

# ALGORITMO PARA RUTEO DE VEHÍCULOS ELÉCTRICOS

Santiago Soto  
Universidad Eafit  
Colombia  
ssotom@eafit.edu.co

Kevyn Santiago Gómez  
Universidad Eafit  
Colombia  
ksgomezp@eafit.edu.co

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

## RESUMEN

En este documento se propone una solución al problema que parte desde cómo organizar y gestionar las rutas que recorrerán algunos camiones eléctricos al visitar un conjunto de clientes en un mapa de tal manera que se consuma la mínima energía y los recorridos se hagan en el menor tiempo posible, actualmente se tienen algunos métodos para resolver este tipo de problemas pero estos no tienen en cuenta las restricciones que implica que sea un vehículo que funciona con energía eléctrica y que este se debe estar cargando constantemente. La solución fue basada en el algoritmo del vecino más cercano con varias modificaciones de variables utilizando búsqueda local. Los resultados que obtuvimos de acuerdo a lo que buscamos solucionar son muy buenos y así mismo en este documento se muestra que pasos seguimos y como hicimos para reducir nuestros tiempos de solución en más de un 10% con pequeñas modificaciones

## Palabras clave

Estructuras de datos, Recorrido de Grafos, Camino más corto.

## Palabras clave de la clasificación de la ACM

Ejemplo: Theory of computation → Design and analysis of algorithms → Graph algorithms analysis → Shortest paths.

## 1. INTRODUCCIÓN

Con la implementación de nuevas tecnologías y el uso de energías renovables actualmente se está viendo en aumento, y con esto el uso de vehículos eléctricos para carga y transporte de pasajeros. Estas nuevas tecnologías nos llevan a resolver nuevos retos y problemas como el de encontrar las rutas óptimas que debe seguir unos camiones de carga eléctricos a unos respectivos clientes con limitantes como los tiempos de carga de energía de los vehículos y pasar por todos los clientes en el tiempo más óptimo.

## 2. PROBLEMA

Diseñar un algoritmo para ruteo óptimo para que un conjunto de camiones eléctricos con carga lineal visiten un conjunto de clientes. La solución de este problema nos permite tener la ruta más óptima que podemos recorrer con el mínimo gasto de energía y tiempo, que en el mundo real representaría un ahorro de tiempo y dinero para la empresa dueña de los camiones eléctricos además también un beneficio para el medio ambiente.

## 3. TRABAJOS RELACIONADOS

### 3.1 Enrutamiento óptimo para vehículos eléctricos

Los vehículos eléctricos alimentados por baterías jugarán un papel importante en el tráfico vial del futuro. Ya que la tarea ahora es determinar la ruta más económica en lugar de la más corta. Específicamente, (i) formalizamos el enrutamiento eficiente en energía en presencia de baterías recargables como un caso especial del problema restringido de ruta más corta (CSPP) con restricciones duras y blandas, y (ii) presentamos una adaptación de un algoritmo de ruta más corto (utilizando un gráfico de energía, es decir, un gráfico con una función de peso que representa el consumo de energía) que respeta las restricciones dadas y tiene una complejidad de caso más desfavorable de  $O(n^3)$ . Los algoritmos presentados se han implementado y evaluado dentro de un sistema de navegación prototípico para el enrutamiento eficiente de la energía.[1]

### 3.2 El problema de enrutamiento de vehículos

El problema de enrutamiento del vehículo (vrp) implica el diseño de un conjunto de rutas de costo mínimo para una flota de vehículos que atiende exactamente una vez a un conjunto de clientes con demandas conocidas. El problema de recogida y entrega con ventanas de tiempo (pdptw) es una generalización del vrp que se refiere a la construcción de rutas óptimas para satisfacer las solicitudes de transporte, cada una de las cuales requiere recogida y entrega por debajo de la capacidad, ventana de tiempo y restricciones de precedencia. Este algoritmo usa un esquema de generación de columnas con una ruta corta más restringida como un subproblema. El algoritmo puede manejar múltiples depósitos y diferentes tipos de vehículos.[2]

### 3.3 Un algoritmo genético para el problema de enrutamiento del vehículo

Este estudio considera la aplicación de un algoritmo genético (GA) al problema básico de enrutamiento de vehículos (VRP), en el que los clientes de demanda conocida se abastecen desde un único depósito. Los vehículos están sujetos a un límite de peso y, en algunos casos, a un límite en la distancia recorrida. Solo se permite un vehículo para abastecer a cada cliente. Los algoritmos genéticos (GA) se han utilizado para abordar muchos problemas combinatorios, incluidos ciertos tipos de problemas de enrutamiento de vehículos. Sin embargo, parece que las GA aún no han tenido un gran impacto en el VRP.[3]

### 3.4 Un algoritmo genético eficiente para el problema del vendedor viajero con restricciones de precedencia

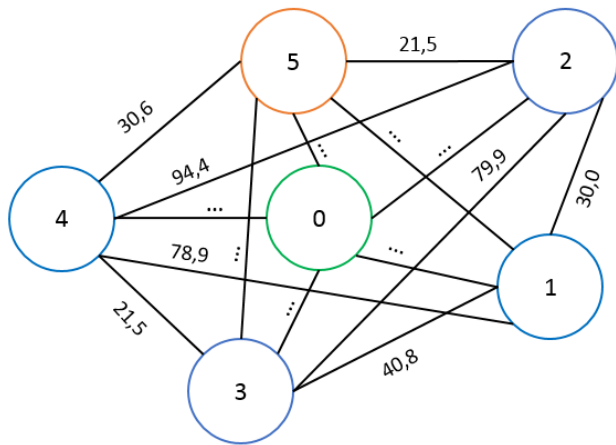
El problema del vendedor ambulante con restricciones de precedencia (TSPPC) es uno de los problemas de optimización combinatoria más difíciles. En este trabajo, se presenta un algoritmo genético eficiente (GA) para resolver el TSPPC. El concepto clave del GA propuesto es un tipo topológico (TS), que se define como un ordenamiento de vértices en un gráfico dirigido. Además, se desarrolla una nueva operación de cruce para la GA propuesta. Los resultados de los experimentos numéricos muestran que la GA propuesta produce una solución óptima y muestra un rendimiento superior en comparación con los algoritmos tradicionales. [4]

## 4. SOLUCIÓN FINAL: VECINO MÁS CERCANO OPTIMIZADO

A continuación, explicamos la estructura de datos y el algoritmo.

### 4.1 Estructura de datos

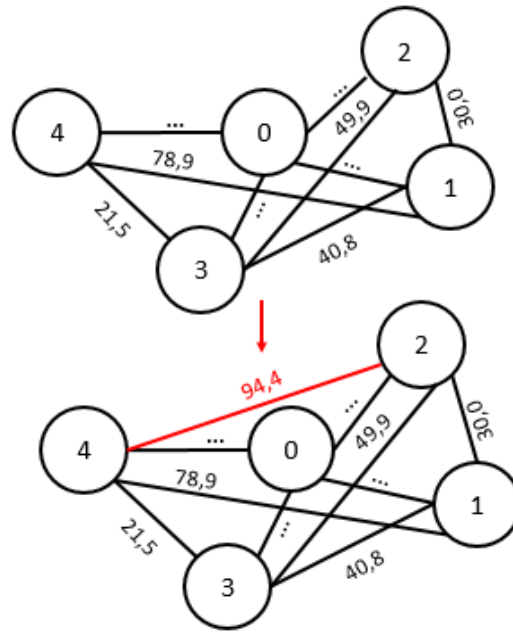
Grafo dirigido con pesos, representado con matriz de adyacencia, los nodos representan los clientes, estaciones de carga, y el depósito, las aristas representan las calles entre los nodo, y el peso representa la distancia.



**Gráfica 1:** Grafo totalmente conexo con pesos; los pesos representan la distancia en km de un nodo a otro.

### 4.2 Operaciones de la estructura de datos Agregar Arista

Esta operación lo que permite es indicar qué distancia hay entre dos puntos.



**Gráfica 2:** Imagen de una operación de agregar arista entre nodos.

### 4.3 Criterios de diseño de la estructura de datos

El grafo es la estructura de datos ideal para representar un mapa de una ciudad, o mejor dicho son casi equivalentes, las calles en el mapa son igual a las aristas en el grafo, las casas o esquinas son igual a los nodos. Cada casa tiene una dirección, dicho de otra forma, cada casa tiene una coordenada, con las coordenadas podemos obtener la distancia de una casa a otra en el grafo este sería el peso de la arista, como los mapas y los grafos tienen estas características en común es la mejor opción resolver este problema.

### 4.4 Análisis de Complejidad

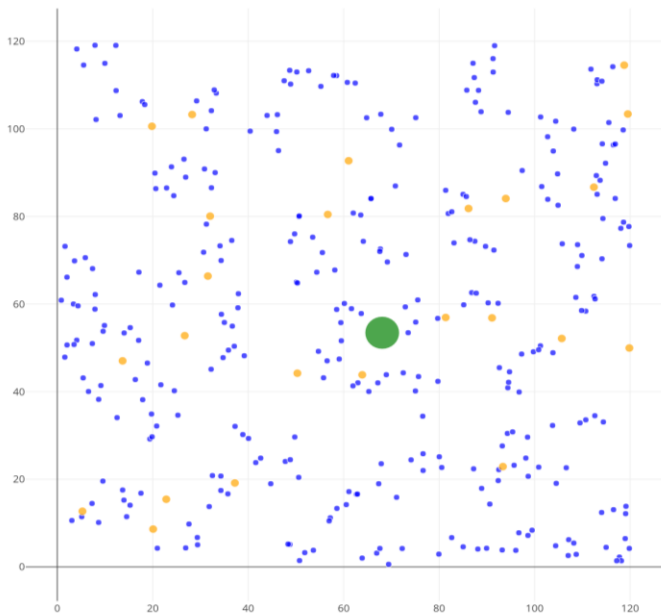
N es el número de clientes sumado el número de estaciones de carga.

Método	Complejidad
Agregar Arista	$O(1)$
Obtener peso entre dos nodos	$O(1)$
Obtener sucesores	$O(n)$

**Tabla 1:** Tabla de complejidades de las operaciones de la estructura tipo grafo.

#### 4.5 Algoritmo

Para resolver este problema hemos implementado, una versión del algoritmo del agente viajero, en este caso estamos utilizando un algoritmo voraz que busca el vecino más cercano, pero además modifica variables del problema para encontrar una mejor respuesta. Para la solución final estamos teniendo en cuenta que un camión no puede realizar un recorrido de más de un tiempo máximo de horas, que un camión tiene una batería limitada, que las estaciones de carga son diferentes, entre otros factores. El algoritmo siempre busca cual es punto más cercano al que se encuentra y verifica que la batería se la suficiente para continuar, si no va a la estación de carga más cercana, el recorrido se finaliza antes de completar el tiempo máximo de cada ruta, al final se realiza una optimización del tiempo que se haya utilizado cargando. Todo esto se realiza una cantidad constante de veces modificando variables como la anticipación con la carga la batería o reduciendo el tiempo máximo de la ruta. Con todas estas modificaciones se pudo mejorar en un 10% aproximadamente la respuesta originalmente entregada.



Gráfica 3: Ejemplo representación mapa

#### 4.6 Cálculo de la complejidad del algoritmo

N es el número de clientes sumado el número de estaciones de carga.

Sub problema	Complejidad
Leer Data Sets	$O(n)$
Crear Grafo	$O(n^2)$
TSP (Vecino más cercano)	$O(k \cdot n^2)$

Complejidad total	$O(k \cdot n^2)$
-------------------	------------------

Tabla 2: Complejidad de cada uno de los sub problemas que componen el algoritmo.

#### 4.7 Criterios de diseño del algoritmo

Es claro que hay algoritmos que darían una mejor solución, o las optima, pero para encontrar las solución óptima habría que utilizar algún algoritmo que calcule todas las combinaciones posibles de rutas, y por el tamaño del problema esto se demoraría una eternidad, es por eso que hemos implementando un algoritmo voraz, que aunque no da la mejor solución la da antes de los 30 sg por el tamaño de los datos, además por la estructura de datos que estamos utilizando es conveniente utilizar este algoritmo. Además la memoria utilizada por el algoritmo también se optimizó demasiado.

#### 4.8 Tiempos de Ejecución

	<i>tc2c320s24cf1</i>	<i>tc2c320s24cf4</i>	<i>tc2c320s24ct0</i>
Mejor caso	5.4 sg	5.5 sg	5.9 sg
Caso promedio	16 sg	7.3 sg	6.8 sg
Peor caso	20.7 sg	10.5 sg	8.2 sg

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

#### 4.9 Memoria

Mencionar la memoria que consume el programa para varios ejemplos

	<i>tc2c320s24cf1</i>	<i>tc2c320s24cf1</i>	<i>tc2c320s24ct0</i>
Consumo de memoria	140 MB	150 MB	135 MB

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

#### 4.10 Análisis de los resultados

Conjunto de Datos	Tiempo(h)
<i>tc2c320s24cf1</i>	274.03815
<i>tc2c320s24cf1</i>	334.3014
<i>tc2c320s24ct0</i>	462.16064

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

#### 5. CONCLUSIONES

La realización de este trabajo fue toda una experiencia ya que al implementar el algoritmo del vecino más cercano tuvimos nuestra primera solución y vimos cómo esta fue mejorando mientras hacíamos modificaciones en variables y veíamos como nuestra respuesta era cada vez mejor; sabíamos que teníamos menos de 30 segundos para encontrar una solución por eso utilizamos un algoritmo voraz ya que la cantidad de datos es alta y nos retorna una solución que no es la óptima al 100% pero es muy buena a comparación de nuestros primeros resultados. Este proyecto podría seguir avanzando hasta tener el algoritmo mucho más óptimo que hoy y aplicarlo en otros sectores como los sistemas de transporte públicos eléctricos, aplicaciones de viaje y movimiento que entreguen la ruta óptima para llegar a una dirección analizando el tráfico y las condiciones viales de la ciudad, camiones recolectores de basura eléctricos e infinidad de ámbitos en los que se puede implementar optimizando procesos, ahorrando tiempo y dinero y además aportándole al medio ambiente.

##### 5.1 Trabajos futuros

A futuro nos gustaría poder implementar el algoritmo para la ciudad de Medellín obteniendo la información de bases de datos de ciudad, además optimizar el algoritmo para que de una respuesta aún mejor.

#### REFERENCIAS

- 1.Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles. Retrieved March 3, 2018 from [https://link.springer.com/chapter/10.1007/978-3-642-16111-7\\_35](https://link.springer.com/chapter/10.1007/978-3-642-16111-7_35)
- 2.Yvan Dumas, Jacques Desrosiers and François Soumis. The pickup and delivery problem with time windows

Retrieved March 3, 2018 from <https://www.sciencedirect.com/science/article/abs/pii/S03772179190319Q>

3. Barrie M. Baker. Un algoritmo genético para el problema de enrutamiento del vehículo. Retrieved March 3, 2018 from

<https://www.sciencedirect.com/science/article/pii/S0305054802000515>

4.Chiung Moon, Jongsoo Kim, and Gyunghyun Choi. Un algoritmo genético eficiente para el problema del vendedor viajero con restricciones de precedencia. Retrieved March 3, 2018 from

<https://www.sciencedirect.com/science/article/abs/pii/S0377221701002272>