

DEVELOPMENT OF AN ALGORITHM FOR THE OPTIMIZATION OF PEDESTRIAN ROUTES BASED ON STREET HARASSMENT OF THE LGBTQ COMMUNITY

Valentina Movil
Universidad EAFIT
Colombia
vmovils@eafit.edu.co

Felipe Gil Macia
Universidad EAFIT
Colombia
fgilm@eafit.edu.co

Andrea Serna
Universidad EAFIT
Colombia
Asernac1@eafit.edu.co

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

ABSTRACT

The acronym LGBTQ has been used to refer to people that identify as lesbian, gay, bisexual, or transgender or those who have doubts about their sexuality and/or gender identity (*queers*) [1]. On the other hand, during 2017, a large part of the violent deaths of LGBTQ individuals in Colombia occurred at Medellín. Then, many of the victims were engaged in low-paid jobs and high exposure to violence, that is, recycling, informal trade, and prostitution [2]. Therefore, the LGBTQ population is mostly exposed to violence on the street. Next, in the Aburrá Valley, 30% of trips are made on foot or by bike [3]. For this, it is necessary to provide security to the citizens, making special emphasis on population groups that, when moving, are more likely to be attacked, for example, LGBTQ individuals and/or groups.

Keywords

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

1. INTRODUCTION

The denomination LGBTQ is an inclusive name that refers to lesbians, gays, bisexuals, transgender, or queers [1]. On the other hand, LGBTQ people have been overwhelmingly subjected to domestic and sexual violence [4, 5]. For example, during 2017 a large part of the violent deaths of LGBTQ people occurred in Medellín. About their trades, many of the victims were engaged in low-paid jobs and high exposure to violence, that is, recycling, informal trade, and prostitution [2]. Additionally, the origin-destination survey carried out in the Aburrá Valley (AMVA- in Spanish acronym) that same year showed that around 30% of the trips in this place are made on foot or by bicycle. Then, with the interest of being a sustainable territory, government entities have promoted the use of these means of transport among the inhabitants of the AMVA. However, vulnerable groups such as women and the LGBTQ community use them the least [3]. For example, it is estimated that LGBTQ people and/or groups are 3 or 4 times more likely to be attacked [5]. For this reason, it is necessary to propose and carry out projects, in this case, in science, technology, and innovation that improve their safety guarantees on the street. Specifically, this paper seeks to point to what has been said through the development of an optimization algorithm for pedestrian routes based on street harassment of people from the LGBTQ community.

1.1. Problem

In the AMVA, the local government promotes walking and cycling as a tool to reduce air pollution. Likewise, it places the pedestrian at the top of priorities in terms of mobility [3]. Currently, however, the main reason for vulnerable communities (women, LGBTQ people, etc.) not to commute sustainably (i.e., walking or cycling) is exposure to street harassment [6]. That is why, in this work, the development of an algorithm for optimizing pedestrian routes through a data structure is presented, which focuses on calculating the shortest path without exceeding a weighted average risk of street harassment for people from the LGBTQ community. In fact, this solution provides a reference framework in ST+I in the reduction of insecurity in terms of sustainable mobility with a gender approach.

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. RELATED PROJECTS

2.1.1. Walking Secure: Safe Routing Planning Algorithm and Pedestrian's Crossing Intention Detector Based on Fuzzy Logic App

This job pretends to improve road safety through artificial intelligence (AI). Actually, AI is now crucial to achieving more secure smart cities. For this reason, another objective of this reference is to develop a mobile app based on the integration of the smartphone sensors, and a fuzzy logic strategy to determine the pedestrian's crossing intention around crosswalks is presented. Also, the app developed also allows the calculation, tracing, and guidance of safe routes thanks to an optimization algorithm that includes pedestrian areas on the paths generated over the whole city through a cloud database (i.e., zebra crossings, pedestrian streets, and walkways). Finally, a total of 30 routes were calculated by the proposed algorithm and compared with Google Maps considering the values of time, distance, and safety along the routes. As a result, the routes generated by the proposed algorithm were safer than the routes obtained with Google Maps, achieving an increase in the use of safe pedestrian areas [8].

2.1.2. A computational model of pedestrian road safety: the long way round is the safe way home

This work proposes a novel linear model of pedestrian safety in urban areas with respect to road traffic crashes that

considers a single independent variable of pedestrian path safety. This variable is estimated for a given urban area by sampling pedestrian paths from the population of such paths in that area and in turn estimating the mean safety of these paths. We argue that this independent variable directly models the factors contributing to pedestrian safety. This contrasts previous approaches, which, by considering multiple independent variables describing the environment, traffic, and pedestrians themselves, indirectly model these factors. In sum, using data about 15 UK cities, it demonstrated that the proposed model accurately estimates numbers of pedestrian casualties [9].

2.1.3. A route navigation algorithm for pedestrian simulation based on grid potential field

This job explains that pedestrian simulation modeling has become an important means to study the dynamic characters of dense populations. In the continuous pedestrian simulation model for complex simulation scenarios with obstacles, the pedestrian path planning algorithm is an indispensable component, which is used for the calculation of pedestrian macro path and microscopic movement desired direction. However, there is less efficiency and poor robustness in the existing pedestrian path planning algorithm. To address this issue, we propose a new pedestrian path planning algorithm to solve these problems in this article. In our algorithm, we have two steps to determine pedestrian movement path, that is, the discrete potential fields are first generated by the flood fill algorithm and then the pedestrian desired speeds are determined along the negative gradient direction in the discrete potential field. Combined with the social force model, the proposed algorithm is applied in a corridor, a simple scene, and a complex scene, respectively, to verify its effectiveness and efficiency. In conclusion, the results demonstrate that the proposed pedestrian path planning algorithm in this article can greatly improve the computational efficiency of the continuous pedestrian simulation model, strengthen the robustness of application in complex scenes [10].

2.1.4. Routes' Safety Evaluation Application Development

For this job, pedestrians and cyclists are the most vulnerable road users. One of the ways to increase their safety is the safest route selection. Nowadays, there are a lot of navigation systems and route planners. However, these routes are not assessed from the viewpoint of their safety. The article describes one of the possible system alternatives to assess the safety of routes for different types of road users. The developed system is based on the multifactorial analysis of information. Correction factors allow considering special features of each type of road user. Finally, the developed algorithm is implemented as an application [11].

2.2. RELATED ALGORITHMS

2.2.1. Dijkstra algorithm

The objective of this algorithm is to travel from vertex A to vertex B by the fastest possible path. But ¿How does it work? This algorithm only has two processes; the first one is to update the estimates and the second one is to choose the next vertex.

At first, it has the weighted graph (i.e., vertex connected by edges that have different weights, those could be the distance or the time that takes going from a vertex to another one). So, the first step is to estimate the time or distance (which is dependent on the weights at the edges) that takes going from the initial vertex to any vertex. It's remarkable that in this step, the only time it knows is the time that takes going from the initial vertex to the same vertex, this is 0. Next, for the other vertexes, it does not know how long it takes to get there. Therefore, it estimates the infinite time in all the unexplored vertexes.

Following, it begins to compare which one of those vertexes has the minimum distance or time to get there. For example, where is an update estimation, must compare all the new vertexes that are known, and must choose the one that has the minimum time or distance to get there (choose the next vector). After, it repeats that process until arrived at the goal vertex by the shortest possible path. It should be notated, that must add a process that saves the edges that were taken to have the shortest path in our memory (See Figure 1.).

Finally, if it is in a vertex and at the time it is updating estimates and if one vertex has an estimate minor then the new estimate that it could have goes by a different edge or path. Then, it must not change the valor of the estimate in that vector. This is, it should only update estimations if the new value that we found is minor then the actual estimation. Too, this algorithm does not work when the weights of the edges are negative.

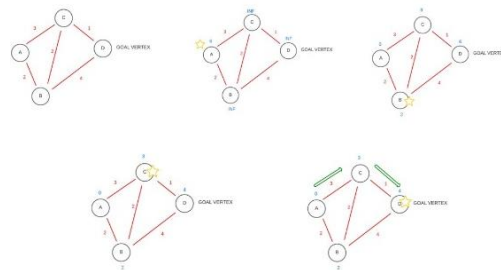


Figure 1: An example of finding the shortest path by Dijkstra algorithm [7].

2.2.2. A-star (A*)

Like before, the objective of this algorithm is to travel from vertex A to vertex B by the fastest possible path. For this reason, the A* algorithm uses the same method as the Dijkstra algorithm with the difference that this algorithm it

uses an additional parameter to compare the shortest path, and that parameter is called the heuristic.

Consequently, while Dijkstra only chose the shortest edge between two vertexes, even if going that way means being further from the goal vertex; the heuristic measures the distance between the vertex that we are estimating and the final vertex. Therefore, the algorithm described so far gives us only the length of the shortest path. To find the actual sequence of steps, the algorithm can be easily revised so that each node on the path keeps track of its predecessor. After this algorithm is run, the ending node will point to its predecessor, and so on, until some node's predecessor is the start node.

In addition, is possible to say that the A* algorithm is more efficient. For example, the following figure shows that if we are standing in A and that vertex relates to B and C, the weight of the edge A to B is 4 and the weight of the edge A to C is 5, and we want to go to D that has a distance of 7 with B and a distance of 5 with C. Then, if we follow the Dijkstra algorithm, we are going by the vertex B but, the best option was vertex C, the A* algorithm knew it because of the heuristic that shows not only how long are the connected vertex but also how long are the connected vertex with the goal vertex.

For all above, as the same problem shown in Figure 1 whose was solved by Dijkstra now it is solved by A*, here the lector can see that the process is faster than in Dijkstra.

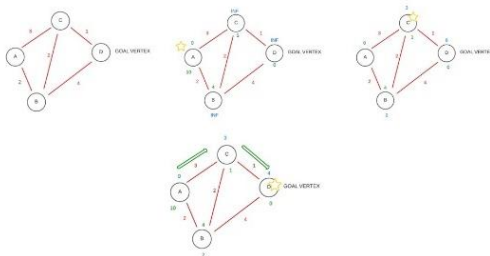


Illustration 2: The example solved in illustration 1, but now solved by A* algorithm [7].

2.2.3 Floyd-Warshall's algorithm

In the same way, the objective of this algorithm is to travel from A to B by the fastest possible path.

In consequence, this algorithm has two processes; the first one is to make a table where we calculate all the distances between all the pairs of vertexes, and the second one is to select the minimum value between two paths that takes to the same vertex, passing by different edges.

Then, to start, the first table (i0) calculates the distances between pairs of vertexes witch pass for a particular pair of vertexes. Hence, if those vertexes do not have any common path, must be assumed infinite distance. After it starts to

iterate, being the first iteration the distance between a pair of vertexes passing by the vertex 1, so it compares the new values of distances with the table of the last iteration and if it found shorter values than those it had before; it proceeds to add those values to the new table.

Finally, it iterates until we get to the table that makes the paths cross the vertex n (being vertex n the last vertex of the table); and in that table, we should have the shortest path for all the pairs of vertexes (See Figure 3.).

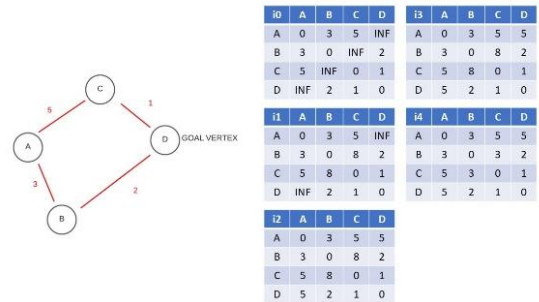


Figure 3: An example of finding the shortest path between vertex by Floyd-Warshall's algorithm [7].

2.2.4 BFS algorithm

The objective of this algorithm is to find paths from one vertex to another one. Hence, this algorithm is a recursive one witch asks the vertex where it is at the beginning and if there is an edge that connects this vertex with the goal vertex and if the answer is false, it calls the vertexes that relate to the initial vertex and makes the same question to each connected vertex until the answer to the question is true or until it asks to all the vertexes of the graph. It is remarkable to say that, this algorithm works wide and not deep because he asks to the first vertex and then to all his related vertex before he asks for deeper vertexes. This algorithm works for unweighted graphs. In sum, the following figure shows a practical view of what was said before.

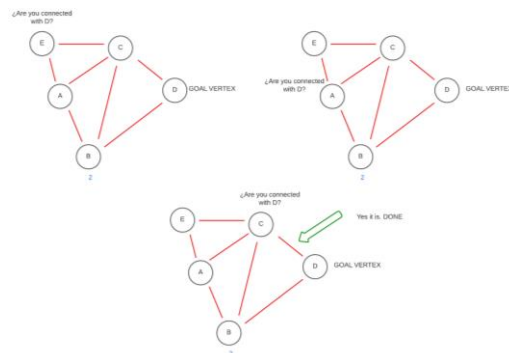


Figure 4: An example of finding a specific node by BFS algorithm [7].

REFERENCES

- [1] Lambda Legal and Child Welfare League of America, "Conceptos básicos sobre ser LGBT", Lambda Legal, 2013.
- [2] S. Kyu Choi, S. Divsalar, J. Flórez-Donado, K. Kittle, A. Ilan H. Meyer and P. Torres-Salazar, "STRESS, HEALTH, AND WELL-BEING OF LGBT PEOPLE IN COLOMBIA Results from a National Survey", 2019.
- [3] Área Metropolitana del Valle de Aburrá, "AVANZA EL PROGRAMA DE CICLOCAMINABILIDAD El Aburrá le camina a la bici", El Metropolitano del Valle de Aburrá, Medellín, 2019.
- [4]"Violence Against LGBTQ People | CARE Office", Care.ucmerced.edu, 2022. [Online]. Available: <https://care.ucmerced.edu/advocacy/violence-against-lgbtq-people>. [Accessed: 23- Feb- 2022].
- [5]"Sexual Assault and the LGBTQ Community", Human Rights Campaign, 2022. [Online]. Available: <https://www.hrc.org/resources/sexual-assault-and-the-lgbt-community>. [Accessed: 23- Feb- 2022].
- [6]S. Aguirre, "¿MEDELLÍN ES UNA CIUDAD HOSTIL PARA LAS CICLISTAS?", Colectivo SiCLas, Medellín, 2021.
- [7]D. Delling, P. Sanders, D. Schultes and D. Wagner, "Engineering Route Planning Algorithms", Algorithmics, vol. 5515, pp. 117–139, 2009. Available: <https://www.itk.itk.edu/extra/publications/dssw-erpa-09.pdf>. [Accessed 23 February 2022].
- [8]J. Lozano Domínguez and T. Mateo Sanguino, "Walking Secure: Safe Routing Planning Algorithm and Pedestrian's Crossing Intention Detector Based on Fuzzy Logic App", Sensors, vol. 21, no. 2, p. 529, 2021. Available: 10.3390/s21020529.
- [9]C. Hannah, I. Spasić and P. Corcoran, "A computational model of pedestrian road safety: The long way round is the safe way home", Accident Analysis & Prevention, vol. 121, pp. 347-357, 2018. Available: 10.1016/j.aap.2018.06.004.
- [10]M. Li, Y. Wei and Y. Xu, "A route navigation algorithm for pedestrian simulation based on grid potential field", Advances in Mechanical Engineering, vol. 11, no. 12, p. 168781401989783, 2019. Available: 10.1177/1687814019897831.
- [11]K. Shubenkova, V. Mavrin, G. Sadygova and P. Buyvol, "Routes' Safety Evaluation Application Development", MATEC Web of Conferences, vol. 334, p. 02010, 2021. Available: 10.1051/mateconf/202133402010.