

### TP3 - Heurísticas para el Problema de Ruteo de Vehículos con Capacidad (CVRP)

---

#### El contexto

El **Problema de Ruteo de Vehículos con Capacidad (CVRP, por sus siglas en inglés)** es uno de los problemas de optimización combinatoria más estudiados debido a su impacto directo en una vasta gama de aplicaciones del mundo real. Se encuentra en el corazón de la logística y la gestión de la cadena de suministro, afectando a cualquier empresa u organización que necesite entregar bienes o proveer servicios a un conjunto de ubicaciones geográficamente dispersas.

Desde un gigante del comercio electrónico como Amazon planificando sus entregas diarias, o un distribuidor local de alimentos abasteciendo a restaurantes, hasta una empresa de gestión de residuos programando sus rutas de recolección, resolver el CVRP de manera eficiente es crítico. Una solución bien diseñada puede conducir a ahorros significativos de costos en términos de combustible, desgaste de vehículos y horas de conductor. Además, mejora la eficiencia operativa, aumenta la satisfacción del cliente a través de entregas puntuales y ayuda a reducir la huella ambiental al minimizar el kilometraje y las emisiones, entre otros beneficios adicionales.

El CVRP pertenece a la clase  $\mathcal{NP}$ -hard, y ha sido estudiado en gran detalle por la comunidad científica durante la últimas décadas. A pesar de su dificultad, actualmente se dispone de algoritmos extremadamente eficientes que permiten soluciones de gran calidad para instancias de miles de clientes en unos pocos minutos. Muchos desarrollos propuestos para el CVRP y sus variantes han sido luego generalizados a otros problemas, permitiendo avanzar el conocimiento no solo para el problema en particular sino para la Optimización Combinatoria en general.

#### El modelo

El Problema de Ruteo de Vehículos con Capacidad (CVRP) simétrico puede definirse de la siguiente forma. Sea  $G = (V, E)$  un grafo completo no dirigido, donde  $V = \{0, 1, \dots, n\}$  es un conjunto de vértices y  $E$  es el conjunto de aristas que los conectan. El vértice 0 representa el **depósito**, donde todos los vehículos comienzan y terminan sus rutas, y  $V' = V \setminus \{0\}$  son los **clientes**. Cada cliente  $i \in V'$  tiene una **demanda**  $d_i \in \mathbb{R}_{\geq 0}$ . Un costo (o distancia)  $c_{ij}$  está asociado con cada arista  $(i, j) \in E$ , y en esta variante asumimos estos costos son simétricos ( $c_{ij} = c_{ji}$ ) y satisfacen la desigualdad triangular. Existe una flota homogénea de vehículos, cada uno con una **capacidad** máxima uniforme  $Q$ . Dada una ruta  $r = (0 = v_0, v_1, \dots, v_l = 0)$ ,  $v_i \in V$ , el costo de una ruta se define como

$$c(r) = \sum_{i=0}^{l-1} c_{v_i v_{i+1}}$$

El objetivo es encontrar un conjunto de rutas de vehículos  $r_1, \dots, r_K$  con un costo total mínimo, definido como la suma del costo de todas las rutas, que satisfaga las siguientes restricciones:

1. Cada ruta comienza y termina en el depósito.
2. Cada cliente es visitado exactamente por un vehículo.
3. La suma de las demandas de los clientes en cualquier ruta individual no debe exceder la capacidad del vehículo  $Q$ , es decir,

$$\sum_{i \in r_k} q_i \leq Q, \text{ para } k = 1, \dots, K$$

4. El número de vehículos a utilizar,  $K$ , puede ser limitado. Si se especifica un límite en el número de vehículos para una instancia, ese límite debe ser respetado. Si no, podemos asumir que la flota está limitada por el número de clientes,  $n$ , ya que un escenario de peor caso implica un vehículo por cliente.

## Los datos

Para probar y evaluar tus algoritmos, utilizarás instancias de benchmark de la **VRPLIB**, una biblioteca estándar para instancias de VRP. Esto permite comparar resultados de distintos algoritmos de manera justa y promoviendo la replicabilidad. Los archivos VRPLIB tienen un formato bien definido que especifica las características del problema, como el número de clientes, la capacidad de los vehículos, y las coordenadas y demandas de cada cliente.

En cuanto a la lectura de las instancias, se provee una clase de C++, `VRPLIBReader`, para procesar estos archivos y, además, es posible utilizar la librería `PyVRP` ([documentación](#)) para aquellos grupos que opten por una implementación en Python

## El trabajo

El trabajo práctico consiste en implementar distintas heurísticas para el CVRP, combinarlas, y evaluarlas utilizando las instancias de benchmark de VRPLIB. Los grupos pueden ser de máximo 3 personas. El trabajo práctico puede ser implementado en Python o C++, decisión a tomar por el grupo y que debe ser correctamente justificada. Se pide:

1. **Heurísticas constructivas.** Implementar (al menos) dos heurísticas constructivas distintas para el CVRP, donde una de ellas deberá ser la **Heurística de Savings de Clarke y Wright**, un clásico para el CVRP y de gran efectividad para su simpleza. Analizar la complejidad de los algoritmos, que debe ser consistente con la implementación.
2. **Operadores de búsqueda local.** Implementar (al menos) dos operadores de búsqueda local para el CVRP. Analizar la complejidad de los algoritmos, que debe ser consistente con la implementación. En caso de querer incorporar operadores adicionales, es posible interactuar con una herramienta de AI para el desarrollo.
3. **Métodos iniciales.** Proponer un método que combine una heurística constructiva con los operadores de búsqueda local.

4. **Experimentación y discusión.** Realizar una experimentación exhaustiva con las instancias de benchmark provistas, comparando los métodos propuestos. Los métodos a comparar deben ser (al menos) los siguientes:

- Cada heurística constructiva de manera independiente.
- Heurística constructiva + operador de búsqueda local. Ejemplo; Clarke & Wright + relocation; Clarke & Wright + swap.
- Heurística constructiva (alguna) + combinación de operadores de búsqueda local.

La experimentación debe considerar la calidad de las soluciones<sup>1</sup> así como también el tiempo de ejecución. Respecto al método propuesto en el ítem anterior, el diseño del mismo debe estar justificado por la experimentación.

5. **Diseño e implementación.** Aplicando conocimientos de materias previas, la entrega debe tener un diseño de clases acorde, definiendo aquellas que considere necesarias. Incluir casos de test para al menos 2 clases. Es posible utilizar herramientas de AI para el diseño y testing (no para la implementación), en cuyo caso se debe documentar claramente qué partes del código fueron generadas o influenciadas por la AI, así como también proporcionar un enlace compartible a la(s) conversación(es) con la herramienta como parte del informe.

6. **Informe, presentación de resultados y delivery del código.** La descripción de los algoritmos, los operadores, las decisiones de diseño, la implementación, el testing realizado, la presentación de resultados, instrucciones de compilación y ejecución.

La realización de los puntos anteriores comprenden el 85% de la nota final (i.e., 8.5 de los 10). Para obtener el 15% restante, se pide diseñar, implementar y evaluar al menos una estrategia de randomización y/o metaheurística.

### **Modalidad de entrega**

Se pide presentar el modelo y la experimentación en un informe de máximo 12 páginas que contenga:

- introducción al problema,
- descripción de los algoritmos implementados, según corresponda,
- consideraciones generales respecto a la implementación, incluyendo dificultades que hayan encontrado,
- resumen de resultados obtenidos en la experimentación,
- conclusiones, posibles mejoras y observaciones adicionales que consideren pertinentes.

Junto con el informe debe entregarse el código con la implementación del modelo. El mismo debe ser entendible, incluyendo comentarios que faciliten su corrección y ejecución.

### **Fechas de entrega**

---

<sup>1</sup>En caso de necesitarlo, en el siguiente [link](#) se encuentran los valores las mejores soluciones conocidas para cada instancia del conjunto  $x$ , y se proveen soluciones conocidas para las instancias restantes.

*Formato Electrónico:* **Viernes 27 de junio de 2025, 23:59 hs**, enviando el trabajo (informe + código) vía el campus virtual.

**Importante:** El horario es estricto. Las entregas recibidas después de la hora indicada serán considerados re-entrega.