

Trabajo Práctico Obligatorio de Programación II

Título del TPO: Diseño de rutas eficientes para camiones de reparto en ciudades.

Alumnos:

Joaquín De Luca - Legajo: 1183805

Agustín Cruspeire - Legajo: 1183963

Lautaro Vita - Legajo: 1186102

Nicolas Jesus Ferrante - Legajo: 1194383

Thiago Garcia Pereira - Legajo: 1189367

Profesor: Perez, Nicolas Ignacio

Carrera: Ingeniería en Informática

Universidad: UADE - Universidad Argentina de la Empresa

Fecha de Entrega: 27 / 06 / 2025

The logo of the Universidad Argentina de la Empresa (UADE) is displayed in a large, bold, blue serif font.

1. Introducción

En la actualidad, la logística urbana enfrenta desafíos significativos para optimizar las rutas de reparto, buscando minimizar costos, tiempos y mejorar la eficiencia operativa. Las empresas de transporte necesitan herramientas computacionales que permitan planificar rutas óptimas para sus camiones, considerando variables como el tráfico, distancias y tiempos de recorrido.

Este trabajo aborda la optimización de rutas para camiones de reparto mediante modelado de grafos ponderados y la aplicación del algoritmo de Dijkstra para encontrar rutas mínimas desde un centro logístico a distintos puntos de entrega en una ciudad.

2. Descripción del problema real

Una empresa de transporte urbano debe organizar diariamente las rutas de entrega desde su depósito central hacia múltiples clientes distribuidos en diferentes sectores. La red urbana se representa mediante un grafo dirigido y ponderado, donde:

- Cada nodo representa una ubicación física (almacén central, cliente, depósito intermedio).
- Cada arista representa una calle o ruta entre dos ubicaciones, con un peso asociado que indica el costo de recorrerla (tiempo estimado, distancia o combustible).

El objetivo es determinar la ruta más corta desde el centro logístico a cada punto de entrega, optimizando el tiempo total de viaje y los recursos empleados. Seleccionar rutas subóptimas puede implicar retrasos, mayores gastos y problemas en el cumplimiento de las entregas.

3. Algoritmo seleccionado: Dijkstra

El algoritmo de Dijkstra es una solución clásica para encontrar el camino más corto desde un nodo origen a todos los demás nodos en un grafo con pesos no negativos. Su funcionamiento básico consiste en:

1. Inicializar la distancia a todos los nodos como infinita, excepto el nodo origen con distancia cero.
2. Utilizar una estructura de datos (cola de prioridad) para seleccionar el nodo con la distancia mínima no procesada.
3. Actualizar las distancias de los nodos vecinos si se encuentra un camino más corto.
4. Repetir hasta que se hayan procesado todos los nodos alcanzables.

Ventajas para nuestro problema:

- Garantiza encontrar la ruta mínima en tiempo real.
- Eficiente en grafos con pesos no negativos (tiempos o distancias).
- Permite calcular rutas desde el depósito a todos los puntos de entrega en una sola ejecución.

4. Implementación bajo TDA (Tipos Abstractos de Datos)

Se implementaron los siguientes TDA para representar la red urbana:

- **Nodo:** Representa cada ubicación con su valor (identificador), lista de vecinos y pesos de las aristas que conectan a esos vecinos.
- **Grafo:** Contiene un conjunto de nodos y métodos para agregar nodos y aristas, además de mostrar la lista de adyacencia.
- **Interfaces (INodo, IGrafo):** Definen los contratos para nodos y grafos, facilitando la abstracción y posible extensión futura.
- **Servicio Dijkstra:** Implementa el algoritmo, utilizando una cola de prioridad para eficiencia y estructuras de datos auxiliares para distancias y nodos visitados.

Esta separación modular permite un diseño limpio y escalable, además de facilitar el testeado y mantenimiento del código.

5. Resultados y testeo

Se realizó un test utilizando la clase TestGrafos con la siguiente configuración:

- Nodos: 0 a 5 (0 = depósito central)
- Aristas con pesos que representan tiempos en minutos entre ubicaciones, simulando rutas reales de la ciudad.

Ejemplo de aristas:

Origen	Destino	Peso (min)
0	1	10
0	2	3
1	3	2
4	5	1

La ejecución del algoritmo Dijkstra mostró las distancias mínimas desde el depósito (nodo 0) a cada punto, permitiendo identificar las rutas óptimas para las entregas.

Impacto práctico: Esta solución reduce tiempos y costos, ayudando a la empresa a cumplir con las entregas dentro de los plazos previstos y a optimizar el uso de combustible.

6. Comparación con otros algoritmos

- A*: Requiere heurística para guiar la búsqueda; ideal para problemas con información adicional (como distancia en línea recta). No se utilizó porque en logística urbana general la heurística puede no ser trivial.
- Floyd-Warshall: Calcula rutas mínimas entre todos los pares de nodos; más costoso computacionalmente. Útil para análisis globales, pero no necesario para rutas desde un único depósito.
- Kruskal y Prim: Algoritmos para árboles de expansión mínima, útiles para diseñar redes con costo mínimo, pero no para rutas específicas de camiones.

7. Conclusión

El algoritmo de Dijkstra, aplicado sobre un grafo ponderado que modela la red urbana, es una herramienta efectiva para optimizar las rutas de reparto de camiones. La implementación bajo TDA asegura claridad y mantenimiento del código, mientras que el testeo con datos simulados confirma su utilidad práctica.

Para futuras mejoras se sugiere integrar datos en tiempo real (tráfico, condiciones climáticas) y evaluar algoritmos heurísticos como A* para rutas más inteligentes.