The best way to reach your destination safely and quickly using area index

Victor Villadiego Alvarez
Universidad Eafit
Colombia
vjvilldia@eafit.edu.co

Julián Agudelo Cifuentes
Universidad Eafit
Colombia
jagudeloc@eafit.edu.co

Andrea Serna
Universidad Eafit
Colombia
asernac1@eafit.edu.co

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

ABSTRACT

For years the human had been trying to find the best possible way to move from one place to another, which makes it necessary for tools that satisfy this in terms of security and clearly quickness, two words that currently have to be the daily bread of individuals, companies and nations, and to improve this, a software will be designed that allows these variables to be evaluated at the same time and contributes to any user a route that gives them peace of mind to reach their destination with any harassment signal or something similar.

Keywords
Shortest route, street sexual harassment, identification of safe routes, crime prevention

1. INTRODUCTION

During the last decades the man has been witness of the great technological advance, and the rise of new software and applications that facilitate the displacement of these, but leaving behind issues of great social importance, such as safety, because although they offer the user a very short and fast route, they do not guarantee the user the risks and conditions that exist when taking certain roads, putting the life of its users at risk.

That is why it is necessary to use an application that allows to combine the two variables in a balanced way and thus to obtain an optimal response from the system, giving the user to understand that we care about their safety and also the speed with which they want to reach their destination.

1.1. The problem

Man as an animal, needs a place of comfort that provides the best options based not only on a variable, but in varied, since very recondite years the human being has tried to find a way to know their environment, and based on a set of probabilities make the best decision, the conquerors had to reach the calmest waters, on the other hand, a tourist to get to your hotel without risk, a worker go to office without having to be harassed, etc., that is why it is necessary that through a medium these people should know what is around them, so they can avoid all kinds of situations that could put at risk their mental health, physical and even their life.

1.2 Solution

We search different possibilities and different algorithms that can solve our problem and the most effective is the algorithm of Dijkstra, it is one of the most effective and used algorithms for search the easier route, we were in mid the distance and the risk for select the best route.

1.3 Structure of the article

Next, in Section 2, we present work related to the problem. Then, in Section 3, we present the datasets and methods used in this research. In Section 4, we present the algorithm design. Then, in Section 5, we present the results. Finally, in Section 6, we discuss the results and propose some directions for future work.

2. RELATED WORK

1.3 Structure of the article

Next, in Section 2, we present work related to the problem. Then, in Section 3, we present the datasets and methods used in this research. In Section 4, we present the algorithm design. Then, in Section 5, we present the results. Finally, in Section 6, we discuss the results and propose some directions for future work.

2. RELATED WORK

Below, we explain four works related to finding ways to prevent street sexual harassment and crime in general.

2.1. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search

This work talks about the use of heat maps, where each zone, divided into certain grid points, will be marked with a different color, from intense red (very unsafe) to intense green (safe), using a mathematical axiom called Bresenham's line drawing method, which allows to draw a straight line and evaluate each point through which it passes.[1]

2.2 Incorporating a Safety Index into Pathfinding

It focuses on many variables related to the intersections between streets, the quality of the pavement on a certain route, the type of vehicle in which the person is trying to move, adding to each route an index that is measured in terms of DRAC and MADR which varies by user, where it measures the possibility of deceleration in the event of a possible collision, giving the user three options, the first one that evaluates safety and speed, the second would be the shortest route and finally the safest one. [2]

2.3 Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data

In this document, the author try to explain how to carry out a displacement between two sites, where variables are evaluated taking into account previous records of cases more exactly in the city of new York with the use of machine learning, to use the data of crime and accident rate, then based on this to obtain an index that measures the degree of security of this area by handling these two components, which for many other software is not relevant.[3]

2.4 Safety-aware routing for motorised tourists based on open data and VGI

This paper seeks to ensure the safety of each of its users, implementing a system based on the collection of crime data, making a certain type of average that allows to assign to sites or polygons as it is called in the study to a certain area, data that will be collected based on police studies of the environment in which it is located, in the same way to obtain this information from people who are already witnesses of what happens in certain areas and thus be more accurate. [4]

3. MATERIALS AND METHODS

In this section, we explain how the data were collected and processed, and then different alternative path algorithms that reduce both the distance and the risk of sexual street harassment.

3.1 Data collection and processing

The map of Medellín was obtained from *Open Street Maps* (OSM)[1] and downloaded using the Python API[2] OSMnx. The map includes (1) the length of each segment, in meters; (2) the indication of whether the segment is one-way or not, and (3) the known binary representations of the geometries obtained from the metadata provided by OSM.

For this project, a linear combination (LC) was calculated that captures the maximum variance between (i) the fraction of households that feel insecure and (ii) the fraction of households with incomes below one minimum wage. These data were obtained from the 2017 Medellín quality of life survey. The CL was normalized, using the maximum and minimum, to obtain values between 0 and 1. The CL was obtained using principal components analysis. The risk of harassment is defined as one minus the normalized CL. Figure 1 presents the calculated risk of bullying. The map is available on GitHub[3] .

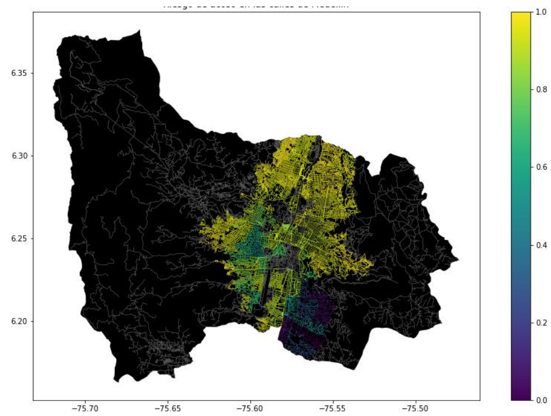

Figure 1. Risk of sexual harassment calculated as a linear combination of the fraction of households that feel unsafe and the fraction of households with income below one minimum wage, obtained from the 2017 Medellín Quality of Life Survey.

3.2 Algorithmic alternatives that reduce the risk of sexual street harassment and distance

In the following, we present different algorithms used for a path that reduces both street sexual harassment and distance.

3.2.1 A*

The A* The algorithm is one of the best and most used for find the best route with point A to point B, but this algorithm tries to use only the best kinds of routes using a mathematical formula reducing a lot of possibilities that it takes to find the best way to arrive at the final node. This algorithm put a

[1] https://www.openstreetmap.org/

[2] https://osmnx.readthedocs.io/

[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets

value for every possibility of move gave by the next mathematics formula:

F(n) = Total Value

g(n) = The value for move for the next node

h(n) = The value of the heuristic for the next node

Heuristic (h(n))

$$h(n) = \sqrt{(x_{(n)} - x_{(nf)})^2 + (Y_{(n)} - Y_{(nf)})^2}$$

$X_{(n)}$= The position in x you will move

$X_{(nf)}$= The position in x of the final node

And is the same with “Y”

After know about the value, this formula gave about all the possibilities of moves, it takes the move who have the min value, until arrive the final node

Complexity

In the worst case of the heuristic the complexity is $O(2^n)$

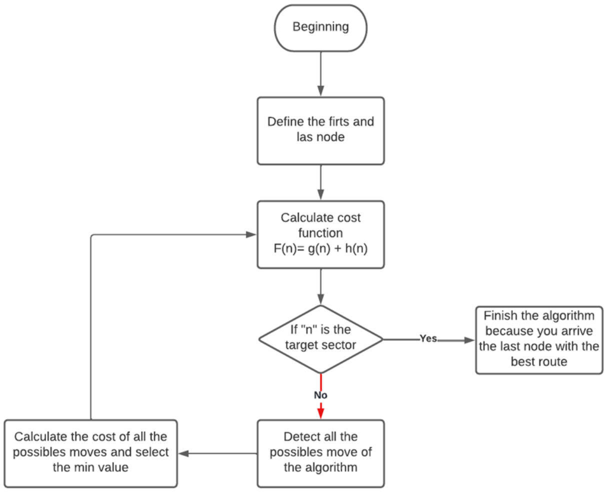
Beginning
Define the firts and las node
Calculate cost function F(n)= g(n) + h(n)
If "n" is the target sector
Yes
Finish the algorithm because you arrive the last node with the best route
No
Detect all the possibles move of the algorithm
Calculate the cost of all the possibles moves and select the min value

Figure 2. Flowchart of the A* algorithm

3.2.2 Dijkstra

The Dijkstra algorithm is a good choice for search the best route, the methodology of this algorithm is based in the interactions, that means this algorithm is better with a small matrix. This defines labels starting in the first node and for the next nodes in the possible routes, these labels contain some information to define which is the best route, this information are the accumulated value, the last node of the route and the number of interactions for arrive at this point. We have two different types of labels, temporals and permanents, the temporals are the different possibilities of the shortest route

Label format:

Label = [accumulate, last node] interactions

At the end all the routes arrive the final node are compared by the accumulate to select the min value.

Complexity

The complexity of this algorithm is $O(n^2)$ and n is the number of vertexes

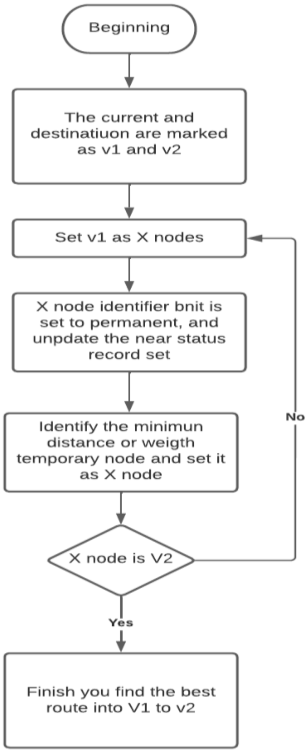
Beginning
The current and destinatiuon are marked as v1 and v2
Set v1 as X nodes
X node identifier bnit is set to permanent, and unpdate the near status record set
No
Identify the minimun distance or weigth temporary node and set it as X node
X node is V2
Yes
Finish you find the best route into V1 to v2

Figure 3. Flowchart of the Dijkstra algorithm

3.2.3 DFS

Depth first search is a algorithm for search routes in a grapho, as it names say this algorithm first go for one branch until arrive the end of the this, The algorithm can work with two ways of programming, those are using stack or recursion. The most efficient mode is using recursion. For begin you need to select the first node like the majority of this algorithms and later the program classifies the nodes in the next way visited or adjacent to the node the program is found, for the next step select one of the nodes adjacent but first it checks if that node was visited before, when the program is located in a node without adjacent nodes, it returns until found a adjacent node that is no visited until found the final node.

Complexity

The complexity of this algorithm is O(v+e), v and e are the total number of vertexes

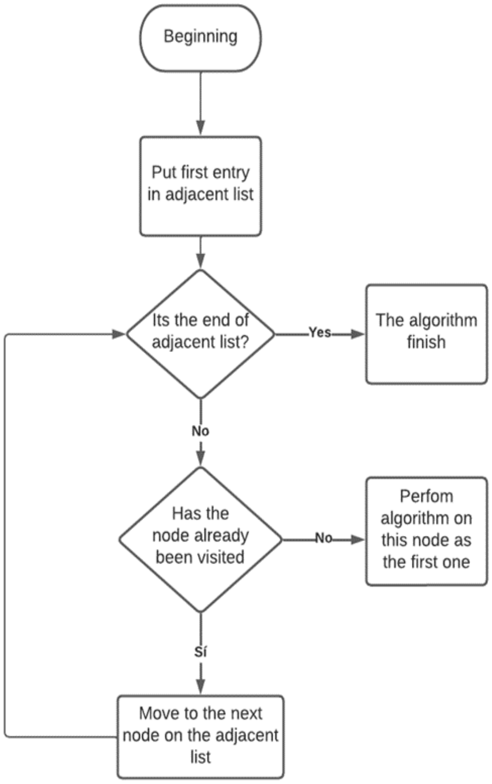
Beginning
Put first entry in adjacent list
Its the end of adjacent list?
Yes
The algorithm finish
No
Has the node already been visited
No
Perfom algorithm on this node as the first one
Sí
Move to the next node on the adjacent list

Figure 4. Flowchart of the DFS algorithm

3.2.4 Floyd

The algorithm Floyd is used for search the best route in a grapho, the unique method it uses is go across all the grapho the number of nodes it haves, that means this algorithm do a lot of operations, this algorithm create two matrix the first have the values until one node to other if those have a direct connection (whit out pass for another node) and the second is the matrix with the different routes, initially the matrix with the route is the same letter of the column

The first matrix (distances or values)

	A	B	C	D
A	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)
B	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)

C	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)
D	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)	Value of the distance into those points (if exist a route if not you put infinite)

The second Matrix(routes)

	A	B	C	D
A	A	B	C	D
B	A	B	C	D
C	A	B	C	D
D	A	B	C	D

Those matrixes are the base of this algorithm in the first as I said you put the values or the distance into the points if those have a direct connection if not put a infinity, later compare the values for get the better route changing the second matrix, finding the best route for connect all the nodes

Complexity

The complexity of this algorithm is $O(n^3)$

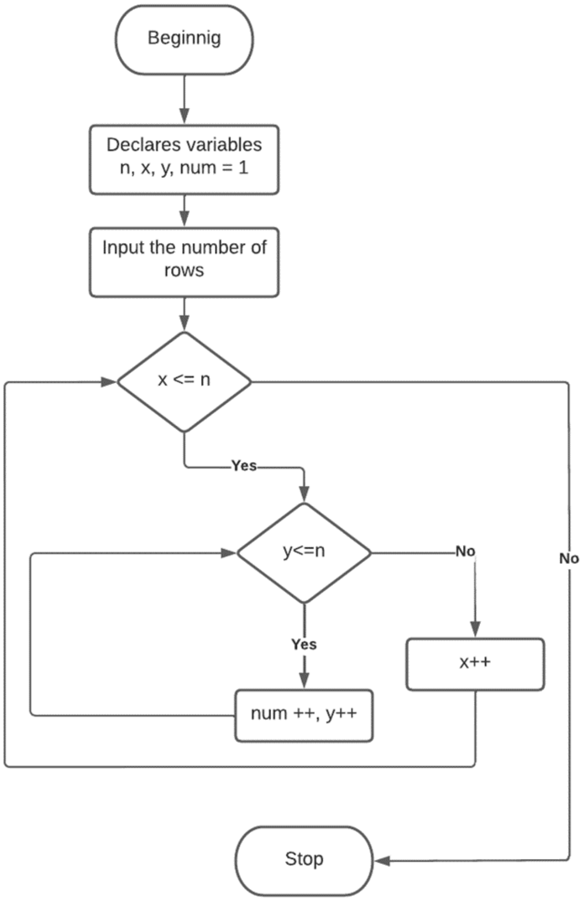

Figure 5. Flowchart of the Floyd algorithm

4. ALGORITHM DESIGN AND IMPLEMENTATION

In the following, we explain the data structures and algorithms used in this work. The implementations of the data structures and algorithms are available on Github[4] .

[4] https://github.com/jac123desu/Project

4.1 Data Structures

The structure we use for represent the map of Medellin it the adjacency list with dictionaries, because in this way we use less memory and it is easier to access of any position in the code, the key of the dictionaries are the coordinates of all the streets of the map and they are associates with a array with all the streets are connected with them, that it's the way we represent the map in the code. The data structure is presented in Figure 6.

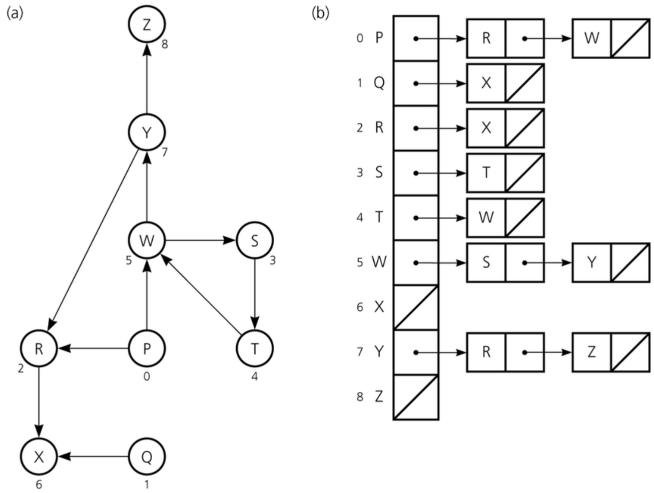

Figure 6: It is a representation of the adjacency list.

4.2 Algorithms

In this paper, we propose an algorithm for a path that minimizes both the distance and the risk of street sexual harassment.

4.2.1 Algorithm for a pedestrian path that reduces both distance and risk of sexual street harassment

We selected Dijkstra algorithm for search the easier route in two points we have in mind two different aspects those are, the risk and the distance, we multiply de risk by 100 because we want to focus in the risk, that is the reason we multiply the risk by 100 later we add the risk and distance and that was the weight we put for each Node, After that the algorithm take one Node and called that current node and the distance for that node Its 0 and the others put "inf" later try with all the connections the Node have in this case "Neighbor's", the put a weight for all the route after that select the Node with the minimum weight and it do that until arrive the destination. The algorithm is exemplified in Figure 6.

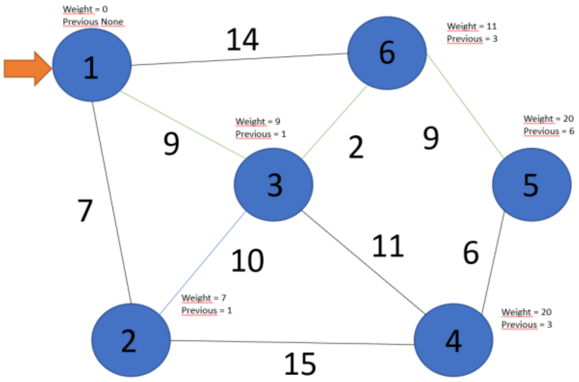

Figure 6: Example of the Dijkstra algorithm

REFERENCES

1. He, Z. and Qin, X. Incorporating a Safety Index into Pathfinding. SAGE Journals. 2017

2. Ma, D. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search. Omdena 2020

3. Gauri, V ans Sandeep C. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data . ResearchGate. 2019

4. Keler A and Damascene, J. Safety-aware routing for motorised tourists based on open data and VGI. Journal of Location Based Services. 2016

5 Aymar, M. and Barahona, J. and Gomez, F.and Martinez, L: INTELIGENCIA ARTIFICIAL (BÚSQUEDA A ESTRELLA), San Simon. 2019

6 Jing-Chao, C. Dijkstra's Shortest Path Algorithm. Donghua University Shanghai, d January 2, 2004

7 Asghar, A. Amir, S. Speeding up the Floyd–Warshall algorithm for the cycled shortest path problem. Applied Mathematics Letters January 2012.

8 Baruch, A.A new distributed Depth-First-Search algorithm. Information Processing Letters. 8 April 1985,