ROUTE OPTIMIZATION FOR PREVENTION OF STREET HARRASMENT IN MEDELLIN

Thomas Martinod
Universidad Eafit
Colombia
tmartinods@eafit.edu.co

Emanuel Serna
Universidad Eafit
Colombia
esernar1@eafit.edu.co

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

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

ABSTRACT

In this publication, street sexual harassment is identified as a degrading social problem that affects women around the world, who do not have access to real-time media to prevent it. After analyzing the rates of harassment and the localities that provide greater vulnerability in the case of the city of Medellin, different data structures are analyzed to know which one of them allows optimizing an algorithm that defines an arbitrary route over the city, considering the variables, distance d and harassment index R.

To solve this situation in an efficient and comprehensive way, we implemented the Dijkstra's Algorithm, an algorithm that calculates the shortest path between two nodes in a graph, in this case, coordinates of the city of Medellín. We presented a small modification to the algorithm that takes a source and a destination, and tries to find the best possible path, considering both distance and harassment risk in the streets of this city. Therefore, we created a new variable that considers both factors, and we defined it in three different ways. Each of the cases gives us a different path where the total distances and the average expected harassment risk variate.

Finally, we calculate the execution times and the complexity of our code, thus, giving us an efficient program that determines the most optimal path between two places in less than a second. In this way, we concluded that Dijkstra's Algorithm and data structures are a good method of programming when we want to solve these types of problems in real life situations.

Key words

Shortest route, street sexual harassment, data structures

1. INTRODUCTION

Gender-based Street harassment occurs when a person is subject to unconsent sexual comments, gestures, and actions by any person in a public space. These types of actions are identified because they are directed at the victim due to their gender or sex.

Actions which classify as street harassment include catcalling, sexist slurs and names, public sexual demands, masturbation, public flashing, and rape [1]. Street harassment is an issue all over the world, with a disproportionately greater effect on women. Although it's a problem described as transnational, developing countries, which often have weak infrastructure, poor lighting conditions in public places and unsafe transport systems, often have worse cases of street sexual harassment.

For instance, a study conducted by YouGov found that Bogota, Mexico City and Lima were among the top five worst cities in terms of verbal harassment [1]. Evidently, street harassment is an issue affecting safety perception in cities, on top of contributing to destructive feelings in victims.

In Medellín specifically, 34.4% of underaged girls affirm that they are victims of street harassment, even several times a day, while 60% of women feel Medellin is not a safe city due to its persistent patriarchal culture [2]. Thus, the constant threat of harassment when walking around a city, including Medellin, leads to fear, and can even stop people from walking or using the city's transportation network.

1.1. The problem

As seen by the statistics exhibited above, street sexual harassment is a deep problematic affecting safety in most cities, including Medellin. However, woman all around the world find themselves defenseless, they do not have access to data that can help them avoid areas of the city where sexual harassment incidents are likely to occur.

Even worse, citizens and tourists usually rely on navigation platforms, such as Waze or Google Maps to obtain routes for traveling around the city. These types of applications calculate routes based solely on distance, which not only fails to inform users of unsafe areas but can also potentially lead people to take risker routes just because they are shorter. Thus, there's a great need to develop an algorithm that calculates routes within the city based on both distance and safety.

In the case of sexual harassment, this means developing a shortest path algorithm that provides routes that are short and have a low risk of sexual harassment. Through the implementation of this algorithm, security perception in the city would improve greatly, more people, especially women, would be confident about walking and taking public transportation, and cases of street harassment hopefully may decrease significantly.

1.2 Solution

To solve this problem, which affects women the most, we decided to design an algorithm that considers both the distance from one place to another but also reduces the street harassment that a woman can potentially receive along the way.

By researching, analyzing, and filtering various algorithms that could be useful to us for this project, we initially concluded that two of them greatly facilitated the solution. Firstly, the Bellman Ford algorithm and, on the other hand, the Dijkstra's algorithm, two quite similar algorithms that, in a nutshell, look for the shortest path from one place to another. Their main difference is that the Bellman Ford algorithm can work with negative values, therefore, it is more complex in terms of time. After studying and testing them we realized that, due to complexity issues and the fact that our program is designed to take only positive values, the best option we could take was to use the Dijkstra's algorithm, modifying it in such a way that it not only finds the shortest path from one point to another, but also shows the list of points that are traveled during that 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 SafeStreet: Empowering women against street harassment using a privacy-aware location-based application

Scientists from the Bangladesh University of Engineering Technology and Dhaka University proposed a crowd-powered privacy-aware location based mobile application called "SafeStreet", in order to reduce and prevent the sexual harassment against women in public places [3].

This app allows women to privately document and share their own experiences, and, on the other hand, find the best path to a destination that has the least harassment hazard.

SafeStreet works mainly with two important aspects, time, and place. Some research and studies made them conclude that sexual harassments are spatial temporal in nature, which means they occur more frequently at a certain place or at a particular time.

According to this fact, SafeStreet app has four major features: visualizing harassments, capturing harassments, reporting harassments, and searching safe routes. The first three of them allow users to capture personal experiences either privately or by sharing them with the rest of users and see other harassment reports of surrounding regions in a map.

Finally, for safe routed searching, the app uses a route search algorithm that finds out the shortest path from the starting location of the user to her destination, considering the harassment records (location and time) to be the weight. With all that information, their server process runs the route search algorithm and returns necessary paths to the mobile user.

2.2 Secure Routing with Crowdsourcing

A group of investigators in Mexico City were concerned with security issues that arise in big cities and urban centers, especially in developing countries where crime is more widespread. Popular mobile systems compute paths based entirely on distance, and don't consider safety hazards that may arise along a recommended route.

Thus, the researchers, developed an approach that integrates crowd-reported crime data with official government data to obtain safer routes in Mexico City for both locals and tourists. The academic paper states that a preliminary experiment was implemented with 75% of effectiveness [4].

The approach proposed by the researchers consists of collecting Tweets that are related to crime and integrating them with crime data from official governmental institutions through an automatic system that considers descriptions and attributes in the data. After this, the Bayes algorithm is used to classify data that couldn't be integrated automatically. Most importantly, this algorithm is used to assign probabilistic crime rates to specific parts of the city.

After collecting, sorting, and classifying the data, a mobile application was developed to display the safest route between two given locations. The safe route is obtained though the Dijkstra algorithm, in which the weight of the nodes is derived from the average number of crimes in the given location. Thus, the algorithm outputs a route that avoids places with higher crime rates.

2.3 A Navigation System for Safe Routing

This novel system was created by scientists from the Indraprastha Institute of Information Technology Delhi, Indian Institute of Technology Ropar and Indira Gandhi Delhi Technical University for Women and was presented in 2021 during the 22nd IEEE International Conference on Mobile Data Management (MDM) [5].

This system consists of an algorithm that recommends routes to move from one place to another by balancing the requirements of increasing the safety and constraining the total length of the path. Their target and focus were the city of Delhi in India. They compute street level safety by collecting information from news, social media and government websites and then applying processing techniques using tools from machine learning, natural language processing, and geocoding to transform the data into a usable form of safe routing.

To make this system as effective as possible, they use multiple data sources including crime data from news websites, crime statistics data from the NCRB website, public transport data, traffic data, image data, road quality data and police stations coordinates.

After processing all the data extracted previously and generating safety scores, a Multiple Segment Replacement algorithm does its job, firstly, finding a seed path from s (the origin) to d (the destination), then, estimating score gain for all segments of the seed path, then, selecting a set of segments for replacement, and finally, replacing the chosen segments in the seed path while checking the budget constraint. With all of these, the user receives the best possible route considering both safety and quickness.

2.4 SafePaths

Finally, Galbrun, Pelechrinis and Terzi [6] were the first to define safe routes with the use of criminal data. Specifically, concerned with growing insecurity in cities, and aiming to take advantage of publicly available

datasets, the investigators sought to improve the quality of life of those traveling around cities by developing an algorithm that uses criminal data to yield navigation options based on both distance and

safety. The preliminary experiment was done using data from Chicago and Philadelphia, which they used to create a city risk model that included the probability of crime on any given road segment.

Overall, their central objective was to find a short and low-risk route between two given locations. However, since these two variables cannot be computed together as a single problem, the investigators developed a solution based on a bi-objective shortest path problem that outputs a set of paths that have varying degrees of safety and distance [6].

Despite their success, this SafePaths problem is based on crime in general, and fails to address specific types of crime, such as street harassment, along a certain path. The investigators working on the SafePaths problem used a deterministic algorithm to determine the best route, which is done by computing all possible routes between two given locations.

Overall, given a fixed pair of origin and destination locations, the algorithm computes two important paths: the safest path and the shortest path. When these two initial paths differ, as it is often the case, a recursive algorithm is then used to consider, in each iteration, an intermediate non-dominated path between the original intervals. This process is repeated until the difference between the shortest and the safest path is minimal [6].

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] .

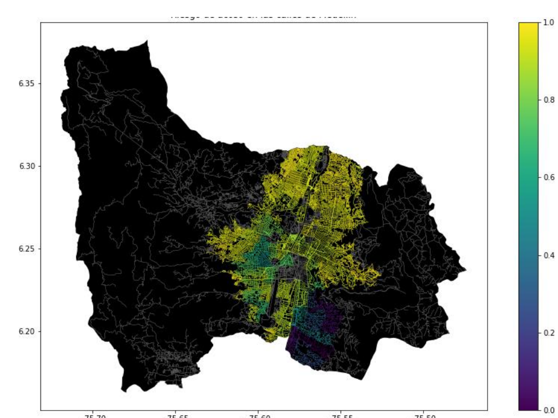

Figure 1. Risk of sexual harassment calculated as a linear combination of the fraction of households that feel unsafe and the fraction of households with income below one minimum wage, obtained from the 2017 Medellín Quality of Life Survey.

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

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

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

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 Dijkstra's Algorithm

Given a graph and a source vertex in the graph Dijkstra's shortest path algorithm is used to find the shortest paths from the source to all vertices in the given graph, mainly regarding distance. As a result, when using Dijkstra's to create risk-free paths for a short distance, the output will most likely be the one with the fewest hops and longer paths will be overlooked by this algorithm [7].

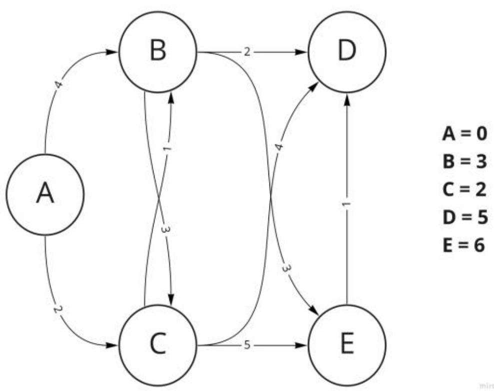

Figure 2. *Graph of Dijkstra's algorithm*

Dijkstra's algorithm starts by labeling each node with the known distance from the source. Before any calculations are made, the source node is assigned a value of 0, and all the other nodes are assigned a value of ∞. The values of the nodes distinct from the source are updated as the algorithm is executed, and the value will always represent the shortest distance found until that point. As the first step, the algorithm "travels" through all the edges connected to the source node and updates the value of the distance taken to get to each of the "secondary" nodes. After this, the algorithm chooses the secondary node with the shortest distance and calculates the new distances through the edges that depart from the choose secondary node. If a new distance is shorter than one found in the previous step, the distance assigned to that node is updated. In addition, the algorithm keeps track of the path followed to obtain the assigned shortest distance. Overall, Dijkstra's algorithm follows 3 main steps: updates node distances, keeps track of paths followed, and chooses the next vertex. Once all vertices on the graph had been visited, the value assigned to each node, with its corresponding route, is the shortest distance and path from the source node [7].

3.2.2. The Bellman-Ford algorithm

The Bellman–Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all the other vertices in a weighted digraph. It is slower than Dijkstra's algorithm for the same problem, but more versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers.

Like Dijkstra's algorithm, Bellman–Ford proceeds by relaxation, in which approximations to the correct distance are replaced by better ones until they eventually reach the solution. In both algorithms, the approximate distance to each vertex is always an overestimate of the true distance and is replaced by the minimum of its old value and the length of a newly found path. However, Dijkstra's algorithm uses a priority queue to greedily select the closest vertex that has not yet been processed and performs this relaxation process on all its outgoing edges; by contrast, the Bellman–Ford algorithm simply relaxes all the edges and does this $|V| - 1$ times, where $|V|$ is the number of vertices in the graph. In each of these repetitions, the number of vertices with correctly calculated distances grows, from which it follows that eventually all vertices will have their correct distances [8].

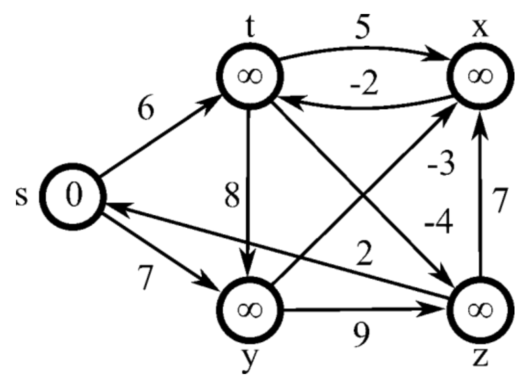

Figure 3. *Visual graph of Bellman algorithm*

3.2.3. Bresenham's line algorithm

Bresenham's line algorithm is a line drawing algorithm that determines the points of an *n*-dimensional raster that should be selected to form a close approximation to a straight line between two points. The general concept of this algorithm is: given a starting endpoint of a line segment, the next grid point it traverses to get to the other endpoint is determined by evaluating where the line segment crosses relative to the midpoint (above or below) of the two possible grid points choices. Basically, this algorithm considers some variables defined previously by the programmer and, regarding these parameters, decides between two different points of destination [9].

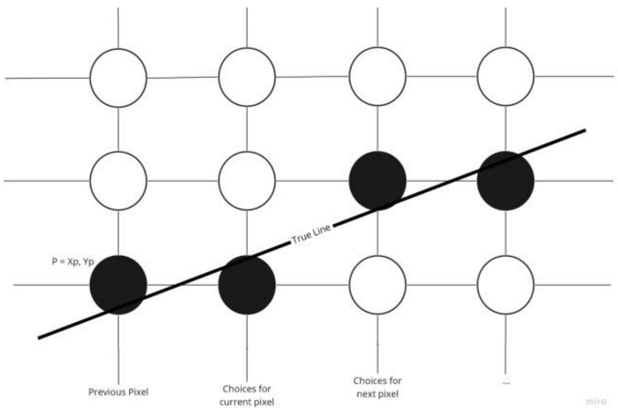

Figure 4. *Trajectory of Bresenham's line algorithm*

3.2.4. Breadth-first search algorithm (BFS)

The breadth-first search algorithm is used for searching tree data structures until a node with a certain condition is found. The algorithm starts at the root node and explores each level of the tree completely before moving on to the next level. This differs from the depth-first search algorithm, which explores a tree branch until its last level before backtracking and exploring other branches.

Among many other things, breadth-first search is used for finding the shortest path from a source vertex to all the other nodes on the graph. Unlike other algorithms, like Dijkstra's and Floyd-Warshall, breadth-first search is used on graphs that are unweighted. This means that edges between nodes are not assigned a specific length. Instead, the distance from a certain node to the source node is determined by how many levels beneath the source node it is found, with each level being assigned a value of one [10].

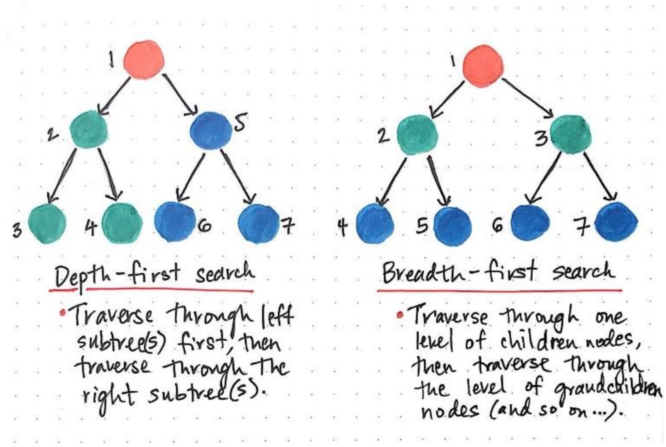

Figure 5. *Differences between DFS and BFS algorithms.*

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

For a proper and complete implementation of our proposed solution, we decided to choose as data structure, to represent the map of the city of Medellin, an adjacent list using dictionaries. First, creating a dictionary where the nodes of the graph are included, which are basically all the data in the "origin" column (in coordinates) previously filtered to discard those that are repeated. After this, we create another dictionary in each of the elements of the large dictionary. These dictionaries are of the form {vertex1: {Adj_vertex1: weight, Adj_vertex2: weight}}, where Adj_vertex are the nodes that have connection with the main node and weight is a tuple that stores the distance (in meters) and the harassment risk index.

We decided to use an adjacent list over an adjacent matrix since each of the nodes is connected to very few others, and it was a waste of time and memory if a huge matrix was created where the nodes only had connection to a few of the thousands that exist. The data structure is presented in Figure 6.

[4] https://github.com/ThomasMartinod/SafeMED

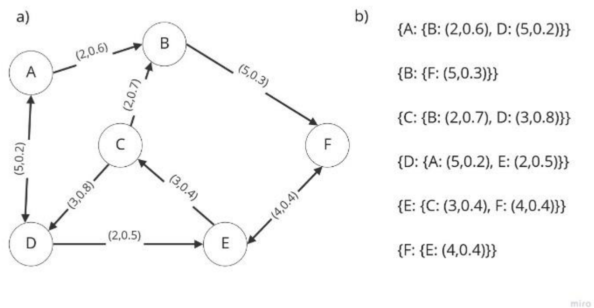

Figure 6: An example street map is presented in (a) and its representation as an adjacency list using dictionaries in (b).

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 Dijkstra's algorithm for a pedestrian path that reduces both distance and risk of sexual street harassment

We decided to choose the Dijkstra's algorithm, applying a small modification to its original and most used form.

The foundation on which this algorithm is based is the principle of optimization: if the shortest path between vertices "u" and "v" passes through vertex "w", then the part of the path that goes from "w" to "v" must be the shortest path among all the paths that go from "w" to "v". In this way, minimum cost paths are successively constructed from an initial vertex to each of the vertices of the network, and the paths achieved are used as part of the new paths.

With our modification, the algorithm will not search for the shortest path to each of the other nodes but will focus on dictating the optimal path from an origin to a target.

We also created a new formula, which we call PathValue and is the one on which the Dijkstra's algorithm is based to take the best path from one node to another. This formula considers both the distance between two points and the risk of street harassment between them. To optimize this value, the formula was expressed as $PathValue = \left(\frac{1}{2}\right) * length + (100) * harassmentRisk$. In this way, both values are being considered when taking the most optimal path.

This algorithm will not only help us to find the optimal route, but it will also give us the solution in less time and consuming less computational memory. Compared to other possible algorithms, Dijkstra's algorithm handles less complexity and when applied to an adjacent list, it becomes much more effective and efficient. The algorithm is exemplified in Figure 7.

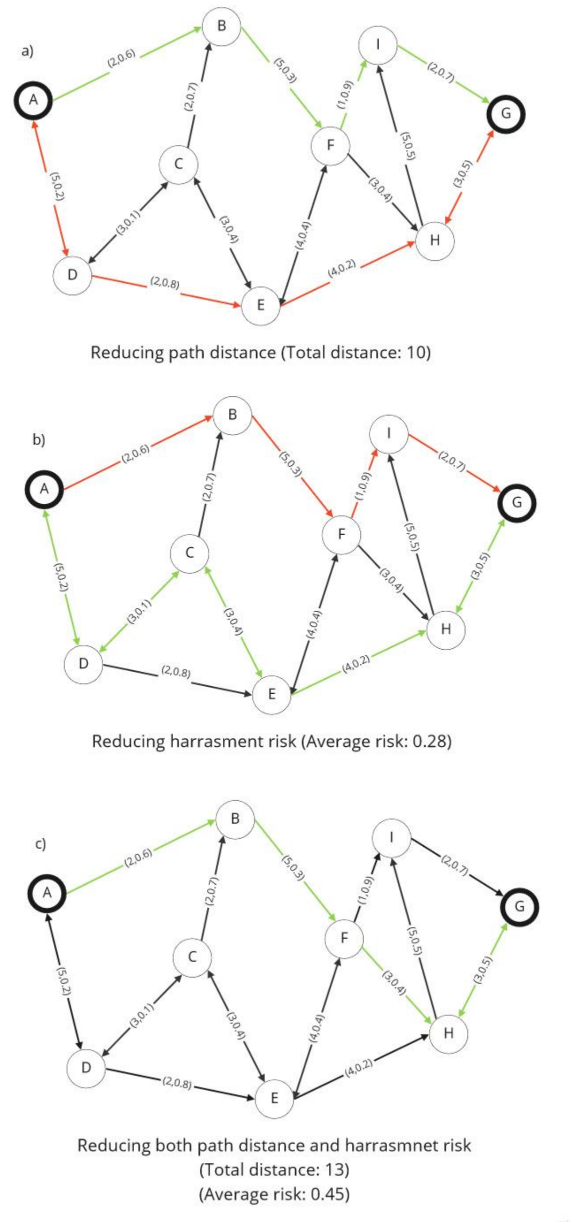

Figure 7: Calculation of a path that reduces both distance and risk of harassment.

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

For the other two paths, we decided to modify the variable that we had named PathValue, to optimize the distance and reduce the harassment in two different ways. Since our first path slightly prioritizes the risk of harassment over distance, our two new variables seek to do something different. One of them seeks to reduce as much distance as possible, but at the same time, the risk of harassment affects it very minimally. On the other hand, we seek to take both into account to the same extent, and we achieve this by taking the distance and raising it to the risk, as these are values between 0 and 1, the more risk there is, the higher the value of the variable and, in turn, if there are two streets with the same risk but with different distance, the algorithm will choose the street with less distance. The algorithm is exemplified in Figure 8.

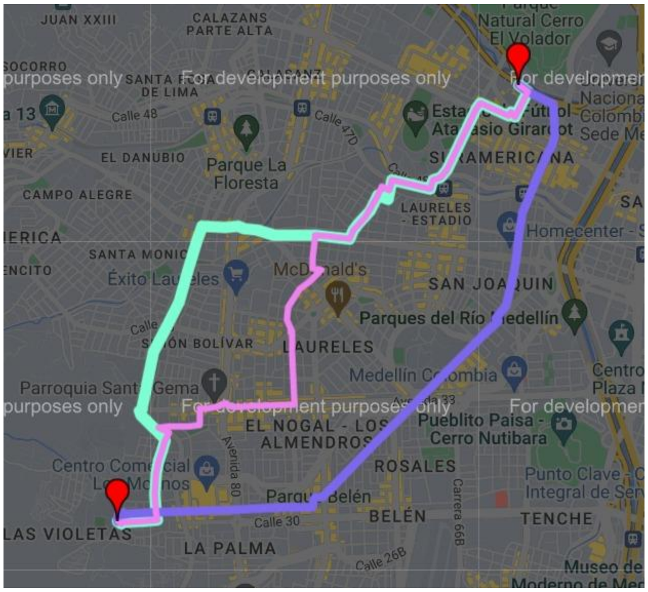

Figure 8: Map of the city of Medellín showing three pedestrian paths that reduce both the risk of sexual harassment and the distance in meters between the University of Luis Amigó and the Medellin University.

4.3 Algorithm complexity analysis

After considering and studying the program several times, and looking for the worst case for both Dijkstra and the adjacency list we used, we came up with the following:

Algorithm	Time complexity
Dijkstra´s Algorithm	$O(E + V \ \log V)$

Table 1: Time complexity of the Dijkstra's algorithm, where V is the number of vertices in the graph and E is the number of edges that make connections between vertices.

Data Structure	Complexity of memory
Adjacency list using dictionaries	$O(V)$

Table 2: Memory complexity of the adjacency list using dictionaries, where V is the number of elements in the dictionary that represent all vertices in the graph.

4.4 Algorithm design criteria

As mentioned before, we worked with Dijkstra's Algorithm and with an adjacency list using dictionaries as our data structure.

Dijkstra because it works in stages and takes at each stage the best solution without considering future consequences. But also, the optimum found at one stage can be modified later if a better solution emerges, what makes the algorithm more efficient and optimal.

The advantage of the adjacency list implementation is that it allows us to compactly represent a sparse graph. The adjacency list also allows us to easily find all links that are directly connected to a particular vertex. By using dictionaries, it is easier and faster to access to the elements that each of the edges contains. In this case, path value.

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	8969.396 m	0.5302
Eafit	Unal	8574.132 m	0.6868
Eafit	Unal	9061.75 m	0.5785

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.

d = distance

r = harassment risk

Calculation of v	Average run times (s)
v = $\left(\frac{1}{2}\right) d + 100r$	0.0488 s
v = $100d + 100r$	0.0317 s
v = d^r	0.0557 s

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

6. CONCLUSIONS

After reviewing and analyzing the three paths that the program generates, we can conclude various things. First, the execution time of the second case is significantly less than in the other two cases. This happens because the variable V is mainly considering only the distance and the harassment risk has less effect on it, so in this order, the program is working with less data and can be executed faster. In addition, among the different proves we did for the three paths, we realize that in some cases the first path gave us better results than the third one, but in other cases, happened the opposite (the third path was better than the first in both distance and average expected harassment). We conclude that this may happen because of some aspects, for example, the real distances between the two places, or the zone of the city that the travel was working on. We also noticed that the last path is the slower one, mainly because it is working with and exponential variable. Nevertheless, the three paths worked over the main problem we were trying to solve and if we could choose or recommend the best option for reducing as much distance as possible and at the same time, avoiding the harassment in a big way, we would recommend using the first algorithm that defines v = $\left(\frac{1}{2}\right)d + 100r$. This algorithm was the best option, in at least, 60% of the cases we try.

6.1 Future work

In the not-too-distant future, with the help of professional tools and experts, it would be a great idea to further develop this program, thus turning it into a far-reaching project that will help minimize this great social problem in which mainly women are affected. In this way, turning a simple code into an easily accessible application for all citizens. However, before this, it is more than necessary to adapt new knowledge and continue our career with constant learning.

ACKNOWLEDGEMENTS

We are deeply grateful for the help of different persons such as Juliana Velez, who helped us with the creation and designation of the different variables used in the algorithm.

Furthermore, we would like to thank Isabel Mora, student of Math Engineering in EAFIT University, for the constant attention and the way she provided different resources that where useful during the development of this project.

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] Stop Street Harassment, 2022. What Is Street Harassment? | Stop Street Harassment. [online] Available at: https://stopstreetharassment.org/about/what-is streetharassment/

[2] Ruta N Medellín, 2022. Reducir el acoso callejero: el reto de la Secretaría de Mujeres - Ruta N. [en línea] Disponible en: https://www.rutanmedellin.org/es/programasvigentes/2-sin categorizar/592-reto-de-mujeres

[3] 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. New York, NY, USA, Article 24, 1–4. https://doi.org/10.1145/2737856.2737870

[4] Mata, F., Torres-Ruiz, M., Guzmán, G., Quintero, R., Zagal-Flores, R., Moreno-Ibarra, M. and Loza, E., 2016. A Mobile Information System Based on Crowd-Sensed and Official Crime Data for Finding Safe Routes: A Case Study of Mexico City. [online] Mobile Information Systems Available at: https://www.hindawi.com/journals/misy/2016/8068209/

[5] R. Kaur et al., "A Navigation System for Safe Routing," 2021 22nd IEEE International Conference on Mobile Data Management (MDM), 2021, pp. 240-243, doi: 10.1109/MDM52706.2021.00047.

[6] Galbrun, E., Pelechrinis, K. and Terzi, E., 2015. Urban navigation beyond shortest route: The case of safe paths. [online] Science Direct. Available at: https://www.sciencedirect.com/science/article/pii/S0306437915001854.

[7] 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 (September 2020), 17 pages. https://doi.org/10.1145/3402818

[8] Bellman–Ford algorithm, Wikipedia, 20-May-2022. [Online]. Available: https://en.wikipedia.org/wiki/Bellman%E2%80%9 Ford_algorithm. [Accessed: 16-Aug-2022].

[9] Ma, D., 2019. Preventing Sexual Harassment by Finding the Safest Route Using Grid Coverage Techniques. [online] Medium. Available at: <https://medium.com/omdena/preventing-sexual-harassment-by-finding-the-safest-route-using-grid-coverage-techniques-77c920b3882> [Accessed 16 August 2022].

[10] Wikipedia, 2022. Breadth-first search. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Breadth-first_search