

Proyecto VRP con algoritmo de optimización.

Armando De Santiago Rodríguez,

Adrian Ruiz Arana, Jorge Ernesto Lopez Arce Delgado,

Moises Sotelo Rodriguez.

CENTRO UNIVERSITARIO DE CIENCIAS

EXACTAS E INGENIERÍAS, (CUCEI, UDG)

armando.desantiago6265@alumnos.udg.mx

adrian.ruiz6211@alumnos.udg.mx

jorge.lopezarce@academicos.udg.mx

moises.sotelo5881@alumnos.udg.mx

Resumen:

El problema de enrutamiento vehicular (VRP) busca encontrar la ruta óptima considerando peso, volumen y limitaciones de tiempo. Se aplicó el algoritmo de Dijkstra para hallar la ruta más corta entre nodos, ignorando pesos negativos. Se desarrolló una interfaz que permite cargar un archivo CSV con información de nodos y distancias, graficando el resultado y mostrando el recorrido y distancia total. Aunque eficiente y escalable, el algoritmo no maneja pesos negativos. La interfaz facilita la implementación práctica del algoritmo en redes de transporte, pero está limitada a datos en formato CSV específico. El proyecto demuestra la efectividad de Dijkstra para VRP en situaciones sin pesos negativos.

Problema de enrutamiento vehicular (VRP).

<https://github.com/Armando-De-Santiago-Rodriguez/Proyecto-Analisis-de-Algoritmos>

1.0v

Python

I. Introducción:

El problema de enrutamiento vehicular (VRP) es un desafío relacionado con el sistema optimo para encontrar el recorrido más corto que depende, del peso, volumen y limitaciones de tiempo. Comúnmente las soluciones más optimas suelen ser inexistentes y por lo tanto se deben de adaptar a la mayoría de las necesidades. [1]

II. Trabajos relacionados:

En la búsqueda de algoritmos de optimización para la solución del problema de enrutamiento vehicular (VRP) se encontraron una diversidad de algoritmos que llegan a solucionar este problema.

Tabla 1.

Comparación de algoritmos de optimización VRP

Algoritmo	Características	Complejidad Temporal	Pesos Negativos	Aplicaciones Típicas
Dijkstra	Encuentra el camino más corto desde un nodo inicial a todos los demás nodos [2]	$O(V^2)$ u $O(E + V \log V)^*$	No	Redes de comunicación, sistemas de navegación, juegos de video, logística
Bellman-Ford	Maneja pesos	$O(V * E)$	Sí	Redes de comunicación

	negativos, detecta ciclos de peso negativo [3]			ón con posibles pesos negativos, problemas de flujos en redes
A*	Usa heurística para mejorar eficiencia, adecuado para búsqueda de rutas [4]	Depende de la heurística*	No	Sistemas de navegación, juegos de video, robótica
Floyd-Warshall	Encuentra caminos más cortos entre todos los pares de nodos, utiliza una matriz [5]	$O(V^3)$	Sí	Análisis completo de grafos, redes de tráfico, aplicaciones de rutas en redes
Johnson	Combina Bellman-Ford y Dijkstra, eficiente para grafos dispersos [6]	$O(V^2 \log V + V * E)$	Sí	Redes de comunicación, problemas de rutas con pesos negativos, análisis de redes
Prim	Encuentra el árbol de expansión mínima, similar a Dijkstra en técnica [7]	$O(E \log V)^{***}$	No	Construcción de redes, diseño de circuitos eléctricos, problemas de conectividad mínima

III. Descripción del desarrollo del proyecto:

El algoritmo utilizado para la resolución del problema (VRP) es el algoritmo de Dijkstra que permite encontrar la ruta más corta entre nodos que contienen un peso, se debe de marcar el nodo de inicio con un 0, y el resto de los recorridos con una distancia infinita, sucesivamente se marcará el nodo más corto y se actualizará la distancia, se marcará como visitado el nodo y se repetirá el proceso hasta llegar con el camino más corto. [2]

Al mismo tiempo se implementó una interfaz para que se ingrese un archivo csv con las siguientes características: Nombre del lugar, nombre del nodo, distancia del lugar al nodo A, B, C, D..., al

momento de que se ingresa con la librería de networkx graficamos el grafo y con la librería de pandas extraemos la información del archivo csv, la interfaz cuenta con 2 input, uno donde se ingresa el nodo de inicio y el 2do donde se ingresa el nodo del destino, al presionar el botón de “Calcular el camino más corto” se genera una ventana emergente donde te indica los nodos que recorrió, la distancia total y el tiempo que se tardo en desarrollar el cálculo.

Figura 1. Interfaz principal

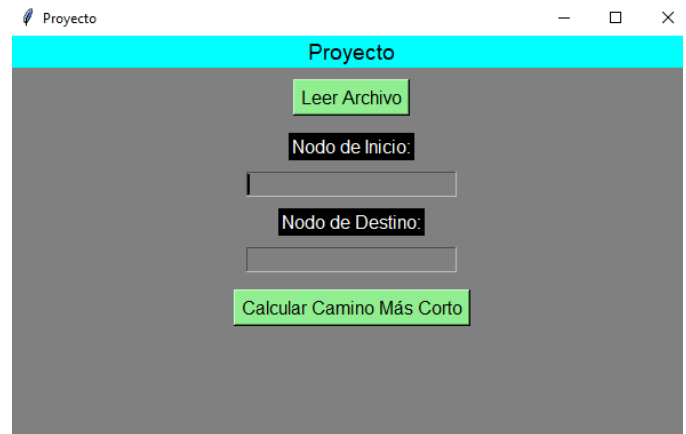


Figura 2. Ejemplo de grafo generado

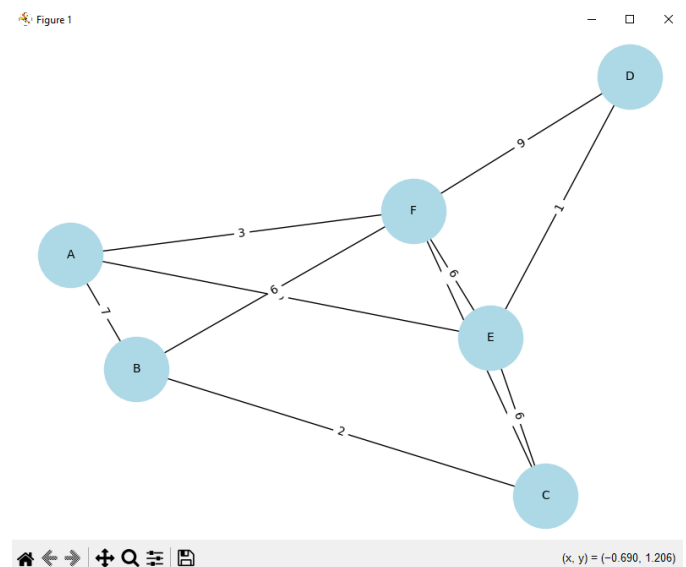
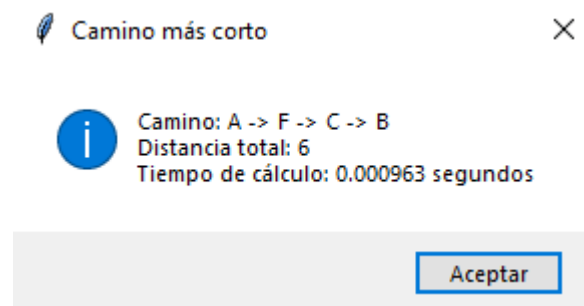


Figura 3. Ejemplo de los resultados



IV. Resultados Obtenidos del proyecto:

Tras la elaboración del código y diversas pruebas realizadas con el mismo, se determinó que el código en cuestión es eficiente para recorrer la distancia más corta en el menor tiempo posible, los problemas que se presentaron al momento de ejecutar el código es que no llega a manejar pesos negativos, por lo tanto ese es un aspecto a tener en cuenta, por otro lado al la escalabilidad del código agregando más nodos con el mismo formato es una opción asequible y funcional, por lo que no se deberían de presentar problemas si se llegan a usar otra base de datos, siempre y cuando el archivo csv cuente con el formato de la Figura 4.

Figura 4. Formato de la base de datos el archivo csv

Lugar	Nodo	Ditancia para A	Ditancia para B	Ditancia para C	Ditancia para D	Ditancia para E	Ditancia para F
Ciudad	A	0	7	0	0	5	3
Pueblo	B	7	0	2	0	0	6
Plaza	C	0	2	0	0	6	1
Parque	D	0	0	0	0	1	9
Zoológico	E	5	0	6	1	0	6
Escuela	F	3	6	1	9	6	0

V. Conclusiones:

Se demuestra la eficiencia del algoritmo Dijkstra para la resolución del problema de enrutamiento vehicular (VRP) de manera óptima y precisa en redes de transporte, tomando en cuenta la distancia existente entre cada uno de los nodos a pesar de las limitaciones de no poder manejar distancias negativas en su peso. A través de la implementación de una interfaz interactiva que permite la carga y procesamiento de archivos CSV con información de nodos y distancias, se facilita el uso práctico del algoritmo en aplicaciones reales. Junto con la posibilidad de escalar la base de datos con un archivo csv pero con el punto negativo de no poder manejar datos fuera de la extensión csv y fuera del formato establecido en la Figura 4.

Referencias:

- [1] <https://www.sciencedirect.com/topics/social-sciences/vehicle-routing-problem>
- [2] <https://www.codingame.com/playgrounds/7656/los-caminos-mas-cortos-con-el-algoritmo-de-dijkstra/el-algoritmo-de-dijkstra>
- [3] <https://jariasf.wordpress.com/2013/01/01/camino-mas-corto-algoritmo-de-bellman-ford/>
- [4] <https://aeia.home.blog/algoritmo-a-estrellas-a/>
- [5] <https://medium.com/algoritmo-floyd-warshall/algoritmo-de-floyd-warshall-e1fd1a900d8>
- [6] <https://complex-systems-ai.com/es/busqueda-de-ruta-de-teoria-de-grafos/algoritmo-johnson-2/>
- [7] <https://runestone.academy/ns/books/published/pythoned/Graphs/AlgoritmoDePrimDelArbolDeExpansion.html>