**ALGORITHM TO FIND THE SHOURTEST ROUTE AND REDUCE HARASSMENT AT THE SAME TIME**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sebastián Cano Rincón  Universidad Eafit  Colombia  Scanor2@eafit.edu.co | Andrea Serna Universidad Eafit Colombia asernac1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

**For each version of this report: 1. Delete all text in red. 2. 2. Adjust the spaces between words and paragraphs. 3. Change the color of all text to black.**

**Red text =** Comments

**Text in black =** Contribution of Andrea and Mauricio

**Text in green** = To complete the first delivery

**Blue text** = To be completed for the 2nd deliverable

**Text in purple** = To be completed for the third deliverable

# **ABSTRACT**

Street harassment is a big problem that everybody could suffer, mostly if we talk about women, It can affect a lot of areas of our lives, for example, your mental health or your state of mind. Based on that, it's possible to say that the harassment create a lot of specific problems because the consequences, for example, suicide, depression or absenteeism

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**

The street harassment is a real big issue, based on statistics one of three women suffer street harassment in the world, the problem is that street harassment can lead to really bad things to the person, for example, can rape you, kill you, affect your physical and mental health, etc. That’s why is so important to find a way to prevent it.

# **1.1. The problem**

The problem is that there’s a lot of people in the streets that harass others causing a lot of problems, making thar person feel uncomfortable. Also, we’re considering reducing the distance because the idea is to optimize everything, there are a lot of ways to go to a specific place, but we mostly don’t know those ways, so the idea is to combinate the algorithm that reduce the distance with the reduction of harassment, solving two problems in one algorithm and, as I said, optimizing the solution.

**1.2 Solution**

The idea is to create a way to reduce the harassment and the distance using an algorithm such as Dijkstra, A\*, etc. Using this algorithm, you’ll be able to choose better for your security, so it really helps people that suffer that kind of situations and if we talk about harassment, it’d be useful for woman, because based on statistic, one of three women suffer harassment. I chose Dijkstra algorithm because it’s considered one of the most effective algorithms to reduce the distance, and it considerate the level of harassment.

**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 Street Harassment**

This article will let you see about the main information you should know about this topic, but also give you tips about the prevents you may take if it happens to you, such as going to a safe place, do what is best for you (like don’t saying anything, just ignoring that person) and reporting the behaviour. This articles, as I said, talk about being quiet when it happens to you, and sometimes that’s quite important, because if you’re alone and that person gets angry with you, maybe it could be quite hazardous for your security, but in the end, it depends on the case.

## **2.2 Preventing and Ending the Cycle of Street Harassment and Sexual Violence**

This article talks about how we could respond in a better way to this kind of problems. They say it’s quite important to collectively shift the culture, making conscious in other, trying to change the norms that create the harassment and sexual violence. Being a person who is working on the problem, like activist and letting know to others that if something like that happens, you don’t have to be quiet about that.

## **2.3 Tips for how to respond to Philly Street harassment and what to do when you witness it.**

# This article is quite interesting because it talks about two situations, who suffer the harassment and who see it, so they say that, if you suffer it, and you feel that the environment is enough safe for you, you can respond to the person is harassing you. Also, they recommend seeing if that person is working in a company, because then you will be able to report it to the company.

If you are the person who is seeing the situation, you can try to create a distraction, call out a supportive comment and even report the situation.

## **2.4 10 Steps You Can Take to Address Street Harassment**

This article focusses on making conscious, for example, they say that if your friend or someone from your family catcalls someone on the street, you should talk with that person about the situation. They say that being a person who share useful information on social media is a good option to help. They say that if it’s safe, you can intervene with the situation and try to help the person.

## **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]](#footnote-1)  and downloaded using the Python API[[2]](#footnote-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]](#footnote-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 DFS Algorithm**

DFS algorithm works in trees in a big profundity and the process finish when the stack is empty. The idea is to visit the initial node, put it in a stack and use the underlying to see the nodes, when you don’t visit the node, you insert it in a stack. You will be able to see step by step how it happens in the following link.

https://lucid.app/lucidchart/395e6d2b-f8aa-41e7-b403-fcb665284b09/edit?invitationId=inv\_59e136cc-f864-49fb-8e62-b15f72c7d290#

**3.2.2 Dijkastra Algorithm**

The idea with this algorithm is to build a graph in which we will know the first node and we will go to the node in which we can access directly. We have to consider that the nodes will be positive, so that’s why it works so well. You can visualize de algorithm in the following link:

[https://lucid.app/lucidchart/bc6d92d6-e3e7-427b-b54f-4a5da6776b4a/edit?viewport\_loc=-11%2C-11%2C2219%2C1108%2C0\_0&invitationId=inv\_ce73409e-dea8-4dfa-97ea-efd116acf11b#](https://lucid.app/lucidchart/bc6d92d6-e3e7-427b-b54f-4a5da6776b4a/edit?viewport_loc=-11%2C-11%2C2219%2C1108%2C0_0&invitationId=inv_ce73409e-dea8-4dfa-97ea-efd116acf11b)

**3.2.3 Bellman-Ford Algorithm**

It’s like Dijkstra, it can solve the problem with negative weights and it’s important to have weights. So, the big difference is that Bellman-Ford do not allow negative cycles. The origin of this algorithm is the origin that you decide. Bellman-Ford relax all the edges and do it E| – 1 time (E is the number of edges of the graph).

https://lucid.app/lucidchart/f5bf0307-0d10-4ccd-9ec1-402bbe8ec8a0/edit?view\_items=hFDXXo3W7FFT&invitationId=inv\_80ea31d7-5935-4288-9091-5b62b3d01704

**3.2.4 A\* Algorithm**

It’s quite similar to Dijkstra, some people say that is a better implementation of Dijkstra algorithm, however it wasn’t the intention, the similarity was just a coincidence. This algorithm run from an origin, then it evaluates the adjacent edge and take a total number, for example, in the graph that you can find on the link below, is evaluated like this

F(B) = 4+2 = 6, then F(C)=4+7=11…. F(B)>F(C)

Then it evaluates F(D) and F(E) and then it does it successively, it depends on the number of nodes.

https://lucid.app/lucidchart/2d4d9845-d8a0-4565-8c2a-73d9440b2281/edit?viewport\_loc=126%2C-36%2C2994%2C1495%2C0\_0&invitationId=inv\_de88fa3f-d966-4392-81a6-e096cdb59d0f

## **4. ALGORITHM DESIGN AND IMPLEMENTATION**

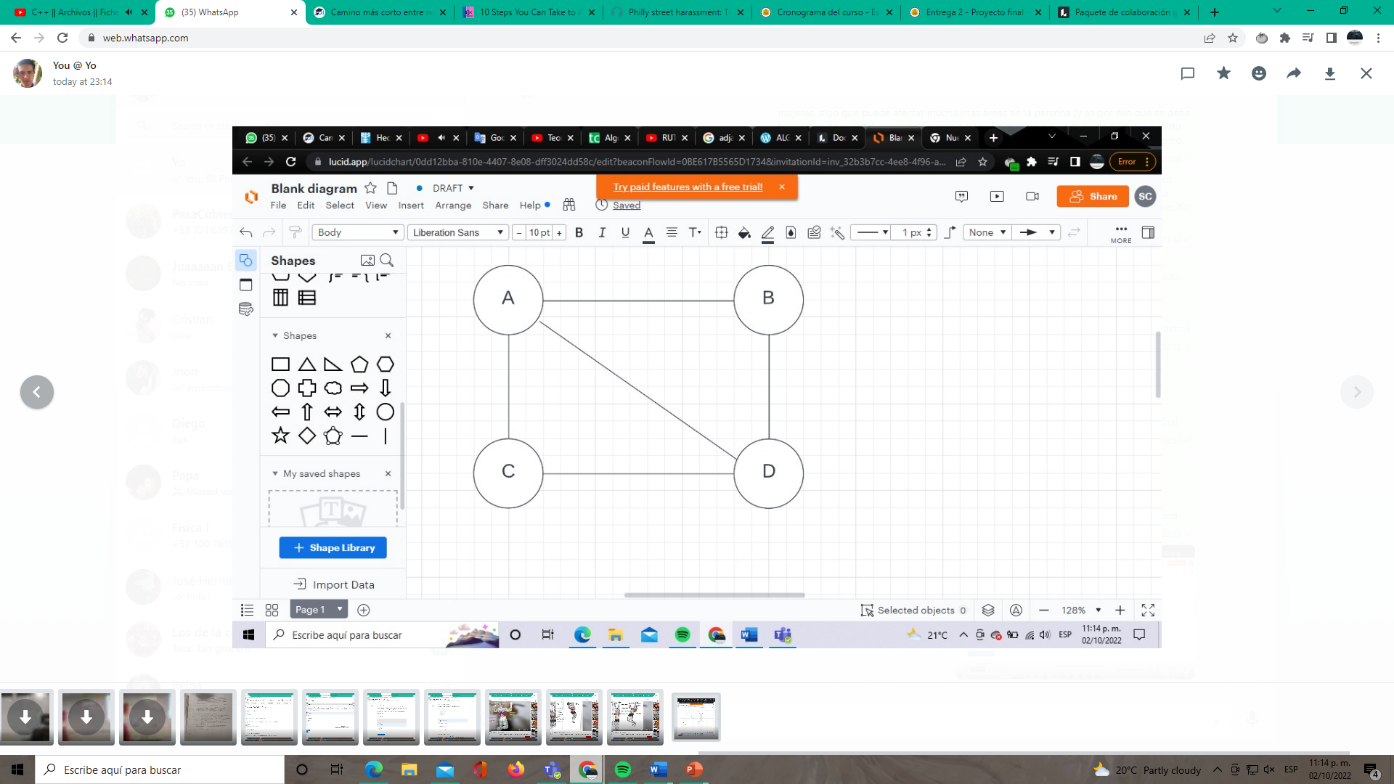
## In the following, we explain the data structures and algorithms used in this work. The implementations of the data structures and algorithms are available on Github[[4]](#footnote-4) .

## **4.1 Data Structures**

## The data Structure used was the adjacency matrix, which can save in rows, as is shown in the picture, some information. It’s used to represent a finite graph. The adjacency matrix is so useful because you can see the information easily and if we consider the problem we’re talking about, it’s even more useful. The data structure is presented in Figure 2.

Interfaz de usuario gráfica, Aplicación, Word, PowerPoint

Descripción generada automáticamente**Figure 2:**



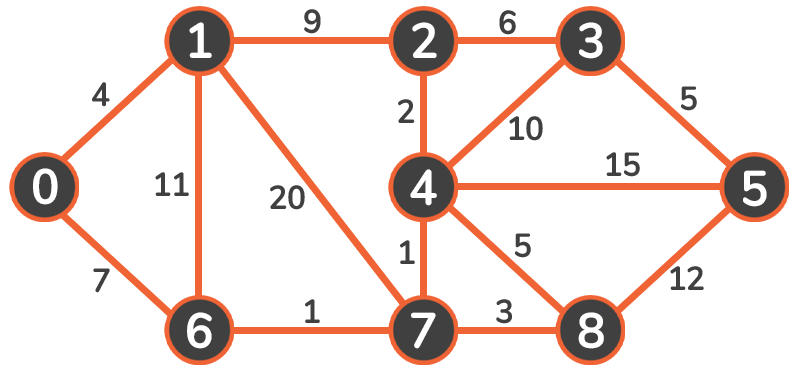
Made with power point and https://lucid.app/lucidchart/0dd12bba-810e-4407-8e08-dff3024dd58c/edit?viewport\_loc=152%2C163%2C870%2C407%2C0\_0&invitationId=inv\_32b3b7cc-4ee8-4f96-a295-165f5b4f5198#

**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 is Dijkstra, one the most popular algorithms to find the shortest ways to reach a place, this algorithm basically evaluates the arcs and determine if an arc is shorter than another one until finally get the answer. The problem with this algorithm is that is slow. The good part of the algorithm is that it considers the harassment risk easily than other algorithms. The algorithm is exemplified in Figure 3.



Took from: <https://stackabuse.com/courses/graphs-in-python-theory-and-implementation/lessons/dijkstras-algorithm/>

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

The paths that reduce both distance and risk of street sexual harassment are the following:

* V = d2r
* V = d10r
* V = d\*r

Dijkstra considers the weight of a graph, so in this case the idea was to combine the distance and the risk harassment to create that weight, so when Dijkstra visit a graph, instead of only consider one thing, it considers two, making it more efficient in terms of security and rentability of time.

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**

Because it iterates n-1 times as maximum. Then you must consider that in each iteration it adds vertices, and you have to consider the Sk number that is the label. So, it would be 2(n-1) then it becomes O(n²)

|  |  |
| --- | --- |
| **Algorithm** | **Time complexity** |
| Dijkstra | O(n²) |

**Table 1:** Time complexity of the Dijkstra Algorithm, where V is vertices and E is edges*.*

|  |  |
| --- | --- |
| **Data Structure** | **Complexity of memory** |
| Adjacency list | O(n) |

**Table 2:** Memory complexity of the adjacency matrix, where V is vertices and E is Edges *because if you consider it, you should multiply each element, so you will obtain data \* data*

**4.4 Algorithm design criteria**

I choose Dijkstra because it’s one of the best algorithms to find the shortest path in a maze, even though is not so good if we talk about time, but I like it a lot because you can manipulate in a really easy way the weigh that each edge has. But even if the time is not the best, if we consider the size of this graph, Dijkstra is one of the most useful algorithms because it works so well with this kind of size

**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 | 8100 |  |
| Eafit | Unal | 8148 |  |
| Eafit | Unal | 8400 |  |

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 = d2r | 100000.2 s |
| v = d10r | 800000.1 s |
| v = d\*r | 8450000 s |

## **Table 4:** *Algorithm* name execution times *is Dijkstra*

## **6. CONCLUSIONS**

In fact, if you see the distance, there’s not a lot of difference between each other. This is quite useful to people, because you can use it to avoid the harassment and to reduce the distance, which is useful for everyone, but mostly for woman. I think that the first path would be great for a mobile phone or web, because it’s mor simple than the others, so, for example, if you use a mobile phone, it will work so much easier.

**6.1 Future work**

I’m not so interesting in this kind of problems as well, but maybe I’ll use them because I would like to be a hacker, and they use a lot of complex algorithms to do their work, so I’d like to improve my programming skills to achieve my dreams and be better in what I want to do. To improve my algorithm application, I have thought that practice with other people, using some web pages and involving myself in some courses would be pretty useful for me to be better in this. But at all, I’m not pretty sure on how I could continue with this project, but I’d say that maybe I’d rather to continue with Artificial Intelligence, using algorithms to create big things in a hacker environment.

# **ACKNOWLEDGEMENTS**

I know this subject is quite hard, when I started this career I didn’t even know what was a programming language, and this course has shown me what difficult could it be, but at the same time, it has shown me that you can do incredible things through the programming languages, I have learning that with your mind you can create basically everything you imagine and that there are a lot of people who could prove that the intelligence has no limits, and that’s what I take from this course and from the teacher Jonathan.

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. *Graphs in Python - Theory and Implementation*. (s. f.). Stack Abuse. Recovered 5/ October/ 2022, de <https://stackabuse.com/courses/graphs-in-python-theory-and-implementation/lessons/dijkstras-algorithm/>
2. Orso, A. (2017, 26 august). Tips for how to respond to Philly street harassment. Billy Penn. Recovered 5th of October/ 2022, from <https://billypenn.com/2017/08/17/tips-for-how-to-respond-to-philly-street-harassment/>
3. Bhat, M. (2017, 5 April). Preventing and Ending the Cycle of Street Harassment and Sexual Violence. PreventConnect.org. Recovered, October 5th / 2022, from <https://www.preventconnect.org/2017/04/preventing-and-ending-the-cycle-of-street-harassment-and-sexual-violence/>
4. Camino más corto entre nodos | Aprende Programación Competitiva. (s. f.).Recovered, 5th October/ 2022, from https://aprende.olimpiada-informatica.org/algoritmia-dijkstra-bellman-ford-floyd-warshall
5. https:\/\/www.gatevidyalay.com\/author\/admin\/#author. (2020, 4 enero). A\* Algorithm | A\* Algorithm Example in AI. Gate Vidyalay. Recovered, 5th October/ 2022, from <https://www.gatevidyalay.com/a-algorithm-a-algorithm-example-in-ai/>
6. ALGORITMO DE BÚSQUEDA: DEPTH FIRST SEARCH (PARTE 1). (2016, 3 noviembre). Algorithms and more. Recovered, 5th October/ 2022, from <https://jariasf.wordpress.com/2012/03/02/algoritmo-de-busqueda-depth-first-search-parte-1/>
7. Tree Data Structure. (s. f.). Recovered, 5th October/ 2022, from [https://www.cs.cmu.edu/%7Eclo/www/CMU/DataStructures/Lessons/lesson4\_1.htm#:%7E:text=A%20tree%20is%20a%20nonlinear,or%20one%20or%20more%20subtrees](https://www.cs.cmu.edu/%7Eclo/www/CMU/DataStructures/Lessons/lesson4_1.htm" \l ":%7E:text=A%20tree%20is%20a%20nonlinear,or%20one%20or%20more%20subtrees).
8. 10 Steps You Can Take to Address Street Harassment. (s. f.). It’s Not A Compliment. Recovered, 5th October/ 2022, from https://itsnotacompliment.org/10-ways-tackle-street-harassment

1. <https://www.openstreetmap.org/> [↑](#footnote-ref-1)
2. https://osmnx.readthedocs.io/ [↑](#footnote-ref-2)
3. https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets [↑](#footnote-ref-3)
4. https://github.com/user23007/ProyectoED [↑](#footnote-ref-4)