

MAPEO DE RUTAS CORTAS CON MENOR RIESGO DE ACOSO

(MRCMRA)

Daniela Álvarez Acevedo
Universidad EAFIT
Colombia
dalvareza1@eafit.edu.co

Tomás Jaramillo Gaviria
Universidad EAFIT
EAFIT
tjaramillg@eafit.edu.co

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

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

RESUMEN

Se requiere buscar la ruta más corta con las calles de Medellín con menor nivel de acoso callejero. Para identificar rutas seguras, prevenir el riesgo de acoso y aumentar la seguridad de las personas. A tal extremo que se pueda reducir también los delitos comunes como lo son los atracos, secuestros, violaciones y entre otros.

Para la solución a este problema se ha propuesto la implementación del algoritmo de Dijkstra para encontrar la ruta más corta de un punto a otro con menor riesgo de acoso. Además, se realizaron varias pruebas como la de encontrar la ruta de la Universidad EAFIT hasta la Universidad de Medellín y se pudo encontrar una ruta óptima sin exceder una distancia de 7km o en su caso de la Universidad Nacional a la Universidad Luis Amigó sin exceder los 10km.

Palabras clave

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

1. INTRODUCCIÓN

En este semestre, la motivación es porque requerimos buscar la ruta más corta sin superar cierto nivel de acoso callejero y a su vez sin superar cierta distancia. En ciertos casos, el nivel de acoso representa un peligro bastante grande para las personas, ya que, dependiendo el caso, puede ser perjudicial.

1.1. Problema

En este semestre, el problema consiste en que no solemos saber cuál puede ser la ruta más corta sin superar cierto nivel de riesgo por acoso sexual y sin superar cierta distancia. Así que sería útil resolver este problema para aumentar la seguridad de las personas y permitir que estas puedan movilizarse sin sentirse inseguras, debido a que la problemática tratada afecta directamente a la integridad de las personas.

1.2 Solución

Para la solución de la problemática dada, decidimos utilizar Dijkstra, debido a que, este algoritmo, tiene la capacidad de resolver el problema de la ruta más corta, tanto para redes cíclicas, como para redes acíclicas. De manera tal que los bucles que presente una red no restringen el uso del algoritmo.

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 Algoritmos para la determinación del circuito de menor longitud en un grafo

Determinar la menor longitud de un circuito en un grafo. Se implementaron diferentes algoritmos como por ejemplo Dijkstra que resuelve el problema con caminos mínimos en un grafo ponderado dirigido de pesos no negativos.

Rayco Hernández D. 2016. Algoritmos para la determinación del circuito de menor longitud en un grafo (julio, 2016).

2.2 Algoritmos para calcular la ruta más corta en la malla vial de la ciudad de Bogotá

Encontrar la ruta más corta entre dos puntos dentro de la ciudad de Bogotá. Se utilizó el algoritmo de A-Estrella, este garantiza que se encuentre el camino más corto entre dos nodos dados.

Leonardo Rodríguez C. 2005. Algoritmos para calcular la ruta más corta en la malla vial de la ciudad de Bogotá (junio, 2005).

2.3 Algoritmos para la determinación del circuito de menor longitud en un grafo

Determinar la menor longitud de un circuito en un grafo. Se utilizó el algoritmo Floyd y Warshall y se basa en lo que hace es buscar directamente las longitudes de los caminos mínimos entre todos los pares de vértices, construyendo lo que se denomina como matriz de distancias mínimas.

Rayco Hernández D. 2016. Algoritmos para la determinación del circuito de menor longitud en un grafo (julio, 2016).

2.4 Algoritmos para calcular la ruta más corta en la malla vial de la ciudad de Bogotá

Encontrar la ruta más corta entre dos puntos dentro de la ciudad de Bogotá. Para este se utilizó el algoritmo Dijkstra, el cual es el más simple para encontrar la ruta más corta entre un nodo y todos los demás nodos pertenecientes al grafo G.

Leonardo Rodríguez C. 2005. Algoritmos para calcular la ruta más corta en la malla vial de la ciudad de Bogotá (junio, 2005).

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) ¹y se descargó utilizando la API ²OSMnx 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 Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub ³.

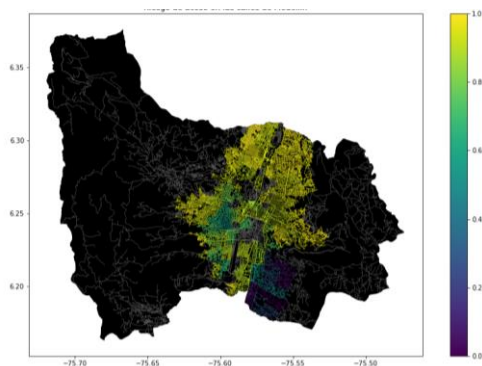


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

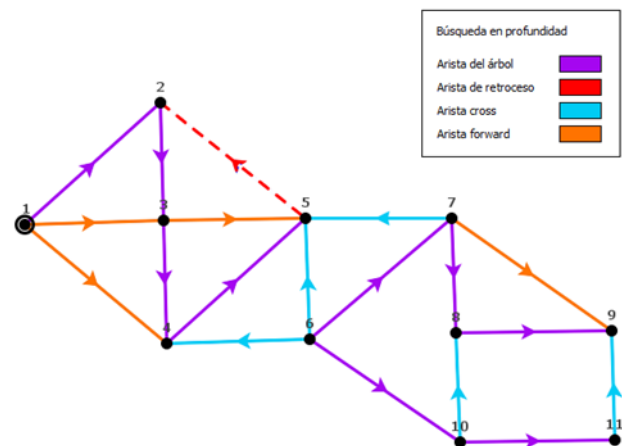
A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido

3.2.1 Algoritmo DFS (Depth First Search)

Es un algoritmo de búsqueda no informada utilizado para recorrer todos los nodos de un grafo o árbol (teoría de grafos) de manera ordenada, pero no uniforme. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto. Cuando ya no quedan más nodos que visitar en dicho camino, regresa (Back tracking), de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

Complejidad del algoritmo DFS: $O(V + E)$

Figura Vectorial



3.2.2 Algoritmo BFS (Breadth First Search)

Es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego

¹ <https://www.openstreetmap.org/>

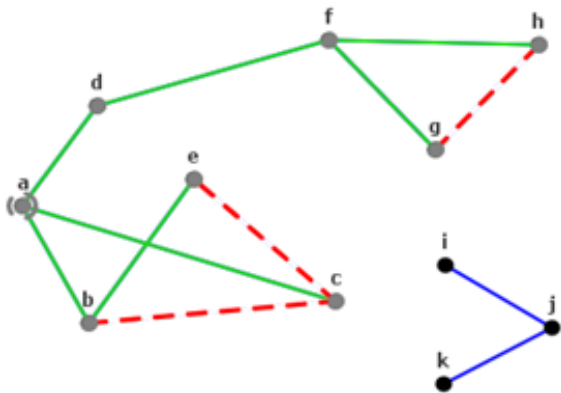
² <https://osmnx.readthedocs.io/>

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

explorar todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

Complejidad del algoritmo BFS: $O(|V| + |E|)$

Figura vectorial

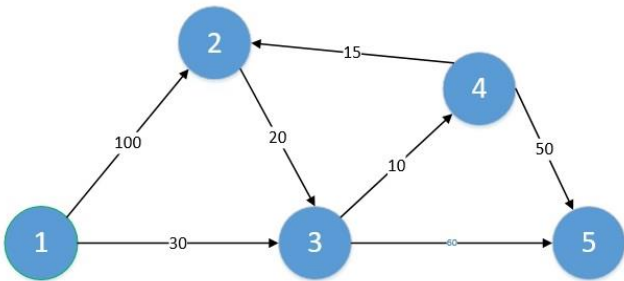


3.2.3 Algoritmo Dijkstra

Es un algoritmo para la determinación del camino más corto, dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista.

Complejidad del algoritmo Dijkstra: $O()$

Figura vectorial

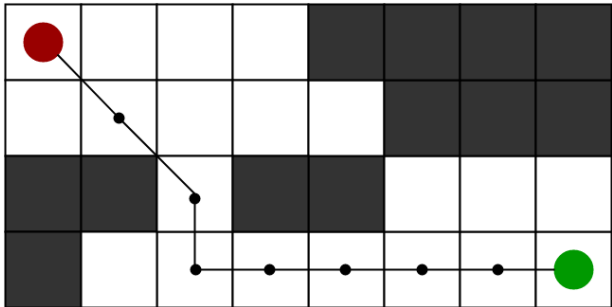


3.2.4 Algoritmo A* (A-Estrella)

Es un algoritmo que encuentra, siempre y cuando se cumplan unas determinadas condiciones, el camino de menor coste entre un nodo origen y uno objetivo.

Complejidad del algoritmo A*(A-Estrella): $O(2^n)$

Figura vectorial



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.

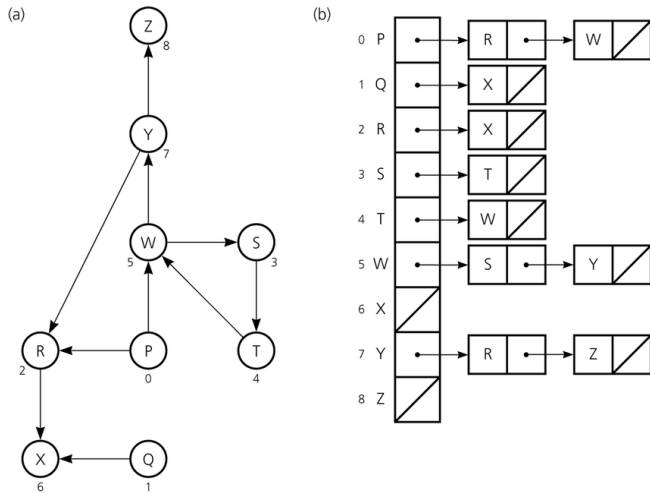
4.1 Estructuras de datos

La estructura de datos utilizada para implementar el algoritmo fue la de un diccionario de pandas.

La siguiente figura explica su funcionamiento.

	Name	Amount
0	Waldon Astling	1.83
1	Catherine MacTerlagh	0.15
2	Gusty Wondraschek	9.19
3	Lois Vaan	1.28
4	Baird Eberts	0.82
5	Amalia Flieg	2.88
6	Leontine Wildbore	9.44
7	Rikki Chasteney	7.01
8	Augustine Papierz	0.22
9	Maynord Lawrance	0.33

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



4.2 Algoritmos

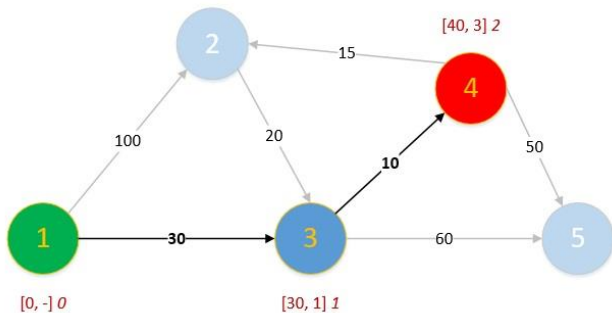
En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de acoso r . El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia d .

4.2.1 Primer algoritmo

El algoritmo de Dijkstra hace uso y define etiquetas a partir del nodo origen y para cada uno de los nodos subsiguientes. Estas etiquetas contienen información relacionada con un valor acumulado del tamaño de los arcos y con la procedencia más próxima de la ruta.

Las etiquetas corresponden a los nodos, no a los arcos. En el algoritmo de Dijkstra, estas etiquetas son temporales y permanentes. Las etiquetas temporales son aquellas que son susceptibles de modificarse mientras exista la posibilidad de hallar para sí, una ruta más corta; de lo contrario, dicha etiqueta pasa a ser permanente.

El algoritmo se ejemplifica en la siguiente figura.



4.4 Análisis de la complejidad de los algoritmos

Análisis, para el peor caso, utilizando la notación O .

La complejidad temporal se refiere al tiempo que puede tardar el algoritmo en dar el resultado.

Algoritmo	Complejidad temporal
Dijkstra	$O(n^2)$

Tabla 1: Complejidad temporal del algoritmo de Dijkstra, donde n representa los vértices.

Estructura de datos	Complejidad de la memoria
Dataframe (Pandas)	$O(n^2)$

Tabla 2: Complejidad de memoria del dataframe en Pandas, donde n es el número de aristas.

4.5 Criterios de diseño del algoritmo

Se decidió utilizar el algoritmo de Dijkstra porque es un algoritmo greedy, trabaja por etapas, y toma en cada etapa la mejor solución sin considerar consecuencias futuras, además, el óptimo encontrado en una etapa puede modificarse posteriormente si surge una solución mejor.

Por otra parte, decidimos trabajar con la estructura de datos de Pandas, ya que es una librería de código abierto que surgió para hacer más fácil todo el ciclo de vida de cualquier dato, desde que este es generado hasta que es aprovechado. Y permite, de forma fácil e intuitiva realizar operaciones capaces de gestionar y manipular cualquier tipo de información sin importar el formato, y sobre todo de una forma rápida y eficaz.

De tal forma que, con la implementación en el programa de este algoritmo y de esta estructura de datos, se dio un resultado óptimo para este problema.

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	6,700 m	0.84
Universidad de Antioquia	Universidad Nacional	1,900 m	0.83
Universidad Nacional	Universidad Luis Amigó	9,600 m	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	0.4	6,800
Universidad de Antioquia	Universidad Nacional	0.2	2,000
Universidad Nacional	Universidad Luis Amigó	0.2	9,600

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.

Los tiempos de ejecución medios fueron de:

	Tiempos medios de ejecución (s)
Universidad EAFIT a Universidad de Medellín	120.4 s
De la Universidad de Antioquia a la Universidad Nacional	81.3 s
De la Universidad Nacional a la Universidad Luis Amigó	167.8 s

Tabla 4: Tiempos de ejecución del nombre del *algoritmo* (Por favor, escriba el nombre del algoritmo, por ejemplo, DFS, BFS, un A* modificado) para las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

Si, son significativamente diferentes debido ya que para que haya menos riesgo de acoso sexual se deben tomar rutas alternas que no siempre se acercan a la ruta más corta. Para la ciudad, una aplicación o una red que permita identificar estas rutas serían de gran ayuda ya que nos permite verificar que tan riesgosa podría ser alguna ruta y aumentaría la seguridad de las personas. Y, a decir verdad, si son razonables los tiempos de ejecución, ya que, si se necesitara esta aplicación en una ocasión real, se podría encontrar la ruta óptima para llegar a su destino.

6.1 Trabajos futuros

Me gustaría mejorar, por lo menos un poco, los tiempos de ejecución, tratando de ayudarme de otras librerías o de apoyarme de otros lenguajes de programación ¿Continuará este proyecto trabajando en la optimización? Si, continuará haciéndolo, para así reducir desconfianzas y aumentar la seguridad de las personas que utilicen el programa ¿Aprendizaje automático? También nos gustaría implementar algo que ayude al programa a actualizar las rutas de forma automática para así facilitar a las personas su uso cotidiano.

AGRADECIMIENTOS

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. Rayco Hernández D. 2016. Algoritmos para la determinación del circuito de menor longitud en un grafo (julio, 2016).
2. Leonardo Rodríguez C. 2005. Algoritmos para calcular la ruta más corta en la malla vial de la ciudad de Bogotá (junio, 2005).