Problema de ruteo de vehículos (VRP) con múltiples restricciones.

Dario Rodríguez Llosa C-412
Alejandro Lamela Delgado C-411

Universidad de la Habana
Diseño y Análisis de Algoritmo

Problema a resolver:

Tras la pandemia de COVID-19, TuEnvio se consolidó como una plataforma clave de comercio electrónico, permitiendo a los ciudadanos acceder a productos sin la necesidad de hacer largas filas. Sin embargo, con el crecimiento de la demanda y la expansión de su red de distribución, la empresa enfrenta importantes desafíos en la gestión eficiente de su cadena de suministro. La operación de TuEnvio abarca múltiples niveles, desde la adquisición de productos hasta su entrega final, utilizando distintos medios de transporte como barcos, camiones y otros vehículos cada uno con sus propias capacidades y costos operativos.

El principal reto de la empresa es garantizar que los productos sean transportados de manera eficiente y a un costo óptimo, cumpliendo con diversas restricciones operativas que afectan la distribución. Entre ellas, la capacidad de carga de los vehículos juega un papel crucial, ya que un uso inadecuado de los transportes puede generar sobrecargas, afectar la seguridad de la mercancía e incrementar los costos operativos debido a viajes adicionales o consumo excesivo de combustible. Además, otro factor crítico es el estado de las carreteras, específicamente la presencia de baches, que impactan directamente la eficiencia del transporte terrestre. Un alto número de baches en una ruta puede ralentizar los tiempos de entrega, aumentar el desgaste de los vehículos y generar costos adicionales en mantenimiento, lo que hace indispensable considerar esta variable al momento de planificar las rutas. Por otra parte, las ventanas de tiempo para la entrega de los clientes finales son una restricción clave en la planificación logística. No solo es importante que los productos lleguen a su destino, sino que lo hagan dentro del tiempo prometido. En el comercio electrónico, la satisfacción del cliente depende en gran medida de la puntualidad de las entregas, por lo que una mala gestión de los tiempos puede afectar la reputación de la empresa y generar insatisfacción entre los usuarios. Asegurar que cada pedido se entregue dentro de su ventana de tiempo requiere una planificación precisa, teniendo en cuenta tanto las distancias como los posibles retrasos derivados de las condiciones de las carreteras y la capacidad de los vehículos.

Dado este panorama, TuEnvio necesita una solución integral que optimice la asignación de productos, la selección de rutas y el uso de la infraestructura de transporte, asegurando entregas puntuales, minimizando costos y mejorando la eficiencia operativa. Solo a través de una planificación inteligente que contemple todas estas restricciones, la empresa podrá garantizar un servicio confiable, competitivo y adaptado a las necesidades del comercio electrónico moderno.

Introducción

En este informe, se modelará el problema de distribución de TuEnvio como un Problema de Enrutamiento de Vehículos (VRP, por sus siglas en inglés), un problema ampliamente estudiado en la optimización combinatoria y la logística. Específicamente, consideraremos el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MVRP), una extensión del VRP clásico en la que los vehículos pueden partir desde diferentes depósitos, añadiendo un nivel adicional de complejidad a la planificación de rutas.

El análisis del MVRP nos permitirá no solo comprender la dificultad inherente de este tipo de problemas, sino también explorar su complejidad computacional y demostrar su clasificación como un problema NP-Difícil. Debido a su naturaleza combinatoria, encontrar soluciones óptimas para instancias grandes del MVRP es computacionalmente costoso, por lo que en este informe exploraremos tanto soluciones exactas basadas en técnicas como Branch and Bound (B&B), como métodos aproximados que buscan obtener soluciones cercanas al óptimo en tiempos computacionales razonables.

1 Demostración de que el Problema MVRP es NP-Difícil

Como el Problema MVRP es prácticamente una generalización del TSP, y es trivial ver que con el MVRP podemos resolver el problema del Viajante (TSP), lo que lo hace NP-Difícil, se expondrá la demostración de que el problema del viajante es NP-Difícil.

Demostración de que el Problema del Viajante de Comercio es NP-Difícil

Para demostrar que el Problema del Viajante de Comercio (TSP, por sus siglas en inglés) es NP-Difícil, realizamos una reducción polinómica desde el problema del *Hamiltonian Cycle* (HC), que se sabe que es NP-Difícil.

1. Problema del ciclo Hamiltoniano (HC)

El problema del ciclo Hamiltoniano consiste en encontrar un ciclo que pase por todos los vértices de un grafo dado exactamente una vez. Formalmente, dada una gráfica $G = (V, E)$, el problema es determinar si existe un ciclo que pase por todos los vértices de V.

Sabemos que el problema HC es NP-Difícil, lo que significa que no hay un algoritmo conocido que lo resuelva en tiempo polinómico, a menos que $P = NP$.

2. Reducción desde HC al TSP

Ahora, consideremos el siguiente grafo $G = (V, E)$ como entrada para el problema del ciclo Hamiltoniano. Construimos una instancia del TSP de la siguiente manera:

- Para cada vértice $v_i \in V$, creamos un nodo en el TSP.
- Para cada arista $(v_i, v_j) \in E$, asignamos un costo de 1.
- Para cada par de vértices no adyacentes en G (es decir, que no están conectados por una arista), asignamos un costo arbitrariamente grande.

Con esta construcción, una solución óptima para el TSP corresponderá a un ciclo Hamiltoniano en G. Si existe un ciclo Hamiltoniano en G, entonces el TSP tendrá una solución de costo $|V|$ (donde $|V|$ es el número de vértices de G), que es la mínima posible dada la construcción de los costos. Si no existe un ciclo Hamiltoniano, entonces no se podrá encontrar un ciclo en el TSP de costo $|V|$.

Dado que hemos demostrado que cualquier instancia del problema HC puede ser reducida en tiempo polinómico al problema del TSP, y dado que HC es NP-Difícil, concluimos que el problema del Viajante de Comercio (TSP) también es NP-Difícil.

Una solución exacta para el problema:

En instancias pequeñas del problema donde el tiempo no sea un problema, se podría utilizar Branch and Bound (B&B).

En el problema de enrutamiento múltiple, B&B se puede utilizar para explorar todas las posibles rutas y eliminar aquellas que no cumplen con nuestras restricciones de optimización. A través de la estrategia de ramificación, se dividen las rutas posibles en subconjuntos más pequeños, mientras que la poda ayuda a descartar opciones ineficientes, reduciendo así el espacio de búsqueda y mejorando el tiempo de solución.

```python
best_cost = float('inf')
best_solution = None
queue = [RootProblem()]  # Start with the root problem

while queue:
    subproblem = queue.pop()  # Get the next subproblem
    if subproblem.lower_bound >= best_cost:
        continue  # Prune this subproblem

# Branch: Create new subproblems by assigning customers to
    vehicles
    for customer in unassigned_customers:
        for vehicle in vehicles:
            if feasible_assignment(customer, vehicle,
                subproblem):
                new_subproblem = create_subproblem(
                    subproblem, customer, vehicle)
                queue.append(new_subproblem)

    # Bound: Compute lower and upper bounds
    lower_bound = compute_lower_bound(subproblem)
    upper_bound = compute_upper_bound(subproblem)

    if upper_bound < best_cost:
        best_cost = upper_bound
        best_solution = subproblem.solution

return best_solution, best_cost
```

Listing 1: Using Branch-and-Bound Algorithm

Solución aproximada para el problema.

Algoritmo Genético (GA):

Nuestro algoritmo genético recibe como entrada los siguientes parámetros:

1. *metaheuristic:* Problema genético que contiene:un conjunto de clientes (identificación y posición)y un conjunto de vehículos (identificación, posición y capacidad de los vehículos).

2. *k:* Cantidad de participantes en el torneo de selección.

3. *ngen:* Número de generaciones.

4. *size:* Tamaño de la población.

5. *ratio cross:* Cantidad de la población que descenderá por cruzamiento.

6. *prob mutate:* Probabilidad de mutación.

7. *distances:* Matriz de distancias.

8. *time windows:* Ventanas de tiempo en el que los clientes pueden recibir los productos.

9. *potholes matrix:* Matriz de la cantidad de baches entre los nodos.

10. *max potholes:* Máxima cantidad de baches que puede pasar un vehículo en su ruta.

Primero se debe crear una población inicial, la cual es elegida aleatoriamente y que está dada por la variable size. Esta será la primera generación y es la que dará origen a generaciones posteriores las cuales serán cada vez más aptas. Posteriormente a la creación de nuestra primera población, debemos crear nuestras posteriores generaciones. En la nueva generación hay una cantidad de individuos que pasan directamente a la siguiente generación, estos descienden mediante selección por torneo en donde la cantidad de descendientes está dada por la cantidad total de la población menos la cantidad de individuos que resultaron mediante cruzamientos. Los individuos que pasan a la generación siguiente directamente lo hacen por elitismo, es decir, losmejores de su población.

El 85% de la población corresponde a hijos que nacieron mediante cruzamientos. Este por ciento se obtiene a partir de una selección por torneo de la siguiente forma:

- Se escogen dos individuos al azar, a los cuales se les calcula su función fitness. El individuo más apto (con mejor fitness) es quién será el padre número uno.

- Posteriormente se repite el paso anterior para obtener el padre número dos.

Por cada hijo existe una probabilidad de que este mute. Esta es una manera para no caer en óptimos locales, dándole paso a soluciones que actualmente parecen no ser las mejores para la siguiente generación. La nueva generación de soluciones más óptimas está dada por:

1. Individuos seleccionados por elitismo cuya cantidad está dada por cantidad total menos cantidad de individuos que se obtendrán por cruzamientos.

2. Individuos obtenidos por cruzamiento.

El algoritmo itera con este procedimiento hasta que el número de generaciones termina. Una vez se obtiene la generación final, el individuo con mayor fitness es el que se escoge como solución final obtenida.

Experimentación:

Se utilizará como prueba un problema de Augerat nombrado como An32k5 con los siguientes parámetros y especificaciones para el algoritmo:

- Cantidad de clientes: 31
- Cantidad de vehículos: 5
- Capacidad de los vehícculos: 100
- Solución óptima: 787
- Cantidad de generaciones: 2500
- Tamaño de la población: 800
- Porcentaje de individuos que heredarán por cruzamiento: 85%
- Probabilidad de mutación: 0.05

Para evaluar el desempeño se emplea una métrica que determina qué tan buena es nuestra solución a partir de una solución óptima conocida para la misma entrada. La métrica de evaluación se define como:

$$\frac{\text{output - optimo}}{\text{optimo} \times 100}$$

Como podemos observar la medida de evaluación muestra el porcentaje de diferencia que el algoritmo varía de acuerdo a la solución exacta del problema.

Luego de 100 iteraciones de nuestra metaheurística basada en algoritmos genéticos para nuestro problema a resolver se obtuvieron los resultados mostrados en la tabla 1.

Tabla1: Estadísticas del Algoritmo Genético

Métrica	**Valor**
Cantidad de iteraciones	100
Mejor solución	797.88
Peor solución	982.41
Promedio de las soluciones	850.67
Mejor tiempo de ejecución (s)	10.467000
Peor tiempo de ejecución (s)	10.932000
Promedio de los tiempos (s)	10.769990
Total de los tiempos (s)	1076.999000
Mejor métrica de evaluación	0.000138
Peor métrica de evaluación	0.002483
Promedio de las métricas	0.000809

Se realizó otro experimento aumentando el tamaño de la población a 800 y el número de generaciones a 2500, lo iteramos 30 veces y los resultados se muestran en la tabla 2.

Tabla2: Estadísticas del Algoritmo Genético

Métrica	**Valor**
Cantidad de iteraciones	30
Mejor solución	798.94
Peor solución	843.91
Promedio de las soluciones	818.30
Mejor tiempo de ejecución (s)	85.006
Peor tiempo de ejecución (s)	87.051
Promedio de los tiempos (s)	86.527899
Total de los tiempos (s)	2595.837
Mejor métrica de evaluación	0.000152
Peor métrica de evaluación	0.000723
Promedio de las métricas	0.0003976

Conclusiones:

Si tomamos en cuenta la metaheurística analizada nos percatamos que se obtuvieron resultados un poco inferior al óptimo, aunque no muy alejado y con un tiempo aceptable pese a no tener equipos de cómputos con grandes capacidades. Además, el tiempo oscila alrededor de los 10 segundos en el primer experimento y en los 86 segundos aproximadamente en el segundo experimento, el cual consideramos como factible para problemas de gran magnitud.

Para obtener mejores resultados en cuanto al costo de la solución, es necesario sacrificar el tiempo de ejecución. Existen dos maneras, no excluyentes, para hacer esto:

1. Aumentando el número de generaciones.

2. Aumentando el tamaño de la población.

Hasta ahora la metaheurística de algoritmos genéticos parecen ser una muy buena herramienta para estimar enrutamiento de vehículos ya que nos provee soluciones muy buenas en tiempo.