PREVENTION OF STREET HARASSMENT USING AN ALGORITHM TO DETERMINE THE SAFEST PATH

José David Toro Franco
Universidad Eafit
Colombia
jdtorof@eafit.edu.co

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

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

ABSTRACT

Sexual harassment is any form of unwelcome sexual behavior, that could be offensive, humiliating or intimidating, this could happen everywhere, however in Medellin a lot of sexual harassment takes place in the streets, where, most of the times, no one can help the people who are affected by it. What is the algorithm you have proposed to solve the problem? What quantitative results have you obtained? What are the conclusions of this work? The abstract should be **at most 200 words**. (*In this semester, you should summarize here the execution times, and the results obtained with the three paths*).

Key words

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

1. INTRODUCTION

Sexual harassment is one of the main problems which affect Medellin, and although the town hall has done a lot of projects to decrease sexual harassment, it doesn't seem to cease.

However, some routes have less sexual harassment than others so the main purpose of this project is to identify the safe routes and, taking into account the distance of the routes, offer some options to the user to help them pick a safer route in Medellin. This could help reduce the sexual harassment while the users are driving, walking or even cycling.

1.2 Solution

The way we are trying to solve this problem is by creating an algorithm that determines the safest route between the start point and a finish point or destination, it will also determine the shortest path between the same points and a combination of a safe path and a short path, for this we are using the Dijkstra's algorithm because it looks at all the possible path options and after it determines the shortest path and it can also determine the safest path.

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 Safest route using smaller crime areas and risk score in New York City

New York City is a huge metropolitan city, and it has a crime rate 28% lower than the national average. However, it has a violent crime rate 41% higher than the national average. This leads to the creation of an algorithm which predicts a safe route taking into account crime, accidents data and distance, and it divides NYC into smaller regions of risk and calculates the risk score of the routes.

This algorithm works by calculating risk scores of all the routes lying between source and destination, the route with the lowest risk score is then suggested as the safest route.

The risk score of the paths are based on the average risk score of the nearby clusters formed using the datasets.[1]

2.2 Safest route for each age and gender

Analysis of criminal activity reveals that a significant number of crimes against civilians occur as people move through a city. This, combined with the increasing rates of crime reports around the world, has motivated some researchers to determine the safest route between the two given locations to improve people's safety when traveling. These researchers developed a mobile app that extracts the safest route personalized to the user's age and gender. The algorithm used by the researchers to achieve the desired result was the Iterative Dichotomiser 3 or ID3 algorithm. Initially, the application receives an input that includes a starting location and a destination. With these two locations, a route is calculated, and all possible streets along that route are analyzed based on the dataset related to the particular user. The safety route between two locations can be calculated by the safety of the streets that make up the route. Thus, the ID3 classifier, which creates a decision tree, checks the safety of all streets and gives the value “Yes” or “No”. Streets marked with "No" are analyzed in more detail and are the streets that determine the overall risk of the route.[2]

2.3 Crowdsourcing to search and report crime incidents

Crowdsourcing has its grounds on the concept that virtually anyone can add valuable information. Many mobile apps are already using crowdsourcing to report community issues, traffic problems, among others. CROWDSAFE is a mobile app which uses crowdsourcing to enable real time, location based crime incident searching and reporting.

This app allows its users to know about the crime information, and has features such as Safety Router and crime analytics. The app was tested in Washington DC and it showed collaboration between citizens and civic authorities..[3]

2.4 Preventing Sexual Harassment by Finding the Safest Route using Nearby Search, Directions API and Grid Coverage Techniques

This algorithm is able to predict locations that have a higher risk of sexual harassment using convolutional neural networks, this creates a heatmap which is very useful to determine the best route to recommend the user.

The algorithm breaks down the problem into three tasks: finding "safespots", find direction to the "safespots" and determine the risk of getting to each "safespot".

The "safespots" are selected using the heatmap which allows it to create them in the safest area according to the map. The algorithm then calculates the potential risk of each route to the destination and recommends the path with less potential risk score..[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] .

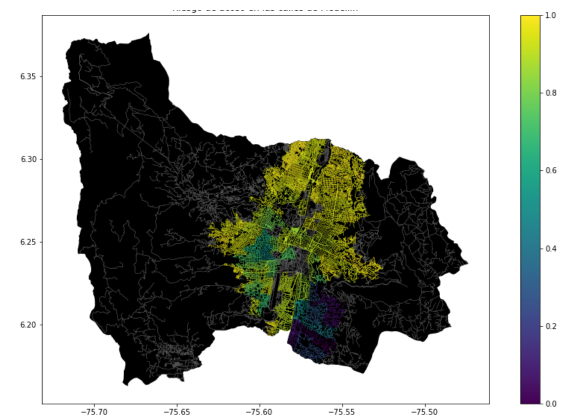


Figure 1. Risk of sexual harassment calculated as a linear combination of the fraction of households that feel unsafe and the fraction of households with income below one minimum wage, obtained from the 2017 Medellín Quality of Life Survey.

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* Algorithm

A* algorithm is a searching algorithm, which searches for the shortest path between the initial and the final state.

This algorithm has 3 parameters:

f: it is the sum of g and h, this means f = g + h.

g: is the sum of all the cells that have been visited since leaving the first cell.

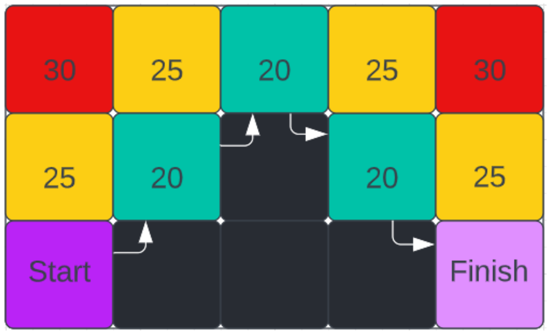
h: this is the estimated cost of moving from the current cell to the final cell, however the actual cost can't be calculated.

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

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

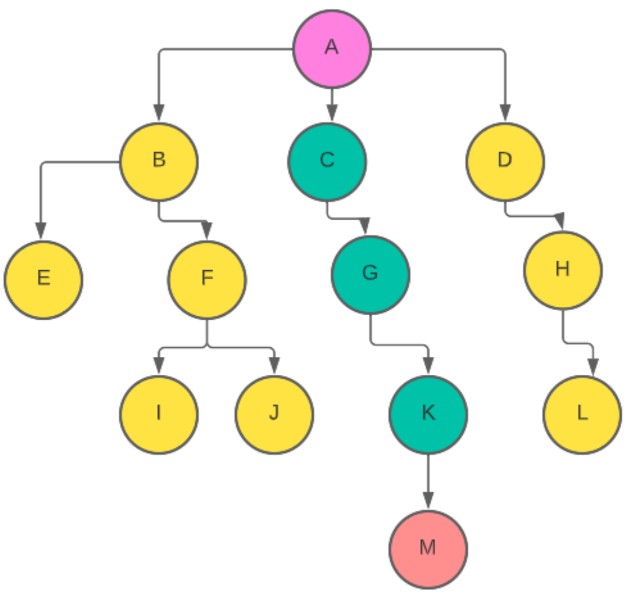
[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets

The way this algorithm calculates the shortest path is by selecting the smallest f-valued cell and then it moves to that cell and repeats itself until it has reached the destination or goal cell..[5]



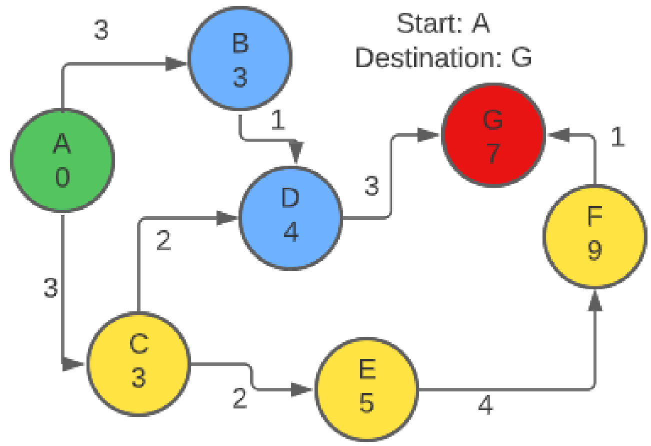
3.2.2 Depth First Search Algorithm

Depth-first search is a recursive algorithm, which uses the backtracking principle. This algorithm searches all nodes by moving forwards, if possible, and backtracking if it's necessary. The algorithm explores data structures, it starts at the root node, which can be any node and examines each path until it reaches the limit and then backtracks to look for other ways to search in order to find the best one..[6]



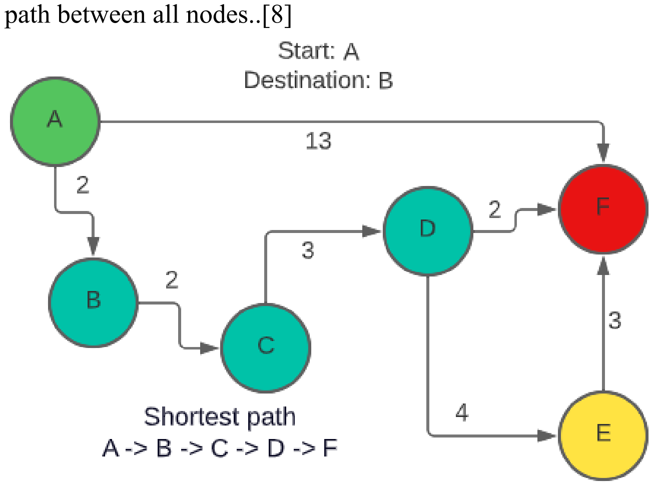
3.2.3 Dijkstra Algorithm

Dijkstra's algorithm allows us to find the shortest path between two nodes of a graph, the way this algorithm works is it analyzes the graph and finds the shortest path between the starting node and all other nodes in the graph, the algorithm explores every option and updates the distance if it finds a shorter path, once it has found the shortest path between the starting node and another node it marks that node as "visited" and adds it to the path, once all nodes have been added to the path it looks for the shortest path between the starting node and the destination node..[7]



3.2.4 Floyd-Warshall Algorithm

Floyd-Warshall algorithm is a shortest path algorithm for graphs, it computes the shortest distance between every pair of vertices in the input graph. This algorithm is an example of dynamic programming. It breaks the initial problem into smaller subproblems, then combines the answers to the subproblems to find the answer to the initial problem. The way this works is the shortest path from A to C is either the quickest path found from A to C so far, or the quickest path from A to B plus the quickest path from B to C. The Floyd-Warshall algorithm is more effective at managing multiple stops on the route because it calculates the shortest path between all nodes..[8]



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

To represent the map of the city of Medellin or any map we can use different data structures, in this case the map of Medellin was represented with an adjacency list using python dictionaries. The dictionaries work using a CSV file which contains the information of the streets, the information contains the distance or length of each street and their risk of sexual harrasment. The data structure is presented in Figure 2.

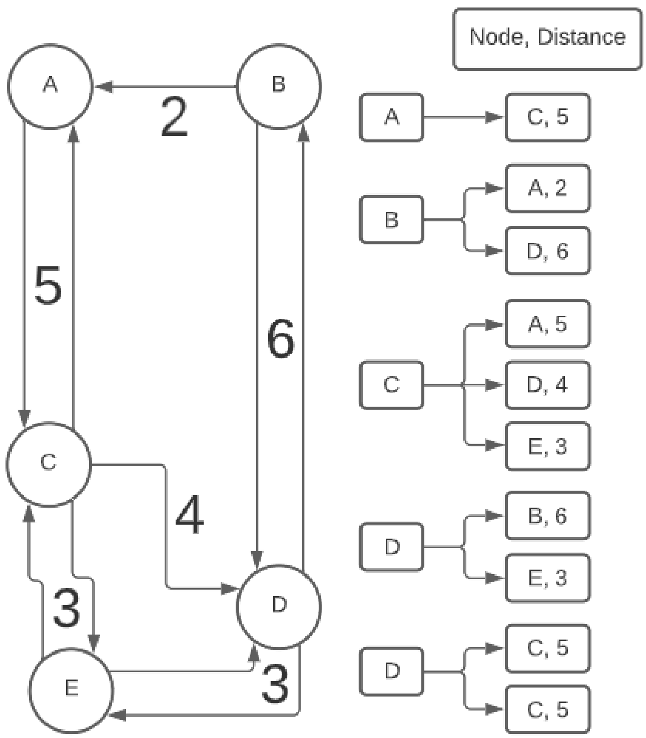


Figure 2: An example street map is presented on the left and its representation as an adjacency list on the right.

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 takes into account the amount of street harassment that has been determined to each street in Medellin and the distance between the nodes, if the risk of a street is greater than the stipulated risk it discards it and looks for another street which does not increase the distance a lot, but is also lower risk than the stipulated. The algorithm is exemplified in Figure 3.

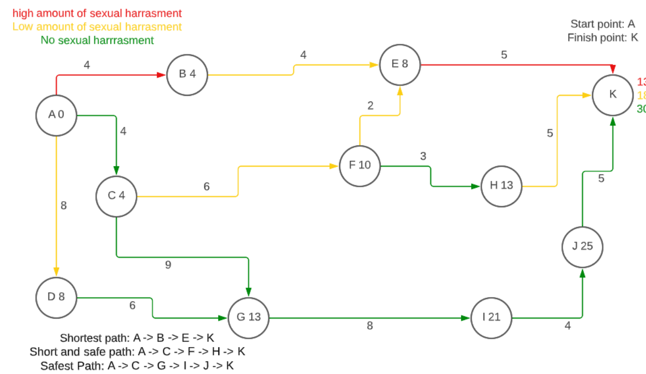


Figure 3: Calculation of a path that reduces both distance and risk of harassment.

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

Explain the other two paths that reduce both distance and risk of street sexual harassment and make your own graph. Do not use graphs from the Internet, make your own. *(In this semester, the algorithm could be DFS, BFS, Dijkstra, A*, among others).*) The algorithm is exemplified in Figure 4.

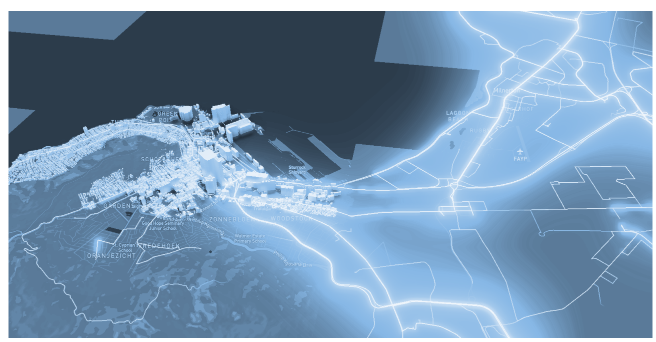


Figure 4: Map of the city of Medellín showing three pedestrian paths that reduce both the risk of sexual harassment and the distance in meters between the EAFIT University and the National University.

4.3 Algorithm complexity analysis

Explain, in your own words, the analysis, for the worst case, using the notation O. How did you calculate these complexities? Explain briefly.

---

[4]https://github.com/josedtf/PREVENTION-OF-STREET-HARASSMENT-USING-AN-ALGORITHM-TO-DETERMINE-THE-SAFEST-PATH

Algorithm	Time complexity
Algorithm name	$O(V^2 * E^2)$
Name of the second algorithm (in case you have tried two)	$O(E^3 * V * 2^V)$

Table 1: Time complexity of the name of your algorithm, where V is.... E is... *(Please explain what V and E mean in this problem). No, do not use 'n'.*

Data Structure	Complexity of memory
Name of the data structure	$O(V * E * 2^E)$
Name of the second data structure (in case you have tried two)	$O(2^{E*} 2^V)$

Table 2: Memory complexity of the data structure name used by your algorithm, where V is.... E is... *(Please explain what V and E mean in this problem). No, don't use 'n'. That is, don't use 'n'. Not 'n'.*

4.4 Algorithm design criteria

Explain why the algorithm was designed that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory. Examples of NON-objective criteria are: "I was sick", "it was the first data structure I found on the Internet", "I did it the last day before the deadline", "it's easier", etc. Remember: This is 40% of the project grade.

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 reduces 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	??	??
Eafit	Unal	???	??
Eafit	Unal	??	??

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 = ??	100000.2 s
v = ??	800000.1 s
v = ??	8450000 s

Table 4: *Algorithm* name execution times *(Please write the name of the algorithm, e.g. DFS, BFS, A*)* for each of the three calculator paths between EAFIT and Universidad Nacional.

6. CONCLUSIONS

Explain the results obtained. Are the paths significantly different? How useful is this for the city? Are the runtimes reasonable to use this implementation in a real situation? Which path would you recommend for a mobile or web application?

6.1 Future work

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its application? Will you continue this project working on optimization? Statistics? Web development? Machine learning? Virtual reality? How?

ACKNOWLEDGEMENTS

Identify the type of thank you you wish to write: to a person or to an institution. Keep the following guidelines in mind: 1. The professor's name is not mentioned because he or she is an author. 2. You should not mention the authors of articles that you have not contacted. 3. You should mention students, teachers of other courses who have helped you.

By way of example: This research has been supported/partially supported by [Name of Foundation, Donor].

We are grateful for help with [particular technique, methodology] to [First name Last name, position, name of institution] for comments that greatly improved this manuscript.

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.Soni, S. Shankar, V, G. Chaurasia, S. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data, 2019. Retrieved August 9, 2022. International Journal of Advanced Science and Technology: https://www.researchgate.net/profile/Venkatesh-Gauri-Shankar/publication/338096313_Route-The_Safe_A_Robust_Model_for_Safest_Route_Prediction_Using_Crime_and_Accidental_Data/links/5dfddeb392851c83648deef2/Route-The-Safe-A-Robust-Model-for-Safest-Route-Prediction-Using-Crime-and-Accidental-Data.pdf

2. Tarlekar, S. Bhat, A. Pandhe, S. Halarnkar, T. Algorithm to Determine the Safest Route, 2016. Retrieved August 9, 2022.) International Journal of Computer Science and Information Technologies: http://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit2016070 3106.pdf

3. Shah, S. Bao, F. Lu, C, T. Chen, I, R. CROWDSAFE: crowd sourcing of crime incidents and safe routing on mobile devices, 2011. Retrieved August 10, 2022. International Symposium on Advances in Geographic Information Systems: https://www.researchgate.net/publication/221589399_CROWDSAFE_crowd_sourcing_of_crime_incidents_and_safe_routing_on_mobile_devices

4. Ma, D. Preventing Sexual Harassment by Finding the Safest Route using Nearby Search, Directions API and Grid Coverage Techniques, 2019. Retrieved August 10, 2022. Omdena: https://medium.com/omdena/preventing-sexual-harassment-by-finding-the-safest-route-using-grid-coverage-techniques-77c920b3882

5. What is the A* algorithm?, Retrieved August 11, 2022.

Educative: https://www.educative.io/answers/what-is-the-a-star-algorithm

6. Your One-Stop Solution To Learn Depth-First Search(DFS) Algorithm From Scratch, 2022. Retrieved August 11, 2022. Simplilearn: https://www.simplilearn.com/tutorials/data-structure-tutorial/dfs-algorithm

7. Navone, E, C. Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction, 2020. Retrieved August 12, 2022. Freecodecamp: https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/

8. Chumbley, A. Moore, k. Yang, J. Ross, E. Khlm, J. Floyd-Warshall Algorithm. Retrieved August 12, 2022. Brilliant: https://brilliant.org/wiki/floyd-warshall-algorithm/