

STATISTICAL STUDY WITH PLANNING ROUTES TO AVOID SEXUAL HARASSMENT IN MEDELLÍN.

Laura Danniela Zarate
Guerrero
Universidad Eafit
Colombia
ldzarateg@eafit.edu.co

Santiago Arias Higuaita
Universidad Eafit
Colombia
sariash@eafit.edu.co

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

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

ABSTRACT

The following problem can be explained by showing the statistics and frequent cases in the social situation that resides on the constant and repeated sexual harassment given on streets, houses, schools, public transport, and places in general. The problem's relevance and importance can be seen on the fact that the kind of moments related to sexual harassment are way to usual or common in Medellín, along with the fact that not much attention is given to this problem. Now, if we talk about the related problems, it can be showed with systems like the ones to ride through streets and avenues avoiding traffic accidents and traffic in cities; or projects like the one in London that is used to calculate heat on the city and search for possible terrorist attacks.

Keywords

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

1. INTRODUCTION

The motivation on this project resides on the fact that it was found very important to help people to search or have routes that can provides them with a path to wherever they may want to go, that doesn't make them handle with way too much sexual harassment but at the same time doesn't make them go through way to long road.

If we're going to talk about history about this problem or passed things about this kind of problems, there's an article named: 'The risk you run' talking about how those women suffered of verbal abuse, threats of sexual violence and discrimination during the wave of anti-government protests that has been spreading through Colombia since April 28.

1.1. Problem

The main problem it's based on an idea to calculate the risk to be sexual harassed and avoid that kind of situation the most while looking for the most short or fastest path to a place having in mind the most important thing: not exceeding certain amount of sexual harassment. This problem has a huge impact on the society because it's important for people to understand that sexual harassment it's not normal and it doesn't have to happen to them; they must feel safe while they're going somewhere. We can conclude that is important the solution of this kind of problems, it's important to make people see that they can be secure and go safe to places.

1.2 Solution

In this semester we are using Dijkstra, along with a dictionary in pandas and polyline to graphic the path. We choose Dijkstra because it has shortest times, some of the other libraries we saw are slower. Also, because we already have used Dijkstra and finally because it uses recursion, something we have been seeing thought this semester so I'll help us to keep learning.

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 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 Path-finding Algorithm for Maximizing On-time Arrival Probability.

- The problem they found it's given by the fact that lots of people are usually traveling somewhere to another through routes; but those people don't know which path maximizes the probability of on-time arrival.
- The results of the study were given through a two-stage solution algorithm that: 1. Finds the upper and lower bound of on-time arrival estimated time. And 2. Tries to find the results to a the most reliable path through a multi-criteria label-setting approaching.
- They found an algorithm that has a remarkable computational advantage over others and efficiently determines the most reliable path to go from a place to other.

2.2 Finding Lonely Routes for Runners and Bikers.

- There's problems with conservating personal health with COVID-19 going around the world and at the same time maintaining good physical and mental health through exercising.

- This problem was solved with the creation of an algorithm that allows the user to choose a start and some end point, and considerate the distance and popularity of paths and finds the most optimal one.
- They reached the conclusion that the things proposed by them provides the user to get a nice workout and at the same time minimize the probability to be exposed to others.

2.3 Real-Time Pathfinding in Multirobot Systems Including Obstacle Avoidance.

- The problem base resides in the idea of trying to approach to the solution of the path problem to make the robot get to the end, including obstacle avoidance in multirobot systems.
- It's a multirobot system based on a few algorithms that have a real method for real-time pathfinding itself that uses a systematic design procedure for multirobot systems which includes the hierarchical coordinator and is based on the nonlinear control approach.
- They approached to demonstrate the efficiency of their real-time pathfinding by showing several cases of practical interest, like the interaction of stationary robots, a collision avoidance between three robots and the movement of robots avoiding obstacles of variable size.

2.4 Increase the Safety of Road Traffic Accidents by Applying Clustering.

- It was found as a problem the increase in the number of traffic accidents and an even more alarming trend the increasing of those accidents with catastrophic consequences of human life health.
- They used the implementation of advanced information systems in existing traffic environment, along with accident clusters, databases of traffic accident, modern ICT technologies, etc.
- The results they obtained were: the increase of road safety thanks to the development of intelligent transport systems as well as the application of information and communication technologies.

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 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. Map is available at GitHub³.

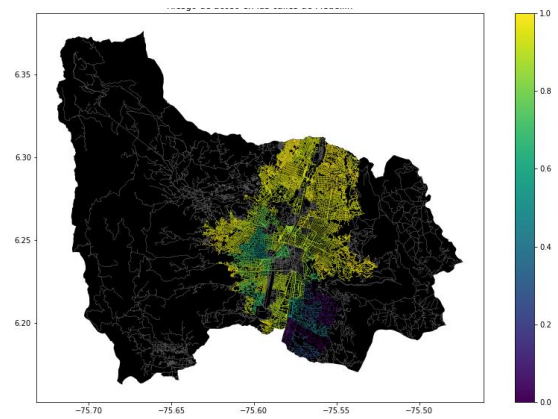


Figure 1. Risk of sexual harassment calculated as a lineal combination of the fraction of households that feel insecure and the fraction of households with income below one minimum wage, obtained from Life Quality Survey of Medellín, in 2017.

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

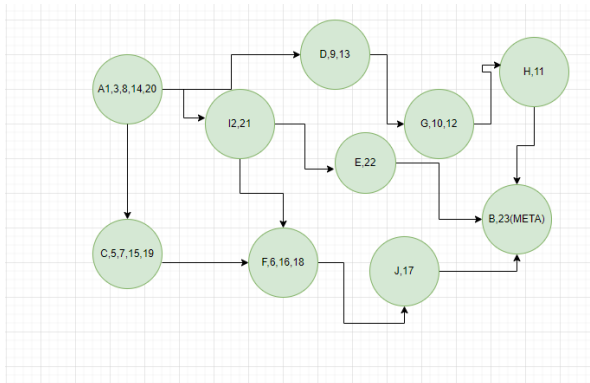
³ <https://github.com/mauriciotoro/ST0245Eafit/tree/master/projecto/Datasets/>

3.2 Constrained Shortest-Path Alternatives

In what follows, we present different algorithms used for constrained shortest path. *(In this semester, examples of such algorithms are DFS, BFS, a modified version of Dijkstra, a modified version of A*, among others).*

3.2.1 Depth first search

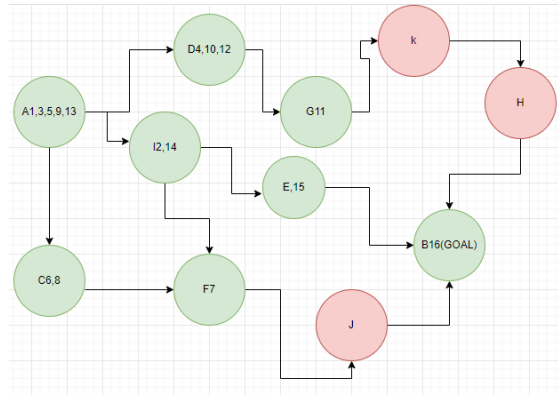
The way this algorithm works can be explained as if it were a blind man touching walls in a large room until he finds what he needs after walking so much. With this algorithm, you start with a random node and then randomly choose a neighbor and travel to it until you reach the desired node.



The letter means the name of the node and the numbers separated by commas are the steps that have been taken and indicate the order of the path that the algorithm would take.

3.2.2 Breadth first search

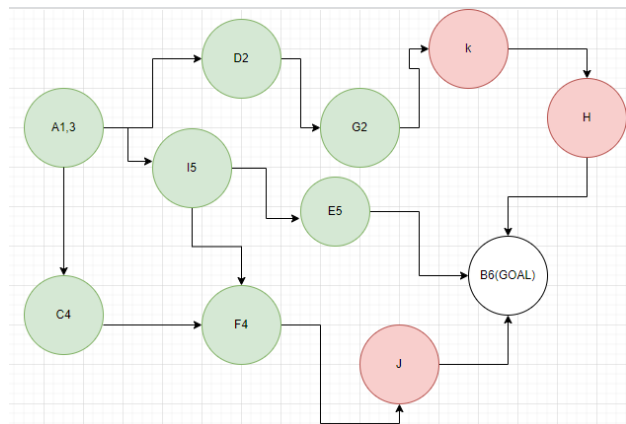
We can use the analogy of the previous algorithm to explain this algorithm as well. Now the blind man has consciousness and a walking stick which means that he can identify in a very short distance where he is standing, and he can remember where he has been and try to remember where he should go until he arrives thanks to his consciousness. This means that this algorithm first evaluates the nearest neighbors and then covers the neighbors of those nodes until reaching the destination one step at a time.



The letter means the name of the node and the numbers separated by commas are the steps that have been taken and indicate the order of the path that the algorithm would take. In addition, the red nodes are those that the algorithm has avoided.

3.2.3 Dijkstra

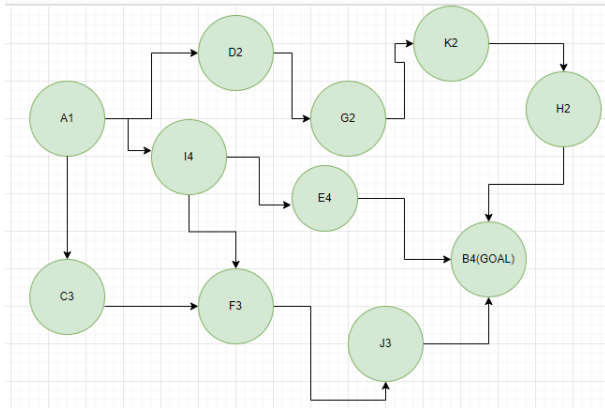
This algorithm can be seen as a completely healthy person with the mind of a curious child trying to explore the shortest paths. This means that what the algorithm does is investigate the shortest paths and then try them until it finds which one guides it to the goal. of the fourth algorithm



The letter means the name of the node and the numbers separated by commas are the steps that have been taken and indicate the order of the path that the algorithm would take. In addition, the nodes are those that the algorithm has avoided. And it can be added that the number of steps has been greatly reduced because the algorithm can evaluate more distance nodes.

3.2.4 A*

The operation of this algorithm it's like the previous one with the only difference that the algorithm decides which path to expand based on both the length of the path and how far from the goal each path would be.



The algorithm goes through all the nodes and in one step per path and was able to successfully choose the shortest path.

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

We are using a panda's dictionary. We have a couple of keys or information (name, origin, destination, length, oneway, harassmentRisk, geometry) on files to organize, it has this data among with the organization of this data on maps to graphic it. Besides the teacher gave us some keys for coding that work well.

Keys	Values
Name	Calle 10
Origin	(-75.5728593, 6.2115169)
Destination Linepoint	(-75.5739582, 6.2081811)
Length	18.479
One Way	True
Harassment Risk	0.5265393126661695
Geometry	LINestring (-

⁴ <http://www.github.com/LauZar12/proyecto/>

	75.5660756 6.2089072, - 75.56591880000001 6.2088496)
--	--

Data structure is presented in Figure 2.

Figure 2: An example of a street map is presented in (a) and its representation as an adjacency list in (b). (Please, feel free to change this Figure if you use a different data structure).

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 starts with a node and analyzes the graph to find the shortest path between that node and all other nodes in the graph.

The algorithm keeps track of the currently known shortest distance from each node to the source node and updates these values when a shorter path is found.

The process continues until all nodes in the graph have been added to the path. This way, we have a path connecting the source node to all other nodes and following the shortest path to each node. Algorithm is exemplified in Figure 3.

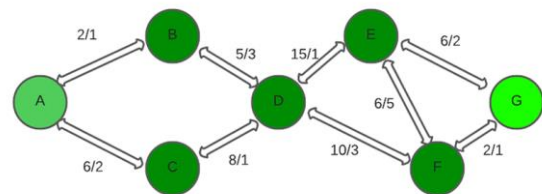


Figure 3: Solving the constrained shortest-path problem with Deep First Search (DFS). (Please, feel free to change this Figure if you use a different algorithm).

4.2.2 Second algorithm

Explain the design of the algorithm to calculate calculates the path with the lowest weighted-average risk of harassment without exceeding a distance d and make your own figure. Do not use figures from the Internet, make your own. (In this semester, the algorithm could be DFS, BFS, a modified version of Dijkstra, a modified version of A*, among others). Algorithm is exemplified in Figure 4.

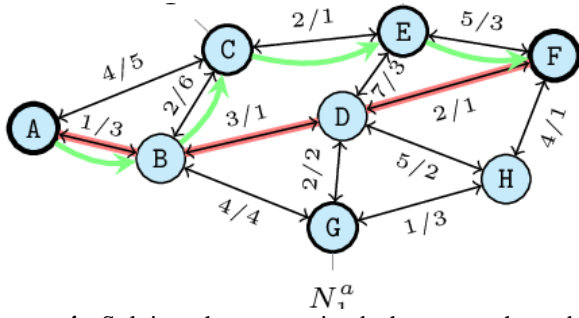


Figure 4: Solving the constrained shortest-path problem with Deep First Search (DFS). (Please, feel free to change this Figure if you use a different algorithm).

4.4 Complexity analysis of the algorithms

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

Algorithm	Time Complexity
Name of the algorithm	$O(V^2 * E^2)$
Name of the second algorithm (in case you 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 do V and E mean in this problem).

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

Table 2: Memory Complexity of the name of the data structure that your algorithm uses, where V is... E is... (Please explain what do V and E mean in this problem).

4.5 Design criteria of the algorithm

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 that I found on the Internet", "I did it on the last day before deadline", "it's easier", etc. Remember: This is 40% of the project grading.

5. RESULTS

In this section, we present some quantitative results on the shortest path and the path with 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	??	0.84
Universidad de Antioquia	Universidad Nacional	???	0.83
Universidad Nacional	Universidad Luis Amigó	??	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	??	5,000
Universidad de Antioquia	Universidad Nacional	???	7,000
Universidad Nacional	Universidad Luis Amigó	??	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.

Compute execution time for the queries presented in Table 3. Report average execution times.

	Average execution times (s)
Universidad EAFIT to	100.2 s

Universidad de Medellín	
Universidad de Antioquia Universidad Nacional	800.1 s
Universidad Nacional to Universidad Luis Amigó	845 s

Table 4: Execution times of the algorithm name (*Please write the name of the algorithm, for instance, DFS, BFS, a modified A**) for the queries presented in Table 3.

6. CONCLUSIONS

Explain the results obtained. Are shortest paths significantly different from paths with lowest harassment-risk? How is this useful for the city? Are execution times reasonable to use this implementation in a real-life situation?

6.1 Future work

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its implementation? Will you continue this projects by further working on Optimization? Statistics? Web development? Machine learning? Virtual Reality? How?

ACKNOWLEDGEMENTS

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved this manuscript.

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:

- Freund E, Hoyer H, Real-Time Pathfinding in Multirobot Systems Including Obstacle Avoidance, Freud E, February 1, 1988
<<https://journals.sagepub.com/doi/abs/10.1177/027836498800700104>>

- Yu C, Shi C, Zhang J, Lam W.H, Li Q, Xiang S. Most reliable path-finding algorithm for maximizing on-time arrival probability. *Transportmetrica B: Transport Dynamics*, Taylor & Francis, Online, 2016, 248-264.
<<https://www.tandfonline.com/doi/abs/10.1080/21680566.2016.1169953>>
- Spicher, Ethan T., "Finding Lonely Routes for Runners and Bikers" (2021). *REU Final Reports*.27. VOL 4208, 2004, 234-245.
<<https://archives.pdx.edu/ds/psu/36294>>
- Kos G, Brlek P, Meic K, Vidovic K, Increase the safety of road traffic accidents by applying clustering, De Gruyter, 2015. (Anonymous, 2021), 4-28
<<https://happycoding.io/tutorials/libgdx/pathfinding>>
- Estefania Cassingena Navone. 2022. Dijkstra's shortest path algorithm - a detailed and visual introduction. (February 2022). Retrieved April 17, 2022 from
<https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/>