

SAFETY-BASED PATH FINDING ALGORITHM FOR REDUCTION OF SEXUAL HARRASSMENT.

Samuel Rico
Universidad Eafit
Colombia
Sricog@eafit.edu.co

Gregorio Bermúdez
Universidad Eafit
Colombia
gbermudezo@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 a daily concern for women in Medellín. 6 out of 10 women reported that they consider that the city is not safe for them [1]. Due to this, is important to implement alternatives, making the city a safer place to live. The goal of this project, is to create an algorithm which provides safer routes to the women, avoiding risk and uncomfortable situations for them.

Keywords

Constrained shortest path, street sexual harassment, secure-path identification, crime prevention.

1. INTRODUCTION

According to SISC [2] in the last month (February 2022) the violence towards women has increased in Antioquia's capital. Femicide is the biggest indicator that violence is rising. There have been 5 cases in the last 19 days and that is a number which worries the authorities. Nowadays, the police is investigating, hoping to find someone they could use as guilty. But is this really the solution? Women home harassment was increasing thanks to the pandemic [3], but since the confinement is over this has move to the city. What should be worrying the government in the first place is finding a suitable way for avoiding street sexual harassment amongst citizens.

Safeness has always been an issue in Antioquia, since the 80s, people levels of concern towards crimes has been rising [4]. Even though the crime rates have decreased lately in Medellín [5], people still feel worried about getting robbed. As an attempt to solve this problematic, this project intends to create a safety-based pathfinding algorithm that protects citizens from high crime zones in Medellín.

1.1. Problem

The problem we are trying to solve is to calculate the shortest path without exceeding a weighted-average risk of harassment and, to find the path with lowest level of danger without exceeding a distance. It is important to return a short path without exceeding a given sexual harassment risk, because it reduces the probability of having uncomfortable and dangerous situations for women, but it is also important to return a path without exceeding a distance, because it may occur that the safest path is not convenient in terms of distance.

1.2 Article structure

In what follows, in Section 2, we present related work to the problem. Later, in Section 3, we present the data sets and methods used in this research. In Section 4, we present the algorithm design. After, in Section 5, we present the results. Finally, in Section 6, we discuss the results and we propose some future work directions.

2. RELATED WORK

In what follows, we explain four related works to path finding to prevent street sexual harassment and crime in general.

2.1 SafeStreet: empowering women against street harassment using a privacy-aware location based application

SafeStreet is a mobile application created to avoid sexual harassment. It was developed by the Department of Computer Science and Engineering of Dhaka University. The application has a Safe Route Search functionality, which allows the user to give the location, destination, and preferred time of travel. SafeStreet, returns a safe path, and an unsafe path, if exists. It measures the safeness of a route depending on the records provided by other users [6]. The Safe Route Search algorithm wasn't specified in the article.

2.2 CROWDSAFE: Crowd Sourcing of Crime Incidents and Safe Routing on Mobile Devices

CROWDSAFE, is a mobile application created by the Department of Computer Science of Virginia Polytechnic Institute and State University. It allows users to search and report new crime information. It has a Safety Router, which provides the safest route between two locations, based on the reports given by other users. This implements the A* and Dijkstra shortest path algorithms, depending on the performance [7].

2.3 SafeRoute: Learning to Navigate Streets Safely in an Urban Environment

This study is an attempt to embrace emerging technologies such as machine learning, in long-existing problematics such as pathfinding. It was developed by researchers of University of California, Santa Barbara. This permitted the creation of a safety-based pathfinding platform in which information given by other users are processed by a deep RL algorithm,

returning the better option regarding safety and travel time [8].

2.4 Safety-based path finding in urban areas for older drivers and bicyclists:

As its title states, this study works with an algorithm aiming for better conditions for elderly drivers and bicyclist. This paper written by researchers in Texas Transportation Institute focus on how to create a methodology that merges safety and pathfinding. Considering both, driver and traffic attributes, they came up with a multi-objective process that studies amongst the safe paths which one is the best one [9].

3. MATERIALS AND METHODS

In this section, we explain how data was collected and processed and, after, different constrained shortest-path algorithm alternatives to tackle street sexual-harassment.

3.1 Data Collection and Processing

The map of Medellín was obtained from Open Street Maps (OSM)¹ and downloaded using Python OSMnx API². The (i) length of each segment, in meters; (2) indication wheter the segment is one way or not, and (3) well-known binary representation of geometries were obtained from metadata provided by OSM.

For this project, we calculated the linear combination that captures the maximum variance between (i) the fraction of households that feel insecure and (ii) the fraction of households with income below one minimum wage. These data were obtained from the quality of life survey, Medellín, 2017. The linear combination was normalized, using the maximum and minimum, to obtain values between 0 to 1. The linear combination was obtained using principal components analysis. The risk of harassment is defined as one minus the normalized linear combination. Figure 1 presents the risk of harassment calculated. Map is available at Github³.

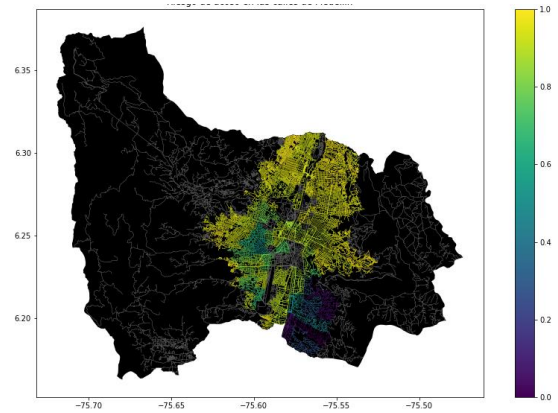


Figure 1. Risk of sexual harassment calculated as a lineal combination of the fraction of households that feel insecure and the fraction of households with income below one minimum wage, obtained from Life Quality Survey of Medellín, in 2017.

3.2 Constrained Shortest-Path Alternatives

In what follows, we present different algorithms used for constrained shortest path.

3.2.1 Extended Dijkstra Algorithm

Dijkstra algorithm is used to find the shortest path in a graph. It starts at the starting point node and calculates the movement cost to each connected node. It marks, or save, the node where the movement cost is the minimum. It repeats this process until it reaches the target node, or the destination. This algorithm requires that every node must be visited. Thus, the search time can be long depending on the number of nodes. Therefore, it can be a good idea to use the Extended Dijkstra Algorithm, in which the algorithm is applied from both directions, this is, from the starting point to the target and vice versa. With this implementation, the algorithm must continue until the search from the starting point and the search from the destination overlap [10].

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

³<https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/>

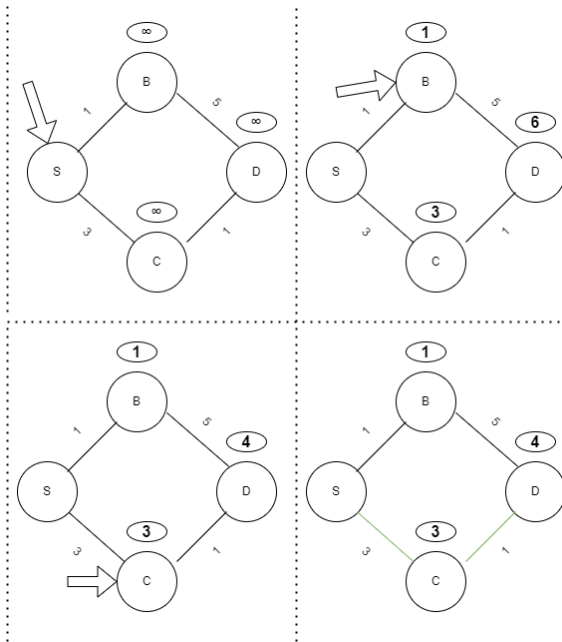


Figure 2. Dijkstra Algorithm.

3.2.2 A Star Algorithm

Also called A* Algorithm. It uses the combination of heuristic and searching based on the shortest path. Each adjacent cell of the current cell is evaluated by the heuristic distance to the goal cell, plus the length of the path from the initial state to the goal state through the selected cells. As in the previous algorithm, it chooses the “next cell” based on the minimum value obtained. This algorithm has some modifications, such as Basic Theta* and Phi* algorithms, which may be better when there is free space between connected cells, because they consider searching in every angle. [11].

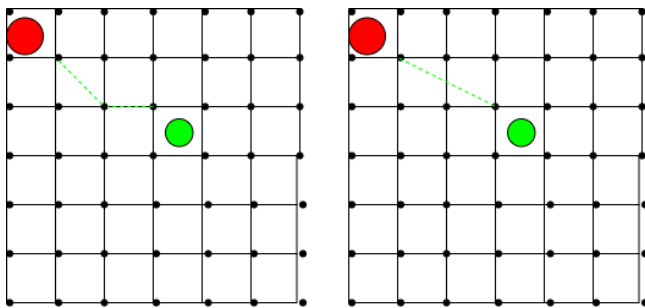


Figure 3. A* Algorithm (left) and a modified A* Algorithm (right).

3.2.3 Breadth first search Algorithm:

BFS algorithm is an algorithm used to examine nodes and edges from a graph. It uses first-in-first-out method in order to find the best path. It works by assigning every node a specific place in the process, every new one goes at the back of the queue. By doing this it examines every path until it arrives to the final node and then performs backtracking for confirm the path. This algorithm ensure that the first path found is the shortest one [12].

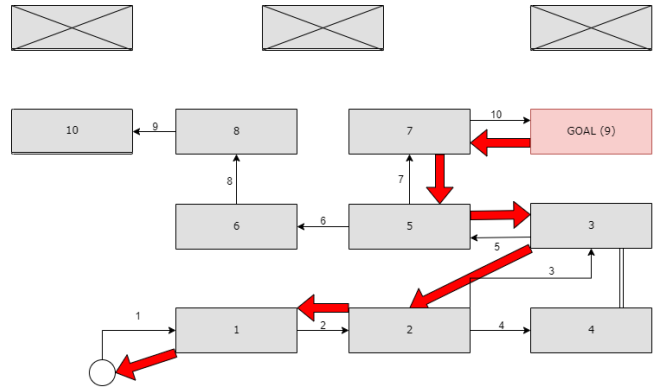


Figure 4. BFS Algorithm.

3.2.4 Depth first search Algorithm:

DFS algorithm is a very similar algorithm to the BFS algorithm, both find a route from a root node to a final node. The main difference between these two is the processed used by one and other. As already mentioned, BFS algorithms use FIFO processing of the data, unlike DFS algorithms use LIFO processing for finding an answer. Due to this DFS algorithm find a path but it isn't ensured to be the best path regarding distance [13].

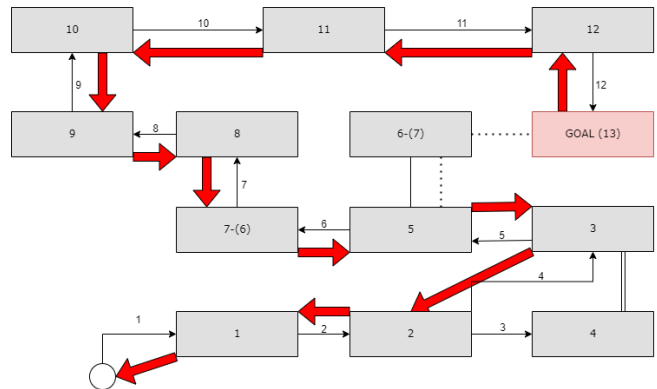


Figure 5. DFS Algorithm.

As seen in the image the path found by the algorithm is not the best. This is thanks to the stacking technique. When it arrives to the node five, it must assign an order to the next two options of pathing. Since the longest path was placed at

the end of the stack (7) it continues to analyze this track. The suitable path is indicated by the dotted line (6-lost node).

4. Algorithm Design and Implementation

In what follows, we explain the data structures and the algorithms used in this work. The implementations of the data structures and algorithms are available at Github⁴.

4.1 Data Structures

The data was given in a csv file. Thus, we use a Pandas DataFrame to clean in. After this, we implemented an adjacency list with dictionaries to represent the graph. The keys are the coordinates of each node of the graph, and the values are the coordinates of the adjacent nodes, with their distance and harassment risk.

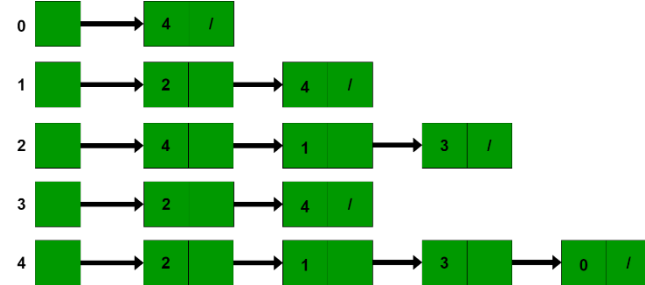


Figure 6. Graph as an Adjacency List.

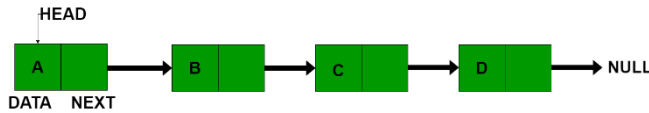


Figure 7. Linked List.

4.2 Algorithms

In this work, we propose algorithms for the constrained shortest-path problem. The first algorithm calculates the shortest path without exceeding a weighted-average risk of harassment r . The second algorithm calculates the path with the lowest weighted-average risk of harassment without exceeding a distance d .

4.2.1 Dijkstra Algorithm

As our problem was to find the shortest path on a weighted graph (distance and harassment risk), we chose Dijkstra Algorithm as the solution.

We used a dictionary, representing the distance from the starting point to every other node, and a queue with all the ‘unvisited’ nodes, meaning that the minimum distance to those nodes was not found yet. At first, that queue had the same number of nodes of the graph, and in the dictionary, the distance to all the nodes was assign to a large number, except from the starting node, that was assign to 0.

Then, we pop the node v in the queue that had the minimum distance value in the dictionary, and we updated the distance of the adjacent nodes of v , if the distance found plus the distance given in the graph was less than the distance in the dictionary. This process was repeated while the queue was not

empty, or if v was equal to the target, meaning that the shortest distance of the target was found.

The same algorithm was used to calculate the path with the minimum harassment risk. *Figure 2*.

ACKNOWLEDGEMENTS

The authors are grateful to Prof. Juan Carlos Duque, from Universidad EAFIT, for providing data from Medellín Life Quality Survey, from 2017, processed into a Shapefile.

REFERENCES

1. Ruta N. Reducir El Acoso Callejero: El Reto De La Secretaría De Mujeres. <https://www.rutanmedellin.org/es/programas-vigentes/2-uncategorised/592-reto-de-mujeres>
2. Semana. (2022, February 21). Aumenta la Violencia contra mujeres en Medellín: Van cinco asesinatos en 15 días. Semana.com Últimas Noticias de Colombia y el Mundo. Retrieved February 23, 2022, from <https://www.semana.com/nacion/articulo/aumenta-la-violencia-contra-mujeres-en-medellin-van-cinco-asesinatos-en-15-dias/202231/>
3. La Pandemia en la sombra: Violencia contra Las Mujeres Durante El Confinamiento. ONU Mujeres. (n.d.). Retrieved February 23, 2022, from <https://www.unwomen.org/es/news/in-focus/in-focus-gender-equality-in-covid-19-response/violence-against-women-during-covid-19>
4. Violencia y crimen en medellín: Pasado, presente y futuro. Medellín Cómo Vamos. (n.d.). Retrieved February 23, 2022, from <https://www.medellincomovamos.org/boletin/violencia-y-crimen-en-medellin-pasado-presente-y-futuro>
5. Investigación Desarrollada en Medellín Revela El impacto de la pandemia sobre la criminalidad en la ciudad. Alcaldía de Medellín. (n.d.). Retrieved February 23, 2022, from <https://www.medellin.gov.co/irj/porta/medellin?NavigationTarget=contenido%2F10823-Investigacion-desarrollada-en-Medellin-revela-el-impacto-de-la-pandemia-sobre-la-criminalidad-en-la-ciudad>
6. Mohammed Eunus Ali, Shabnam Basera Rishta, Lazima Ansari, Tanzima Hashem, and Ahamad Imtiaz Khan. 2015. SafeStreet: empowering women against street harassment using a privacy-aware location based application. In Proceedings of the Seventh International Conference on Information and Communication Technologies and Development (ICTD '15). Association for Computing Machinery, New York, NY, USA,

- Article 24, 1–4.
DOI:<https://doi.org/10.1145/2737856.2737870>
7. Sumit Shah, Fenyao Bao, Chang-Tien Lu, and Ing-Ray Chen. 2011. CROWDSAFE: crowd sourcing of crime incidents and safe routing on mobile devices. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11). Association for Computing Machinery, New York, NY, USA, 521–524. DOI:<https://doi.org/10.1145/2093973.2094064>
 8. Sharon Levy, Wenhan Xiong, Elizabeth Belding, and William Yang Wang. 2020. SafeRoute: Learning to Navigate Streets Safely in an Urban Environment. *ACM Trans. Intell. Syst. Technol.* 11, 6, Article 66 (December 2020), 17 pages. DOI:<https://doi.org/10.1145/3402818>
 9. Chandra, S. (2014, September 18). Safety-based path finding in urban areas for older drivers and bicyclists. *Transportation Research Part C: Emerging Technologies*. Retrieved February 23, 2022, from <https://www.sciencedirect.com/science/article/pii/S0968090X14002344>
 10. M. Noto and H. Sato, "A method for the shortest path search by extended Dijkstra algorithm," *Smc 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0, 2000, pp. 2316-2320 vol.3, doi: 10.1109/ICSMC.2000.886462.*
 11. Duchoň, F., Babinec, A., Kajan, M., et al. 2014. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*. <https://www.sciencedirect.com/science/article/pii/S187770581403149X>.
 12. Khan Academy. (n.d.). The breadth-first search algorithm (BFS) (article). Khan Academy. Retrieved February 23, 2022, from <https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/the-breadth-first-search-algorithm>
 13. Elgabry, O. (2017, October 12). Path finding algorithms. Medium. Retrieved February 23, 2022, from <https://medium.com/omarelgabrys-blog/path-finding-algorithms-f65a8902eb40>
 14. Brilliant.org. 2022. *Dijkstra's Shortest Path Algorithm* | Brilliant Math & Science Wiki. [online] Available at: <https://brilliant.org/wiki/dijkstras-short-path-finder/> [Accessed 17 April 2022].
 15. Anon. 2022. Dijkstra's algorithm for adjacency list representation: Greedy Algo-8. (April 2022). Retrieved April 17, 2022 from <https://www.geeksforgeeks.org/dijkstras-algorithm-for-adjacency-list-representation-greedy-algo-8/>