The fast greedy search. (Project for an algorithm that reduces the risk of sexual harassment)

Maximiliano Sanchez Lara
Universidad Eafit
Colombia
Msanchezl2@eafit.edu.co

Santiago Celis
Universidad Eafit
Colombia
scelisl@eafit.edu.co

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

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

Integrant extra (Con el permiso del profesor): Felipe Gomez Daza

Universidad Eafit
Colombia
fgomezd1@eafit.edu.co

ABSTRACT

In this project we want to take a look into sexual harrasment in the streets. Sexual harrasment isn't only limited to fear dividing itself into other problems like work deficiency, suicides, and unwanted pregnancies. Creating an algorithm that creates the shortest routes to avoid and lower these risks is the objective in this project.

Key words

Shortest route, street sexual harassment, identification of safe routes, crime prevention

1. INTRODUCTION

Sexual harrasment is when people get harrased by the opposite sex and it can happen everywhere, some examples being public spaces like the streets and private spaces. Sexual harrasment in the streets could be easily prevented if they knew what routes are safe to take to get to their destination. This could be implemented by creating a new navigation algorithm that calculates the shortest and safest paths to reach a certain destination by taking into account already researched data on the paths with the most sexual harrasment and the distances between them.

1.1. The problem

The current problem is to calculate three different routes that reduce distance and risk. In order to give the user a variety of routes depending on their current needs and lower the amount of victims to sexual harrasment. Solving this problem would be an advance to current navigation systems where getting you to your location is their only objective instead of considering your full safety.

1.2 Solution

We chose a dijkstra algorithm in order to find the shortest possible distances or risk values possible from one location to another. The reason we chose dijkstra was because the time complexity is one of the shortest possible ones. We also have the advantage of no negative weights in our graph allowing it to work perfectly and faster than other algorithms that take it into account.

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 Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians

The project consists of the route navigation systems today that don't consider other factors instead they only consider the shortest route. The shortest route isn't everything for a quality navigation system. There is so much more than just the shortest route like safety, scenery, and others. This project wishes to recognize routes and give them weights depending on what they are needed for on the variables. For this reason they considered using dijkstra's algorithm in order to examine all of these subdivided routes. Since there doesn't exist a measuring unit that measures all variables universally and can be compared then they had to divide the responses into different categories which we will only be talking about the responses on visited locations. After which they obtained pretty good results when they weighted routes based on popular spots based on popular opinion based on social media but they did not seem to please the people in the minorities like disabled people who don't often appear in those types of sources[1].

2.2 Preventing Sexual Harrasment Through a Path Finding Algorithm Using Nearby Search

The problem in this related work is to find a way to avoid sexual harrasment using a heatmap. The heatmap they create will contain the areas where sexual harrasment has presented itself making them use Bresenham's algorithm to find the safest route between each spot that seems to be safe in the heatmap. After this they apply a scoring system from how safe the route is to how dangerous it can be and choose the safest route from all the previous generated ones. Since this is only a research work on a solution that can be applied it has not gone into testing whether it works or it doesn't.[2]

2.3 **Route - The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data**

People who are new to the city, have no idea about the safe routes. While traveling, you should be aware and choose the safest route. Safety and security became the top most priority of people due to the rising number of crimes in cities. For this situation, they proposed a solution divided in three algorithms. The first algorithm is called "Data Preprocessing Mask Algorithm", that consists of data collection, between them are the arrest data (crime data) and the accident data. Later, they used parts of code, as they are RM (is a command from Unix family of operating systems used to remove files and directories from the file system. Unix family is a portable operative system), Na.rm (is an expression that consists in a condition that is if it minds values in the variable, it should remove them), Accident score of a point (this consists in the count of pedestrians, cyclists and motorists injured and killed), among other things. [3]

The second algorithm is called "K-Mean" or "K-Means" that is an unsupervised classification (clustering) algorithm that groups objects into k groups based on their characteristics. The grouping is done by minimizing the sum of distances between each object and the centroid of its group or cluster. Quadratic distance is often used.[3]

The algorithm consists of three steps:

Initialization: once the number of groups, k, is chosen, k centroids are established in the data space, for example, choosing them randomly.[3]

Assignment of objects to centroids: each data object is assigned to its closest centroid.[3]

Centroid update: the position of the centroid of each group is updated, taking as the new centroid the position of the average of the objects belonging to said group.[3]

And finally, the third and last algorithm is K Nearest Neighbor, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.[3]

2.4 Safety-aware routing for motorized tourists based on open data and VGI

The project consists in explain the situation of the actual political conflicts and global terrorism. These terms are around the world that appear as parallel societies inside countries with great economic power due to social segregation. These areas imply high degree of crimes connected with violence, drug abuse and prostitution. The perception of urban danger often goes together with globally supported stereotypes. To recover the security and avoid the criminal acts, they decided to use the "OPTICS"algorithm (Ordering points to identify the clustering structure).[5]

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

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

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

harassment is defined as one minus the normalized CL. Figure 1 presents the calculated risk of bullying. The map is available on GitHub[3] .

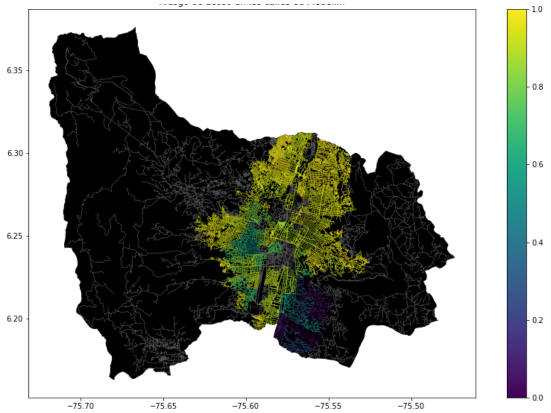


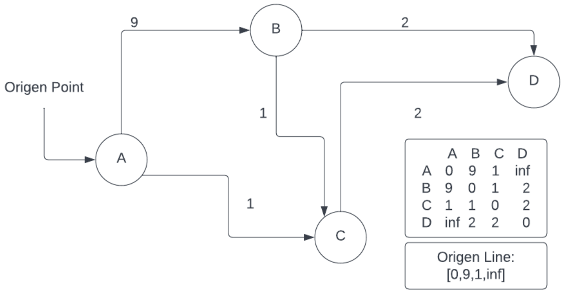
Figure 1. Risk of sexual harassment calculated as a linear combination of the fraction of households that feel unsafe and the fraction of households with income below one minimum wage, obtained from the 2017 Medellín Quality of Life Survey.

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

Dijkstra's algorithm consists in finding all distances from an origin point to different points or nodes you could call them. It consists in turning all of the distances into a NxN matrix where N is gonna be the number of nodes that we are going to have, the matrix consists of placing the distance from an origin point to all the others. After we have all that we are going to grab an origin point in the NxN matrix and then find the minimum value in the origin Nx1 line we have. After that we compare the sum of that minimum value with the sum of a number in the matrix in the minimum position in a k position. If this is less than the number in our origin line in the k position then the number in the origin in that k position gets replaced by the sum of the number in the origin line in the minimum position and the number in the matrix in the minimum position in the k position.The restrictions are that you can't compare the same minimum position more than ounce and k cant take its value anymore after you use it. If we also don't know the distance from one point to another point then we just place the distance as infinite since the route doesn’t exist and will be replaced later. The weights can be applied to risk values or distance values and then compared in order to reach our location.



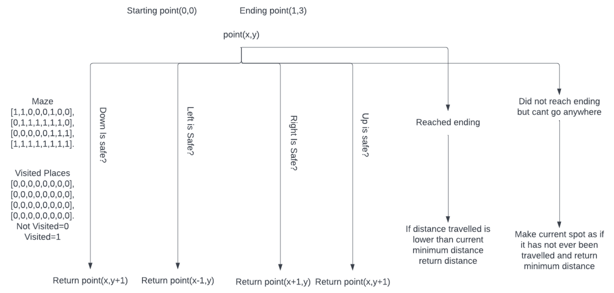
	A	B	C	D
A	0	9	1	inf
B	9	0	1	2
C	1	1	0	2
D	inf	2	2	0

Origen Line:
[0,9,1,inf]

3.2.2 Find the shortest path in a maze

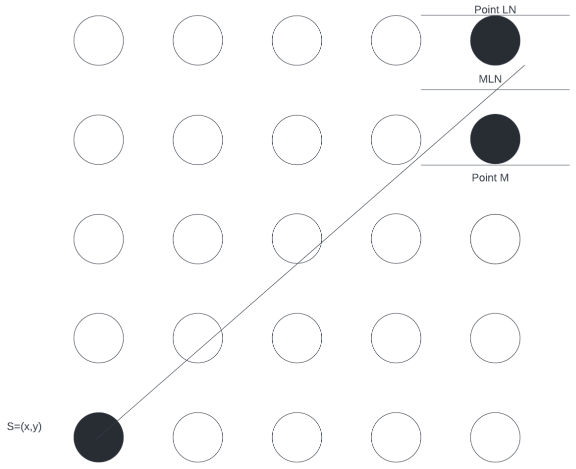
The shortest path in a maze algorithm consists in checking all selected paths through a backtracking technique until we find that minimum amount of distance again. The backtracking technique consists of creating an array with the same dimensions as the one in our maze or beginning array as you would call it. If we have already visited that place we would avoid and take another route by which if we reach nothing at the end of one of the routes we take we are going to set the end point as not visited and we are going to return the value minimum distance. In order to travel every single possible path we are going to create multiple parallel universes where each universe would have gone up, down, left, or right in order to reach the same destination through a different but shorter path. After this we only need our starting point to our ending point. The same idea applies to a map with color like a maze where all colors different to black are a route to follow towards a destination and it can also be applied towards risk factors where every color different than black is a risk towards our location.

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



3.2.3 Bresenham's line algorithm

Bresenham's line algorithm is a line drawing algorithm that determines the we take in a dimensional plane with a point and a final point. For this algorithm we have to create a line from a starting point in the plane to a finishing point in the plane from the left to the right. The point S is the initial grid point and has already been determined to be traversed through. We then need to determine the current grid point to be shaded with regard to the previous point. To do so, we also consider the grid points M and LN (for East and Northeast of the current grid point respectively). MLN is the middle point between the M and LN (Northeast and East).[2]



3.2.4 OPTICS

"OPTICS"(Ordering points to identify the clustering structure) algorithm is an algorithm for finding density-based clusters in spatial data). This is defined with a graphic representation with points and distance, where exists distances such as:

Core Distance: It is the minimum value of radius required to classify a given point as a core point. If the given point is not a Core point, then it's Core Distance is undefined.

Reachability Distance: It is defined with respect to another data point q(Let). The Reachability distance between a point p and q is the maximum of the Core Distance of p and the Euclidean Distance(or some other distance metric) between p and q. Note that The Reachability Distance is not defined if q is not a Core point.

The minimum points are the minimum amount of points that could fit on the space.

[4]

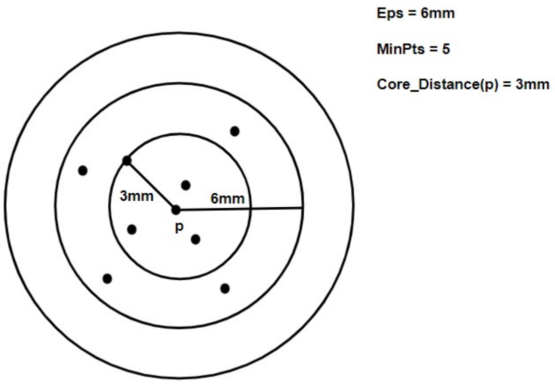
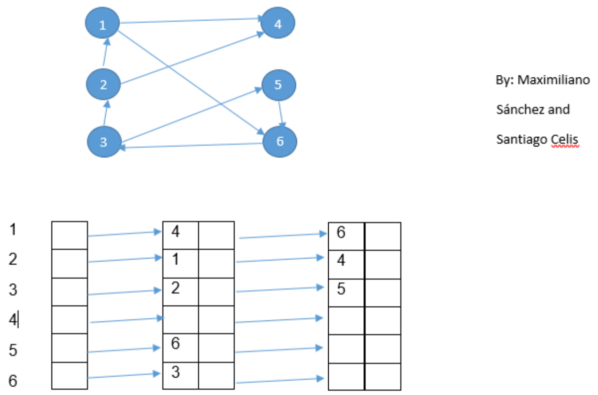


Image inspired by GeeksForGeeks[4]

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



[4] http://www.github.com/ ????????? /.../project/

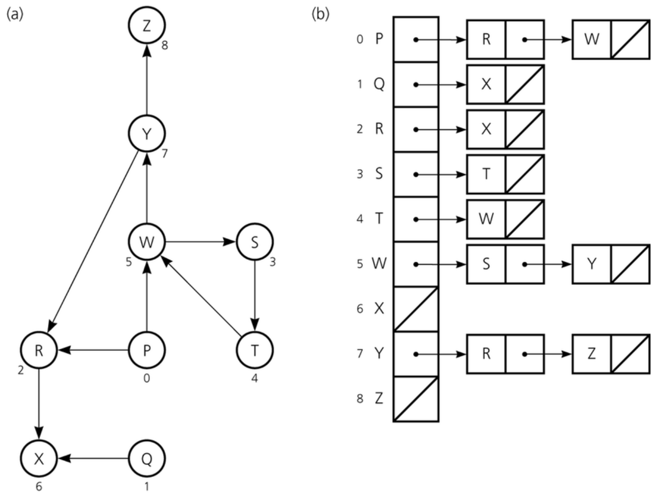


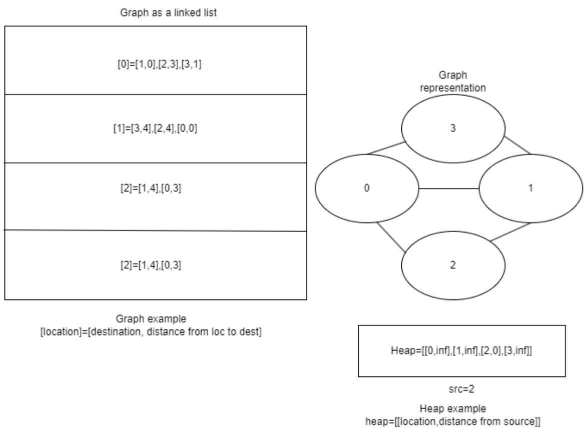
Figure 2: An example street map is presented in (a) and its representation as an adjacency list in (b). (*Please feel free to change this graph if you use a different data structure*).

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

For this semester we used a dijkstra algorithm on a linked list sorting values with a heap, so that we can make use of a priority que. Like any dijkstra algorithm we first define our distance from our starting to any other location as infinite in the heap like shown in the graph. After this we define our starting location(AKA src) to our starting location distance as 0 since we start there in the heap. After this we target the lower distances first in the heap except the ones that are equal to infinite. This way we can start removing them and focus on going around the graph and then calculating the distance from that point and check if it's lower than previously calculated from another place in order to change it. Also as a side note we take distance as a sum between both sexual harrasment and distance, taking a ratio from each and completing it. We keep doing this until the heap is empty and once it's done we have every single distance from that one place src.


Graph as a linked list

[0]=[1,0],[2,3],[3,1]

[1]=[3,4],[2,4],[0,0]

[2]=[1,4],[0,3]

[2]=[1,4],[0,3]

Graph example
[location]=[destination, distance from loc to dest]

Heap=[[0,inf],[1,inf],[2,0],[3,inf]]

src=2

Heap example
heap=[[location,distance from source]]

Figure 3: Calculation of a path that reduces both distance and risk of harassment (please feel free to change this figure if you use a different algorithm).

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

For the two other paths we are going to do the same previous procedure but focus only on distance on the one that minimizes distances and only add risk for the one that only focuses on risk. These paths use the same algorithm but with different distance values.

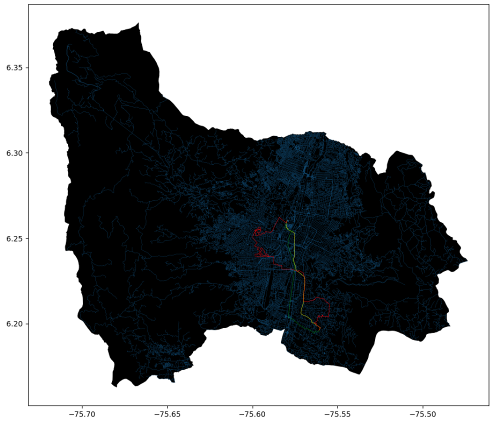


Figure 4: Map of the city of Medellín showing three pedestrian paths that reduce both the risk of sexual harassment and the distance in meters between the EAFIT University and the National University.

4.3 Algorithm complexity analysis

We calculated these complexities by taking the worst case into account. First we saw a for inside of a while and said this should be two different variables multiplied together since the sizes of them aren't equal. After which the last straw was the decrease value making the variable decrease into O(Log(V)). To this we also notice that this is similar to a BFS to which we do O(N+V). After all of this we have O(Log(V))*O(E+V).

Algorithm	Time complexity
Dijkstra	O((E+V)*Log(V))

Table 1: In our algorithm E means the nodes in our Adjacency list graph which can vary from the position in v and V represents the nodes in our heap.

Data Structure	Complexity of memory
Adjacency List	O(V)

Table 2: This is because we specify every single node in our algorithm which in turn takes up that much memory also known as a linear memory complexity. It should be obvious because it is but a simple adjacency list in the worst case.

4.4 Algorithm design criteria

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

The Dijkstra algorithm

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
Entrada a el Tesoro	UN Medellin	8809.722999999998	0.6595981235274482
Entrada a el Tesoro	UN Medellin	9839.547	0.6731124501596933
Entrada a el Tesoro	UN Medellin	23194.528999999995	0.6462234930749653

Distance in meters and risk of sexual street harassment (between 0 and 1) to walk from the Entrance to the Tesoro 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=8809.722999999998	0.67s
v=9839.547	0.57s
v=0.6462234930749653	0.82s

Table 4: *Algorithm* name execution times *(Please write the name of the algorithm, e.g. DFS, BFS, A*)* for each of the three calculator paths between EAFIT and Universidad Nacional.

6. CONCLUSIONS

The first path which prioritizes distance is significantly different to the other two paths which prioritize safety, and distance combined with safety. This is useful for the city because it provides fast, safe, and a combination of both for an optimal path that citizens can use. Runtimes for this algorithm are reasonable since 1 route takes approximately 1 second to execute and this is applicable in real time situations like navigation. For the mobile application and web application I would recommend the route that takes both risk and distance into account so that they have an enjoyable experience while traveling in the city because the shortest and safest route have similar values to this one.

6.1 Future work

From this project, we were thinking about the things that we could improve in the future. In this project we have been thinking that we would like to improve the interface of the project in order to make it more user friendly since we don't even allow the user to interact with it. To improve the graphic interface, we can use libraries in the code which allow us to develop objects called GUI’s (Graphic User Interface). For the project we would also like to improve the optimization of the algorithm in order to use bigger databases and change the complexity of ELog(N) to a lower one like Log(N). We would like to apply machine learning in the future to this project by adding a prediction in the changes of risk in the routes over time. Another thing we would like to add to this would be for the program to grab its own data instead of having already prepared data for it in the databases course we are going to take in the semesters to come.

ACKNOWLEDGEMENTS

Thanks to teacher Alexander de Jesus Narvaez, Universidad EAFIT, for providing examples from code they had on heaps helping understand how they end up working. Also thanks to Student Gregorio Bermudez , Universidad EAFIT, for providing information sources to build up the program we made.

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. Siriaraya, P., Zhang, Y., Wang, Y., Wakamiya, S., Jeszenszky, P., Kawai, Y. and Jatowt, A. Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians, 2020.Retrieved August 14,2022, from ResearchGate: https://www.researchgate.net/publication/343251733_

2. Ma, D., 2020. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search, 2020. Retrieved August 15,2022, from Omdena: https://omdena.com/blog/path-finding-algorithm

3.Soni, S., Sandeep, C. and Gauri Shankar, V.Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data, 2019. Retrieved August, 15,2022, from ResearchGate: https://www.researchgate.net/publication/338096313_Route-The_Safe_A_Robust_Model_for_Safest_Route_Prediction_Using_Crime_and_Accidental_Data

4. GeeksforGeeks. ML | OPTICS Clustering Explanation - GeeksforGeeks, 2019. Retrieved August 15,2022, from GeeksforGeeks : https://www.geeksforgeeks.org/ml-optics-clustering-explanation

5.Keler, A. and Damascene Mazimpaka, J. Safety-aware routing for motorised tourists based on open data and VGI, 2016. Retrieved August 15,2022, from ResearchGate: https://www.researchgate.net/publication/301566519_Safety-aware_routing_for_motorised_tourists_based_on_open_data_and_VGI