

PROYECTO FINAL - INTELIGENCIA ARTIFICIAL

Integrantes:

Daniel Arias

André Ramirez

Santiago Rodriguez

Contexto:	2
Small-vrp:	2
Contiene la ubicación de los clientes y sus demandas, resuelve el problema con una flota de 2, 3 y 4 vehículos con sus respectivas capacidades.....	2
Medium-vrp:	4
Los vehículos tienen capacidad de 100.....	5
Análisis con Flota Limitada a Cuatro Rutas:.....	7
Punto de Inviabilidad con Capacidad de 150 Unidades:.....	8
Large-vrp:	10
Flota de 10 vehículos.....	10
Modificaciones en el número de vehículos.....	10
¿Cuáles serían las rutas y la flota óptima para este problema?.....	10
Referencias:	12

Contexto:

Una empresa de entregas dispone de una flota de vehículos con capacidad limitada y debe repartir paquetes a un conjunto de clientes ubicados en distintas direcciones. Todos los vehículos parten desde un único depósito y deben regresar al mismo al finalizar su ruta.

Cada cliente debe ser visitado exactamente una vez por un solo vehículo, y la cantidad de paquetes que transporta un vehículo no debe superar su capacidad.

Objetivo: Diseñar las rutas para todos los vehículos de manera que la distancia total recorrida sea mínima, cumpliendo con todas las restricciones de capacidad y asignación.

Se plantea un dataset en la carpeta VRP/data, resolver los problemas de acuerdo con:

Small-vrp:

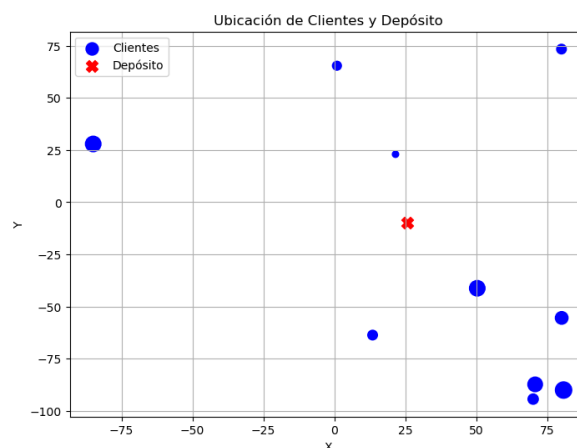
Contiene la ubicación de los clientes y sus demandas, resuelve el problema con una flota de 2, 3 y 4 vehículos con sus respectivas capacidades.

Se inició cargando la información de los clientes desde el archivo vrp.csv, la cual incluye sus coordenadas y demandas. Esta información fue convertida a un DataFrame para facilitar su manipulación y análisis. Se representó visualmente la distribución espacial de los clientes y el depósito para obtener una perspectiva general del problema.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   ID      11 non-null      int64
 1   X        11 non-null      float64
 2   Y        11 non-null      float64
 3   Demand  11 non-null      int64
dtypes: float64(2), int64(2)
memory usage: 484.0 bytes
```

	ID	X	Y	Demand
0	0	25.660036	-9.793401	0
1	1	80.011644	-55.493543	12
2	2	13.439953	-63.729201	7
3	3	21.501643	22.994351	3
4	4	-84.928462	27.941901	18

Gráfico 1: Describen el estado de los DataFrame.



Gráficos 2: Ubicación de los clientes con respecto al Depósito.

Posteriormente, se calcularon las demandas totales de los clientes y se compararon con las capacidades definidas en los archivos 2.vehicles.csv, 3.vehicles.csv y 4.vehicles.csv. En cada caso, la heurística agrupó clientes en rutas que respetan las capacidades asignadas.

Para el caso de 2 vehículos, cada uno con capacidad de 60, se cubrió la demanda total de 115, de acuerdo con el siguiente gráfico:

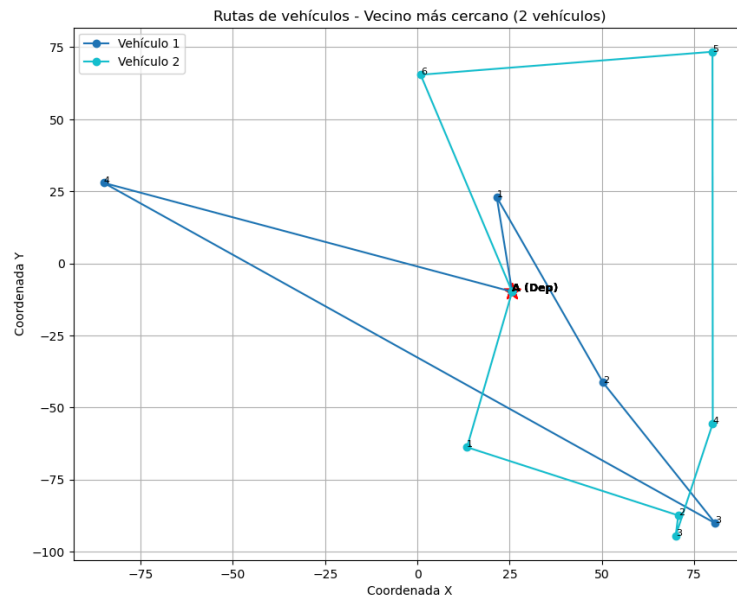


Gráfico 3: Primer caso - Ruta con 2 vehículos

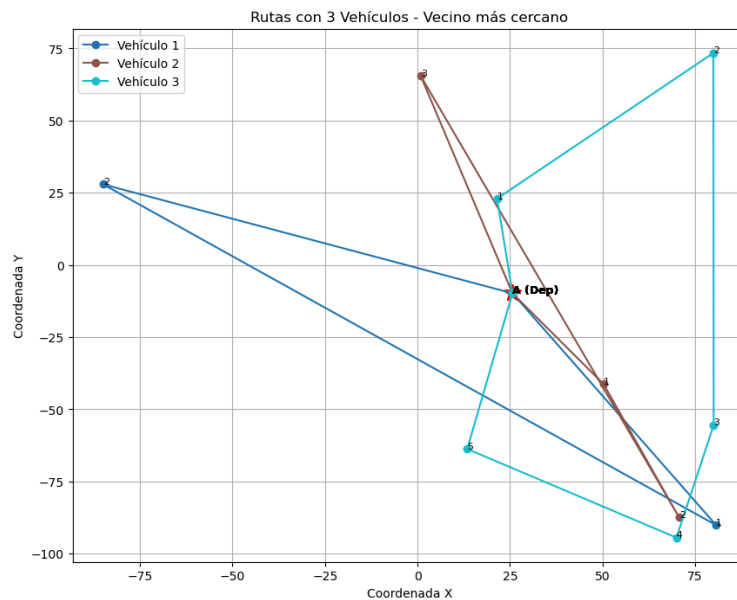


Gráfico 4: Segundo caso - Ruta con 3 vehículos (capacidad de carga 40 c/u)

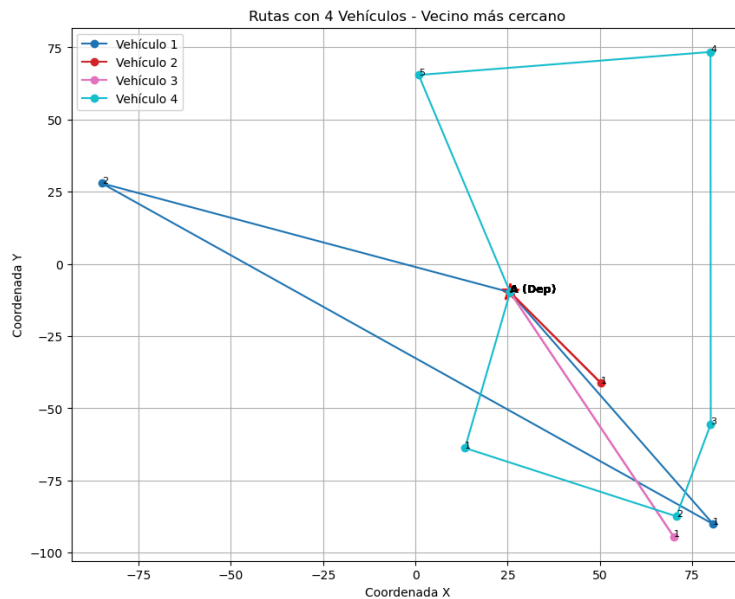


Gráfico 5: Tercer caso - Ruta con 4 vehículos (capacidad variada)

Las restricciones de capacidad influyen directamente en la cantidad de clientes que cada vehículo puede atender. A pesar de que un mayor número de vehículos puede mejorar el balance de carga, también incrementa los recorridos al depósito. Como se observa, el escenario con 2 vehículos resultó en la menor distancia total recorrida:

```
Comparativa de longitudes de ruta total:
2 Vehículos: 933.35 distancia recorrida
3 Vehículos: 1154.21 distancia recorrida
4 Vehículos: 1126.65 distancia recorrida
```

Gráfico 6: Suma total de la distancia recorrida.

Conclusiones:

- El vecino más cercano ofrece soluciones rápidas para VRP pequeños, pero su enfoque local limita su desempeño en problemas grandes. Para escalabilidad y mayor calidad de solución, se prefieren algoritmos genéticos o colonia de hormigas.
- Incrementar el número de vehículos no siempre mejora los resultados.
- La demanda, ubicación de los clientes y capacidad vehicular influyen directamente en la eficiencia de las rutas.
- La configuración con 2 vehículos de 60 unidades fue la más óptima en términos de distancia recorrida.

Medium-vrp

Cuál sería la flota necesaria para obtener las rutas más óptimas si se asume:

Los vehículos tienen capacidad de 100

El código híbrido que combina la heurística de vecinos cercanos con la optimización exacta mediante Pyomo ofrece soluciones potencialmente más óptimas en términos de distancia total recorrida, especialmente para los casos donde la capacidad de los vehículos está cerca del límite mínimo requerido (preguntas 1.1 y 1.2). Sin embargo, este enfoque tiene un costo computacional significativamente mayor y puede volverse inviable para instancias más grandes. En contraste, la versión puramente heurística proporciona soluciones rápidas y factibles, aunque no garantiza la optimalidad, siendo especialmente útil para obtener respuestas aproximadas en tiempos muy cortos. Para la pregunta 1.3 (punto de inviabilidad), ambos enfoques coinciden en el análisis teórico basado en la capacidad total y la demanda.

Los resultados muestran que no hay diferencias significativas entre las soluciones obtenidas con el enfoque híbrido (vecinos cercanos + Pyomo) y el puramente heurístico. Esto se debe principalmente a dos razones:

1. Calidad de la solución heurística inicial: La heurística de vecinos cercanos ya proporciona soluciones bastante eficientes para este problema en particular, dejando poco margen de mejora para la optimización exacta. Las rutas generadas por la heurística cumplen bien con las restricciones de capacidad y minimizan la distancia de manera intuitiva, por lo que Pyomo no encuentra mejoras significativas al refinar la solución inicial.
2. Complejidad del problema y limitaciones del solver: El solver GLPK utilizado con Pyomo tiene limitaciones para manejar problemas de optimización combinatoria complejos como el VRP dentro de tiempos razonables. Es probable que el solver no haya podido explorar suficientemente el espacio de soluciones para encontrar mejoras respecto a la solución heurística inicial, especialmente con los parámetros de tiempo y tolerancia (mipgap) configurados.

Conclusión práctica:

En este caso específico, la heurística de vecinos cercanos fue suficiente para obtener soluciones de calidad similar a las del enfoque híbrido, pero con un costo computacional mucho menor.

A continuación se presentan los resultados para este ejercicio:

```
Modelo híbrido - vecinos cercanos y pyomo

1.1. Flota necesaria con capacidad de 100:
Heurística: Solución con 7 vehículos, distancia 2467.917474105577
Número mínimo de vehículos necesarios: 7
Rutas óptimas:
Vehículo 1: [0, 14, 35, 40, 41, 12, 18, 22, 13, 42, 3, 36, 11, 0]
Vehículo 2: [0, 19, 17, 15, 6, 29, 28, 20, 0]
Vehículo 3: [0, 39, 24, 16, 34, 30, 37, 5, 0]
Vehículo 4: [0, 32, 2, 38, 21, 45, 43, 9, 25, 0]
Vehículo 5: [0, 33, 27, 7, 31, 48, 23, 1, 0]
Vehículo 6: [0, 26, 4, 47, 49, 46, 10, 0]
Vehículo 7: [0, 8, 44, 0]

1.2. Capacidad necesaria para 4 rutas:
Heurística: Solución con capacidad 160, distancia 1830.485411937124
Capacidad mínima requerida: 160
Rutas óptimas:
Vehículo 1: [0, 14, 35, 40, 41, 12, 18, 22, 13, 42, 3, 36, 19, 17, 11, 25, 31, 7, 48, 20, 0]
Vehículo 2: [0, 39, 24, 16, 34, 30, 37, 47, 49, 4, 26, 0]
Vehículo 3: [0, 29, 6, 15, 23, 1, 10, 9, 8, 44, 28, 0]
Vehículo 4: [0, 32, 2, 38, 21, 45, 43, 46, 33, 5, 27, 0]

1.3. Punto de inviabilidad con capacidad máxima de 150:
El problema se vuelve inviable cuando se requieren más de 4 vehículos

Flota ideal para capacidad de 150:
Heurística: Solución con 5 vehículos, distancia 2016.6013587596688
Número ideal de vehículos: 5
Rutas óptimas:
Vehículo 1: [0, 14, 35, 40, 41, 12, 18, 22, 13, 42, 3, 36, 19, 17, 11, 25, 31, 7, 5, 20, 0]
Vehículo 2: [0, 39, 24, 16, 34, 30, 37, 47, 49, 4, 0]
Vehículo 3: [0, 29, 6, 15, 23, 1, 10, 9, 8, 44, 0]
Vehículo 4: [0, 32, 2, 38, 21, 45, 43, 46, 26, 33, 0]
Vehículo 5: [0, 27, 48, 0]
```

Gráfico 7: resolución utilizando modelo híbrido

```

1.1. Flota necesaria con capacidad de 100:
Solución con 7 vehículos, distancia total: 2467.92
Número mínimo de vehículos necesarios: 7
Rutas encontradas:
Vehículo 1: ['Depósito (0)', 'Cliente 14 (d:10)', 'Cliente 35 (d:8)', 'Cliente 40 (d:6)', 'Cliente 41 (d:3)', 'Cliente 12 (d:4)', 'Cliente 18 (d:11)', 'Cliente 22 (d:8)', 'Cliente 13 (d:14)', 'Cliente 42 (d:13)', 'Cliente 3 (d:3)', 'Cliente 36 (d:14)', 'Cliente 11 (d:5)', 'Depósito (0)']
Vehículo 2: ['Depósito (0)', 'Cliente 19 (d:12)', 'Cliente 17 (d:10)', 'Cliente 15 (d:20)', 'Cliente 6 (d:18)', 'Cliente 29 (d:20)', 'Cliente 28 (d:14)', 'Cliente 20 (d:4)', 'Depósito (0)']
Vehículo 3: ['Depósito (0)', 'Cliente 39 (d:20)', 'Cliente 24 (d:13)', 'Cliente 16 (d:18)', 'Cliente 34 (d:19)', 'Cliente 30 (d:16)', 'Cliente 37 (d:8)', 'Cliente 5 (d:6)', 'Depósito (0)']
Vehículo 4: ['Depósito (0)', 'Cliente 32 (d:17)', 'Cliente 2 (d:7)', 'Cliente 38 (d:18)', 'Cliente 21 (d:8)', 'Cliente 45 (d:19)', 'Cliente 43 (d:20)', 'Cliente 9 (d:8)', 'Cliente 25 (d:3)', 'Depósito (0)']
Vehículo 5: ['Depósito (0)', 'Cliente 33 (d:13)', 'Cliente 27 (d:17)', 'Cliente 7 (d:7)', 'Cliente 31 (d:9)', 'Cliente 48 (d:15)', 'Cliente 23 (d:19)', 'Cliente 1 (d:12)', 'Depósito (0)']
Vehículo 6: ['Depósito (0)', 'Cliente 26 (d:9)', 'Cliente 4 (d:18)', 'Cliente 47 (d:18)', 'Cliente 49 (d:20)', 'Cliente 46 (d:17)', 'Cliente 10 (d:16)', 'Depósito (0)']
Vehículo 7: ['Depósito (0)', 'Cliente 8 (d:20)', 'Cliente 44 (d:12)', 'Depósito (0)']
Distancia total: 2467.92

1.2. Capacidad necesaria para 4 rutas:
Solución con capacidad 160, distancia total: 1830.49
Capacidad mínima requerida: 160
Rutas encontradas:
Vehículo 1: ['Depósito (0)', 'Cliente 14 (d:10)', 'Cliente 35 (d:8)', 'Cliente 40 (d:6)', 'Cliente 41 (d:3)', 'Cliente 12 (d:4)', 'Cliente 18 (d:11)', 'Cliente 22 (d:8)', 'Cliente 13 (d:14)', 'Cliente 42 (d:13)', 'Cliente 3 (d:3)', 'Cliente 36 (d:14)', 'Cliente 19 (d:12)', 'Cliente 17 (d:10)', 'Cliente 11 (d:5)', 'Cliente 25 (d:3)', 'Cliente 31 (d:9)', 'Cliente 7 (d:7)', 'Cliente 48 (d:15)', 'Cliente 28 (d:4)', 'Depósito (0)']
Vehículo 2: ['Depósito (0)', 'Cliente 39 (d:20)', 'Cliente 24 (d:13)', 'Cliente 16 (d:18)', 'Cliente 34 (d:19)', 'Cliente 30 (d:16)', 'Cliente 37 (d:8)', 'Cliente 47 (d:18)', 'Cliente 49 (d:20)', 'Cliente 4 (d:18)', 'Cliente 26 (d:9)', 'Depósito (0)']
Vehículo 3: ['Depósito (0)', 'Cliente 29 (d:20)', 'Cliente 6 (d:18)', 'Cliente 15 (d:20)', 'Cliente 23 (d:19)', 'Cliente 1 (d:12)', 'Cliente 10 (d:16)', 'Cliente 9 (d:8)', 'Cliente 8 (d:20)', 'Cliente 44 (d:12)', 'Cliente 28 (d:14)', 'Depósito (0)']
Vehículo 4: ['Depósito (0)', 'Cliente 32 (d:17)', 'Cliente 2 (d:7)', 'Cliente 38 (d:18)', 'Cliente 21 (d:8)', 'Cliente 45 (d:19)', 'Cliente 43 (d:20)', 'Cliente 46 (d:17)', 'Cliente 33 (d:13)', 'Cliente 5 (d:6)', 'Cliente 27 (d:17)', 'Depósito (0)']
Distancia total: 1830.49

1.3. Punto de inviabilidad con capacidad máxima de 150:
El problema se vuelve inviable cuando se requieren más de 4 vehículos

Flota ideal para capacidad de 150:
Solución con 5 vehículos, distancia total: 2016.60
Número ideal de vehículos: 5
Rutas encontradas:
Vehículo 1: ['Depósito (0)', 'Cliente 14 (d:10)', 'Cliente 35 (d:8)', 'Cliente 40 (d:6)', 'Cliente 41 (d:3)', 'Cliente 12 (d:4)', 'Cliente 18 (d:11)', 'Cliente 22 (d:8)', 'Cliente 13 (d:14)', 'Cliente 42 (d:13)', 'Cliente 3 (d:3)', 'Cliente 36 (d:14)', 'Cliente 19 (d:12)', 'Cliente 17 (d:10)', 'Cliente 11 (d:5)', 'Cliente 25 (d:3)', 'Cliente 31 (d:9)', 'Cliente 7 (d:7)', 'Cliente 5 (d:6)', 'Cliente 28 (d:4)', 'Depósito (0)']
Vehículo 2: ['Depósito (0)', 'Cliente 39 (d:20)', 'Cliente 24 (d:13)', 'Cliente 16 (d:18)', 'Cliente 34 (d:19)', 'Cliente 30 (d:16)', 'Cliente 37 (d:8)', 'Cliente 47 (d:18)', 'Cliente 49 (d:20)', 'Cliente 4 (d:18)', 'Depósito (0)']
Vehículo 3: ['Depósito (0)', 'Cliente 29 (d:20)', 'Cliente 6 (d:18)', 'Cliente 15 (d:20)', 'Cliente 23 (d:19)', 'Cliente 1 (d:12)', 'Cliente 10 (d:16)', 'Cliente 9 (d:8)', 'Cliente 8 (d:20)', 'Cliente 44 (d:12)', 'Depósito (0)']
Vehículo 4: ['Depósito (0)', 'Cliente 32 (d:17)', 'Cliente 28 (d:14)', 'Cliente 2 (d:7)', 'Cliente 38 (d:18)', 'Cliente 21 (d:8)', 'Cliente 45 (d:19)', 'Cliente 43 (d:20)', 'Cliente 46 (d:17)', 'Cliente 26 (d:9)', 'Cliente 33 (d:13)', 'Depósito (0)']
Vehículo 5: ['Depósito (0)', 'Cliente 27 (d:17)', 'Cliente 48 (d:15)', 'Depósito (0)']
Distancia total: 2016.60

```

Gráfico 8: resolución utilizando modelo únicamente con heurística

La configuración más eficiente con vehículos de 100 unidades de capacidad requiere **siete unidades**. Estos vehículos cubren un total de **2,467.92 unidades de distancia**, distribuyendo la demanda de manera relativamente uniforme entre ellos. Específicamente:

- El **Vehículo 1** atiende a 12 clientes, transportando un total de 99 unidades.
- El **Vehículo 2** sirve a 7 clientes, con una carga de 98 unidades.
- El **Vehículo 7**, con solo 32 unidades para 2 clientes (los remotos 8 y 44), ilustra que la dispersión geográfica, más que la capacidad de los vehículos, es un factor limitante en algunos casos.

La ruta más extensa, la del Vehículo 1 con 14 paradas, se caracteriza por clientes cercanos con necesidades individuales pequeñas. En contraste, rutas como la del Vehículo 3 consolidan clientes con demandas elevadas cerca del punto de origen.

Análisis con Flota Limitada a Cuatro Rutas:

Para completar todas las entregas utilizando únicamente **cuatro vehículos**, cada uno debe tener una capacidad mínima de **160 unidades**, resultando en una distancia total recorrida de **1,830.49 unidades**. Los datos clave son:

1. El **Vehículo 2** opera al límite de su capacidad, transportando exactamente 160 unidades (la suma de las demandas de sus 10 clientes).
2. Las rutas se han optimizado por zonas geográficas:
 - El Vehículo 1 cubre la región noreste (clientes 3 al 48).

- El Vehículo 4 se encarga del sector suroeste (clientes 2 al 46).
3. El Vehículo 3 gestiona la mayor demanda acumulada, 159 unidades, incluyendo clientes con necesidades significativas como el 29 y el 15 (ambos con una demanda de 20 unidades).

Punto de Inviabilidad con Capacidad de 150 Unidades:

El problema no puede resolverse con **cuatro o menos vehículos** si cada uno tiene una capacidad de 150 unidades. La razón principal es que la **demanda total (627 unidades) excede la capacidad combinada de cuatro vehículos (600 unidades)**, dejando 27 unidades sin entregar.

Una solución viable se encuentra al utilizar **cinco vehículos** (con una capacidad total de 750 unidades). En este escenario:

- El **Vehículo 2** se acerca al límite de su capacidad con 149 unidades.
- El **Vehículo 5** transporta una carga mucho menor (32 unidades), lo que sugiere nuevamente que la distribución geográfica de los clientes es una restricción más importante que la capacidad en sí misma.
- La distancia total recorrida aumenta a **2,016.60 unidades** en comparación con la solución de cuatro vehículos de mayor capacidad, lo que evidencia una compensación entre el tamaño de la flota y la eficiencia en términos de distancia.

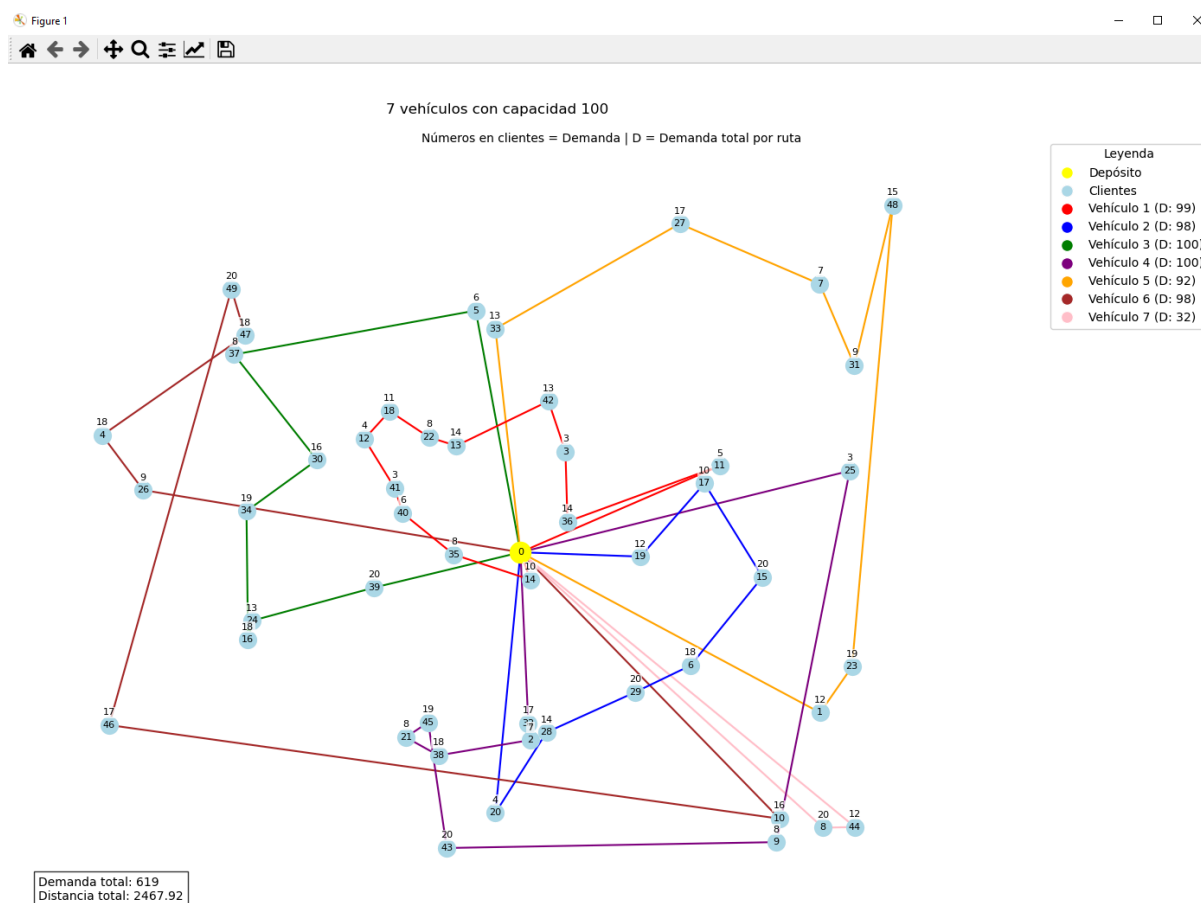


Gráfico 9: grafo de resolución pregunta 1 (7 vehículos)

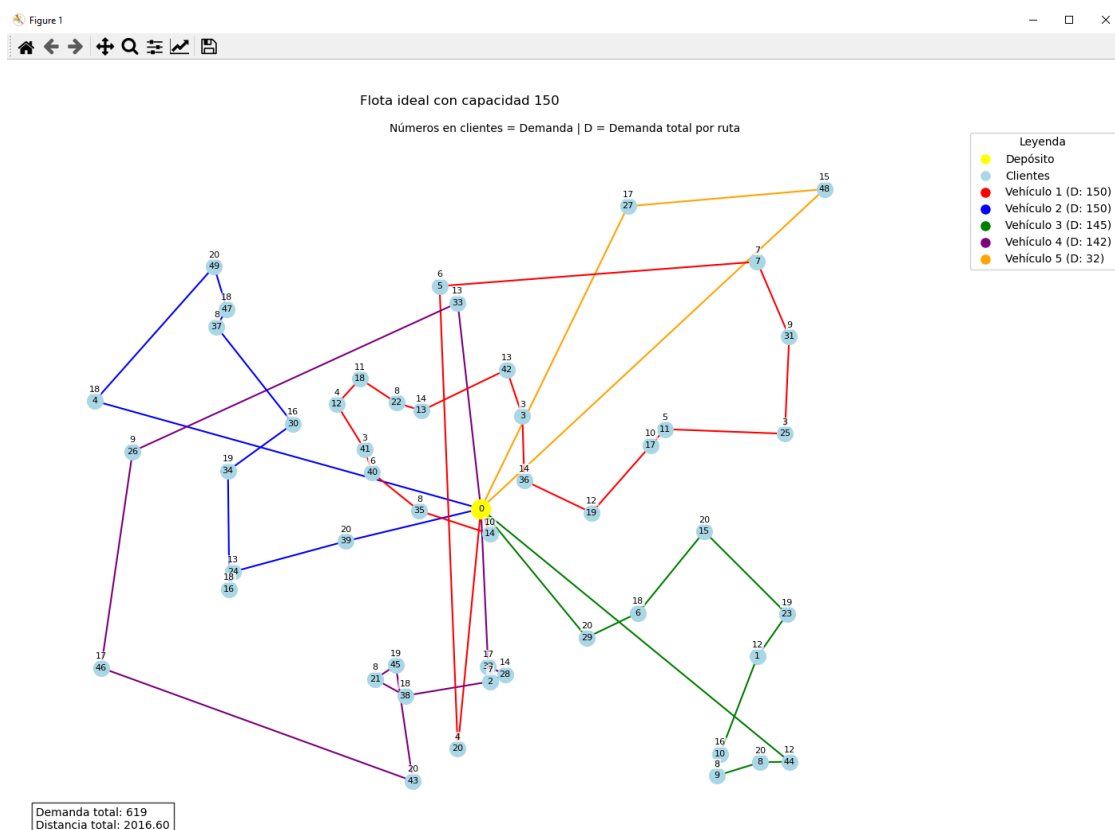
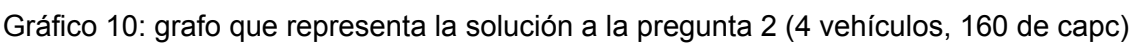
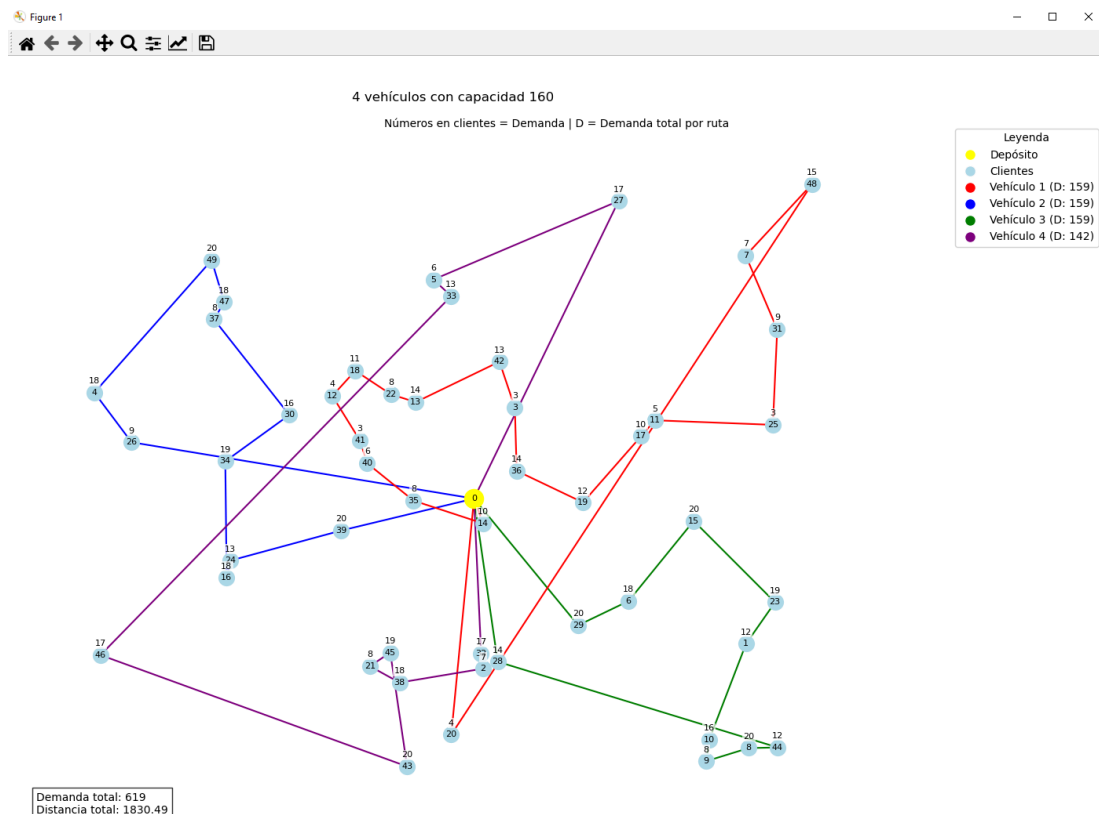


Gráfico 11: grafo que representa la solución a la pregunta 3 (4 o menos vehículos)

Large-vrp:

Aplicar cualquier metaheurística o algoritmo que permita resolver el problema en tiempo razonable:

Flota de 10 vehículos

Para resolver el problema utilizando una flota de 10 vehículos, se determinó que cada uno debe contar con una capacidad mínima de 119 unidades. No obstante, este valor puede reducirse si se incrementa la capacidad individual de los vehículos.

Modificaciones en el número de vehículos

Aumentar el número de vehículos con la misma capacidad no afectaría directamente la eficiencia de las rutas, pero sí podría representar un desperdicio de recursos para la empresa, ya que no se estaría aprovechando completamente la capacidad de cada unidad. En cambio, reducir la cantidad de vehículos disponibles volvería inviable la solución del problema, dado que la demanda total de los clientes superaría la capacidad máxima que la flota podría cubrir.

¿Cuáles serían las rutas y la flota óptima para este problema?

En este ejercicio se implementaron dos enfoques. El primero consistió en modelar el problema como un programa de optimización utilizando Pyomo y el solver GLPK. Sin embargo, al aumentar la cantidad de ciudades y vehículos, la región factible del problema crece exponencialmente, lo que incrementa la carga computacional. Por esta razón, se optó por una segunda estrategia basada en la metaheurística Clarke-Wright Savings (Clark y Wright, 1964). Aunque este enfoque no garantiza una solución óptima, permite obtener resultados viables en tiempos reducidos. Para aproximarse a una solución óptima con este tipo de algoritmos, es fundamental establecer condiciones realistas y coherentes con la capacidad operativa de la flota.

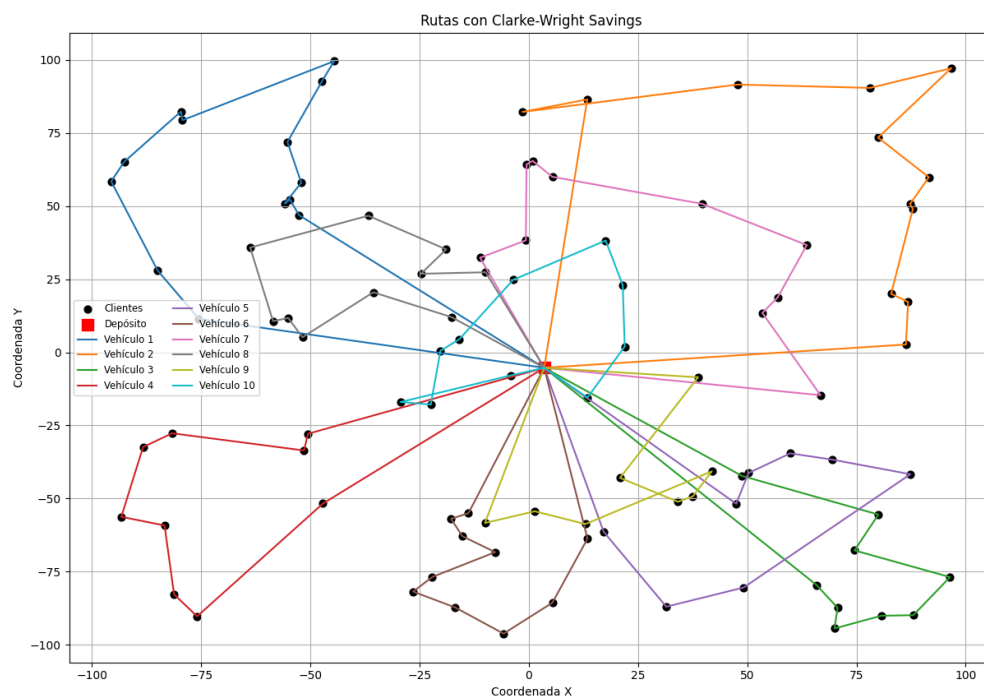


Gráfico 12: VRP resuelto para una flota de 10 vehículos.

```

+ 13455: mip =      not found yet >= 1.195817097e+03      (95; 0)
Time used: 307.0 secs.  Memory used: 180.8 Mb.
+ 13486: mip =      not found yet >= 1.195817097e+03      (97; 0)
+ 13548: mip =      not found yet >= 1.195817097e+03      (99; 0)
+ 13560: mip =      not found yet >= 1.195817097e+03     (101; 0)
+ 13660: mip =      not found yet >= 1.195817097e+03     (103; 0)
+ 13735: mip =      not found yet >= 1.195817097e+03     (104; 0)
+ 13852: mip =      not found yet >= 1.195817097e+03     (105; 0)
+ 13870: mip =      not found yet >= 1.195817097e+03     (107; 0)
+ 13932: mip =      not found yet >= 1.195817097e+03     (109; 0)
+ 14016: mip =      not found yet >= 1.195817097e+03     (111; 0)
+ 14030: mip =      not found yet >= 1.195817097e+03     (113; 0)
Time used: 369.6 secs.  Memory used: 181.3 Mb.
+ 14044: mip =      not found yet >= 1.195817097e+03     (115; 0)
+ 14064: mip =      not found yet >= 1.195817097e+03     (117; 0)
+ 14112: mip =      not found yet >= 1.195817097e+03     (119; 0)
+ 14129: mip =      not found yet >= 1.195817097e+03     (121; 0)
+ 14155: mip =      not found yet >= 1.195817097e+03     (123; 0)
+ 14168: mip =      not found yet >= 1.195817097e+03     (125; 0)
+ 14366: mip =      not found yet >= 1.195817097e+03     (126; 0)
+ 14542: mip =      not found yet >= 1.195817097e+03     (128; 0)
+ 14576: mip =      not found yet >= 1.195817097e+03     (130; 0)
Time used: 431.6 secs.  Memory used: 181.4 Mb.
+ 14605: mip =      not found yet >= 1.195817097e+03     (132; 0)
+ 14707: mip =      not found yet >= 1.195817097e+03     (134; 0)
+ 14756: mip =      not found yet >= 1.195817097e+03     (135; 0)
+ 14771: mip =      not found yet >= 1.195817097e+03     (136; 0)
+ 14784: mip =      not found yet >= 1.195817097e+03     (137; 0)
+ 14845: mip =      not found yet >= 1.195817097e+03     (139; 0)
+ 14951: mip =      not found yet >= 1.195817097e+03     (140; 0)
+ 14975: mip =      not found yet >= 1.195817097e+03     (142; 0)
+ 14999: mip =      not found yet >= 1.195817097e+03     (144; 0)
Time used: 493.6 secs.  Memory used: 181.6 Mb.
+ 15070: mip =      not found yet >= 1.195817097e+03     (146; 0)
+ 15134: mip =      not found yet >= 1.195817097e+03     (148; 0)
+ 15148: mip =      not found yet >= 1.195817097e+03     (150; 0)
+ 15168: mip =      not found yet >= 1.195817097e+03     (152; 0)
+ 15187: mip =      not found yet >= 1.195817097e+03     (154; 0)

```

Gráfico 13: Registro del tiempo de optimización empleado por el solver GLPK.

Referencias

Clarke, G., y Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4), 568–581. <https://doi.org/10.1287/opre.12.4.568>