Project for the creation of fast and safe routes to avoiding sexual harassment and violence

Mariana Gutierrez
Universidad Eafit
Colombia
mgutierre@eafit.edu.co

Esteban Giraldo
Universidad Eafit
Colombia
egiraldo@eafit.edu.co

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

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

ABSTRACT

In this project, we'll talk about street sexual harassment, violence, and insecurity in general when picking a route to get to our destination we'll also review some of the related work, algorithms, and possible solutions to street sexual harassment, these solutions will be based on coding, data treatment, and route-generating programs.

Also, we'll take conclusions about the most efficient algorithm or program for our goal and analyze the related work and its results.
To solve the problem, we proposed the implementation of Dijkstra's algorithm using a priority queue, the code can calculate a safe route within seconds (0.38, 0.69, and 0.70 seconds each).

Keywords

The shortest route, street sexual harassment, identification of safe routes, crime prevention

1. INTRODUCTION

In today's world is well known that sexual harassment is an unsolved issue that affects pedestrians (mostly females), without them having much to do about it, therefore we want to create a code and algorithm to calculate paths that reduce the risk, time, and distance to easily avoid sexual harassment.

1.1 The problem

Street sexual harassment is a non-illicit practice that affects women around the world, this behavior is wrong because it negatively affects pedestrians, meaning they cannot feel safe when going out by themselves or even accompanied.

Street sexual harassment is described as the action of harassing, physically or verbally, performing acts of exhibitionism, touching, or any other sexual behavior, may it be a public or semi-public space, like urban transport, the sidewalk, or the streets in general.

1.2 Solution

Explain, briefly, your solution to the problem *(In this semester, the solution is a pedestrian algorithm to reduce both the distance and the risk of harassment. Which algorithms did you choose? Why?)*

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 Avoiding crime while traveling

The problem this model looks up to prevent is criminality while traveling, developed by Shivangi Soni, Venkatesh Gauri Shankar, and Sandeep Chaurasia, the model focuses on a more specific problem than simply traveling fast, as software like Google maps would do, they aimed to develop a software based on NYC OpenData for the data some machine learning algorithms to determine the risk scores of the regions.

This model is a well-planted solution for crime in general, the authors have said they would like to take into consideration more factors for the optimization of the algorithm, but what they don't propose is more specific kinds of crime, like, and based on our objective street sexual harassment.

The investigators that proposed this model used a kNN algorithm which is a Machine Learning algorithm that uses regression to determine the most efficient/fastest route, they implemented this algorithm to return the score of the routes. This model gets information from a database with information about criminality rates, then it receives a user input, equivalent to the destination and the starting point, then it calculates possible routes to the destination, the next thing is to determine if there is more than one route if there is it will display it if there are more it'll use the kNN regression to look the scores of the routes and then return the most efficient.

The proposed model proved to be efficient by creating paths that avoid reported crimes, a downside of this model would be the fact that it uses information that is kind of outdated, this is because it takes data from arrest and accident datasets, meaning this info is linked directly to the efficiency of the NYPD, this doesn't mean is badly implemented, but as the investigators say it would be more efficient if it had more factors and more updated data.

2.2 Safest routes from home to school in Madrid

The motivation for this project started because of a study by the City of Madrid in 2007, under the name of "Camino seguro al cole" or "Safe trip to school" searching to improve the child's safety and autonomy while going to school, thus improving pedestrians' safety.

They used diverse types of data processing: CHIC and barycentric coordinates were used to obtain a safety rating for each street, then these ratings will be used in Dijkstra's algorithm to find the shortest path.

This work creates a map, this map shows the safest route, helping pedestrians (Mostly children) to get to school safely, meaning it solves in a reduced way issues related to crime and harassment, but it doesn't mean that the problem is completely solved, it provides an alternative to the common routes, which is why is related to our project.

2.3 School patrol program and emergency routes based on Shortest Path Problem

School violence is an issue that produces negative effects on its victims and society, victims of school violence tend to develop a lack of trust in the system or other people, other effects could be violence against others, depression, or even suicide.

This project aims to create software that helps police patrols get to a place with a report in the shortest time possible, meaning a patrol would be on the place quickly.

The algorithm used is the Shortest Path Algorithm and the routes were selected by the Floyd-Warshall algorithm. This algorithm is very similar to Dijkstra's algorithm which is a regression algorithm that calculates possibilities regarding routes and then selects the most efficient one.

The results provided by this method proved to be useful in reducing the time invested to get to the destination, thus improving the treatment of the problem of school violence.

Even though this method is useful in various scenarios, it is not directly related to street sexual harassment, but it shows tangible results proving the value of the Shortest Path Algorithms.

2.4 Safe urban mobility using crowdsourcing Incident Data:

Urban areas aren't as safe as they should be when taking into consideration the fact that they should be safer because of police, vigilance, etc.

We should think about risk when picking a route to get to our destination, and route-generating programs such as Google Maps don't consider incidents, crime, or any other variables.

They used and described the Safe Commuting System (SCS) the current prototype uses safety incident and emergency alerts through a panic button integrated into a mobile app, this app sends information to a database, then this information is processed, analyzed, and classified into the system, that information will be later compared with similar reports to prove its veracity.

Then the information on the database is used in an algorithm like the ones mentioned before, it creates a risk score, then Dijkstra's algorithm returns the safest and most efficient route.

Taking into consideration the things said before, this is a very efficient solution, but it lacks reliability because of its dependence on people and the information they provide.

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

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

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

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] .

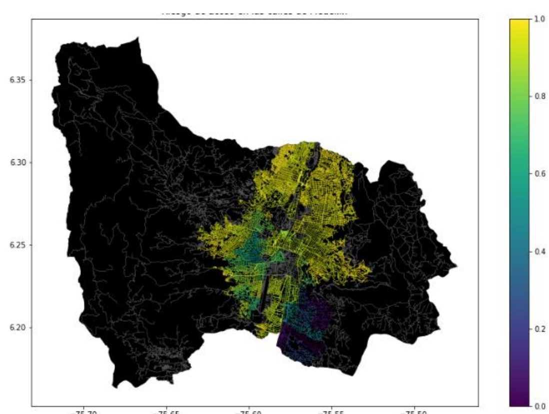


Figure 1. The risk of sexual harassment was calculated as a linear combination of the fraction of households that feel unsafe and the fraction of households with income below one minimum wage, obtained from the 2017 Medellín Quality of Life Survey.

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. *(In this semester, examples of such algorithms are DFS, BFS, Dijkstra, A*, Bellman, and Floyd, among others).*

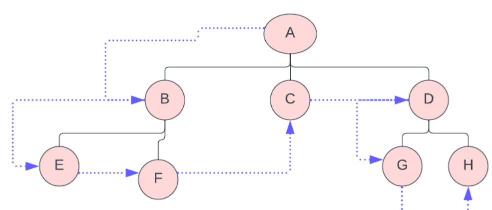
[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets

3.2.1 DFS (Depth First Search):

This kind of searching is based on expanding only one route from a starting node. This means that for each node it expands recursively on a specific path, and when it is possible to visit other nodes this way, it goes back and repeats the process for each of the brother nodes of the processed node.

This algorithm is used in situations when is required to find if a solution among others meets certain requirements.

The complexity is one of O (V + E) where V and E are the total numbers of vertex and edges on the graph

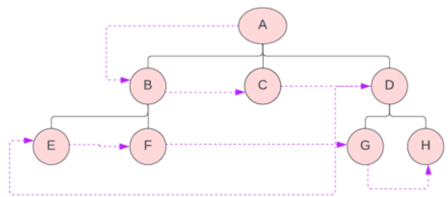


Using the DFS (Depth First Search) algorithm, the route that is carried out would be: A,B,E,F,C,D,G,H

3.2.2 BFS (Breadth-First search):

For this type of searching, it'll start from a starting node, it proceeds to explore all the neighbor nodes. Then, for each of the neighbors, it looks at their adjacent neighbors and so on until it has traveled the whole graph. This kind of searching algorithm is used to choose the best possible path in each part of the way.

The complexity is one of O (V + E) where V and E are the total numbers of vertex and edges on the graph

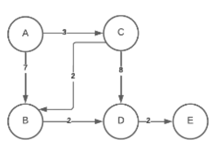


Using the BFS algorithm (Breadth-First search), the route that is carried out would be: A,B,C,D,E,F,G,H

3.2.3 Dijkstra's

In this type of search, the starting point is an initial vertex, and it is calculated which is the closest one, the vertex that is not yet fixed and that is at a smaller distance from the initial one is fixed. Then it is checked if using the last fixed vertex as an intermediate point, new vertices are reached, or a shorter path is found to those that we have already accessed.

The Dijkstra's complexity will change depending on if we use the priority queue or not, if we use the priority queue the complexity will be of O ((|A|+|V|) log |V|) = O (|A| log |V|), in the other case O(|V|2+|A|,) = O(|V|2).



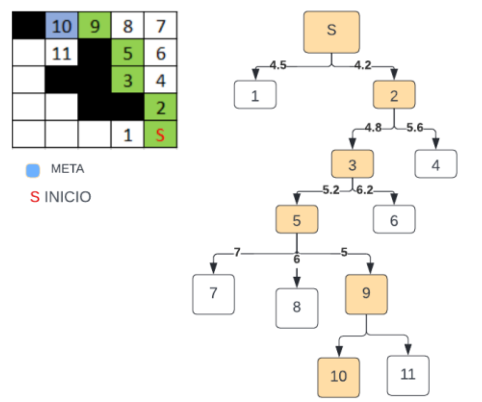
STEP	A	B	C	D	E
0	0	∞/-	∞/-	∞/-	∞/-
1	-	7/A	3/A	-	-
2	-	5/C	-	11/C	-
3	-	-	-	7/B	-
4	-	9/D	-	-	9/D

STEP 0: we are at A. The shortest path to A has length 0
STEP 1: We can get to B and C. The closest is C, so we fix it
STEP 2: We can get to D and also find a shorter path from A to B than the one we had. We set this path
STEP 3: We can get to D, so we fix this path
STEP 4: We set the shortest path to E

3.2.4 A*

This is a pathfinding algorithm that is used to find the shortest path between two nodes while avoiding obstacles. This algorithm is a heuristic search, for it to be optimal it is necessary that the remaining distance between the present node and the target node is not overestimated. Two auxiliary data structures are used for the operation of this algorithm, an open one, which serves as a priority queue, and a closed one, where the information of the nodes that have already been visited is stored. As the algorithm progresses, the node that is first in open is expanded, and in case this is not a target node, the function f(n) of all neighbors is calculated, they are inserted in open, and the evaluated node becomes closed.

The complexity comes from the heuristic quality used on the problem, in the worst-case scenario with a heuristic of very low quality the complexity would be exponential.



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] .

4.1 Data Structures

[4] https://github.com/gutim1011/ST0245-001

We implemented an adjacency list, which is the most efficient way to represent the map of Medellin. For the implementation, the adjacency list will contain the information of each vertex, allowing us to individually look at each one of them and allowing Dijkstra's algorithm to execute, also the adjacency list occupies way lesser memory complexity than other similar options.

The data structure is presented in Figure 2.

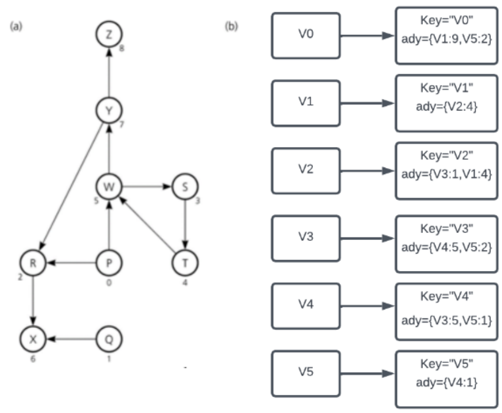


Figure 2: An example street map is presented in (a) and its representation as an adjacency list using dictionaries in (b).

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

The algorithm we choose is Dijkstra's algorithm which starts from a vertex and calculates the closest one, then it analyzes if the closest vertex creates a path that is smaller than the other one using a different vertex comparing and fixing along the process, it also checks if using the fixed vertex as an intermediate point it can find a shorter path for the ones already accessed.

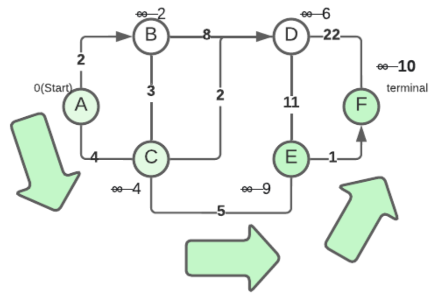


Figure 3: Calculation of a path that reduces both distance and risk of harassment

4.2.2 Calculation of two other paths to reduce both the distance and the risk of sexual street harassment

We still choose Dijkstra's algorithm, but for the generation of the two other paths we changed the value between the nodes, one using the multiplication of both values, generating a path that takes into consideration both the risk and the distance but gives more value to the risk, creating a longer but a safer path, and for the last path we only considered the distance, making it the shortest path but not the safest.

The algorithm is exemplified in Figure 4.

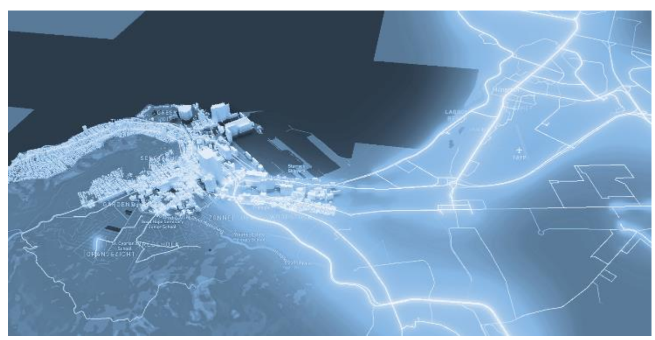


Figure 4: Map of the city of Medellín showing three pedestrian paths that reduce both the risk of sexual harassment and the distance in meters between the EAFIT University and the National University.

4.3 Algorithm complexity analysis

The worst-case scenario would be the algorithm visiting all the nodes and edges, giving us a complexity of O((|A|+|V|) log |V|) on each iteration it'll identify the minor vertex that is not marked as visited.

Algorithm	Time complexity
Dijkstra	O((|E|+|V|) log |V|)

Table 1: Time complexity of the name of your algorithm, where V is the vertex and E is the edges

Data Structure	Complexity of memory
Adjacency list	$O(V^2)$

Table 2: Time complexity of the name of your algorithm, where V is the vertex and E is the edges

4.4 Algorithm design criteria

We choose an adjacency list because when compared to adjacency matrixes it is way more efficient in referring to the memory space that is required, this is because an adjacency matrix in the worst-case scenario will occupy a space of O (V^2), on the other hand, an adjacency list will occupy on the worst-case scenario a space of O (d) where d is the grade of the vertex i. this is why we choose this data structure for our project.

The reason why we choose Dijkstra was that while doing the research for the third section, we found out that its complexity is lower than other similar options, which makes it a more viable and optimal option for what we are aiming for with the project.

For the optimization of the code, we decided to implement a priority queue making the algorithm more efficient.

5. RESULTS

In this section, we present some quantitative results on the three pathways that reduce both the distance and the risk of sexual street harassment.

5.1 Results of the paths that reduce both distance and risk of sexual street harassment

Next, we present the results obtained from *three paths that reduce both distance and harassment,* in Table 3.

Origin	Destination	Distance	Risk
Eafit	Unal	12.464 m	0.69
Eafit	Unal	9.407 m	0.70
Eafit	Unal	18.439 m	0.38

Distance in meters and risk of sexual street harassment (between 0 and 1) to walk from EAFIT University to the National University.

5.2 Algorithm execution times

In Table 4, we explain the ratio of the average execution times of the queries presented in Table 3.

Calculate the execution time for the queries presented in Table 3.

Calculation of v	Average run times (s)
v = Safest	0.1406 s
v = Fastest	0.1874 s
v = Both	0.1367 s

Table 4: *Dijkstra* execution times for each of the three calculator paths between EAFIT and Universidad Nacional.

6. CONCLUSIONS

With this code we developed a way to easily calculate paths using an algorithm, allowing users to find ways to travel both fast and safely to their destinations, our code is very useful because of the insecurity that we must face every day in this city and country, also this code can be implemented in other places by only changing the data and the location. The execution times are very applicable to real life, they are very short, allowing the user to calculate the route within seconds.

For an app or a web, we would recommend the path that uses both the risk and the distance, but for the value of freedom, we would give the user the option to ask for any of the three.

6.1 Future work

We would like to upgrade the code to make it more efficient, both in complexity and execution times, the data provided is from 2017 so using more updated data would make it more efficient, maybe use the code on an app or the web.

ACKNOWLEDGEMENTS

For Mariana Gutierrez:

I am very grateful for my scholarship, which has allowed me to study o this university. To my family and friends for being a big support when needed. And finally, to my partner for the great teamwork we achieved.

For Esteban Giraldo:

I'm grateful to the university for giving me the space to learn about what makes me happy. To my family for giving me the support that I needed, and the economical aid to achieve my goals. To my friends and girlfriend for being there on the rough times. And to my partner for the teamwork and mutual respect and support

The authors thank Professor Juan Carlos Duque, Universidad EAFIT, for providing the data from the 2017 Medellín Quality of Life Survey, processed in a *Shapefile*.

REFERENCES

[1]. Shivangi, S. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data. School of Computing and IT, Manipal University Jaipur, India, International Journal of Advanced Science and Technology, India, 2019, 33, 1415-1448.

[2]. Colina Torres, Rocío (2013) Optimización de rutas en ‘Camino Seguro al Cole’. Construcción y evaluación de rutas seguras. [Trabajo de curso]

[3]. Alves, R. Modelo matemático para seleção de rotas de patrulhamento escolar: o caso da patrulha escolar de Ponta Grossa. Ponta Grossa, Universidade Tecnológica Federal do Paraná, 2015, 70

[4]. C. V. Chavez et al., "Towards Improving Safety in Urban Mobility Using Crowdsourcing Incident Data Collection," 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 626-631, DOI: 10.1109/ISC246665.2019.9071714.

[5]. Búsqueda de Caminos,2011. Obtenido el 14 de agosto de 2022 de EcuRed: https://www.ecured.cu/B%C3%BAsqueda_de_caminos

[6]. Algoritmo A*,2005. Obtenido el 14 de agosto de 2022 de Instituto Tecnológico de Nuevo Laredo: http://www.itnuevolaredo.edu.mx/