

# OPTIMIZACIÓN DE RUTEO PARA VEHÍCULOS DISTRIBUIDORES ELÉCTRICOS

Alejandro Arroyave Bedoya  
EAFIT  
Colombia  
Aarroyaveb@eafit.edu.co

Santiago Castrillón Galvis  
EAFIT  
Colombia  
Scastrillg@eafit.edu.co

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

Luis Javier Palacio Mesa  
EAFIT  
Colombia  
Ljpalaciom@eafit.edu.co

## Palabras clave

TSP, grafos, algoritmo voraz, ruteo de vehículos, estructura de datos, vehículos eléctricos con batería (VEB), K-means.

## Palabras claves de la clasificación de la ACM

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

## 1. INTRODUCCIÓN

Con la creciente globalización, la manera eficiente para el transporte de productos surge como una necesidad ante la progresiva demanda de los mismos. Se deben diseñar rutas para que vehículos de carga distribuyan en el menor tiempo posible, y es por esto que constantemente vemos nuevos modelos de vehículos, con mayor capacidad de carga. Sin embargo, al existir un número altísimo de clientes que complican la solución manual, es evidente que la velocidad y capacidad de carga no son los únicos factores que determinan la efectividad en el tiempo de las entregas. Por ello es que la ruta obtiene tanta relevancia, porque se debe calcular el orden de entrega de los productos de manera que se visite a todos los clientes y se consuma el menor tiempo posible.

No obstante, dado que el petróleo es un recurso limitado, pensando en el futuro, se deben buscar soluciones a este problema con un nuevo detalle: Se deben usar vehículos eléctricos. Esto presenta un reto puesto que los mismos tienen poca capacidad de autonomía si se compara con un vehículo que use petróleo.

## 2. PROBLEMA

Cuando hacemos un envío de varios productos a diferentes puntos, o queremos recorrer varios lugares determinados en el menor tiempo posible, se vuelve necesario trazar la ruta más corta que nos optimice el tiempo de viaje, pero también debemos tener en cuenta que los medios de transporte utilizan combustible, por lo que se hace necesario tener en cuenta el tiempo de recarga y el desplazamiento hacia este punto. Así que es necesario determinar bajo estos factores, cuál es la ruta más corta que podemos escoger para lograr el menor tiempo posible.

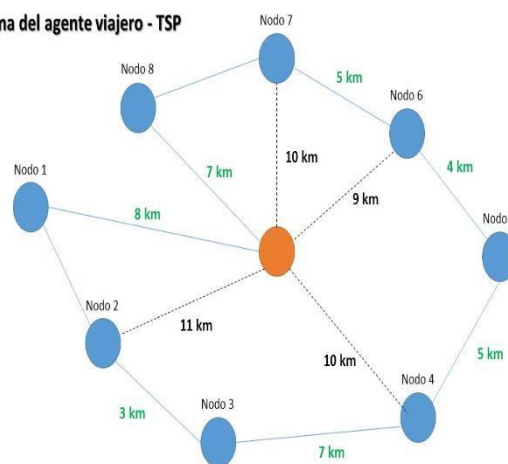
## 3. TRABAJOS RELACIONADOS

**3.1 Problema del agente viajero (TSP)** El problema del agente viajero consiste en encontrar la mejor ruta (minimizando distancia recorrida o tiempo recorrido) para un vehículo de reparto que pase por todos los vértices de un grafo(clientes) sin repetir, y volviendo al punto de origen.

Solución exacta: Una solución exacta sería hacer todas las posibles permutaciones del grafo (fuerza bruta), sin embargo, la complejidad es  $n!$ , el algoritmo es muy ineficiente para un grafo mediano o más grande (más de 15 nodos).

Solución heurística: Una solución heurística sería la llamada el “vecino más cercano”, que consiste en escoger para el nodo inicial, el nodo más próximo, y a su vez, de manera recursiva, este último, a su más cercano, excluyendo al ya visitado. Cuando todos los nodos hayan sido visitados, se debe agregar el camino hacia el nodo inicial.

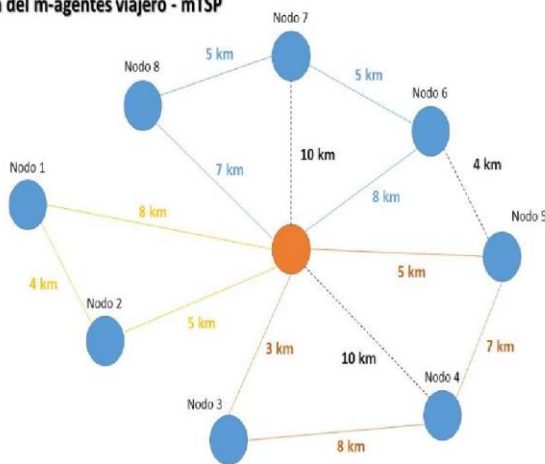
Problema del agente viajero - TSP



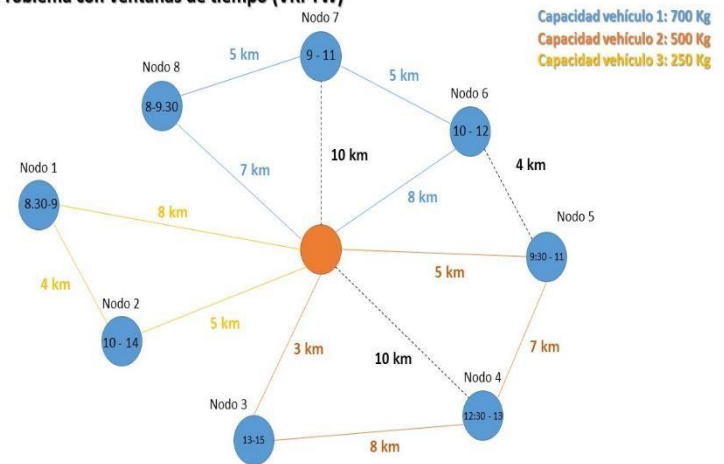
**3.2 Problema del m-agentes viajeros (m-TSP)** Esta es una variación del problema tradicional del agente viajero. En este caso se tienen  $m$  vehículos de reparto, y un nodo inicial y final visto como la bodega de una empresa, se deben realizar  $m$  rutas para los  $m$  vehículos y se debe garantizar

que cada cliente sea visitado exclusivamente una vez por uno sólo de estos vehículos, encontrando las rutas que generen el menor tiempo o distancia posible.

### Problema del m-agentes viajero - mTSP



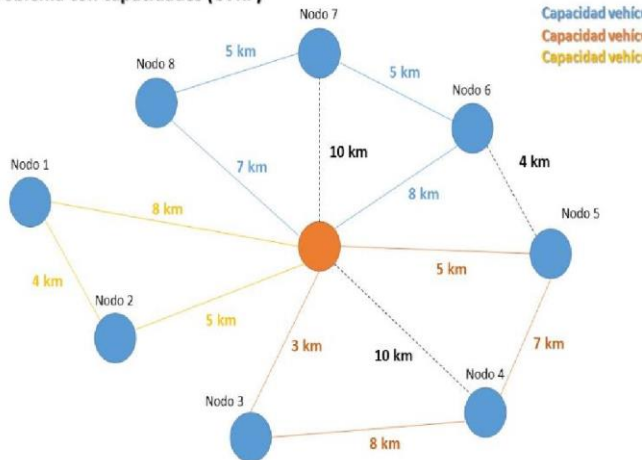
### Problema con ventanas de tiempo (VRPTW)



### 3.3 Problema con capacidades (CVRP)

Este es una derivación del problema m-agentes viajeros, en las que se introducen dos nuevos conceptos. El primero es la demanda de cada cliente, y se refiere a la cantidad de bienes que cada uno requiere, y el segundo es la capacidad de los bienes de cada vehículo de reparto.

#### Problema con capacidades (CVRP)

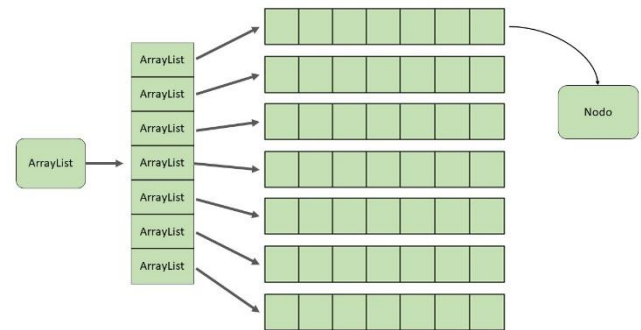


## 4. TÍTULO DE LA PRIMERA SOLUCIÓN DISEÑADA

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

### 4.1 Estructura de datos

La estructura de datos principal utilizada es un grafo



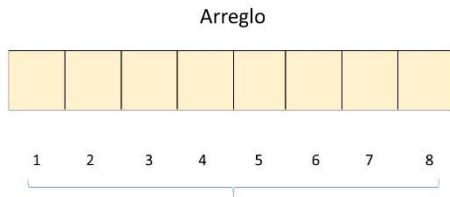
**Gráfica 1:** Grafo representado por listas de adyacencia

Dentro de los algoritmos del proyecto se encuentran otras estructuras de datos que son:

### 3.4 Problema con ventanas de tiempo (VRPTW)

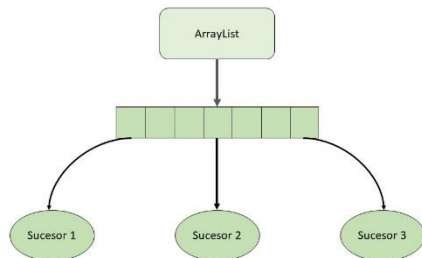
En este problema, además de tener capacidades y demandas (CVRP), hay un tiempo de visita válido para cada cliente, en el que se restringe las horas en las que cada camión puede llevar el pedido a cada uno de ellos. Además, cada cliente tarda un tiempo determinado para ser despachado.

## - Arreglos



Gráfica 2: Representación gráfica de arreglo

## - ArrayList



Gráfica 3: Representación gráfica de ArrayList

## - Pair

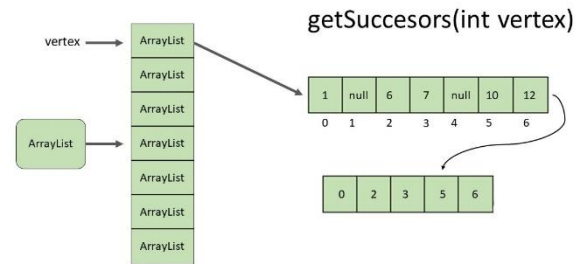


Gráfica 4: Representación gráfica de pair

## 4.2 Operaciones de la estructura de datos

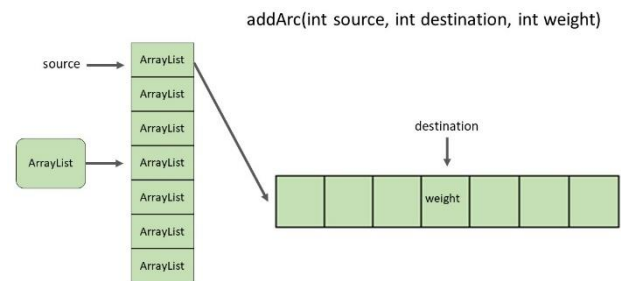
Las operaciones de la estructura de datos principal son:

- **getSuccesors (int vertex):** Esta operación devuelve un arreglo con los sucesores del nodo vertex.



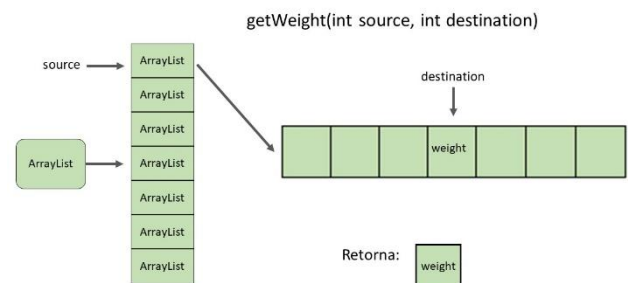
Gráfica 5: Método `getSuccesors (int vertex)`

- **addArc (int source, int destination, int weight):** Esta operación inserta un arco desde el nodo source al nodo destination con un peso weight.



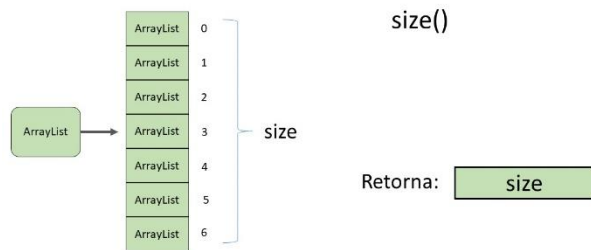
Gráfica 6: Método `addArc (int source, int destination, int weight)`

- **getWeight (int source, int destination):** Esta operación devuelve el peso del arco que hay entre source y destination.



Gráfica 7: Método `getWeight(int source, int destination)`

- **size():** Esta operación retorna el tamaño del grafo que tenemos.



Grafica 8: Método size()

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

### 4.4 Análisis de Complejidad

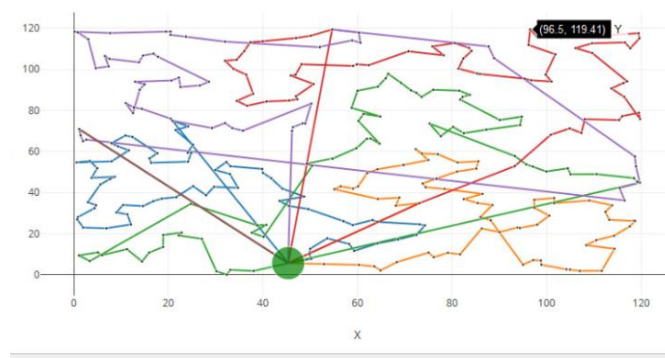
Método	Complejidad
getSuccesors	$O(n \log n)$
addArc	$O(n)$
getWeight	$O(n)$
Size	$O(1)$

**Tabla 1:** Tabla para reportar la complejidad, donde n es el número de nodos que hay en el grafo.

### 4.5 Algoritmo

En este primer algoritmo se busca generar varias rutas que no superen una duración de “Tmax” horas. Para esto se usa el algoritmo del vecino más cercano con una condición que lo detiene cada que se va a superar esta constante del problema. De esta manera se obtienen n rutas que corresponden a n buses. Este algoritmo implica que las últimas rutas incluyan puntos bastante lejanos del depósito, y serán éstas las que determinan el tiempo de solución. No se tiene en cuenta las estaciones de carga ni la batería de los autos.

A continuación, un gráfico que muestra el algoritmo:



Grafica 9: Solución al algoritmo

### 4.8 Tiempos de ejecución

	Mejor caso (en ms)	Caso promedio (en ms)	Peor caso (en ms)
<b>Conjunto de datos 1</b>	144	171	172
<b>Conjunto de datos 2</b>	257	317	350
<b>Conjunto de datos 3</b>	44	60	64
<b>Conjunto de datos 4</b>	59	67	71
<b>Conjunto de datos 5</b>	46	52	60
<b>Conjunto de datos 6</b>	25	34	46
<b>Conjunto de datos 7</b>	28	37	38
<b>Conjunto de datos 8</b>	29	32	35
<b>Conjunto de datos 9</b>	24	37	39
<b>Conjunto de datos 10</b>	32	37	42
<b>Conjunto de datos 11</b>	33	36	52
<b>Conjunto de datos 12</b>	56	70	67

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

### REFERENCIAS

1. Alfredo Olivera. Heurísticas para Problemas de Ruteo de Vehículos. Universidad de la República, Montevideo, Uruguay. Agosto 2004.

<https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0408.pdf>

2. Bryan Salazar López. PROBLEMA DEL AGENTE VIAJERO – TSP. [www.ingenieriaindustrialonline.com](http://www.ingenieriaindustrialonline.com). Abril 2016.

<https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3nde-operaciones/problema-del-agente-viajero-tsp/>

3. Wikipedia. Problema del viajante. Wikipedia.  
Enero 2018.  
[https://es.wikipedia.org/wiki/Problema\\_del\\_viajante](https://es.wikipedia.org/wiki/Problema_del_viajante)