THE SAFEST PATH

Nicolás Moreno López
Universidad Eafit
Colombia
nmorenol1@eafit.edu.co

Luciana Pineda Yepes
Universidad Eafit
Colombia
lpineday@eafit.edu.co

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

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

ABSTRACT
The principal problem is street sexual harassment. This problem is important because there are a lot of cases of this issue and especially women are seriously affected, even to the point of changing their way of dressing. Other issues related to harassment can be dangerous routes, sexual harassment in several places, and insecurity in the streets, which can result in robberies and deaths.

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

1. INTRODUCTION
Women's safety is the principal motivation because many women feel insecure and have fear of walking alone or in transit for some places where they can be harassed or even outraged, something that happens every day against thousands and thousands of women. Colombian prosecutor's office has expressed that from just 4 complaints they got in 2008, then we're more than 1.656 complaints. They also said that the number of harassment cases was 11.098 in January of 2018.

1.1. Problem
What we're trying to solve is the street sexual harassment finding the best path without exceeding a weighted-average risk of harassment r, and with the lowest weighted-average risk of harassment without exceeding a distance d. Solving this problem can bring trust and security to the most affected people, that in this case are mostly women. Finding this solution can reduce the possibility of suffering from harassment on your way home.

1.2 Solution
We chose Dijkstra's algorithm for the solution because the map shows clearly the paths with less street sexual harassment or less distance, according to the destination. It can be appreciated which places are dangerous and which are not. Also, we believe it was easier to implement when programming.

1.3 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 propose some future work directions.

2. RELATED WORK
In what follows, we explain four related works to pathfinding to prevent street sexual harassment and crime in general.

3.1 Safety-aware routing for motorized tourists based on open data and VGI
They solved the problem about the dangerous paths that tourists can take at night in Los Angeles, not knowing that they're taking the risk of being robbed, something that locals already know. So, they decided to create an algorithm that shows the best route with less risk, and if they go far away, even the shortest path. They got the results using volunteered geographic information, governmental open data for detecting properties and functionalities of the urban infrastructure, and historical crime data from police departments for detecting crime hot spots.

3.2 Route – The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data
They create a method to find the safest route having the lowest risk score for travelers. They got the results using updated crime and accident data from NYC Open Data and using Machine Learning algorithms that generate the risk score average of a path. This algorithm can help travelers and new people in the city to recognize paths with lower risk in the regions.

3.3 Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians
What we can see in this related problem is Dijkstra's algorithm. It is used to find the shortest path for pedestrians, but even to enjoy the walk, so this algorithm provides a safe and attractive path. Then the result is the best route quality that includes safety, wellbeing, effort, exploration, and pleasure for the pedestrian.

3.4 Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search
The algorithm that they used to find a path that prevents Street sexual harassment was a Nearby search. They also used a heatmap to identify where the harassment cases were in Mumbai, India. The map showed the hospital's location nearby too, in case you need it.

3. MATERIALS AND METHODS

This section explains 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 API[2]. The (i) length of each segment, in meters; (2) indication of whether 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. The map is available at GitHub[3].

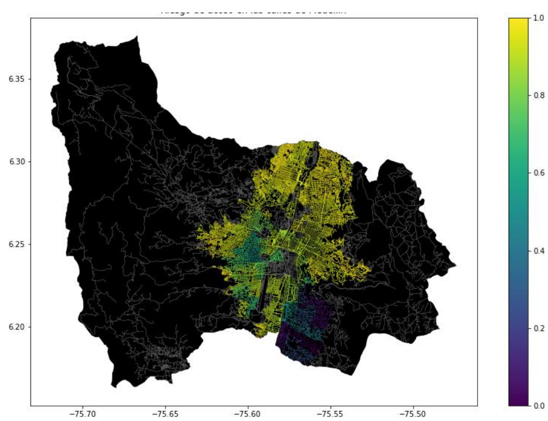


Figure 1. The risk of sexual harassment is calculated as a linear combination of the fraction of households that feel insecure and the fraction of households with income below one minimum wage, obtained from the Life Quality Survey of Medellín, in 2017.

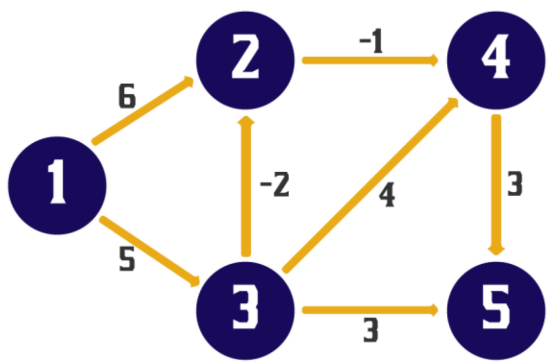
3.2 Constrained Shortest-Path Alternatives

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

3.2.1 Bellman-Ford Algorithm

[6] It is similar to Dijkstra's algorithm but it can work with graphs in which edges can have negative weights.

The Bellman-Ford algorithm works to find the shortest path between a given node and all other nodes in the graph. Though it is slower than the other algorithms. It's important to note that if there is a negative cycle – in which the edges sum to a negative value – in the graph, then there is no shortest or cheapest path. Meaning the algorithm is prevented from being able to find the correct route since it terminates on a negative cycle. Bellman-Ford can detect negative cycles and report on their existence.



3.2.2 Floyd-Warshall Algorithm

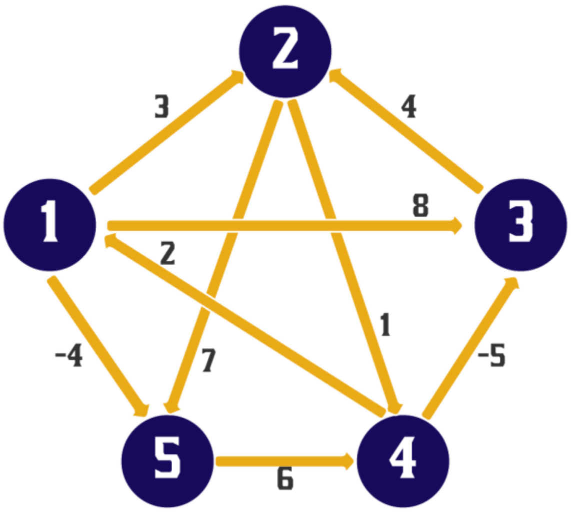
The Floyd-Warshall algorithm calculates the shortest distance between every pair of nodes in the graph, rather than only calculating from a single node. It works by breaking the main problem into smaller ones, then combining the answers to solve the main shortest path issue. [2] This algorithm works for both the directed and undirected weighted graphs. But it does not work for the graphs with negative cycles.

Floyd-Warshall is extremely useful when it comes to generating routes for multi-stop trips as it calculates the shortest path between all the relevant nodes. For this reason, many route planning software will utilize this algorithm as it will provide you with the most optimized route from any given location. Therefore, no matter where you currently are, Floyd-Warshall will determine the fastest way to get to any other node on the graph.

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

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

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

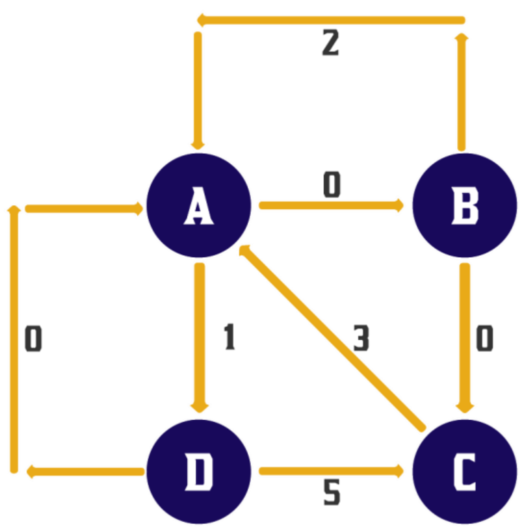


3.2.3 Johnson's Algorithm

Johnson's algorithm works best with sparse graphs, the fewer edges, the faster it will generate a route, as its runtime depends on the number of edges.

This algorithm relies on two other algorithms to determine the shortest path. First, it uses Bellman-Ford to detect negative cycles and eliminate any negative edges. Then use Dijkstra's algorithm to calculate the shortest paths in the original graph that was inputted.

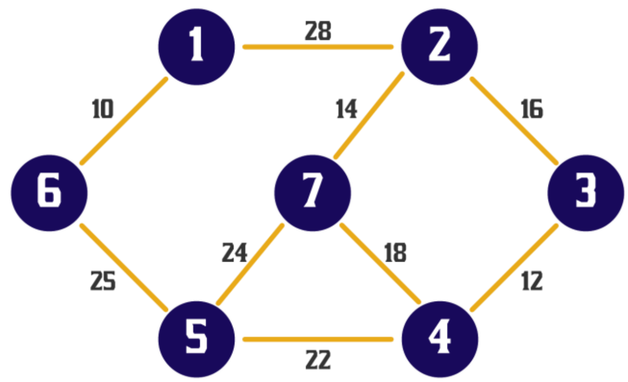
[8] The idea of Johnson's algorithm is to re-weight all edges and make them all positive, then apply Dijkstra's algorithm for every vertex.



3.2.4 Prim's Algorithm

Prim's algorithm of finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. But can't find MST Forest like Krushkal's Algorithm does.

[9] Kruskal's algorithm is another popular minimum spanning tree algorithm that uses a different logic to find the MST of a graph.



The complexity of all algorithms.

Algorithms	Complexity
Bellman Ford Algorithm	Best Case Complexity O(E) Average Case Complexity O(VE) Worst Case Complexity O(VE)
Floyd Warshall Algorithm	Best Case Complexity O(V^3) Average Case Complexity O(V^3) Worst Case Complexity O(V^3)
Johnson's Algorithm	Case Complexity O(V2log V + VE)
Bellman Ford Algorithm	Case Complexity O(E log V)

4. ALGORITHM DESIGN AND IMPLEMENTATION

4.1 Data Structures

First, we use pandas Python library to read the data that we have about street sexual harassment. Next, it got converted into a graph by the Networkx Python library, to later use Dijkstra's algorithm to get a visual map with the data algorithm.

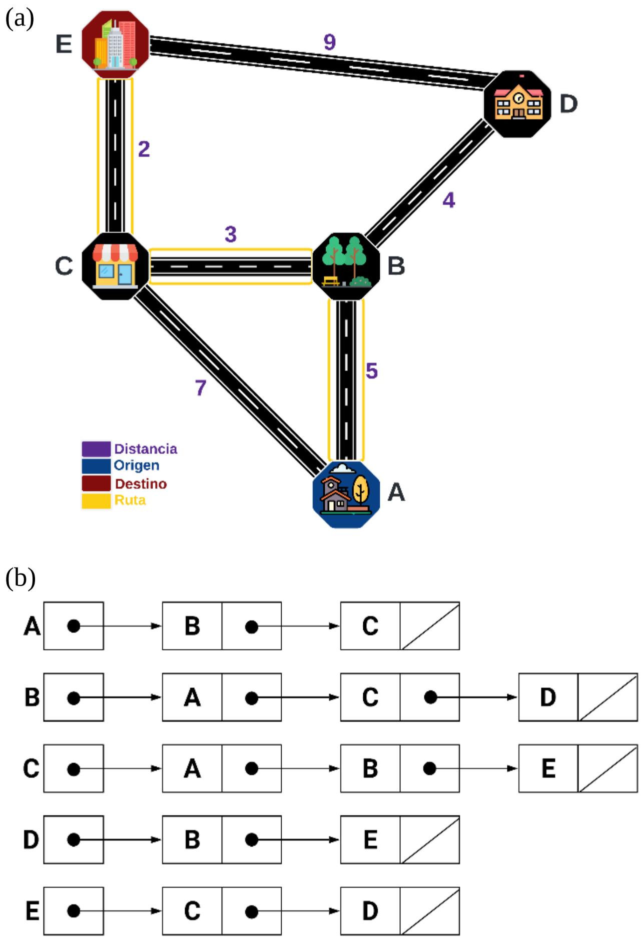


Figure 2: An example of a street map is presented in (a) and its representation as an adjacency list in (b).

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 First algorithm

Dijkstra's algorithm finds the shortest path between nodes in a graph, looking to the adjacent nodes for the one with less weight, in this case, they are distance and risk. This continues working until found the node that satisfies the requirement, getting so the best path for the planning route.

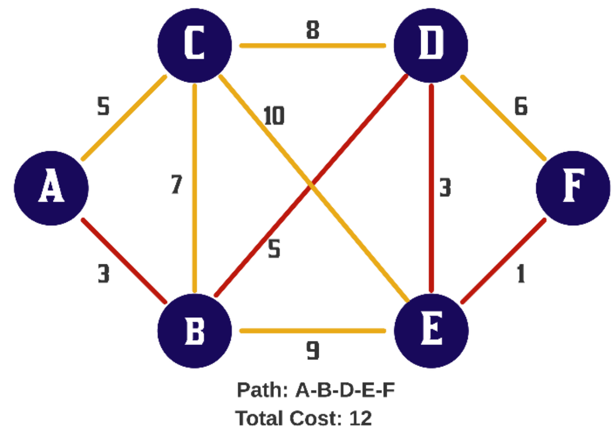


Figure 3: Solving the constrained shortest-path problem with the Dijkstra algorithm.

4.2.2 Second algorithm

Dijkstra's algorithm finds the shortest path between nodes in a graph, looking to the adjacent nodes for the one with the lowest weighted-average risk of harassment without exceeding a distance d. This continues working until found the node that satisfies the requirement, getting so the best path for the planning route.

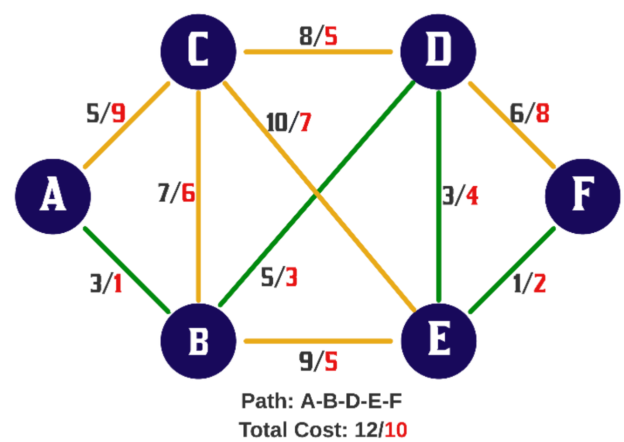


Figure 4: Solving the lowest weighted-average risk of harassment without exceeding a distance d problem with the Dijkstra algorithm.

4.4 Complexity analysis of the algorithms

We calculated the O notation by investigating Dijkstra's algorithm and many hours of working on this project gave us the experience to know how our code is being executed.

Algorithm	Time Complexity
Dijkstra Algorithm	(E+VlogV)

Table 1: Memory complexity of the Dijkstra Algorithm. Where E is the number of edges in the graph and V is the number of vertices in the graph.

Data Structure	Memory Complexity
Dijkstra Algorithm	$O(V^2)$

Table 2: Memory complexity of the Dijkstra Algorithm. Where E is the number of edges in the graph and V is the number of vertices in the graph.

4.5 Design criteria of the algorithm

We decided to design our path with Dijkstra algorithm because other algorithms can take up large amounts of space, but this one takes reasonable space and it's not the faster but it has an acceptable time of execution. Unlike other algorithms that can have a long time of execution. Additionally, it clearly shows the results on the map.

5. RESULTS

In this section, we present some quantitative results on the shortest path and the path with the lowest risk.

5.1.1 Shortest-Path Results

In what follows, we present the results obtained for the shortest path without exceeding a weighted-average risk of harassment *r* in Table 3.

Origin	Destination	Shortest Distance	Without Exceeding *r*
Universidad EAFIT	Universidad de Medellín	6088 m	0.84
Universidad de Antioquia	Universidad Nacional	2000 m	0.83
Universidad Nacional	Universidad Luis Amigó	800 m	0.85

Table 3. Shortest distances without exceeding a weighted-average risk of harassment *r*.

5.1.2 Lowest Harassment-Risk Results

In what follows, we present the results obtained for the path with lowest weighted-average harassment risk without exceeding a distance *d* in Table 4.

Origin	Destination	Lowest Harassment	Without Exceeding *d*
Universidad EAFIT	Universidad de Medellín	0.63	5,000
Universidad de Antioquia	Universidad Nacional	0.32	7,000
Universidad Nacional	Universidad Luis Amigó	0.24	6,500

Table 3. Lowest weighted-average harassment risk without exceeding a distance *d* (in meters).

5.2 Algorithm Execution-Time

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

Routes	Average execution times (s)
Universidad EAFIT to Universidad de Medellín	9.43 s
Universidad de Antioquia to Universidad Nacional	9.4 s
Universidad Nacional to Universidad Luis Amigó	8.36 s

Table 4: Execution times of the Dijkstra algorithm.

6. CONCLUSIONS

The results are great, the route changes with harassment and distance in long distances, but it's not a huge difference in meters. Otherwise in short distances, the path is the same taking harassment or distance. These results are significant for future projects about security and street sexual harassment, it can be useful to create an app that shows the securest path for all the people that feel insecure walking in the city. The algorithm takes a reasonable time, but it depends of the device that you're using and the space that it has.

6.1 Future work

This topic is so important nowadays, since street sexual harassment cases have increased exponentially in the past years, therefore insecurity must be combated and women should arrive to their homes safe and sound. Also, we would like to improve the time that the algorithm takes to show the best path. In conclusion, if we continue with this project, we would like to implement statistics to know what people think about the efficacy of the algorithm and if it works for them, but furthermore, we would like to use web development so the program does not work only in mobile apps, but also on internet, so you can use it on your phone and on your computer.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to all finance alliances of EAFIT, for letting me be part of this incredible team of students. Further, I would like to thank all the students for the thoughtful comments and recommendations on this project. I am also thankful to the School of Engineering and all its member's staff for all the considerate guidance. To conclude, we 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.Porras M, J. El Acoso Callejero a las Mujeres en Colombia: ¿Más Penas o Más Educación?. Revistalevel.com.co, 2019. https://www.revistalevel.com.co/contenido/el-acoso-callejero-a-las-mujeres-en-colombia-mas-penas-o-mas-educacion.

2.Andreas Keler & Jean Damascene Mazimpaka (2016) Safety-aware routing for motorised tourists based on open data and VGI, Journal of Location Based Services, 10:1, 64-77, DOI: 10.1080/17489725.2016.1170216

3. Soni, Shivangi & Gauri Shankar, Venkatesh & Sandeep, Chaurasia. (2019). Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data. 28. 1415-1428.

4. P. Siriaraya et al., "Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians," in IEEE Access, vol. 8, pp. 135569-135590, 2020, doi: 10.1109/ACCESS.2020.3011924.

5.Ma, D. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search. Omdena | Building AI Solutions for Real-World Problems, 2020. https://omdena.com/blog/path-finding-algorithm/.

6.Bellman Ford's Algorithm. Programiz.com, 2020. https://www.programiz.com/dsa/bellman-ford-algorithm.

7.Floyd-Warshall Algorithm. Programiz.com, 2020. https://www.programiz.com/dsa/floyd-warshall-algorithm.

8.Johnson's algorithm for All-pairs shortest paths - GeeksforGeeks. GeeksforGeeks, 2017. https://www.geeksforgeeks.org/johnsons-algorithm/.

9.Prim's Algorithm. Programiz.com, 2020. https://www.programiz.com/dsa/prim-algorithm.