. [1]

## **3.2 Random walk algorithm**

The problem in this algorithm is to find, after some fixed time, the probability [distribution function](https://www.britannica.com/science/distribution-function) of the distance of the point from the origin. So it solution is the Random walk algorithm, which is a process for determining the probable location of a point subject to random motions, given the probabilities of moving some distance in some direction. [2]

## **3.3 The traveling salesman problem**

The traveling salesman problem consists you finding the shortest path that visits each possible city and returns to the original city. [3]  
  
The problem has multiple solutions and is a NP- hard problem since the number of possibilities for the solution when given a big number of cities is immense. Due to that solving it through brute Force is not recommended but to find an optimization method. Solutions to this type of problem are starting to be solved by swarm type algorithms inspired in bees, where the traveling variable (the salesman in this problem) has certain Independence yet a communication with the hive. [4]

**3.4 Floyd-Warshall algorithm**

The Floyd-Warshall algorithm is a problem that looks to find the shortest path between every pair of vertices in a given edge weighted directed Graph. The condition for the problem to work is that it must be as mentioned before a weighted graph, it can be directed or undirected, but it does not work with negative cycles (The sum of the edges in the cycle is negative). Such algorithm can be of help since for a graph made of arrays it would find the fastest, or cheapest path from each vertex in the graph. The way it works is that for each vertex it tries all the possible paths including the vertexes it passes through and if its smaller than the specified it is replaced in the matrix.[5][6]

**You should mention the fourth algorithmic problem and its solution.**

# **REFERENCES**

[1] Pennington, L., n.d. *Dijkstra's Algorithm: Definition, Applications & Examples*. [video] Available at: <https://study.com/academy/lesson/dijkstra-s-algorithm-definition-applications-examples.html>

[Accessed 14 February 2021].

[2] Encyclopedia Britannica. 2008. *Random walk | mathematics and science*. [online] Available at: <https://www.britannica.com/science/random-walk>

[Accessed 14 February 2021].

[3] *Travelling Salesman Problem | Set 1 (Naive and Dynamic Programming) - GeeksforGeeks*. *GeeksforGeeks.com*, 2018.

Available at: <https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/>

[Accessed 15 February 2021].

[4] Nunes de Castro, L. and A. S. Masutti, T. *hindawi.com*, 2017. *Bee-Inspired Algorithms Applied to Vehicle Routing Problems: A Survey and a Proposal.* [online] Available at: <https://www.hindawi.com/journals/mpe/2017/3046830/>

[Accessed 15 February 2021].

[5] *Floyd-Warshall Algorithm*. [online] Available at: <https://www.programiz.com/dsa/floyd-warshall-algorithm>

[Accessed 15 February 2021].

[6] GeeksForGeeks. 2021.*Floyd Warshall Algorithm | DP-16 - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/.>

[Accessed 15 February 2021]