

ENRUTAMIENTO SEGURO Y RAPIDO PARA LOS PEATONES

Isabela Osorio
Universidad Eafit
Colombia
iosoriob@eafit.edu.co

Karol Cuello
Universidad Eafit
Colombia
kvcuellor@eafit.edu.co

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

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

RESUMEN

En Colombia los peatones corren un gran riesgo de acoso sexual al andar por ciertas zonas, siendo víctimas de diferentes actos no consentidos como manoseo, insultos, burlas discriminatorias, etc. por razón de su género u orientación sexual; y Medellín no es la excepción para todos estos fenómenos. Estos acosos no solo incomodan a sus víctimas, traen otras consecuencias bastante graves a la sociedad, tales que impiden el libre desarrollo de la vida íntegra, estando ligados a otros crímenes como el hurto, la trata de personas e incluso el homicidio, es por esto que resulta indispensable poner manos a la obra y encontrar una solución para este problema, que garantice la seguridad de los peatones.

El algoritmo es dijkstra, utilizado en más de 50 países por su utilidad y rapidez. Su complejidad es $O(n^2)$, utiliza preferencialmente programación lineal, y su función es hallar el camino más corto. De este trabajo podemos concluir que, a pesar de los conflictos y peligros, siempre hay una salida, como los algoritmos, el aprendizaje con grafos y análisis más precisos y extendidos, para hallar soluciones a este tipo de problemas, se debe tener una investigación detallada y sólida, y todo esto pudimos implementarlo a medida que pasaba el curso.

Palabras clave

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

1. INTRODUCCIÓN

El acoso sexual callejero no es algo que nació hace poco, sino que viene ocurriendo desde tiempo atrás, y alrededor de todo el mundo, por ejemplo, en India se han hecho varios algoritmos para reducir el riesgo de acoso callejero a las mujeres al tomar rutas teniendo en cuenta mapas de calor, e implementando integración y análisis de datos. Los peatones, sean hombres o mujeres, necesitan una ruta que los lleve de manera segura pero también rápida a su lugar de destino, pues si bien es cierto que la seguridad es primero, no es conveniente despreciar el tiempo en la toma de decisiones.

1.1. Problema

Este es un problema que no afecta solo a los peatones, debemos observar que es un problema social que nos incluye a todos y es una amenaza para ti y para mí, pero también para nuestros queridos. El acoso sexual trae consigo muchas consecuencias negativas, como: delincuencia, robos, atracos,

etc.; miedo en las víctimas, lo que impide su libre desarrollo en la sociedad, así como también su desarrollo personal, incluso al punto de inducir el suicidio; atentados contra la vida, ya que, muchos acosos son más que eso, pueden incluir abusos y agresiones físicas, e incluso el asesinato; alto riesgo en niños y adolescentes, ¿a quién le gustaría que sus hijos salieran aun sabiendo que corren peligro?, los pequeños son el futuro de la sociedad, por lo que está en nuestras manos garantizar un futuro seguro y sano para ellos.

1.2 Solución

Ya pudimos ver a fondo el problema, junto con sus causas y posibles alternativas de solución. Ahora necesitamos concretar cuál será la solución, por lo que hemos escogido el algoritmo de Dijkstra en contra del acoso sexual. Ya que el Algoritmo de Dijkstra permite encontrar el camino más corto entre dos vértices de un grafo. Yendo desde la definición de un grafo, como la representación gráfica de un conjunto de nodos o vértices unidos por enlaces llamados aristas o arcos, y esto es muy importante para la solución del problema.

Además, 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.

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 RUTAS CONSCIENTES

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

2.1 Seguridad para los turistas

En primer lugar, tenemos rutas conscientes de la seguridad para turistas motorizados basadas en datos abiertos y VGI.

Los propietarios: Andreas Keler y Jean Damascene Mazimpaka

El problema solucionado fue el generar rutas cocientes, hacia personas vulnerables o en peligro, principalmente sobre sus destinos y los riesgos que pueden correr a medida que avanzan.

Usando VGI y Open Data.

La geriatría geriátrica integral (VGI) se define como “un proceso diagnóstico evolutivo multidimensional e interdisciplinario, diseñado para identificar y cuantificar los problemas físicos, funcionales, psicológicos y sociales que puedan presentar los adultos mayores, con el propósito de desarrollar un plan individualizado de atención integral, lo que permite una acción preventiva, terapéutica, rehabilitadora y de seguimiento, con la óptima utilización de los recursos, para alcanzar el mayor grado de autonomía y mejorar su calidad de vida”.

Multidimensional, porque incluye la evaluación de varias dimensiones que determinan el estado de salud (funcional, afectivo, cognitivo, social y físico), e interdisciplinario, porque requiere el aporte de varios profesionales para la evaluación y manejo de los problemas de los adultos mayores. El equipo básico de salud responsable de la evaluación y cuidado de los ancianos debe estar integrado por un médico, una enfermera, un profesional de servicio social y un fisioterapeuta.

y, por último, Open Data o datos abiertos es una iniciativa global que busca poner a disposición de la sociedad en su conjunto los datos recogidos, producidos y difundidos por las administraciones públicas de forma abierta y en formatos reutilizables.

El sector empresarial también se suma a esta iniciativa ya que la integración de ambos tipos de datos (públicos y privados) permite una explotación con resultados nunca antes alcanzados.

Destacamos también la iniciativa Open Data Science, cuyo principal objetivo es que los datos y herramientas obtenidos para la investigación se publiquen libremente para que puedan ser reutilizados por cualquier persona u organización.

Desde la Cátedra impulsamos actividades de investigación, formación y divulgación de datos abiertos. En los próximos años, la reutilización de datos abiertos experimentará un importante crecimiento y lo veremos en diferentes entornos que nos rodean.

Este documento quiere tratar el tema de la inseguridad en los recorridos que más realizan las personas, generalmente en zonas urbanas, utilizando información voluntaria (VGI), datos históricos, antecedentes penales, etc. presentando conceptos de mapeo, ruteo, SIG.

Los puntos calientes se consideran obstáculos de enrutamiento.

Los conductores de vehículos que no conocen las zonas peligrosas de la ciudad pueden utilizar la ruta menos peligrosa, que se basa en nuestro índice de seguridad calculado. Finalmente, discutimos la efectividad de nuestro método y consideramos extensiones adicionales utilizando geodatos disponibles gratuitamente.

Sobre el mapeo del crimen, debemos decir que es una herramienta cartográfica utilizada por los analistas de las fuerzas de seguridad para mapear, visualizar y analizar patrones delictivos en un lugar determinado, ya sea con fines estadísticos, así como para predecir y prevenir futuras acciones delictivas.

El número total de puntos inspeccionados en el área de investigación y los pesos correspondientes. Con pesos fijos seleccionados para los tres tipos de puntos de datos basados en los supuestos anteriores sobre el nivel de seguridad asociado a los tres tipos de puntos y después de realizar una serie de pruebas.

2.2 Análisis de datos para rutas seguras

El problema a solucionar en este trabajo es la inseguridad en uno de los países más violentos del mundo para las mujeres, India, y las maneras en las que se pueden resolver.

Y en cuanto al algoritmo se estará utilizando la cola de prioridad (por ejemplo, un montón binario o un árbol binario equilibrado). Por otro lado, si se usa un montón de Fibonacci, sería $O(|V| \log |V| + |A|)$.

Un sistema de integración y análisis de datos para la planificación segura de rutas

Propietarios: Aryan Gupta y Bhavye Kethan

El artículo trata principalmente del peligro que atraviesan las mujeres, siendo el sexual uno de los más horribles y frecuentes, se reportan más de 67.000 casos desde el 2013 al 2018.

El acoso sexual es cualquier comportamiento, verbal o físico, de naturaleza sexual que tenga el propósito o el efecto de atentar contra la dignidad de una persona, en particular cuando se cree un entorno intimidatorio, degradante u ofensivo.

Es ilegal acosar a una persona (empleado o solicitante de empleo) debido al sexo de esa persona. El acoso puede incluir "acoso sexual" o insinuaciones sexuales no deseadas, solicitudes de favores sexuales y otro tipo de acoso verbal o físico de naturaleza sexual.

El acoso no tiene que ser exclusivamente de naturaleza sexual, sino que también puede incluir comentarios ofensivos sobre el género de una persona. Por ejemplo, es ilegal acosar a una mujer haciendo comentarios ofensivos sobre las mujeres en general.

Tanto la víctima como el acosador pueden ser una mujer o un hombre, y la víctima y el acosador pueden ser del mismo sexo.

Si bien la ley no prohíbe las bromas simples, los comentarios improvisados o los incidentes únicos que no son muy graves, el acoso es ilegal cuando es tan frecuente o grave que crea un ambiente de trabajo hostil u ofensivo, o cuando da como resultado una decisión negativa de empleo (p. ej., que la víctima sea despedida o colocada en un puesto inferior).

La persona que acosa puede ser el supervisor de la víctima, un supervisor en otra área, un compañero de trabajo o alguien que no trabaja para el empleador, como un cliente.

Por eso, queremos encontrar una alternativa a esto, y sobre todo ayudar a las mujeres, que son las que más sufren este acoso, por tanto, una forma de reducirlo es encontrar los caminos y rutas más adecuadas y seguras. Así, al preguntar a varias mujeres qué les haría sentir más seguras en este tipo de situaciones, tomando como referencia la India...

Pero, ¿por qué India?, según un artículo de "EL PAÍS": "Los riesgos a los que se enfrentan las mujeres en India van más allá de la violencia sexual, según el estudio de la Fundación Thompson Reuters, que considera que el país asiático es también el más peligroso en Gupta relación con la trata de personas, incluida la esclavitud sexual y doméstica. En consecuencia, la fundación subraya el impacto económico de la violencia machista: "Al menos 20 millones de mujeres han dejado sus trabajos en la tercera economía más grande de Asia desde 2005 debido, en parte, al maltrato [sufren]". Entonces podemos tomar como referencia a un indio porque es el peor país en el que puede estar una mujer.

Las respuestas de las mujeres sirvieron para hacer comparaciones y tratar de obtener nuevas soluciones o implementar sistemas de seguridad. El problema con esto es que no considera nuevas áreas que podrían ser puntos críticos.

El artículo nos habla de "Influencias en la seguridad de un viaje" y dice que hay factores bastante predominantes, los cuales son:

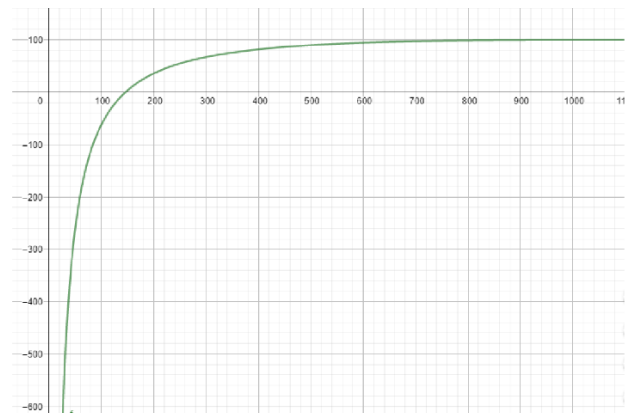
Gente de la zona.

Algo importante a considerar es cuál es el tamaño de las personas en un lugar a la vez, para que, si una mujer es víctima de violencia sexual, alguien pueda escucharla y acudir en su ayuda. La cuestión es que una persona a 500 m escucha esto y probablemente huirá o se retirará, a 100 m podría detectar a la víctima. Entonces un punto adecuado sería el intermedio, es decir unos 200m porque así en vez de salir irán a ver y podrían detectar a la víctima. Lo malo de esto es que, si tienes una multitud, sería difícil capturar a una persona que se escapa y se dispersa entre la gente...

Iluminación

La iluminación es importante, ya que este tipo de cosas pueden ocurrir en la noche, ya que la víctima no podría verla y la causa de esto podría escapar más fácilmente, por lo que esta es una pieza clave. Entonces se construyó un gráfico en

el que se mostraba la iluminación adecuada, y se limitaba que la luz de las calles sería la adecuada y se graficó:



Graphic of the article. Lightning.

Número de casos

El lugar presenta casos demasiado altos para saber exactamente cuántos hay. Se demostró que a pesar de que hay gente cerca, y buena iluminación, los casos no bajan y siguen aumentando, por lo que debe haber más factores que influyan en las comunidades.

Cuando tomamos situaciones simuladas para esos factores, tomamos el número de casos en esa región como 0, pero se descubrió que esas ubicaciones no eran seguras, por lo que se manejarán por separado...

Un factor muy influyente es el clima, ya que puede ser seguro durante el día, pero no por la noche.

La planificación de rutas

En este artículo se utilizó el algoritmo de Dijkstra, también llamado algoritmo de la ruta más corta, que es un algoritmo para determinar la ruta más corta, dado un vértice de origen, al resto de los vértices en un gráfico que tiene pesos en cada borde.

La idea subyacente en este algoritmo consiste en explorar todos los caminos más cortos que parten del vértice de origen y que conducen a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen hasta el resto de vértices que componen el grafo, el algoritmo se detiene. Es una especialización de búsqueda de costo uniforme y, como tal, no funciona en gráficos con bordes de costo negativos (al elegir siempre el nodo con la distancia más pequeña, los nodos que en iteraciones futuras reducirían el costo total pueden ser excluidos de la búsqueda). del camino al pasar por una arista con coste negativo)

Orden de complejidad del algoritmo:

$O(|V|^2 + |A|) = O(|V|^2)$, sin usar cola de prioridad, $O((|A| + |V|) \log |V|) = O(|A| \log |V|)$ utilizando la cola de prioridad (por ejemplo, un montón binario o un árbol binario equilibrado).

Por otro lado, si se usa un montón de Fibonacci, sería $O(|V| \log |V| + |A|)$.

La complejidad computacional del algoritmo de Dijkstra se puede calcular contando las operaciones realizadas:

- El algoritmo consta de $n-1$ iteraciones, como máximo. En cada iteración, se agrega un vértice al conjunto distinguido.
- En cada iteración se identifica el vértice con la etiqueta más pequeña entre los que no están en S_k . El número de estas operaciones está acotado por $n-1$.
- Además, se realiza una suma y una comparación para actualizar la etiqueta de cada uno de los vértices que no están en S_k .

Luego, en cada iteración, se realizan como máximo 2 ($n-1$) operaciones.

Después:

Teorema: el algoritmo de Dijkstra realiza operaciones $O(n^2)$ (sumas y comparaciones) para determinar la longitud del camino más corto entre dos vértices de un grafo simple, conectado, ponderado no dirigido con n vértices.

2.3 Viaje seguro – Con datos sobre delitos

En este problema se quiere resolver la inseguridad de la mujer en sus rutas. Y el preprocesamiento será lo ideal para resolver estos problemas

Ruta: la caja fuerte: un modelo robusto para la ruta más segura, utilizando datos sobre delitos y accidentes

Propietarios: Venkatesh Gauri Shankar y chaurasia Sandeep

Teniendo en cuenta los artículos anteriores, esta habla de la inseguridad que hay todos los días sobre las mujeres. Esto nos da a entender cosas sobre rutas seguras.

Dice que existen aplicaciones móviles para encontrar el camino más seguro, pero no solo obtener el camino más corto y rápido, sino que tenemos la ruta más corta posible, pero también la más segura, utilizando los antecedentes penales de las rutas y la opinión popular, las aplicaciones. asigne caminos teniendo en cuenta el crimen, pero puede ser difícil optimizarlos para un objetivo secundario.

Para el preprocesamiento de datos se realizó lo siguiente: el conjunto de delitos y accidentes, de estos dos se eliminaron las columnas que no servían, si existían las filas que tenían nulo o nan se eliminaron, se cambiaron los nombres para mejor comprensión, etc.

Ahora sigue el diseño del modelo: agrupamiento anidado, el agrupamiento de K-media se realizó en uno de los agrupamientos que se formarían usando el método coso, los centroides de los agrupamientos recién formados se encontraron usando K-medias.

Metodología

El artículo lo expresa con un diagrama que explicaré a continuación:

Al iniciar la aplicación se inician los mapas, se cargan los datos, se hace el preprocesamiento explicado anteriormente, se detectan rutas peligrosas con los algoritmos, se realiza el agrupamiento anidado (Nesting (llamado nesting en inglés) es la práctica de incorporar llamadas (llamadas) a funciones o procedimientos (unos) dentro de otros, al incluir varios niveles de paréntesis.), luego se obtendrá la puntuación promedio de estos grupos y finalmente se obtendrá el resultado.

Los conjuntos de datos utilizados en esta solución son el conjunto de datos de Accidentes y el conjunto de datos de Arrestos. Y usaron un conjunto de datos de la ciudad de Nueva York como ejemplo.

Lo siguiente es iniciar el APL de mapas de Google y luego programarlo, se utiliza para realizar varias operaciones.

Luego, los datos del bloqueo se cargan y van al sitio web NYC OpenData, que se actualiza en tiempo real.

Este conjunto de datos de arresto brinda detalles del crimen, es decir, su latitud, longitud, ubicación, etc.

Después de esto sigue el procesamiento de datos, que es el paso más importante de todos, ya que se realiza para eliminar los datos faltantes y no normales.

Luego se etiquetan y se le da una puntuación al accidente. el conjunto de detenciones se utiliza para averiguar qué delito ocurrió.

Finalmente, la ruta más segura, el paso final de este modelo. Este paso va con condiciones, si solo hay una ruta con menos riesgo, se toma como la más segura, de lo contrario, si más de una ruta tiene puntajes positivos, se realizará el análisis de distancia.

2.4 Ruta más corta

En este problema tenemos el acoso sexual y su a través de un algoritmo de búsqueda de rutas usando la búsqueda cercana

Propietarios: Casos de Estudio y Proyectos, Seguridad Pública

Este artículo utiliza mapas de calor sobre la violencia contra las mujeres, específicamente el acoso sexual. Calculando el riesgo asociado a tomar cada ruta y buscando la más segura a través del análisis de mapas de calor para prevenir casos de acoso sexual, es que cuando hay momentos de estrés y se necesita una decisión rápida esas habilidades cognitivas son necesarias. Así que pensaron mejor en el problema y dijeron "¿Cómo podemos determinar la seguridad de las rutas?"

Nuevamente, trabajando desde soluciones simples hasta soluciones más complejas, podemos:

Organizar los datos de puntajes con mayor riesgo asociado, determinar la seguridad en las rutas, si la red es buena o no, si el historial delictivo es muy largo.

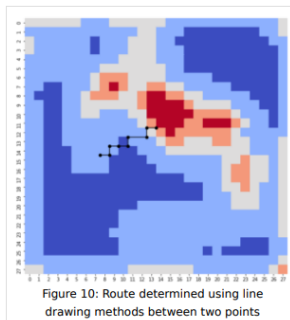
La primera solución no es compleja ya que tenemos coordenadas y la cuadrícula más cercana, y la que se supone que tiene menos riesgo es el camino más corto, la distancia euclidiana. El problema es que podría verse afectado por obstáculos físicos o también por un lugar demasiado arriesgado para que la gente pasara, por lo que recurrieron al método de dibujo lineal de Bresenham.

El algoritmo de línea de Bresenham es un algoritmo de dibujo de línea que determina los puntos de un ráster n-dimensional que debe seleccionarse para formar una aproximación cercana a una línea recta entre dos puntos.

Aplicamos esta cobertura de cuadrícula, calculando puntuaciones medias de las rutas, siempre primando la seguridad sobre la velocidad, con mapas de calor.

Análisis para prevenir casos de acoso sexual.

Este método se aplicó al mapa de calor para ilustrar cómo una ruta podría verse como:



3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de caminos que reducen tanto la distancia como el riesgo de 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 API² OSMnx de Python. El mapa incluye (1) la 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 obtenidas de los metadatos proporcionados por OSM.

Para este proyecto, se calculó una 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 GitHub³.

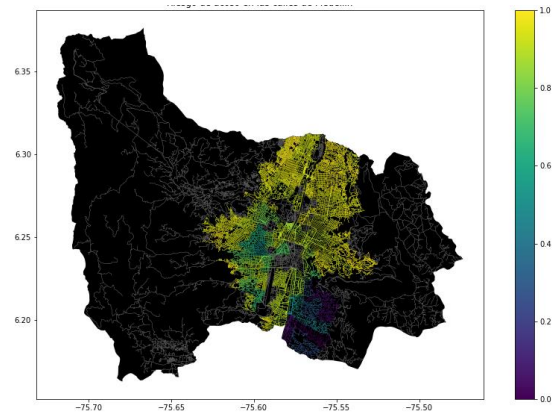


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenidas de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de caminos que reducen el riesgo de acoso sexual callejero y distancia

A continuación, presentamos diferentes algoritmos utilizados para un camino que reduce tanto el acoso sexual callejero como la distancia.

3.2.1 Shortest Path using Backtracking

Este algoritmo resuelve un problema similar, el cual consiste en encontrar el camino más corto de $1s$ dada una matriz de unos (1) y ceros (0), una posición inicial y una posición objetivo. Este algoritmo utiliza la recursión para encontrar, si existe, el camino más óptimo. En la matriz solo podemos desplazarnos vertical u horizontalmente, lo que aplicado a nuestro caso serían las carreras y las calles. Primero se comprueba que el movimiento sea válido (sea arriba, abajo, derecha e izquierda) y que la celda no haya sido recorrida en esa misma ruta, si no hay movimiento válido se retrocede a la anterior posición para elegir otro camino distinto, hasta agotar todas las posibilidades y retornar el camino más corto. La complejidad de este algoritmo corresponde al número de nodos (intersección entre calles y/o carreras, más puntos donde se terminan los caminos), ya que, a la final recorre todos los nodos, es decir, $O(k)$ con $k=\#total$ de nodos. O en su defecto, dada una matriz a la que se aplica este algoritmo,

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

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

su complejidad correspondería al tamaño de dicha matriz, es decir, $O(m \cdot n)$ siendo $n = \#$ de filas y $m = \#$ de columnas.

Diagrama de flujo:

https://lucid.app/lucidchart/15da7b40-7941-4a20-81dd-5eeb76caf6d5/edit?invitationId=inv_6abf6e71-27f2-42e1-ab3d-73d9dd504647#

3.2.2 Algoritmo de Dijkstra

El Algoritmo de Dijkstra, también llamado Algoritmo de caminos mínimos, se clasifica dentro de los algoritmos de búsqueda. Su objetivo, es determinar la ruta más corta, desde el nodo o vértice origen al resto de vértices en un grafo que tiene pesos en cada arista; visto aplicado en nuestro problema, el peso de la arista estaría dado por un número que represente tanto el acoso como la distancia. El algoritmo recorre todos los caminos posibles a un nodo (lugar de destino) y guarda el que tenga menor peso de todos (el mínimo de acoso sexual y distancia posible). En Python los grafos se trabajan en forma de matrices de adyacencia, y en este caso el algoritmo de Dijkstra tiene una complejidad $O(n^2)$, donde “ n ” es el número de vértices, pues, en el peor de los casos, se encuentra el camino con menos peso desde un nodo origen a todos los demás nodos del grafo. Dato curioso: Fue diseñado por el holandés Edsger Wybe Dijkstra en 1959.

3.2.3 Algoritmo A*

Este algoritmo es la forma avanzada del algoritmo BFS (Breadth-first search), que busca primero la ruta más corta que las rutas más largas. Es una solución completa y óptima para resolver problemas de ruta y cuadrícula, utilizando inteligencia artificial. El algoritmo de búsqueda A* utiliza el costo de la ruta heurística, el costo del punto de partida y el punto final, además es el algoritmo más conocido y usado para resolver problemas de encontrar el camino más corto. Este algoritmo trabaja teniendo en cuenta 3 “valores” importante, que son, $g(n)$: La ruta de costo real desde el nodo de inicio hasta el nodo actual- $h(n)$: La ruta de costo real desde el nodo actual hasta el nodo objetivo- $f(n)$: La ruta de costo real desde el nodo de inicio hasta el nodo de meta. En este algoritmo se crean dos listas vacías Open y Closed y el nodo origen se coloca en Open, luego, se pasa de Open a Closed y se compara si el nodoActual con menor valor de f es igual al nodoObjetivo para retornar g del nodoActual. Si no es así, se procede con los nodos adyacentes del nodoActual así: si nodoAdyacente está en la lista Closed se repite el proceso del nodoActual teniendo en cuenta los valores de g, h y f para dicho nodo, y si está en Open, se compara si g del nodoActual es menor que g del que está en Open y finalmente se agrega a la lista Open. La complejidad computacional del algoritmo está íntimamente relacionada con la calidad de la heurística que se utilice en el problema. En el caso peor, con una heurística de pésima calidad, la

complejidad será exponencial $O(2^n)$, mientras que, en el caso mejor, con una buena heurística, el algoritmo se ejecutará en tiempo lineal ($O(n)$)

Diagrama de flujo:

https://lucid.app/lucidchart/4505587d-6e8a-44ba-98b2-782a9b25eec3/edit?invitationId=inv_4ac32de1-92b8-4a96-97a9-26e155c5e442#

3.2.4 Algoritmo de Bellman Ford

El algoritmo de Bellman Ford nos ayuda a encontrar el camino más corto desde un vértice hasta todos los demás vértices de un grafo ponderado. Es similar al algoritmo de Dijkstra, pero puede trabajar con gráficos en los que los bordes pueden tener pesos negativos (lo que resulta muy conveniente en la solución de algunos problemas de finanzas o química). El algoritmo de Bellman Ford funciona sobreestimando la longitud del camino desde el vértice inicial hasta todos los demás vértices. Luego relaja iterativamente esas estimaciones al encontrar nuevos caminos que son más cortos que los caminos sobreestimados anteriormente; Al hacer esto repetidamente para todos los vértices, podemos garantizar que el resultado esté optimizado. La complejidad de este algoritmo depende del número de aristas multiplicado por el número de vértices $O(m \cdot n)$, siendo m =número de aristas y n =número de nodos, que es mucho más alta que el de Dijkstra.

Diagrama de flujo: https://lucid.app/lucidchart/3fab6703-d32a-4a50-b87c-2eb52a9f43be/edit?invitationId=inv_42fef99d-5bc9-43d6-9b7f-8447230119a8#

4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

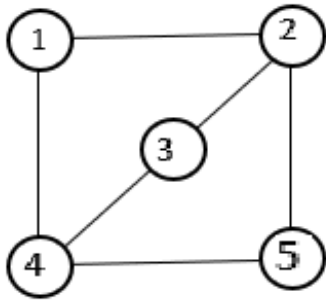
A continuación, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en Github⁴.

4.1 Estructuras de datos

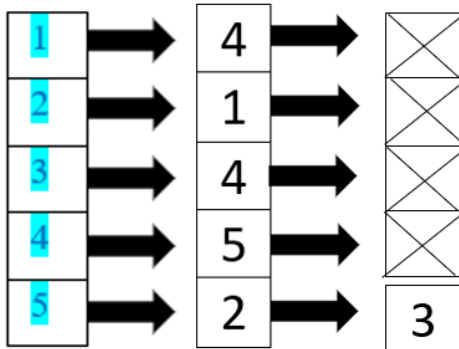
La estructura que se utilizó es la lista de adyacencia. En teoría de grafos, una lista de adyacencia es una representación de todas las aristas o arcos de un grafo mediante una lista. Si el grafo es no dirigido, cada entrada es un conjunto o multiconjunto de dos vértices conteniendo los dos extremos de la arista correspondiente. Si el grafo es dirigido, cada entrada es una tupla de dos nodos, uno denotando el nodo fuente y el otro denotando el nodo destino del arco correspondiente. Típicamente, las listas de adyacentes no son ordenadas.

⁴<https://github.com/kvcuellor/Proyecto-Datos-y-Algoritmos.git>

Un ejemplo simple sería el siguiente realizado en Word:



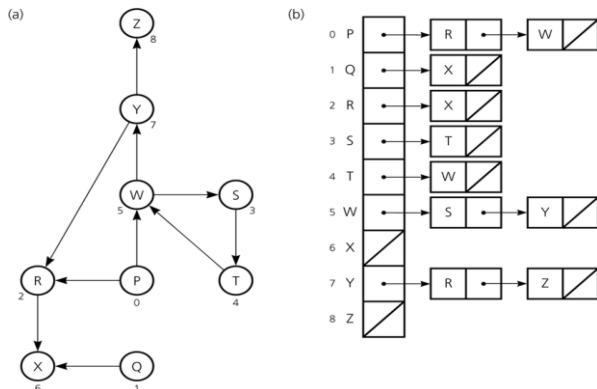
A partir de eso se hace esta lista realizada en Word:



Se asocia con cada vértice ‘i’ del grafo una lista que contenga todos aquellos vértices ‘j’ que sean adyacentes a él. Así que solo almacena memoria para arcos adyacentes a ‘i’. Por tanto, el grafo se representa con un vector de n componentes, así que cada componente va a ser una lista de adyacencia correspondida a cada vértice. Si el grafo es etiquetado, habrá que añadir un segundo campo para mostrar el valor de la etiqueta.

Y podemos tomar esto como caminos, por ejemplo, con el 5, este se dirige a 2 y a 3, por lo tanto, si necesitamos el camino más corto... ¿Cuál debemos escoger? Claramente el 2.

Figura 2



Este ejemplo es exactamente lo mismo, pero con mayor complejidad. Podemos observar que este tiene además de números letras, por lo que cada vértice tiene un nombre, con su respectivo peso. ¿La calle ‘T’ puede ir a ‘S’? No directamente, pero si por otros medios, la calle ‘W’, entre otros detalles. Por lo tanto, tenemos mayor complejidad.

4.2 Algoritmos

En este trabajo, proponemos un algoritmo para un camino que minimiza tanto la distancia como el riesgo de acoso sexual callejero.

4.2.1 Algoritmo para un camino que reduce tanto la distancia como el riesgo de acoso sexual callejero

Para alcanzar el objetivo de este proyecto, teniendo en cuenta las diferentes opciones de algoritmos sencillos y prácticos, elegimos el algoritmo de Dijkstra, el cual nos permite por medio de un grafo (en este caso hecho con listas de adyacencia) construir un “mapa de Medellín” que tenga los datos que necesitamos, tales como: puntos de partida y puntos de llegada (que se hacen de acuerdo a la dirección del lugar), dirección (en una sola dirección o en ambas), distancia y riesgo de acoso (los cuales se combinan para generar la variable V , la cual representa los pesos de las aristas y es la que buscamos minimizar); en forma de listas para cada nodo, y por medio del algoritmo hallar el camino más corto y seguro.

En nuestro algoritmo usamos Grafo, pandas, deque y math (infinito), para poder implementarlo de la manera correcta y mejor posible dadas nuestras capacidades. Con estas herramientas podemos tratar los datos anteriormente mencionados; a V le damos valor de infinito inicialmente, y luego, sabiendo el riesgo de acoso y la distancia entre los nodos, se calcula un nuevo valor para la variable y se reemplaza el anterior si el actual es menor, se recorren todos los nodos no visitados, desde todos los caminos posibles, y de esta forma se garantiza que se encuentra el camino más óptimo entre todos.

El algoritmo se ejemplifica en la Figura 3.

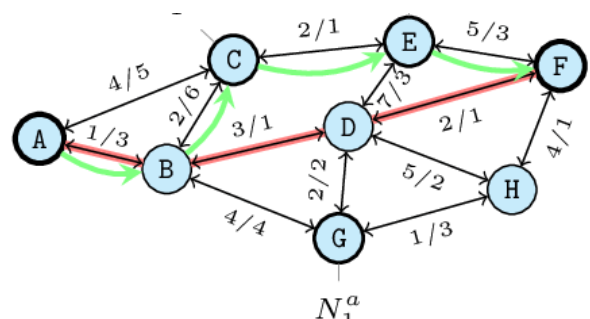
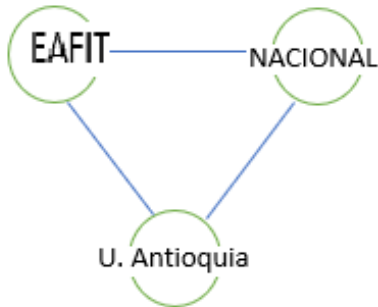


Figura 3: Cálculo de un camino que reduce tanto la distancia como el riesgo de acoso. Se recorre el grafo y se retorna el camino que corresponde a la mejor solución del problema.

4.2.2 Cálculo de otros dos caminos para reducir tanto la distancia como el riesgo de acoso sexual callejero.

Se hará el cálculo con otros dos caminos, los cuales serán de EAFIT a la universidad Nacional, en los que reduciremos la distancia como el acoso sexual



Ya que en la segunda entrega utilizamos la U de Antioquia como referencia decidimos implementarle a en ese ejemplo simple de dijkstra, aunque en la imagen no muestra los pesos, utilizamos la distancia, por ejemplo, de EAFIT a la Nacional, son 80 kilómetros buscando el camino más corto. Por lo que hay menos posibilidad de abuso sexual. El algoritmo se ejemplifica en la Figura 4.



Figura 4

Podemos adjuntar esta imagen en caso de ser un recorrido más corto.

Pero si hablamos de algo más extenso, podemos adjuntar esta imagen la cual muestra 4 caminos distintos a recorrer en Medellín. Hecha por “ViaMichelin”.



4.3 Análisis de la complejidad del algoritmo

La complejidad temporal en el peor de los casos para el Algoritmo de Dijkstra es equivalente al cuadrado del total del número de nodos que debe visitar. En el peor de los casos, debe visitarlos a todos y, todos se relacionan con todos, por lo que el número de iteraciones y la complejidad temporal serían de V^2 y la complejidad en memoria sería $O(V)$ pues es el número total de vértices.

| Algoritmo | Complejidad temporal |
|-----------|----------------------|
| Dijkstra | $O(V^2)$ |

Tabla 1: Complejidad temporal del algoritmo de Dijkstra, donde V es el número de vértices o nodos del grafo, es decir, todos los puntos donde los caminos se cruzan o terminan.

| Estructura de datos | Complejidad de la memoria |
|---------------------|---------------------------|
| Grafo | $O(V)$ |

Tabla 2: Complejidad de memoria del grafo, donde V es el número de nodos o vértices del mismo.

4.4 Criterios de diseño del algoritmo

Ya que no trabajamos con valores de peso negativos, el algoritmo dijkstra es el más adecuado para este problema de acoso sexual que exige el camino más corto. Además, sabemos que es uno de los más eficientes, pero ya hablamos mucho de esto anteriormente, su complejidad por la forma en la que lo implementamos es $O(V^2)$. en caso de querer ir a un lugar y no saber qué camino tomar, el algoritmo se tardaría **alrededor de 1 minuto** o en el peor de los casos **casi 90 segundos** en ejecutar, en cambio, el usuario tendría que buscar en internet o preguntar a conocidos y desconocidos, lo que tardaría en el peor de los casos **20 minutos**. Es decir, el tiempo ahorrado fueron **18 minutos con 30 segundos**. Algo fundamental en el algoritmo dijkstra es iniciar todas las distancias en D con un valor infinito relativo pero **la x debe tomar el valor de 0**.

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre los tres caminos que reducen tanto la distancia como el riesgo de acoso sexual callejero.

5.1 Resultados del camino que reduce tanto la distancia como el riesgo de acoso sexual callejero

A continuación, presentamos los resultados obtenidos de *tres caminos que reducen tanto la distancia como el acoso*, en la Tabla 3.

| Origen | Destino | Distancia | Riesgo |
|--------|---------|-----------|--------|
| Eafit | UdeA | 8284.35 m | 0.9215 |
| Eafit | UdeA | 6184.01 m | 0.9482 |
| Eafit | UdeA | 7516.07 m | 0.9215 |

Tabla 3. Distancia en metros y riesgo de acoso sexual callejero (entre 0 y 1) para ir desde la Universidad EAFIT hasta la Universidad de Antioquia caminando.

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.

| Cálculo de v | Tiempos medios de ejecución (s) |
|------------------|---------------------------------|
| v = riesgo | 72 s |
| v = distancia | 83 s |
| v = d+(r*100000) | 70 s |

Tabla 4: Tiempos de ejecución del algoritmo de Dijkstra para cada uno de los tres caminos calculadores entre EAFIT y Universidad de Antioquia.

6. CONCLUSIONES

A lo largo del proyecto pudimos aprender, analizar, investigar y resolver muchos conceptos y en especial el problema y razón principal por la que se realizó el proyecto, el acoso sexual callejero. Pudimos encontrar caminos más cortos y más seguros para que las personas en especial las mujeres tuvieran más seguridad.

Por lo que también podemos destacar que los caminos son bastante diferentes, y esto le aporta bastante a la ciudad, ya que no solo será utilizado en el enfoque del acoso sexual, puede utilizarse para los ancianos que deben llegar rápido a sus casas por hurtos o por dificultades físicas, así que sería increíblemente útil para la seguridad y orientación de los ciudadanos. Aunque debemos admitir, que la rapidez que hemos conseguido no alcanza a la necesaria, ya que debemos ejecutarlo rápido por la seguridad del usuario, pero en caso de que la ejecución fuera en un tiempo mucho más rápido, sería más razonable. Por ultimo quisiéramos agregar que, en vez de elegir solo el camino más corto, recomendaría caminos seguros, con calles bien estructuradas, o lugares con más iluminación, tomando en cuenta patrullas de policía que hay cerca para la seguridad, tal y como la aplicación “Waze” hace

6.1 Trabajos futuros

Nos encantaría mejorar la compresión hacia otros algoritmos, ya que optamos por Dijkstra y solo conocemos este. Y acerca del código, aunque es funcional, no es tan veloz, así que mejorar el tiempo de ejecución sería lo ideal. Además, pensamos en seguir optimizando el código, ya sea en realidad virtual y desarrollo web, serían retos bastante interesantes para realizar, ¿Cómo? Implementando conocimientos de **Sensores de posición**. Dispositivos que se colocan en la habitación dónde se utilice el sistema de realidad virtual. El cual permite al sistema determinar la ubicación del usuario en todo momento. De esta forma se da la posibilidad de moverse libremente por el espacio virtual, utilizando su movimiento en la habitación, y podría usarse para que el usuario pueda ver el lugar al que se dirige... en caso de un encuentro solicitado por un extraño el usuario podría vivir literalmente el ir allá, y que caminos tomar. Hay bastante variedad.

AGRADECIMIENTOS

Queremos dar profundos agradecimientos a profesores de otros cursos de programación, los cuales han ayudado a aclarar dudas. Este es el profesor Édison Valencia Diaz, profesor de lenguajes de programación en la universidad EAFIT

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 archivo *Shapefile*.

REFERENCIAS

1. Pyle, D., Data preparation for data Mining. ed. Morgan kaufmann Publishers, Los Altos, California, 1999
2. Gupta, A. Un sistema de integración y análisis de datos para la planificación segura de rutas. ed. Department of Computer and Information Sciences. Towson University, Towson, 2018.
3. Esther Galbrun, Urban Navigation Beyond shortest route, recuperado en 2008.
<https://daneshyari.com/article/preview/397293.pdf>
4. Shortest path in a maze – Lee Algorithm, Techie Delight, N.D. <https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>
5. Cristian Bastidas, Algoritmo de Dijkstra en Python, Recuperado el 20 de abril del 2021.
<https://cristianbastidas.com/my-blog/algorithms/python/tutorial/wiki/2021/04/20/python-dijkstra.html#encontrar-los-caminos-m%C3%A1s-cortos>
6. The Insider's Guide to A* Algorithm in Python, Python Pool, N.D. <https://www.pythonpool.com/a-star-algorithm-python/#:~:text=A%20Algorithm%20in%20Python%20or%20in%20general%20is,the%20starting%20point%20%80%99s%20cost%20and%20the%20ending%20point>
7. Bellman Ford's Algorithm, Programiz, N.D.
<https://www.programiz.com/dsa/bellman-ford-algorithm>