

OPTIMIZACIÓN DE ENTREGAS CON VEHÍCULOS ELÉCTRICOS

Jacobo Rave Londoño
Universidad Eafit
Colombia
jravel@eafit.edu.co

Diego Alejandro Vanegas González
Universidad Eafit
Colombia
davanegasg@eafit.edu.co

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

RESUMEN

Usar vehículos eléctricos para repartir mercancía a un conjunto de clientes con las rutas más óptimas. La importancia de lograr llevar una mercancía en las rutas más óptimas es bastante, ya que provee grandes beneficios a los dueños de la mercancía. Por ejemplo, un mejor y más rápido servicio a sus clientes, disminución en consumo de energía y aumento en la productividad.

1. INTRODUCCIÓN

Los vehículos eléctricos son el futuro ya que le hacen daño a la naturaleza, de esta forma, sus desventajas son pocas tales como rango de conducción limitado y el tiempo de carga de la batería es relativamente alto. Por esta razón, es importante hallar una forma de encontrar las rutas más óptimas posibles para entregar las mercancías.

Es la justificación de las condiciones en el mundo real que llevan al problema. En otras palabras, es hablar sobre qué va a tratar el documento e incluir la historia de este problema.

2. PROBLEMA

En pocas palabras escriban cuál problema están resolviendo, además de responder ¿para qué resolver este problema?

3. RELATED WORK

3.1 Optimización Combinatoria para el mantenimiento de vehiculos electricos (EnSPP)

El EnSPP consiste en encontrar rutas de origen-destino óptimas para EV maximizando la carga de la batería de los vehículos en el nodo de destino.

3.2 Vehicle Routing Problem (VRP)

Is a combinatorial Optimization and integer programming problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set de clientes? por lo que el tamaño de los problemas que se pueden resolver de manera óptima mediante programación matemática u optimización combinatoria puede ser limitado. Por lo tanto, los solucionadores comerciales tienden a usar heurísticas debido al tamaño y

la frecuencia de los VRP del mundo real que necesitan resolver.

3.3 ElectricVehicle Routing Problem (e-VRP)

Problema de enrutamiento de vehículos eléctricos (EVRP) que encuentra la estrategia de enrutamiento óptima con un costo de tiempo de viaje y un costo de energía mínimos, así como el número de vehículos eléctricos despachados. Eso se puede resolver utilizando la formulación de programación lineal de enteros mixtos MILP (que se puede adaptar para modelar diferentes supuestos de carga) y un conjunto de instancias pequeñas con información real sobre

3.4 Traveling Salesman Problem (TSP)

El problema del vendedor ambulante (TSP) es el desafío de encontrar la ruta más corta pero más eficiente para que una persona la tome dada una lista de destinos específicos. Es un problema algorítmico bien conocido en los campos de la informática y la investigación de operaciones. Algunas de sus soluciones son el método de fuerza bruta, el método de bifurcación y delimitación, el método del vecino más cercano.

4. Optimización de ruteo de Vehículos Eléctricos v1.0

4.1 Estructura de datos

Para resolver este problema, utilizamos varias estructuras de datos. Tales como, Listas de listas, Matrices de adyacencia, Listas de Arreglos.

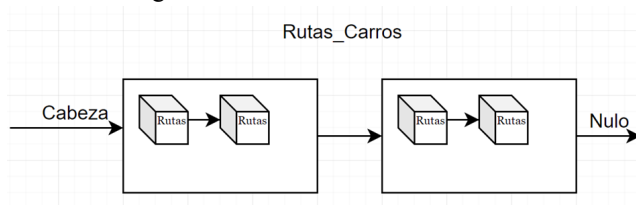


Gráfico 1. Listas de Listas, para almacenar las rutas de los diferentes carros

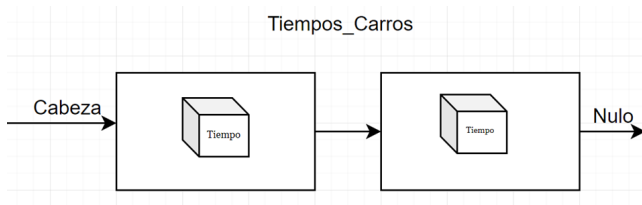


Gráfico 2. Listas de Listas, para almacenar los tiempos que se demora cada carro en su respectiva ruta.

	Deposito	c0	c1
Deposito	null	{D, T}	{D, T}
c0	{D, T}	null	{D, T}
c1	{D, T}	{D, T}	null

Gráfico 3. Matriz de Adyacencia, en las que almacenamos Pares de datos Distancia - Tiempo.

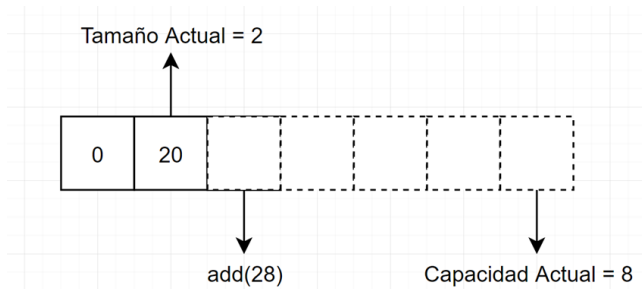


Gráfico 4. Listas de Arreglos, en los que almacenamos la ruta general.

4.2 Operaciones de la estructura de datos

Diseñen las operaciones de la estructura de datos para solucionar el problema eficientemente. Incluyan una imagen explicando cada operación

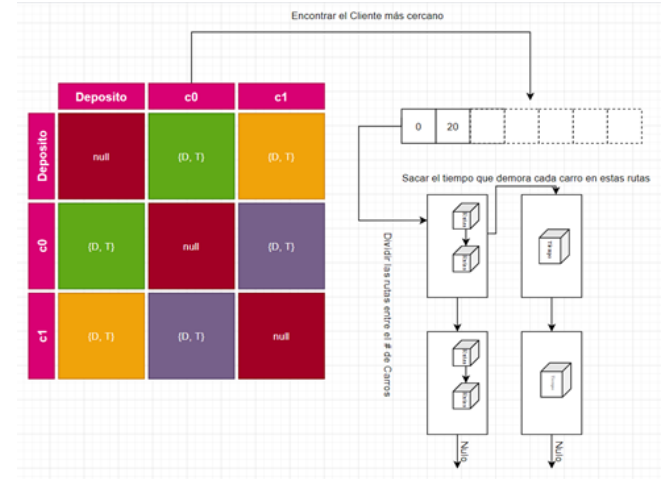
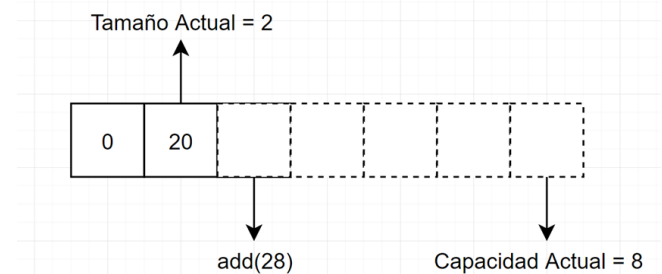
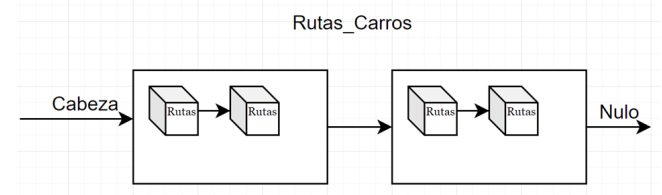


Gráfico 5. Imagen resumen sobre las operaciones del Algoritmo.

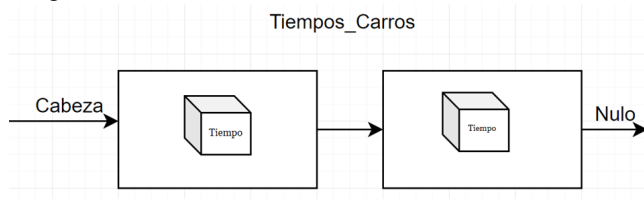
Creación de ruta general, creamos una ruta grande con todos los nodos, teniendo en cuenta los mejores tiempos para así hallar el cliente más cercano y dirigirse a su posición. Después, organizamos todos los clientes por orden de tiempo y de esta forma creamos una ruta gigante con todos los clientes con menor tiempo de viaje entre cada uno y añadimos esto a un ArrayList el cual almacena la ruta gigante.



Después, dividimos esta ruta en varias partes dependiendo del número de vehículos que vayamos a usar y les asignamos una partición de estas rutas para visitar siempre y cuando se cumpla con la restricción del tiempo.



Luego, separamos los tiempos de cada ruta individualmente por sus Vehículos. Para así saber cuánto se va a demorar en completar su ruta



4.3 Criterios de diseño de la estructura de datos

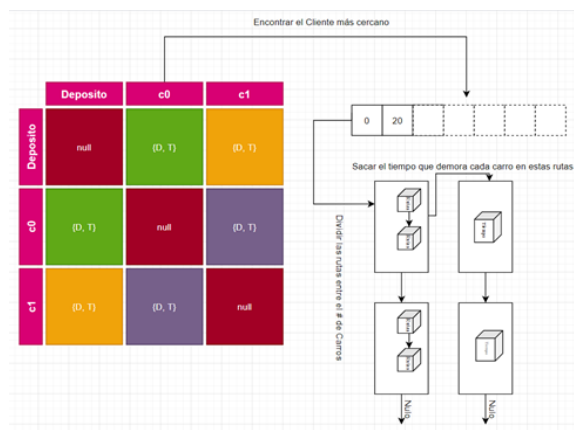
Seleccionamos esta estructura de datos por su fácil acceso y además de su increíble eficiencia en tiempo y memoria.

4.4 Análisis de Complejidad

Método	Complejidad
Leer	$O(N)$
generarGrafoDT	$O(N*M)$
calcularRuta	$O(N)$
calcularTiempo	$O(N^2)$
generarTiempoTotal	$O(N)$
generarCarros	$O(1)$
bestClient	$O(N)$
asignarRutas	$O(N)$

Tabla 1: Tabla para reportar la complejidad

4.5 Algoritmo



Gráfica 3: Paso a paso cómo se generan las rutas para cada carro

4.6 Cálculo de la complejidad del algoritmo

Sub problema	Complejidad
Crear el grafo con las distancias y los tiempos	$O(N*M)$
Encontrar el mejor cliente	$O(N)$
Dividir la ruta entre carros	$O(C)$
Generar los vehículos para las rutas	$O(1)$
Calcular el tiempo que se demora cada vehículo en su ruta	$O(A^2)$
Complejidad Total	$O(N^2)$

Tabla 2: complejidad de cada uno de los sub problemas que componen el algoritmo. Sea N y M nodos de coordenadas, C es el número de Carros y A es el número de clientes a visitar.

4.7 Criterios de diseño del algoritmo

Este algoritmo tiene una gran eficiencia en memoria y tiempo. Ya que nos permite hacer la creación y asignación de rutas en $O(N)$, además de calcular los tiempos para las rutas y filtrarlos en $O(A^2)$. Lo cual nos brinda una muy buena optimización en memoria y tiempo para resolver este problema

4.8 Tiempos de Ejecución

Calculen, (I) el tiempo de ejecución y (II) la memoria usada del algoritmo, para el Conjunto de Datos que está en el ZIP:

Tomen 100 veces el tiempo de ejecución y memoria de ejecución, para cada conjunto de datos

	Conjunto de Datos 1	Conjunto de Datos 2	...Conjunto de Datos n
Mejor caso	9.9 sg	10 sg	5 sg
Caso promedio	9.8 sg	9.7sg	6 sg
Peor caso	11 sg	11 sg	8 sg

Tabla 3: Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

4.9 Memoria

	<i>Conjunto de Datos 1</i>	<i>Conjunto de Datos 2</i>	<i>...Conjunto de Datos n</i>
Consumo de memoria	257MB	257 MB	257 MB

Tabla 4: Consumo de memoria del algoritmo con diferentes conjuntos de datos

4.10 Análisis de los resultados

Valores durante la ejecución del código

Tabla 5: Análisis de los resultados obtenidos con la implementación del algoritmo

	Matriz	LinkedList	ArrayList
Espacio en el Heap	0,9MB	0,16MB	0,04MB
Tiempo creación ruta	0	0	30MS
Tiempo creación número Carros	0	29MS	0
Tiempo creacion rutas-Carros	0	33MS	0
Impresión Matriz	7535MS	0	0

REFERENCIAS

1. Nora Touati MOUNGLA and Vincent JOST. 2011.(January 2011). Retrieved February 21, 2021 from <http://www.icrepq.com/icrepq'11/504-touati.pdf>
2. Suzanne Ma. 2020. Understanding the Travelling Salesman Problem (TSP). (January 2020). from <https://blog.routific.com/travelling-salesman-problem#:~:text=To%20solve%20the%20TSP%20using,this%20is%20the%20optimal%20solution.&text=This%20method%20breaks%20a%20problem,solved%20into%20several%20sub-problems>.
3. Alejandro Montoya. 2017. Electric Vehicle Routing Problems : models and solution approaches. (January 2017).from <https://tel.archives-ouvertes.fr/tel-01441718/document>
4. Anon. 2021. Vehicle routing problem. (January 2021). Retrieved February 22, 2021 from https://en.wikipedia.org/wiki/Vehicle_routing_problem