

Ruta eléctrica: La mejor ruta para entregas con vehículos eléctricos.

Isaías Labrador Sánchez
Universidad Eafit
Colombia
ilabradors@eafit.edu.co

Santiago Hincapié Murillo
Universidad Eafit
Colombia
shincapiem@eafit.edu.co

Andrés Almanzar Restrepo
Universidad Eafit
Colombia
aalmanzarr@eafit.edu.co

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

RESUMEN

Se trata de optimizar las rutas que los camiones eléctricos de entregas toman para realizar su trabajo teniendo en cuenta factores como: la duración de la batería, la velocidad de recarga para cada estación de recarga de batería, la duración máxima que una ruta puede tener, la velocidad del camión, entre otras. De fondo es un problema de grafos con unas restricciones específicas. Al tratarse de un mundo en el que constantemente se trata de automatizar y/o optimizar cada proceso para así ser más efectivo, este tipo de problemas va a ser muy común, por lo que, ir teniendo un primer contacto con lo que son rutas y restricciones es un buen primer paso para adaptarse a lo que, muy probablemente, va a ser nuestro futuro.

La solución de se plantea para este problema se basa en el algoritmo del vecino más cercano con algunas modificaciones.

Palabras clave

Ruta, grafos, vehículos eléctricos, vecino más cercano.

Palabras clave de la clasificación de la ACM

Theory of computation → Design and analysis of algorithms

1. Graph algorithms analysis → Shortest paths

1. INTRODUCCIÓN

A través de la historia siempre se ha buscado la solución al problema del camino más corto, incluso en las sociedades primitivas esta búsqueda del camino más corto era esencial. Hoy en día la búsqueda del camino más corto en busca de un mejor beneficio se puede ver evidenciado en las empresas de envíos como Fedex ya que entre menor y más eficiente sea su tiempo de entrega sus clientes estarán más satisfechos y por ende lo usaran nuevamente. En este proyecto buscamos crear un algoritmo que nos permita calcular la ruta

óptima entre dos o más puntos, siendo estos los lugares para entregar mercancía a clientes.

2. PROBLEMA

Se trata de optimizar las rutas tomadas por camiones eléctricos repartidores teniendo en cuenta varios factores que influyen en la toma de decisiones en cuanto a donde se

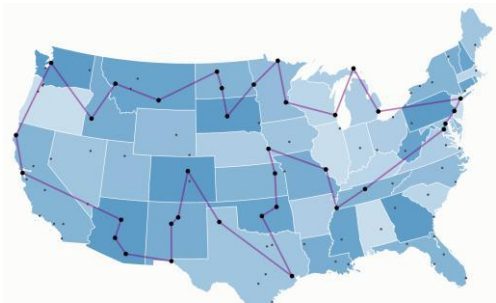
dirige, tales como: la velocidad del camión, la duración de la batería, las múltiples estaciones de recarga y su velocidad de recarga y la duración máxima de cada ruta. Esto es un problema que hoy en día con la automatización de los envíos con drones y los carros automatizados, es muy común y nos va acercando a lo que se consideraría un salto en el futuro en el que diseñar estas soluciones va a ser fundamental si se quiere trabajar en el campo tecnológico.

3. TRABAJOS RELACIONADOS

A continuación se presentan 4 problemas algorítmicos similares:

3.1 Problema del viajante (Traveling Salesman Problem)

Este problema se basa en un vendedor que se ve en la necesidad de visitar distintas ciudades, partiendo de una en específico y después de visitar cada una de las ciudades, regresar a la ciudad desde la cual partió.

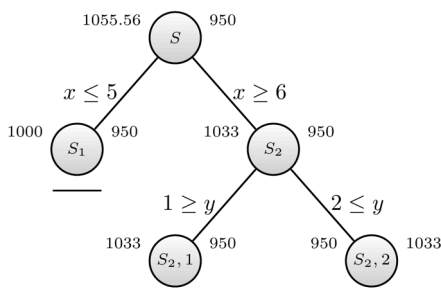


Grafica 1. Recorrido de solución para TSP

una de las soluciones más directas sería ver todas las posibles rutas, y ver cual de éstas es la menor (distancia). Para ello se utilizaría fuerza bruta o Brute Force. Pero haciendo uso de fuerza bruta para resolver este problema, puede llegar a ser demasiado lento, por ello se diseñó el

algoritmo Held-Karp. Este algoritmo al buscar una solución del problema del viajante simétrico, utiliza los mínimos árboles de expansión tanto en el proceso de ramificación como en el de acotamiento.

3.2 Método de branch and bound. El método de branch and bound (ramificación y poda), proporciona una solución óptima del problema del agente viajero, calculando mediante el algoritmo simplex la solución del modelo. A medida que aumenta el tamaño de la red el método puede tardar gran cantidad de tiempo en resolverse, sin embargo, para redes de mediano tamaño es una excelente alternativa.



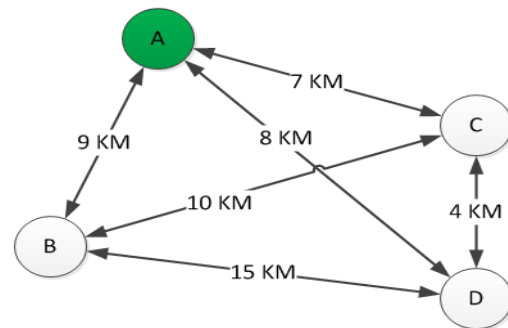
Grafica 2. Método de branch and bound.

3.3 Algoritmo Simplex

El Método Simplex es un método iterativo que permite ir mejorando la solución en cada paso. La razón matemática de esta mejora radica en que el método consiste en caminar del vértice de un poliedro a un vértice vecino de manera que aumente o disminuya (según el contexto de la función objetivo, sea maximizar o minimizar), dado que el número de vértices que presenta un poliedro solución es finito siempre se hallará solución.

3.4 Algoritmo del vecino más cercano

El Algoritmo del vecino más cercano (NN por sus siglas en inglés) o también llamado algoritmo voraz (greedy) permite al viajante elegir la ciudad no visitada más cercana como próximo movimiento. Este algoritmo retorna rápidamente una ruta corta. Para N ciudades aleatoriamente distribuidas en un plano, el algoritmo en promedio retorna un camino de un 25% más largo que el menor camino posible.²⁰ Sin embargo, existen muchos casos donde la distribución de las ciudades dada hace que el algoritmo NN devuelva el peor camino.



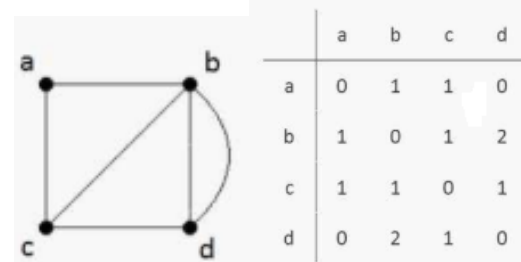
Grafica 3. Ruta utilizando vecino más cercano

4. VECINO MÁS CERCANO MODIFICADO

A continuación, se explicará la estructura de datos y el algoritmo que se usó para la solución del problema planteado.

4.1 Estructura de datos

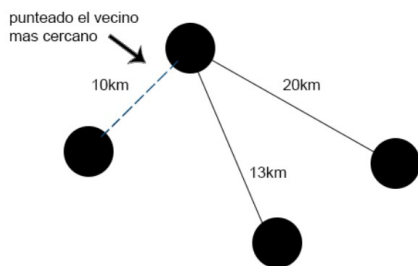
La estructura de datos que se maneja es un grafo que se basa en matrices de adyacencia, elegimos este modelo ya que desde un nodo se puede ir a cualquier otro nodo que pertenezca al grafo. Comúnmente llamado como grafo no dirigido.



Grafica 4. Representación de grafo con matriz de adyacencia

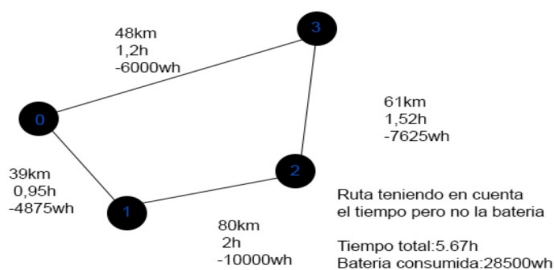
4.2 Operaciones de la estructura de datos

Buscar vecino más cercano



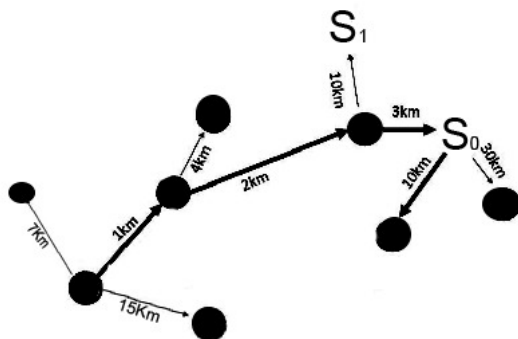
Gráfica 5: Vecino más cercano.

Buscar vecino más cercano con restricción de tiempo



Gráfica 6. Recorrido que se realiza cuando se tiene tiempo y no batería en un vehículo.

Buscar vecino más cercano con restricción de tiempo y batería:



Gráfica 7. Recorrido que hace cuando se tiene tiempo y batería.

4.3 Criterios de diseño de la estructura de datos

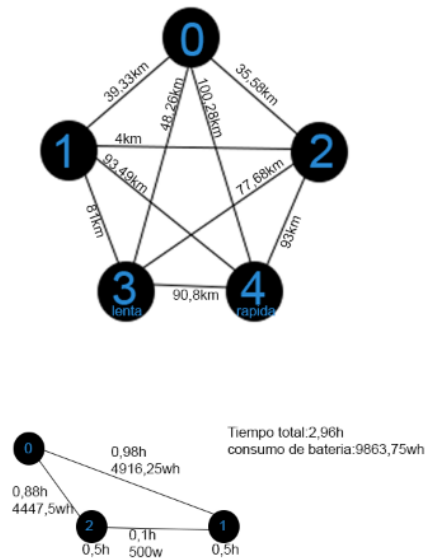
Después de analizar diferentes soluciones al problema, nosotros concluimos que una solución basada en un grafo dirigido usando una matriz de adyacencia era la mejor opción dado que un nodo A esta conectado con todos los nodos B del grafo, y de esta manera podríamos almacenar todas las distancias entre nodos para posteriormente acceder a ellas con una complejidad de $O(1)$.

4.4 Análisis de Complejidad

Funcion	Complejidad	Significado
Nodo	$O(n)$	Numero de nodos
Carro	$O(n)$	Numero de carros
Graph	$O(n^2)$	Numero de nodos
Algoritmo	$O(n^2)$	Numero de clientes
LinkedList	$O(n)$	Numero de clientes
Final	$O(n^2)$	

4.5 Algoritmo

El algoritmo se basa principalmente en el algoritmo de vecino más cercano con algunas modificaciones para poderlo recorrer con varios vehículos en algunos casos ya que existen algunas limitaciones como el tiempo y la batería. Además, siempre se mantiene control de volver al nodo inicial (deposito) desde cualquier nodo visitado con los recursos disponibles.



Grafica 8. Sistema para dos clientes, en el que se es más eficiente enviar un camión que vaya a 2 luego a 1 y vuelva al depósito. Al enviar dos camiones el tiempo total sería mayor, al igual que el consumo de batería.

4.6 Criterios de diseño del algoritmo

Después de analizar diferentes soluciones al problema, nosotros concluimos que una solución basada en el vecino más cercano es una buena solución al problema de encontrar un óptimo ruteo para vehículos eléctricos en un mapa dado. Mientras se garantice que desde el depósito (Nodo inicial) se pueda ir a todos los nodos y volver al mismo, teniendo en cuenta la batería y el tiempo del recorrido, el algoritmo funciona eficientemente. Uno de los factores claves es que podemos acceder a la distancia entre dos nodos en complejidad de $O(1)$.

4.7 Tiempos de Ejecución

Consumo de tiempo y memoria			
DataSet	Descripción	Memoria(Mb)	Tiempo (s)
1	345 nodos	35,8	24,68
2	345 nodos	36,1	34,33
3	345 nodos	35,3	27,98
4	345 nodos	35,4	23,98
5	345 nodos	35,4	34,39
6	345 nodos	38,4	33,6
7	359 nodos	38,5	24,9
8	359 nodos	38,6	35,88
9	359 nodos	38,7	34,21
10	359 nodos	38,7	31,88
11	359 nodos	38,9	32,99
12	359 nodos	38,5	32,7
Total	12 archivos	448,3	371,52

Tabla 2: análisis de los resultados obtenidos con la implementación del algoritmo

5. CONCLUSIONES

Para concluir, luego de analizar algunas estructuras de datos y ver la solución planteada vemos que la solución obtenida es optima y en tiempo bueno: La ruta obtenida en la solución final, satisface los parámetros dados. Por otra parte, pudimos deducir que el cambio de factores tales como: el tiempo de carga, el nodo seleccionado para visitar, entre otros pueden cambiar la solución. Este proyecto tiene muchos enfoques ya que con pocas modificaciones se puede proyectar a empresas del medio local para hacer sus rutas comerciales óptimas.

5.1 Trabajos futuros

En un futuro se desea implementar una mejor optimización del código e incluir métodos de búsqueda más rápidos, para así obtener menor consumo de memoria y de tiempo, y posiblemente obtener una ruta más óptima.

Agradecimientos

A la universidad EAFIT que nos brindo diferentes espacios para poder llevar a cabo la realización de este proyecto, así como diversas fuentes de materiales como libros digitales para investigar a la hora de encontrar información relevante.

REFERENCIAS

[1] Montoya A., Guérethttps C. [En línea]
Disponible en: <https://hal.archives-ouvertes.fr/hal-01245232/document>

[2] Wikipedia, El Problema del Viajante.

[En línea] Disponible en:

https://es.wikipedia.org/wiki/Problema_del_viajante

[3] Ingeniería Industrial Online, El Problema del Agente Viajero-TSP. [En línea] Disponible en:

<https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/problema-del-agente-viajero-tsp/>

[4] Ingeniería Industrial Online, El Método Simplex. [En línea] Disponible en:

<https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/m%C3%A9todo-simplex/>

<https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/m%C3%A9todo-simplex/>

[5]

https://es.m.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*#/media/Archivo%3APathfinding_A_Star.svg

[6] <http://idelab.uva.es/algoritmo>

[7]

<https://www.csd.uoc.gr/~hy583/papers/ch11.pdf>

[8]

https://upload.wikimedia.org/wikipedia/commons/f/f4/Pathfinding_A_Star.svg

[9]

<https://revistas.ucm.es/index.php/CESE/article/viewFile/CESE9696110267A/10650>