



REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DEL PODER POPULAR PARA LA
EDUCACIÓN UNIVERSITARIA,
CIENCIA Y TECNOLOGÍA
POLITÉCNICO SANTIAGO MARIÑO
EXTENSIÓN CARACAS

CÁTEDRA: OPTIMIZACIÓN DE SISTEMAS DE INFORMACIÓN

Optimización de Rutas para la Distribución diaria de una empresa

Profesor:

Yancelis Noguera

Autor:

Rúbel Maneiro

Caracas, 01 de diciembre de 2025

Este proyecto busca resolver la ineficiencia en la distribución diaria de la empresa ficticia "ElectroEntrega S.A.". Actualmente, las rutas se planifican manualmente, resultando en recorridos largos, consumo excesivo de combustible y una utilización desigual de la flota. El objetivo es implementar un modelo de **Problema de Enrutamiento de Vehículos Capacitado (CVRP)** para minimizar la distancia total recorrida, respetando las capacidades de carga de los vehículos.

Fase 1: Análisis del Sistema Actual (Diagnóstico)

1.1. Descripción del Proceso

"ElectroEntrega S.A." opera un almacén central desde el cual distribuye productos electrónicos pequeños a clientes en una zona metropolitana.

- **Entradas:** Pedidos diarios de clientes con ubicaciones conocidas y cantidad de paquetes.
- **Recursos:** Una flota de 4 camionetas de reparto idénticas.
- **Salidas:** Rutas diarias para cada conductor y entrega exitosa de paquetes.
- **Proceso Actual:** El gerente de logística asigna pedidos a los conductores basándose en zonas geográficas aproximadas cada mañana. Los conductores deciden el orden de visita empíricamente.

1.2. Definición del Problema y Cuello de Botella

El problema principal es la **planificación subóptima de rutas**.

- *Síntomas:* Vehículos que cruzan sus trayectorias, algunos vehículos regresan casi vacíos mientras otros están sobrecargados (teniendo que hacer segundos viajes), y un alto kilometraje total diario.
- *Objetivo de la Optimización:* Minimizar la distancia total recorrida por toda la flota para entregar la totalidad de la demanda diaria.

1.3. Recolección de Datos (Escenario Ficticio)

Para esta prueba piloto, se utilizarán los siguientes datos:

- **Almacén Central (Depósito):** Coordenada [20, 20].
- **Flota:** 4 Vehículos.
- **Capacidad:** Cada vehículo puede transportar hasta 15 unidades.

- **Clientes y Demanda:** 16 clientes dispersos.

Fase 2: Modelado de Optimización

Se modela el problema como un **CVRP (Capacitated Vehicle Routing Problem)**. Este es un problema de optimización combinatoria que busca el conjunto de rutas más eficiente.

2.1. Variables de Decisión

Aunque herramientas avanzadas manejan esto internamente, conceptualmente las variables principales son:

- X_{ijk} : Variable binaria que es igual a 1 si el vehículo k viaja directamente desde el punto i al punto j . Es 0 en caso contrario.
- Y_{ik} : Variable auxiliar que indica la carga acumulada en el vehículo k después de visitar al cliente i .

2.2. Función Objetivo

El objetivo es minimizar la suma de las distancias de todos los trayectos realizados por todos los vehículos.

$$\text{Minimizar } \sum (Distancia_{ij} \times X_{ijk})$$

2.3. Restricciones del Sistema

1. **Visita Única:** Cada cliente debe ser visitado exactamente una vez por un solo vehículo.
2. **Flujo de Ruta:** Cada vehículo que sale del depósito debe regresar eventualmente al depósito.
3. **Capacidad del Vehículo:** La suma de las demandas de los clientes asignados a una ruta no puede exceder la capacidad máxima del vehículo (15 unidades).
4. **Subtours:** Deben evitarse rutas que no conecten con el depósito (ciclos aislados entre clientes).

Fase 3: Estrategia de Implementación y Solución

Para resolver este problema, que es computacionalmente complejo (NP-hard), no utilizaremos un solver genérico de Programación Lineal Entera (como Excel Solver básico), ya que se vuelve ineficiente con muchas restricciones de "subtours".

Utilizaremos **Google OR-Tools** en Python. Es una librería de nivel industrial diseñada específicamente para problemas de enrutamiento (Routing Model), que utiliza metaheurísticas avanzadas (como Búsqueda Tabú o Recocido Simulado) para encontrar soluciones casi óptimas en tiempos muy cortos.

3.1. Herramientas

- **Lenguaje:** Python 3.x
- **Librería de Optimización:** `ortools.constraint_solver`
- **Librería de Visualización:** `matplotlib.pyplot` (para generar las ventanas visuales del mapa).

3.2. Proceso del Algoritmo

1. Calcular una matriz de distancias entre todos los puntos (depósito y clientes).
2. Inicializar el modelo de enrutamiento de OR-Tools con la cantidad de vehículos y el nodo inicial.
3. Añadir la restricción de capacidad (Dimensión de Demanda).
4. Ejecutar el solver utilizando una heurística de primera solución (ej. "Path Cheapest Arc") y luego una metaheurística de búsqueda local para refinarla.
5. Extraer la solución y graficarla en un plano cartesiano.

Fase 4: Conclusiones Esperadas y Análisis

Tras la implementación del modelo propuesto, se esperan los siguientes resultados en comparación con el sistema manual:

1. **Reducción de Distancia:** Se estima una reducción de entre el 15% y el 25% en el kilometraje total diario de la flota. Esto se traduce directamente en ahorro de combustible y menor desgaste de vehículos.
2. **Balanceo de Carga:** Las rutas generadas aprovecharán la capacidad de los vehículos de manera más equitativa, evitando que un conductor trabaje en exceso mientras otro está subutilizado.

3. **Automatización y Estandarización:** El tiempo de planificación se reducirá de horas a segundos, eliminando la dependencia de la experiencia empírica de un solo planificador.

Análisis de Sensibilidad (Ejemplo Teórico):

Si la empresa adquiriera vehículos más grandes (capacidad 25 en lugar de 15), el modelo probablemente reduciría el número de vehículos necesarios para la misma demanda, pasando quizás de usar 4 camionetas a solo 3, lo que representa un ahorro significativo en costos operativos (un conductor menos, un vehículo menos en mantenimiento).