

Algoritmo rutas seguras contra el acoso a la mujer en Medellín

Moisés Arrieta
Universidad Eafit
Colombia

mdarrietah@eafit.edu.co

Sara Cardona
Universidad Eafit
Colombia

smcardonav@eafit.edu.co

Juan López
Universidad Eafit
Colombia

jmllopezs@eafit.edu.co

Andrea Serna
Universidad Eafit
Colombia

asernac1@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia

mtorobe@eafit.edu.co

RESUMEN

En ciertas zonas de la ciudad de Medellín se presenta una alta probabilidad de acoso sexual y delincuencia hacia las mujeres, debido a esto su resolución es muy importante para garantizar a las ciudadanas, que a la hora de desplazarse por la ciudad estas se sienta seguras y tranquilas cuando se dirige a su lugar destino.

Algunos problemas relacionados con este son la inseguridad vial, agresión física, las amenazas e intimidación y otros tantos que se analizarán en este informe.

El algoritmo propuesto para la resolución del problema es el Dijkstra algorithm, el cual nos permite encontrar el camino más corto con menor acoso o el camino con menor acoso sin superar una distancia x .

Los resultados se podrán observar a lo largo de este informe específicamente en el punto 5, al igual que las conclusiones en el 6to punto.

Palabras clave

Camino más corto restringido, acoso sexual callejero, identificación de rutas seguras, prevención del crimen.

1. INTRODUCCIÓN

Cada día es más común que una mujer cuando se desplaza de un lugar a otro sea víctima de algún tipo de acoso callejero tales como palabras soeces e incómodas, miradas persecutorias, insinuaciones sexuales entre otros, un estudio del Centro de Atención Integral a Víctimas de Abuso Sexual de la Fiscalía General de la Nación, indica que, en el año 2020, en la capital antioqueña 1.647 mujeres fueron víctimas de delitos sexuales.

Por lo anterior el objetivo de este proyecto es encontrar las rutas más seguras de desplazamiento dentro de la ciudad de Medellín teniendo en cuenta un ponderado de acoso, lo cual mitiga que una mujer pueda vivir situaciones tan deplorables.

1.1. Problema

El problema consiste en encontrar una estructura de datos que permita calcular el camino más corto sin superar un riesgo de acoso entre dos distancias.

Resolver este problema aportaría a disminuir índices de inseguridad con respecto a delitos sexuales y delincuencia los cuales día a día es un problema en común para toda la sociedad.

1.2 Solución

La solución que presentamos para esta problemática, es implementar un algoritmo donde el usuario pueda interactuar con este, para llegar a su destino de una manera más segura, evitando considerablemente la posibilidad de agresión sexual, el algoritmo consiste en que el usuario le brinda los datos de la posición donde se encuentra y al lugar donde quiere llegar, por lo cual este se encargará de brindarle el camino más corto y con el menor riesgo de acoso.

1.3 Estructura del artículo

A continuación, en la Sección 2, presentamos trabajos relacionados con el problema. Posteriormente, en la Sección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuro.

2. TRABAJOS RELACIONADOS

A continuación, explicamos cuatro trabajos relacionados con la búsqueda de caminos para prevenir el acoso sexual callejero y la delincuencia en general.

2.1 Algoritmo para determinar la ruta más segura

Los incidentes criminales están en aumento en todo el mundo. Con el avance de la tecnología y los datos fácilmente disponibles relacionados con el crimen, es posible analizar estos datos para aumentar la conciencia y la seguridad de los ciudadanos. Usamos el conjunto de datos de San Francisco para proponer un modelo para determinar el camino más seguro de un lugar a otro.

El conjunto de datos consta de todos los delitos que han ocurrido en los últimos 12 años. Estos delitos van desde infracciones menores de tránsito hasta asalto e intento de asesinato y cada uno tiene una clasificación de seguridad “SI” o “NO”

calle	Asalto	Sexual ofensas	Conducir bajo el influencia	Secuestro	Robo	Vehículo hurto	Seguridad
1	ALTO	ALTO	ALTO	BAJO	BAJO	ALTO	NO
2	BAJO	ALTO	ALTO	BAJO	ALTO	BAJO	NO
3	BAJO	BAJO	BAJO	ALTO	ALTO	BAJO	SI
4	ALTO	BAJO	BAJO	ALTO	ALTO	ALTO	SI
5	ALTO	BAJO	BAJO	ALTO	BAJO	BAJO	SI
6	BAJO	ALTO	BAJO	BAJO	ALTO	ALTO	SI

Se crean perfiles de calle para cada calle. Un perfil de calle contiene el nombre de la calle y el recuento de las ocurrencias de cada categoría individual de delito cometido en una dirección que contiene la calle correspondiente

1) Calculation of risk weight of street_{ij}:

$$\text{Risk weight of street}_{ij} = \sum_{n=1}^{\text{total}} (\text{count}_{ijn} * \text{risk factor}_{ijn})$$

Where,

Total = the number of types of crimes

Count = the count of criminal incidents for crime_n

for street_{ij}

Crime factor = the intensity of the crime

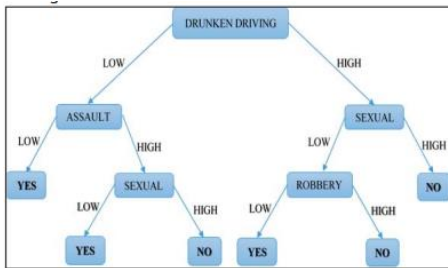
2) Calculation of risk of a route:

$$\text{Risk of route}_i = \sum_{k=1}^{tu} (\text{Risk weight of street}_{ik} * \text{distance}_{ijk})$$

Where,

tu = total number of unsafe streets on that route.

El árbol de decisiones se utiliza para encontrar la ruta más segura entre múltiples rutas posibles.



2.2 Detección de la ruta más segura a través del cálculo del índice de peligro y la agrupación de K-Means

Formular una solución para identificar la ruta más segura entre dos ubicaciones geográficas ingresadas. Usando el conjunto de datos de la ciudad de Nueva York, que nos proporciona estadísticas de delitos con etiquetas de ubicación; estamos implementando diferentes algoritmos de agrupamiento y analizamos los resultados comparativamente para descubrir el más adecuado.

El conjunto de datos utilizado para el trabajo propuesto es de la ciudad de Nueva York tomado de los recursos en línea NYC Open-Data.

El cálculo de la puntuación de accidentes para cada coordenada utilizando la información de los conjuntos de datos se puede estimar como:

Accident Score (As) = $P_k * 2 + C_k * 2 + M_k * 2 + P_i + C_i + M_i$ (1) where, P_k = Pedestrians Killed, C_k = Cyclists Killed, M_k = Motorcyclists Killed, P_i = Pedestrians Injured, C_i = Cyclists Injured and M_i = Motorcyclists Injured

Asimismo, para los datos de delincuencia, le hemos dado a cada delito una calificación para considerar su ponderación.

Tabla 1: Puntajes delictivos asignados a diferentes delitos para el cálculo del índice de peligrosidad

Crimen	ponderación	Crimen	ponderación
Violación	15	armas peligrosas	7
Otros delitos sexuales	15	Delito grave de asalto	6
Homicidio involuntario	14	delito grave no clasificado	6
Prostitución y delitos relacionados	13	Para otras autoridades	6
Delitos relacionados con niños	13	Robo	5
Secuestro y delitos relacionados	12	Herramientas de ladrón	5
Abandono/falta de mantenimiento infantil	11	Conducción en estado de embriaguez/en estado de ebriedad	4
Juegos de azar	10	Robo-fraude	4
Conducta desordenada	10	Infracciones de movimiento	3
robo de servicios	9	Leyes de tráfico y vehículos	3
Incendio premeditado	9	Falsificación/acoso fraudulento	2
Otras leyes estatales	8	Hurto mayor	1

Hemos implementado la agrupación en dos capas, la primera en función del índice de peligro y la segunda en función de las coordenadas para aumentar aún más nuestra precisión y permitir que las rutas se marquen con mayor precisión con el índice de peligro.

El resultado mostrado comprende la distancia, el tiempo de duración requerido para recorrer esa ruta y el índice de peligro calculado por el algoritmo propuesto. De esta manera, brinda al usuario una visión general de la seguridad en el camino que está a punto de recorrer.

2.3 Recomendación de ruta segura específica del usuario sistema

Las actividades delictivas han alcanzado niveles sin precedentes en casi todas las partes del mundo. Momentos desesperados como estos requieren medidas desesperadas para garantizar la seguridad de las personas, especialmente de las personas que necesitan viajar diariamente a lugares, conocidos y desconocidos.

Utilizamos la base de datos UCR [17] del FBI, que contiene registros bien equilibrados y seleccionados de víctimas relacionadas con delitos penales. Esta base de datos abarca un período de tiempo de más de 15 años.

Para la siguiente fase del algoritmo, es decir, el perfilado de rutas, utilizamos el conjunto de datos de San Francisco que contiene incidentes delictivos ocurridos durante 12 años.

1) Red de decisión

$$P(\text{Risk}) = P(\text{Age}) * P(\text{Sex}) * P(\text{Time}) \quad (1)$$

2) Cálculo del índice de riesgo y puntuación delictiva

$$\text{Risk Index} = P(\text{Risk}) * \text{Crime Score} \quad (2)$$

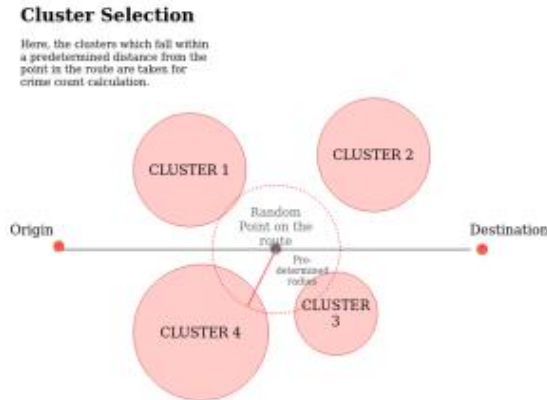
3) Información geoespacial reducida

Almacenamos los datos de conteo de delitos del grupo con el centroide del grupo correspondiente, obtenido del agrupamiento de K-Means como un atributo de índice.

4) Perfilado de ruta óptimo

El sistema toma la ubicación del origen y el destino como entrada del usuario para la recomendación de ruta. Luego, utilizando la API de Google Maps, se distinguen las tres rutas más cortas desde el origen hasta el destino. El sistema propone la ruta óptima

en función del índice de seguridad y la longitud de la ruta.



2.4 Be-Safe Travel, una aplicación geográfica basada en la web para explorar una ruta segura en un área

Esta investigación propone BeSafe Travel, una aplicación basada en la web que utiliza la API de Google a la que pueden acceder las personas a las que les gusta conducir en un área, pero que aún no conocen los caminos que están a salvo de la delincuencia.

Los datos necesarios para la aplicación Be-Safe Travel son la ubicación de los delitos y los tipos de delitos

Para calcular la distancia entre el lugar de origen y el lugar de destino, usamos la fórmula de Haversine

$$a = \sin^2 \frac{\Delta\Phi}{2} + \cos \Phi_1 * \cos \Phi_2 * \sin^2 \frac{\Delta\Lambda}{2} \quad (1)$$

$$c = 2 * \arcsin 2(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

$$\text{distance} = R * c \quad (3)$$

El programa comienza con la inicialización de mapas que define un contenedor en una etiqueta div HTML. Este contenedor mostrará Google Maps para la ruta deseada. La aplicación presentada utiliza DirectionService, que es la clase de la API de Google Maps para mostrar rutas.

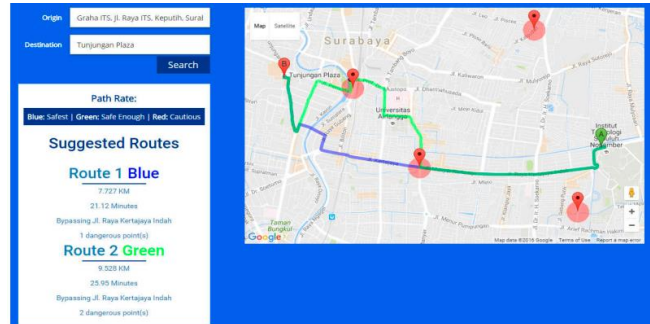
El siguiente paso es la inicialización de puntos peligrosos y radio, que contiene dos subprocesos.

El primer subproceso es conectarse a la base de datos. La aplicación se conecta a la base de datos MySQL y hay dos tipos de consultas. Primero, una consulta para recuperar el nombre y las coordenadas de ubicación expresadas por latitud y longitud. En segundo lugar, una consulta para obtener el waypoint definido como puntos peligrosos en los mapas.

El segundo subproceso creará el radio peligroso que marca la ubicación más cercana con los puntos del crimen como área peligrosa.

En primer lugar, el procedimiento contará cuántos puntos de delincuencia atraviesan las rutas posibles

Luego, el procedimiento crea una matriz en las secuencias de visualización de direcciones y variables incluidas en la matriz para determinar su secuencia.



3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos del camino más corto restringido para abordar el acoso sexual callejero.

3.1 Recogida y tratamiento de datos

El mapa de Medellín se obtuvo de Open Street Maps (OSM) ¹ y se descargó utilizando la API ² OSMnx de Python. La (i) longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías se obtuvieron de los metadatos proporcionados por OSM.

Para este proyecto, se calculó la combinación lineal (CL) que captura la máxima varianza entre (i) la fracción de hogares que se sienten inseguros y (ii) la fracción de hogares con ingresos inferiores a un salario mínimo. Estos datos se obtuvieron de la encuesta de calidad de vida de Medellín, de 2017. La CL se normalizó, utilizando el máximo y el mínimo, para obtener valores entre 0 y 1. La CL se obtuvo mediante el análisis de componentes principales. El riesgo de acoso se define como uno menos la CL normalizada. La

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub³.

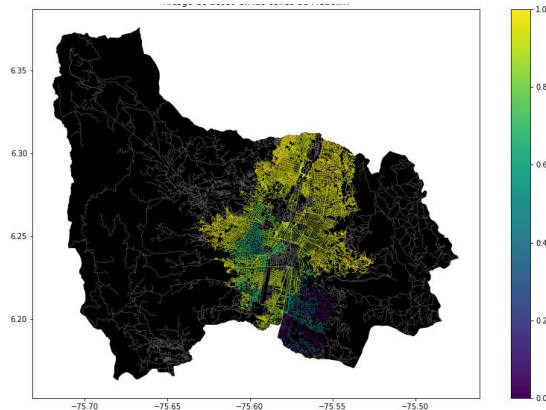


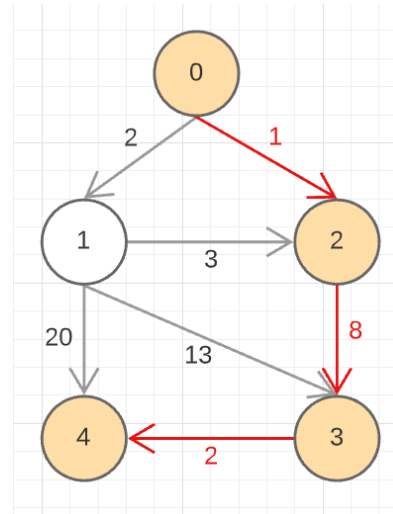
Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido. *(En este semestre, ejemplos de dichos algoritmos son DFS, BFS, una versión modificada de Dijkstra, una versión modificada de A*, entre otros).*

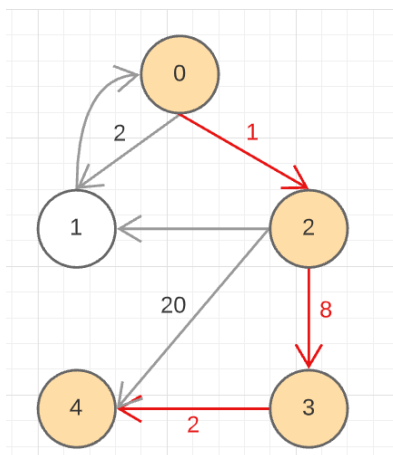
3.2.1 Dijkstra algorithm

El algoritmo de Dijkstra puede encontrar la ruta más corta desde un nodo llamado origen a todos los demás nodos de un grafo, produciendo un árbol de ruta más corta, este básicamente comienza en el nodo origen, luego analiza la ruta más corta desde ese nodo a todos los demás nodos del grafo, después ejecuta un seguimiento de la distancia más corta conocida actualmente desde cada nodo hasta el nodo principal u origen, y actualiza estos datos cada vez que encuentre una ruta más corta, prontamente de que el algoritmo haya encontrado la ruta más corta entre el nodo origen y otro nodo, este nodo se marca como visitado por lo cual se agrega a la ruta, este proceso continua hasta que todos los nodos del grafo se hayan agregado a la ruta. De esta forma tenemos una ruta que conecta el nodo de origen con los demás, siguiendo la ruta más corta posible, para llegar a cada uno de los nodos.



3.2.2 Back tracking algorithm

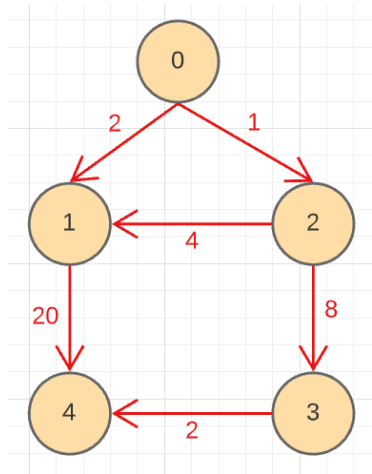
Es un método de búsqueda de soluciones sobre grafos, el cual se acelera mediante la poda de ramas poco prometedoras, normalmente se suele implementar como un procedimiento recursivo, también se generan funciones de cota, de forma que no se generan algunos estados si no conducen a ninguna solución, de esta forma se ahorra espacio en memoria y tiempo de ejecución, para diseñar este tipo de algoritmos se tienen en cuenta los siguientes elementos: una función de poda para eliminar partes del árbol de búsqueda, construir la solución del problema en distintas etapas, en cada caso se elige un posible candidato y se añade a la solución parcial, y si no quedan más valores por probar, se retrocede al candidato anterior, se desecha y se selecciona otro candidato.



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

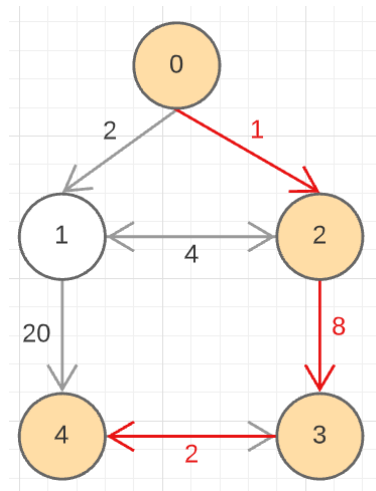
3.2.3 Algoritmo de búsqueda profunda aleatorizada

Este algoritmo busca aleatoriamente una ruta para llegar a una casilla final y verifica en cada caso si hay más de un camino que se pueda recorrer, si ese es el caso este guarda la posición para que cuando encuentre un camino que no tenga salida, regrese a la casilla anterior y continúe la búsqueda desde esa posición, este proceso se repite hasta encontrar el camino que mejor se ajuste a la necesidad.



3.2.4 Bellman Ford Algorithm

Este algoritmo genera el camino más corto desde un nodo origen al resto de nodos de un grafo como lo hace el algoritmo de Dijkstra, sin embargo, la diferencia es que este algoritmo soluciona el problema de la ruta mas corta, de una forma más general que Dijkstra, ya que permite valores negativos, en pocas palabras este algoritmo se utiliza únicamente cuando hay aristas negativas presentes en el grafo, lo que lo hace un poco mejor ya que admite longitudes negativas en los arcos.

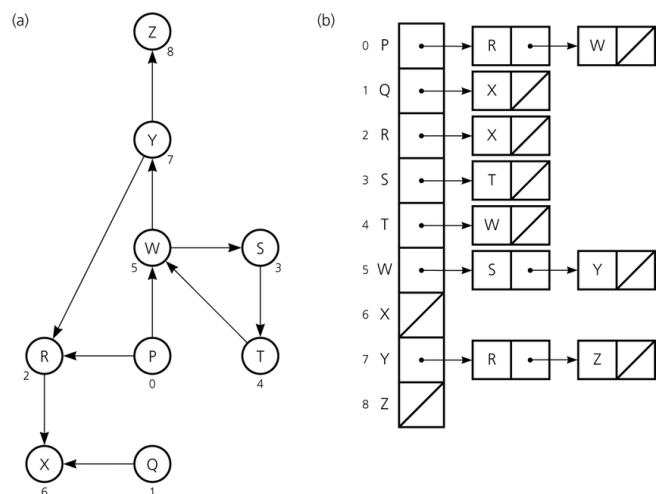


4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

A continuación, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en GitHub⁴.

4.1 Estructuras de datos

Para el algoritmo implementamos una lista de adyacencia la cual nos permite asociar cada vértice de un grafo a una lista que contenga aquellos vértices adyacentes a él, en este caso hacemos uso de esta estructura para los nodos únicos de origen y sus posibles destinos.



4.2 Algoritmos

En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de

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

acoso r . El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia d .

4.2.1 Primer algoritmo

El algoritmo encuentra la ruta más corta desde un nodo llamado origen a todos los demás nodos de un grafo, produciendo un árbol de ruta más corta, este básicamente comienza en el nodo origen, luego analiza la ruta más corta desde ese nodo a todos los demás nodos del grafo, después ejecuta un seguimiento de la distancia más corta conocida actualmente desde cada nodo hasta el nodo principal u origen, y actualiza estos datos cada vez que encuentre una ruta más corta, básicamente primero leemos el csv en un detaframe y guardamos los destinos de cada nodo origen, se crea un dataframe para el camino con el menor acoso, luego se crea una lista adyacente con los nodos únicos de origen y sus posibles destinos, el cual por ultimo imprimimos. El algoritmo se ejemplifica en la Figura 3.

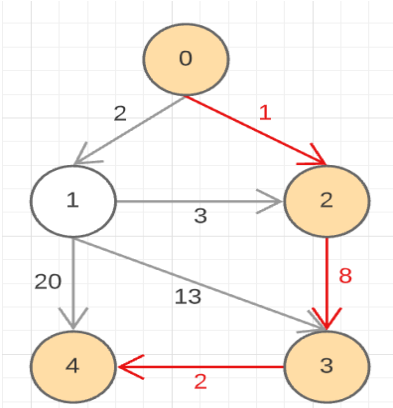
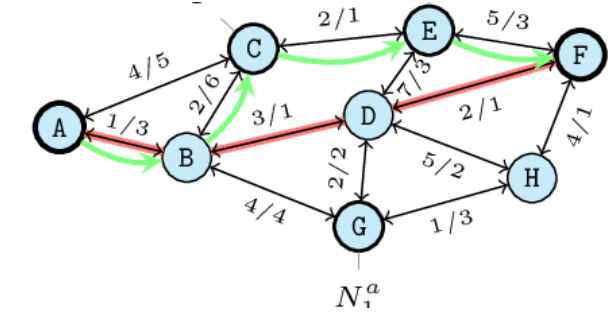


Figura 3: Resolución del problema del camino más corto restringido con Dijkstra.

4.2.2 Segundo algoritmo

Al igual que el primer algoritmo, se utilizó el Dijkstra algorithm el cual encuentra la ruta más corta desde un nodo llamado origen a todos los demás nodos de un grafo, produciendo un árbol de ruta más corta, este básicamente comienza en el nodo origen, luego analiza la ruta más corta desde ese nodo a todos los demás nodos del grafo, después ejecuta un seguimiento de la distancia más corta conocida actualmente desde cada nodo hasta el nodo principal u origen, y actualiza estos datos cada vez que encuentre una ruta más corta. A diferencia del primero, este está un poco modificado para no solo encontrar una ruta con un acoso ponderado, sino para encontrar una ruta la cual no exceda una cierta distancia límite.

El algoritmo se ejemplifica en la Figura 4.



4.4 Análisis de la complejidad de los algoritmos

Algoritmo	Complejidad temporal
Dijkstra algorithm	$O(V^2)$
Dijkstra algorithm modificado	$O(E \log V)$

Tabla 1: Complejidad en tiempo Dijkstra algorithm, la V representa los vértices o los nodos, en este caso las direcciones, y la E representa las aristas, quiere decir la ruta más corta

Estructura de datos	Complejidad de la memoria
Dijkstra algorithm	$O(V)$
Dijkstra algorithm modificado	$O(V)$

Tabla 2: Complejidad en tiempo y memoria de Dijkstra, la V representa los vertices o los nodos, en este caso las direcciones, y la E representa las aristas, quiere decir la ruta mas corta

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre el camino más corto y el camino con menor riesgo.

5.1.1 Resultados del camino más corto

A continuación, presentamos los resultados obtenidos para el camino más corto, sin superar un riesgo medio ponderado de acoso r ; en la Tabla 3.

Origen	Destino	Distancia más corta	Sin exceder r
Universidad EAFIT	Universidad de Medellín	6130.01	0.84
Universidad de Antioquia	Universidad Nacional	2192.820	0.83
Universidad Nacional	Universidad Luis Amigó	1457.79	0.85

Tabla 3. Distancias más cortas sin superar un riesgo de acoso medio ponderado r .

5.1.2 Resultados de menor riesgo de acoso

A continuación, presentamos los resultados obtenidos para el trayecto con menor riesgo de acoso medio ponderado, sin superar una distancia d , en la Tabla 4.

Origen	Destino	Acoso más bajo	Sin exceder d
Universidad EAFIT	Universidad de Medellín	R = 0.720 D = 4440.174	7,000
Universidad de Antioquia	Universidad Nacional	R = 0.872 D = 705.249	800
Universidad Nacional	Universidad Luis Amigó	R = 0.856 D = 1246.877	1600

Tabla 3. Menor riesgo de acoso ponderado sin superar una distancia d (en metros).

5.2 Tiempos de ejecución del algoritmo

En la Tabla 4, explicamos la relación de los tiempos medios de ejecución de las consultas presentadas en la Tabla 3.

	Tiempos medios de ejecución (s)
Universidad EAFIT a Universidad de Medellín	5.30 segundos
De la Universidad de Antioquia a la Universidad Nacional	4.86 segundos

De la Universidad Nacional a la Universidad Luis Amigó	4.90 segundos
--	---------------

Tabla 4: Tiempos de ejecución del nombre del *algoritmo* par Dijkstra algorithm a las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

Como se puede observar tanto en la presentación del proyecto, las diapositivas y en este informe, las rutas son diferentes, algunas mas significativas que otras, pero en general cambian. Gracias a un algoritmo como este la ciudad se puede beneficiar mucho, en especial las mujeres que caminan solas en ella, debido a que si se utiliza se pueden ver mejoras a la hora de ver los números en cuanto al acoso a la mujer en las calles de Medellín.

Los tiempos de ejecución de este algoritmo son muy razonables pues aún teniendo afán se puede utilizar, ya que como vemos tienen una ejecución promedio de 5 segundos en el cual te ayudara a ubicar la ruta mas corta o con menor acoso.

6.1 Trabajos futuros

El proyecto se puede llegar a mejorar, haciéndole optimizaciones y aplicándolo a alguna de las áreas de la programación, como por ejemplo en el desarrollo de software, donde se podría implementar como una aplicación móvil.

AGRADECIMIENTOS

Agradecemos a Víctor Daniel y a Samuel Lopera, por las sugerencias presentadas al grupo para el mejoramiento del algoritmo, a los monitores de la clase por sus ayudas, a el Icetex por ser medio para la beca de uno de los miembros del grupo en el componente de excelencia.

Agradecemos la ayuda con [técnica particular, metodología] a [Nombre Apellido, cargo, nombre de la institución] por los comentarios que mejoraron en gran medida este manuscrito.

Los autores agradecen al profesor Juan Carlos Duque, de la Universidad EAFIT, por facilitar los datos de la Encuesta de Calidad de Vida de Medellín, de 2017, procesados en un *Shapefile*.

REFERENCIAS

1. Alcaldía de Medellín, Se fortalecen las acciones de prevención de la violencia y el acoso contra las mujeres en el Centro de Medellín. Retrieved May 3, 2021, from Alejandro Noreña Jaramillo <https://www.medellin.gov.co/irj/portal/medellin?NavigationTarget=contenido/9737-Se-fortalecen-las-acciones-de-prevencion-de-la-violencia-y-el->

acoso-contra-las-mujeres-en-el-Centro-de-Medellin

2. International Journal of Computer Science and Information Technologies, Algorithm to Determine the Safest Route, Retrieved 2016
<http://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit20160703106.pdf>
3. Computers, Materials & Continua, Safest Route Detection via Danger Index Calculation and K-Means Clustering, Retrieved 2021
<https://www.techscience.com/cmc/v69n2/43879/pdf>
4. International Journal of Engineering Research & Technology, User Specific Safe Route Recommendation System, Retrieved 2020
<https://www.ijert.org/research/user-specific-safe-route-recommendation-system-IJERTV9IS100268.pdf>
5. Actas de la conferencia AIP1867, Be-safe travel, una aplicación geográfica basada en la web para explorar una ruta segura en un área, Retrieved August 01, 2017
<https://aip.scitation.org/doi/pdf/10.1063/1.4994426>