Optimal route: An algorithm that simplifies the distribution of merchandise

Carla Daniela Rendón
Universidad Eafit
Colombia
cdrendonb@eafit.edu.co

Juan Diego Gutierrez
Universidad Eafit
Colombia
jdgutierrez@eafit.edu.co

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

ABSTRACT
We are in the era of technology and information, and many of the problems that arise in real life are trying to solve by using computer artifacts and algorithms, for purposes of this project we will make an algorithm that finds the most optimal route to distribute a specific merchandise in certain points of a bidimentional map

1. INTRODUCTION

In real life, especially in the industry of distribution of merchandise, it is very important to find the most optimal route, since this represents a lower cost, time saving and above all customer satisfaction. For the realization of this project, we will create an algorithm that allows us to find the most optimal route between points.

2. PROBLEM

The problem to be solved consists in making an algorithm to find the most optimal route, considering a limited set of electric vehicles, a list of clients located in a bidimensional map and a specific quantity of merchandise to be distributed.

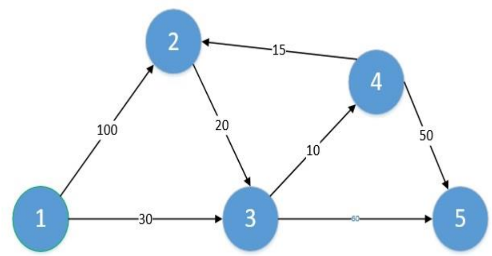
3. RELATED WORK

In this section we present 4 works related to the problem initially proposed.

3.1 The problem of the shortest path

In the theory of graphs, the problem of the shortest path is the problem of finding a path between two vertices (or nodes) in such a way that the sum of the weights of the edges that constitute it is minimal. An example of this is finding the fastest way to go from one city to another on a map. In this case, the vertices would represent the cities and the edges the roads that join them, whose weighting is given by the time used to cross them.

For the solution of this problem we propose the use of the Dijkstra algorithm, also called the minimum path algorithm, it is an algorithm for the determination of the shortest path, given a vertex origin, towards the rest of the vertices in a graph that has weights in each edge. Its name refers to Edsger Dijkstra, a computer scientist from the Netherlands who first described it in 1959.

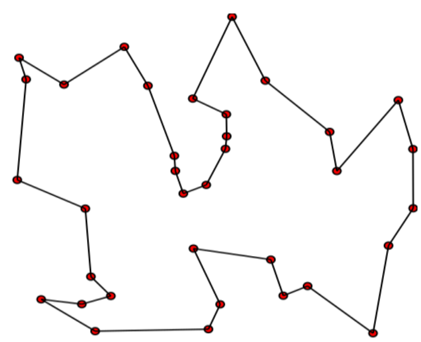


Taken from: [4]

3.2 Traveling Salesman Problem

Traveling Salesman Problem (TSP) answers the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route each city visits exactly once and at the end return to the origin city? This is a NP-Hard problem in combinatorial optimization, very important in operations research and in computer science.

The TSP has various applications even in its simplest formulation, such as: planning, logistics and in the manufacture of electronic circuits. A little modified, it appears as: a sub-problem in many areas, such as in the DNA sequence. In this application, the concept of "city" represents, for example: customers, welding points or DNA fragments and the concept of "distance" represents the travel time or cost, or a measure of similarity between the DNA fragments. In many applications, additional restrictions such as the resource limit or the time windows make the problem considerably difficult. The TSP is a special case of traveling problems (traveling purchaser problem).

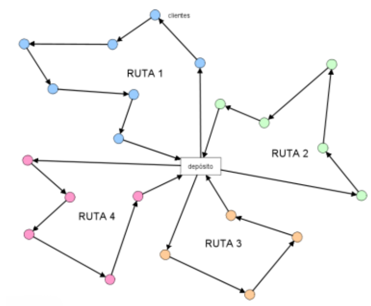


Taken from: [2]

3.3 Vehicle Routing Problems

Vehicle Routing Problems (VRP) are a broad set of variants and customizations of problems. From those that are simpler to some that are still a research subject today.

In them in general, it is about finding out the routes of a transport fleet to service some customers. This type of problems belongs to combinatorial optimization problems. In the scientific literature, Dantzig and Ramser were the first authors in 1959, when they studied the real application in the distribution of gasoline for fuel stations.



Taken from: [1]

3.4 The problem of the Königsberg bridges

The problem of the Königsberg bridges, also called more specifically the problem of the seven bridges of Königsberg, is a famous mathematical problem, solved by Leonhard Euler in 1736 and whose resolution gave rise to the theory of graphs.1 His name is due to Königsberg, the city of East Prussia and then Germany that since 1945 would become the Russian city of Kaliningrad.

The problem, originally formulated informally, was to answer the following question: "Given the map of Königsberg, with the Pregel River dividing the plane into four different regions, which are linked through the seven bridges, is it possible to take a walk starting from any of these regions, going through all the bridges, traveling only one each time, and returning to the same starting point?" The answer is negative, that is, there is no route with these characteristics. The problem can be solved by applying a brute force method, which implies testing all possible existing routes.

AUTHOR KEYWORDS

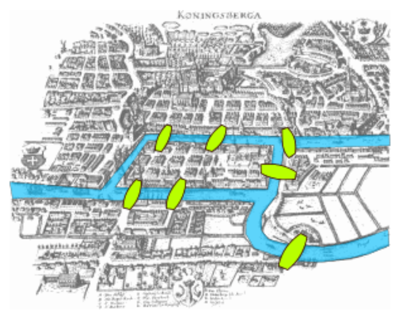
Algorithm: a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Data structure: is a data organization, management and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.

Graph(data structure): A set of items connected by edges. Each item is called a vertex or node. Formally, a graph is a set of vertices and a binary relation between vertices, adjacency. A graph G can be defined as a pair (V,E), where V is a set of vertices, and E is a set of edges between the vertices E ⊆ {(u,v) | u, v ∈ V}. If the graph is undirected, the adjacency relation defined by the edges is symmetric, or E ⊆ {{u,v} | u, v ∈ V} (sets of vertices rather than ordered pairs).

ACM KEY WORDS

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



Taken from: [3]

DESIGN OF THE DATA STRUCTURE:

For the design of the data structure we consider certain criteria, first we apply an array of fixes; to divide the map into different quadrants. Clients and recharging stations that belong to a quadrant will be stored in an array

To each vehicle we assign a quadrant to travel, where the vehicle will visit all customers in that quadrant including charging stations, we apply a greedy algorithm in each quadrant to travel the nodes to optimize the path of each vehicle (when we reduce the number of nodes by dividing them in quadrants we optimize the execution time of the greedy algorithm)

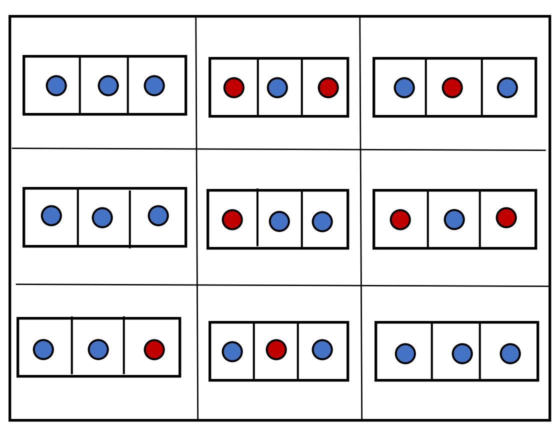


Figure 1.

DESIGN OF THE DATA STRUCTURE OPERATIONS

To divide the map into rows and columns we apply an algorithm to calculate the two factors closest to each other that multiplied give us the number of quadrants.
To create a node, we assign the proposed coordinates, the type of node and its id; and then it is added in the corresponding quadrant.
As we mentioned earlier in the design of the data structure, each vehicle corresponds to a quadrant, the route of that quadrant is made by always visiting the nearest node. If there is a charging station inside the dial, it will always be visited, and the battery will be recharged. If the battery is less than 30% and there is not a recharge station within the quadrant, the one closest to the quadrant will be evaluated and it will be visited to recharge.
Once a quadrant is traveled the route will be stored in a matrix that represents the quadrants. To calculate the time traveled by a vehicle we use the formula t = v / d.

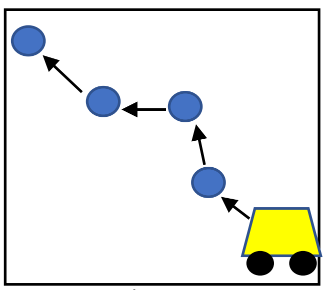


Figure 2.

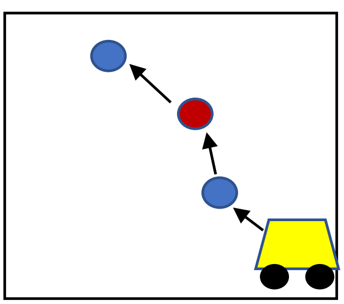


Figure 3.

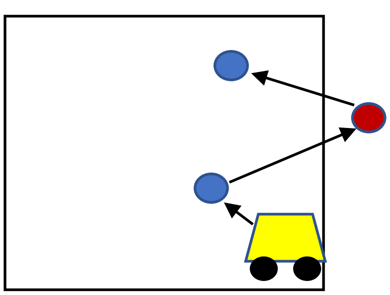


Figure 4.

COMPLEXITY OF THE ALGORITHM OPERATIONS

OPERATION	COMPLEXITY
Read the file	O(N), N being the number of lines
Divide into Quadrants	O(N), N being the number of vehicles
Create Quadrants	O(N x M), N being the rows and M the columns
Create Node	O(1)
Add Node	O(N), N being the index in the array where the node is stored
Travel	O(N x M x P), N being the rows, M the columns and P the number of nodes in the array
Calculate Distance	O(1)
Calculate Time	O(1)
Finish Travel	O(N), N being the number of clients in the quadrant
Add Routes	O(N), N being the index of the array where the node is stored
Print Routes	O(N), N being the number of quadrants
Print Time	O(1)
Total Complexity	O(N x M x P), N being the rows, M the columns and P the number of nodes in the quadrant. N x M it tends to be N ^ 2 since N and M are very close or equal

DATA STRUCTURE DESING CRITERIA

We decided to apply this data structure as we consider that it is optimal to solve the problem. In the given problem we have a map with many nodes, directly applying a greedy algorithm, it would not be the most efficient solution, but, by dividing the map into quadrants, we create smaller maps where we reduce the number of nodes, thus optimizing the execution of greedy algorithm, applying it to quadrants with few nodes.

Also in our data structure we implement an algorithm that optimizes the division of the map into quadrants, calculating the two closest factors that multiplied give us the number of quadrants.

RESULTS OBTINED

Number of Vehicles	Execution Time
6	1068 ms
20	2001 ms
42	2069 ms
100	2015 ms
200	2070 ms
300	2082 ms

CONCLUTIONS

In real life, especially in the industry of distribution of merchandise, it is very important to find the most optimal route, since this represents a lower cost, time saving and above all customer satisfaction. In this project we create an algorithm where we apply an array of fixes.

We can improve the algorithm carried out, by taking the map of nodes as a circle and considering the center point of the circle as the deposit of vehicles and divide from that point center by angles, so the routes of each car will be divided more equally

THANKS TO

This research was supported supported by R.C.J. Services. S.A.S.

REFERENCES

1. Vehicle routing Problem, taken from, https://en.wikipedia.org/wiki/Vehicle_routing_problem
2. Traveling Salesman Problem (TSP), taken from, https://en.wikipedia.org/wiki/Travelling_salesman_problem
3. Seven Bridges of Königsberg, taken from, https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg
4. Shortest path problem, taken from, https://en.wikipedia.org/wiki/Shortest_path_problem

TEAMWORK

Since the beginning of the semester, the work was distributed equally, Juan Diego Gutierrez, researched, for the first 2 solutions, **The problem of the shortest path** and **Traveling Salesman Problem,** Carla Rendón, investigated **Vehicle Routing Problems** and **The problem of the Königsberg bridges**. The introduction, the abstract and the approach of the problem, was carried out by both team members. Carla Rendón was responsible for making the author keywords and Juan Diego Gutierrez the ACM keywords. The desing of the data structure, the complexity of the algorithm, the data structure desing criteria, the conclusions and final notes were made by both team members.