

**Universidad de Guadalajara**  
**Centro Universitario de Ciencias Exactas e Ingenierías**  
**Ingeniería de Computación**



**Análisis de Algoritmos (D01)**  
**Proyecto final - Documentación General**

**Integrantes:**

Carlos Andres Chico Aguayo  
Gael Emiliano Anaya Garcia  
Naylea Danae Silva Penaloza  
Carlos Humberto Avila Sanchez  
Nahomi Itzel Luna Ornelas

**Profesor:**

Jorge Ernesto Lopez Arce Delgado

26/NOV/2023

## **Problema: Optimización de Rutas para un Servicio de Entrega Local**

### **Descripción del Problema:**

Su tarea es crear un algoritmo de optimización de rutas eficiente para un servicio de entrega local. La compañía cuenta con una flota de vehículos de entrega y una lista de puntos de entrega (ubicaciones de clientes) que deben visitarse. El objetivo es encontrar las rutas más cortas para que los vehículos entreguen paquetes a todos los clientes y regresen al almacén central minimizando la distancia total recorrida.

El problema abordado es el de la optimización de rutas logísticas en un sistema de gestión de entregas. La solución propuesta se basa en un diseño orientado a objetos, donde entidades clave como "Camión", "Entrega" y "Ruta" se modelan como clases independientes. La estrategia de planificación logística se centra en la presencia de dos vehículos con características distintas.

Se divide el mapa en dos secciones, la cercana y la lejana, asignando a cada vehículo la responsabilidad de una sección específica. La división se realiza en función de la ubicación geográfica de los puntos de entrega para optimizar la cobertura de cada vehículo. Todas las ubicaciones se representan mediante coordenadas en un plano cartesiano, facilitando la planificación de rutas y coordinación espacial.

Cada vehículo carga todos los envíos posibles de su sección asignada sin exceder su límite de peso. Esta estrategia busca maximizar la capacidad de carga de cada vehículo y reducir la necesidad de viajes innecesarios. El proceso de entrega implica que un vehículo, una vez que ha entregado todos los pedidos en su sección, regrese al almacén para recargarse y continuar con los pedidos pendientes. Este ciclo continúa hasta que se han entregado todos los pedidos planificados.

La solución busca equilibrar la carga de trabajo entre los dos vehículos y optimizar las rutas de entrega. La división del mapa y la coordinación basada en coordenadas facilitan la asignación eficiente de pedidos, minimizando el tiempo y los recursos necesarios para completar las entregas. La estrategia prioriza la eficiencia y coordinación, maximizando la capacidad de carga de cada vehículo en su área designada.

Se ha optado por utilizar un Algoritmo Voraz en el sistema de planificación de rutas logísticas. Este enfoque voraz busca una solución óptima local en cada paso, seleccionando la mejor opción disponible en ese momento. La estrategia se centra en seleccionar pedidos por peso para maximizar la capacidad de carga del vehículo en cada asignación, reduciendo así el número total de rutas necesarias.

Es importante destacar que este algoritmo se enfoca exclusivamente en determinar los pedidos que formarán parte de una ruta y no aborda la planificación detallada del camino de la ruta en sí. La ventaja de este enfoque es su eficiencia y simplicidad, ya que permite una asignación rápida y eficiente de pedidos.

**Cálculo de rutas:** La planificación de rutas se basa en la representación de puntos como nodos en un grafo. El proceso comienza seleccionando el punto más lejano con respecto al origen, creando un grafo donde las aristas representan distancias euclidianas entre puntos. El algoritmo avanza seleccionando el siguiente nodo más cercano en cada paso, asegurando que todos los puntos sean visitados antes de regresar al almacén para obtener una nueva combinación de envíos.

**Procedimiento para el Cálculo de Rutas:**

**Selección del Punto de Partida:** Utilizando el algoritmo voraz, se selecciona el punto más lejano con respecto al origen como punto de partida.

**Creación del Grafo:** Se crea un grafo con todos los puntos obtenidos por el algoritmo voraz como nodos, y las aristas se ponderan con las distancias euclidianas entre los puntos.

**Asignación de Rutas:** Con un punto de partida determinado, se busca el siguiente nodo más cercano en el grafo, repitiendo este proceso hasta que todos los puntos sean visitados.

**Registro de Puntos Visitados:** Se mantiene un registro de los puntos visitados y no visitados para garantizar que cada punto sea alcanzado una vez y solo una vez.

**Regreso al Almacén:** Una vez que todos los puntos han sido visitados, se regresa al almacén para obtener una nueva combinación de envíos y repetir el proceso.

Al utilizar los datos de entrada donde se representa información sobre ubicaciones de clientes, la cantidad y peso de los artículos que deben entregarse a cada cliente. Cada cliente está asociado con un nombre, una ubicación en coordenadas y una lista de artículos con sus respectivos pesos. Aquí hay un ejemplo de datos de entrada:

**Nahomi Luna** (-16,7):

**Ubicación:** Coordenadas (-16, 7).

**Artículos y Pesos:**

Computadora: 20 unidades.

Consola: 15 unidades.

Estantería: 30 unidades.

Mesa: 20 unidades.

**José Cabrera** (20, -20):

**Ubicación:** Coordenadas (20, -20).

**Artículos y Pesos:**

Lavadora: 60 unidades.

Ropero: 90 unidades.

Puerta: 20 unidades.

Cama: 140 unidades.

**Carlos Rodríguez** (8,18):

**Ubicación:** Coordenadas (8, 18).

**Artículos y Pesos:**

Refrigerador: 120 unidades.

Bocina: 30 unidades.

Pantalla: 80 unidades.

Mochila: 26 unidades.

Cada conjunto de datos de entrada representa un cliente con su nombre, ubicación y una lista de artículos que se deben entregar, junto con la cantidad correspondiente de cada artículo.

Una vez que ingresemos los datos de entrada nos mostrará de salida algo como esto



Donde se podrá ver una visualización de resultados del pedido.

Esta clasificación se establece considerando la proximidad geográfica de los usuarios al punto de origen.

Por lo que podemos decir que el algoritmo voraz tiene de ventaja:

- Eficiencia: El enfoque voraz permite una asignación rápida y eficiente de pedidos.
- Simplicidad: La simplicidad del algoritmo facilita su implementación y comprensión.
- Optimización Local: Cada paso busca la mejor solución local, contribuyendo a la eficiencia general.

Pero su limitación es:

Optimización Global: No garantiza la solución global óptima, ya que la optimización se realiza en pasos locales.

Mientras que el algoritmo de cálculo de rutas tiene de ventajas:

- Eficiencia de Distancia: Utiliza distancias reales entre puntos para optimizar la eficiencia en las rutas.
- Optimización Local: El enfoque voraz garantiza soluciones óptimas locales en cada paso.

Pero sus limitaciones son:

- Optimización Global: No asegura la solución global óptima en términos de la planificación total del sistema de rutas.
- Complejidad del Grafo: La creación del grafo y el cálculo de distancias pueden volverse intensivos en recursos para conjuntos de datos muy grandes.

Con esto en mente se hicieron varios análisis de rendimiento dándonos lo siguiente: Evaluación del Algoritmo.

Se utilizó un algoritmo Voraz divide y vencerás junto con Fuerza Bruta y a continuación se muestra la complejidad.

• **Divide y vencerás:** Divide el problema en subproblemas más pequeños de manera recursiva  $O(\log(n))$ .

• **Fuerza Bruta en subproblemas pequeños:** Cuando los subproblemas alcanzan un tamaño manejable, aplica un enfoque de fuerza bruta para encontrar la solución óptima.

