Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Ingeniería logística: problema de ruta de vehículos con capacidades y consistencias temporales

Logistics engineering: vehicle route problems with capacity and time consistencies.

Antonella Sofía García Alvarez

La Laguna, *12* de julio de 2023

D. **Juan José Salazar González**, con N.I.F. 43.356.435-D profesor Catedrático de Universidad adscrito al Departamento de Matemáticas, Estadística e Investigación Operativa de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

"Ingeniería logística: problema de ruta de vehículos con capacidades y consistencias temporales"

Ha sido realizada bajo su dirección por Dña**. Antonella Sofía García Alvarez**, con N.I.F. 43.858.324-S.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 14 de junio de 2023

Agradecimientos

Primeramente, a mi tutor de trabajo de fin de grado ya que me ha ayudado enormemente a mantenerme en página para obtener todo a tiempo y con esmero.

También a mis padres por el apoyo todos estos años que he estado en la carrera tanto financiero como mental.

A mi pareja que me ha apoyado en todo.

A mis compañeros de clase por cada día brindarme risas y ayuda ya que esta carrera sin amigos y compañeros es mucho más complicada

A mis profesores a lo largo de la carrera ya que me han brindado una educación excelente y me han llevado a convertirme en una ingeniera capaz.

Licencia

Resumen

La presente memoria de Trabajo de Fin de Grado abordará el problema de ruta de vehículos con restricciones de capacidad y días consistentes. Este problema consiste en encontrar las mejores rutas para un conjunto de vehículos que deben visitar destinos teniendo en cuenta las limitaciones de capacidad de los vehículos disponibles y los diferentes días de entrega establecidos para mantener una programación de entrega consistente.

Para afrontar el problema, se utilizarán técnicas de programación de alto nivel en un lenguaje llamado julia. Julia nos ofrece herramientas especializadas como paquetes y librerías propias del lenguaje para solucionar problemas de optimización. Esto nos permitirá encontrar soluciones óptimas que cumplan con todas las restricciones mencionadas, mientras que también se minimizan los costos totales de las rutas, esto implica considerar factores como la capacidad de carga de los vehículos, como se ha mencionado anteriormente, los diferentes días de entrega requeridos y las distancias entre los destinos y el depósito.

En resumen, se resolverá el desafío del enrutamiento de vehículos con restricciones de capacidad y días consistentes. Utilizando el lenguaje de programación Julia, especializado en operaciones matemáticas y optimización, se buscará encontrar soluciones óptimas que cumplan las restricciones mencionadas y optimicen el proceso de entrega de los vehículos a los clientes.

Palabras clave: Enrutamiento de vehículos, Restricciones, Programación de alto nivel, Lenguaje Julia, Optimización, Soluciones óptimas.

Abstract

This bachelor's Thesis report will address the problem of vehicle routes with capacity restrictions and consistent days. This problem consists of finding the best routes for a set of vehicles that must visit destinations taking into account the capacity limitations of the available vehicles and the different delivery days established to maintain a consistent delivery schedule.

To deal with this problem, high-level programming techniques will be used in a language called julia. Julia offers us specialized tools such as language packages and libraries to solve optimization problems. This will allow us to find optimal solutions that comply with all the mentioned restrictions, while also minimizing the total costs of the routes, this implies considering factors such as the loading capacity of the vehicles, as mentioned above, the different delivery days required and the distances between the destinations and the depot.

In summary, the challenge of routing vehicles with capacity constraints and consistent days will be solved, using the Julia programming language, specialized in mathematical operations and optimization, it will seek to find optimal solutions that meet the aforementioned restrictions and optimize the delivery process of vehicles to customers.

Keywords: Vehicle routing, Constraints, High-level programming, Julia language, Optimization, Optimal solutions.

Índice general

Índice de figuras

Índice de tablas

Capitulo 1.

Introducción

1.1 ¿Qué es el VRP?

El problema de enrutamiento de vehículos o Vehicle Routing Problem (VRP) es un problema muy interesante que se encuentra en el campo de la logística y la gestión de vehículos, el cual utiliza Optimización Combinatoria. Básicamente, se trata de encontrar la mejor ruta para un grupo de vehículos que tiene para visitar varios lugares o clientes para entregar mercancías o servicios. Una característica implícita en todo VRP es que los clientes tienen demandas conocidas y los vehículos capacidades máximas. También existen otros factores que en algunas aplicaciones particulares (no en este TFG) puede convenir tener en cuenta, como por ejemplo, las ventanas de tiempo en las que se deben hacer las entregas ya que pueden haber limitaciones donde los clientes no quieren tener entregar inconsistentes o puede que existan horarios en los cuales los establecimientos están cerrados, la disponibilidad temporal de vehículos, o de distancia máxima que puede recorrer cada vehículo.

Figura 1.1: Representación general del problema de ruta de vehículos

La resolución eficiente de este problema tiene un impacto significativo en la reducción de costos de transporte, la optimización de la utilización de los vehículos y la mejora de la eficiencia en la distribución de bienes y servicios. Hay diversos enfoques y algoritmos para abordarlo. Desde métodos exactos hasta métodos heurísticos y metaheurísticos, que buscan soluciones óptimas en un tiempo razonable. En definitiva, el VRP es útil porque permite mejorar la productividad y la eficiencia de las operaciones de transporte y distribución, lo que se traduce en ahorros de costos y una mejor calidad de servicio para las empresas.

1.1.1 Antecedentes del VRP

El VRP surgió debido a la necesidad de resolver un desafío fundamental de la logística: optimizar las rutas de los vehículos para satisfacer la demanda de los clientes de manera eficiente. A medida que las empresas crecían y se volvían más complejas, surgieron dificultades para planificar las rutas de entrega, recolección o transporte de manera manual.

Se originó en la década de los 50, cuando investigadores y profesionales de la logística comenzaron a abordar el problema de cómo asignar y programar eficientemente un conjunto de vehículos para atender a múltiples destinos con diversas restricciones. La complejidad del problema radica en encontrar la mejor combinación de rutas, minimizando los costos.

El crecimiento del comercio, transporte de mercancías y los servicios de entrega ha aumentado con gran rapidez y relevancia los últimos años en diferentes industrias. A medida que las empresas buscan reducir costos, mejorar la eficiencia y satisfacer las expectativas de los clientes, el VRP se ha convertido en una herramienta invaluable para la planificación estratégica y la toma de decisiones de las industrias en cuanto a la logística.

Con el avance de la tecnología y la disponibilidad de poder computacional, han surgido numerosos enfoques y algoritmos para resolver el VRP de manera más rápida y eficiente. Estos métodos incluyen técnicas de optimización, algoritmos genéticos, heurísticas y enfoques basados en inteligencia artificial.

1.1.2 Visión del VRP en el mundo real

El VRP se puede aplicar a diversos escenarios reales que añadirán la tarea de buscar una solución más compleja ya que se ata a restricciones más limitadas, como la ventana de tiempo en las que no se puede visitar a un cliente. Aquí hay algunos ejemplos de cómo se puede ver esto en un escenario real:

- Distribución de bienes: Utilizan el VRP para planificar las rutas de los vehículos y entregar la mercancía de manera eficiente. Esto implica que las rutas sean lo más cortas y rápidas posibles, minimizandolos costos de transporte y aprovechando al máximo la capacidad de los vehículos.
- Recolección de residuos: Estas empresas utilizan el VRP para optimizar las rutas de sus camiones de recolección, para recoger de forma eficiente, minimizando la distancia y los costos operativos, al mismo tiempo que garantiza que todos los contenedores se visitan al menos una vez dentro del horario previsto.
- Servicios de entrega: Empresas de entrega a domicilio, como las de comida o comercio electrónico, aplican el uso de esta herramienta para optimizar las rutas de los repartidores, para que los tiempos de espera de los clientes sean mínimos.
- Transporte de pasajeros: El objetivo es planificar las rutas de los vehículos para maximizar la eficiencia y para minimizar los tiempos de viaje garantizando las paradas requeridas dentro del horario establecido.

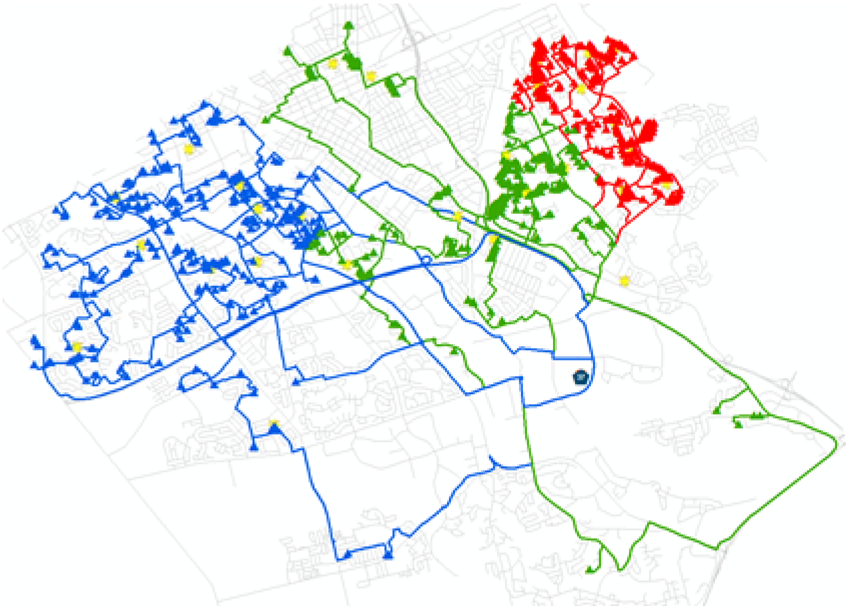

Figura 1.2: Representación en la vida real del VRP

El VRP se ve en escenarios reales como una herramienta para optimizar la planificación de rutas de vehículos mejorando la eficiencia, ya sea para la distribución de bienes, la recolección de residuos, los servicios de entrega o el transporte de pasajeros, entre otros.

1.1.3 Estado actual

Existen varias investigaciones actuales y se han realizado avances significativos en este campo. El avance más significativo siempre cae sobre programas o software que ayude a resolver este problema ya que tiene gran dificultad matemática y aún sigue activo en investigación, ya que siempre surgen nuevos puntos a tener en cuenta como por ejemplo, teniendo en cuenta el tiempo real, la duración de los servicios, el costo del combustible de los vehículos, entre otros.

Para poner un ejemplo, se puede mencionar las herramientas de Google o Google OR-Tools, es cual es un paquete de software de código abierto que sirve para solucionar problemas de optimización como el VRP. Permite utilizar múltiples solucionadores para resolver estos problemas y soporta varios lenguajes de programación como python, C++, java o C#. Es un software muy útil, si se sabe usar, para encontrar soluciones eficientes y rentables para las operaciones logísticas y de transporte.

1.1.4 Optimización combinatorial en la informática

Este área de la ciencia se dedica a encontrar las mejores soluciones posibles clasificándolas dentro de un conjunto de varias soluciones. En el ámbito de la ingeniería informática, tiene un uso fundamental para encontrar resultados en problemas de alta complejidad.

Capitulo 2.

VRP con consistencias temporales

2.1. Definición del problema

En el VRP con restricciones de capacidad y días consistentes se busca determinar las rutas óptimas para un conjunto de vehículos que deben realizar entregas a múltiples clientes. Las restricciones de capacidad implica que cada vehículo tiene una capacidad máxima de carga que no puede ser excedida. Al definir el problema, denotamos un conjunto de clientes como C, y la capacidad máxima de cada vehículo como Q, entonces debemos asegurarnos de que la carga total de cada vehículo en las rutas no supere la capacidad Q.

Por otro lado, las restricciones de días consistentes indican que existen diferentes días de entrega con tiempos consistentes. Cada cliente tiene una ventana de tiempo específica en la cual debe ocurrir la visita del vehículo desde el depósito que debe ser siempre la misma dentro de un margen. Además, cada cliente tiene una demanda que representa la cantidad de productos que se deben entregar respectivamente.

El objetivo de este problema es encontrar las rutas más óptimas que mantengan las restricciones establecidas minimizando los costos totales, como la distancia o el tiempo empleado. Además, se deben tener en cuenta otros factores como la distancia entre los clientes, la distancia desde el punto de partida, el tiempo necesario para cada entrega, el volumen de los vehículos y las demandas de los clientes, como se ha mencionado anteriormente.

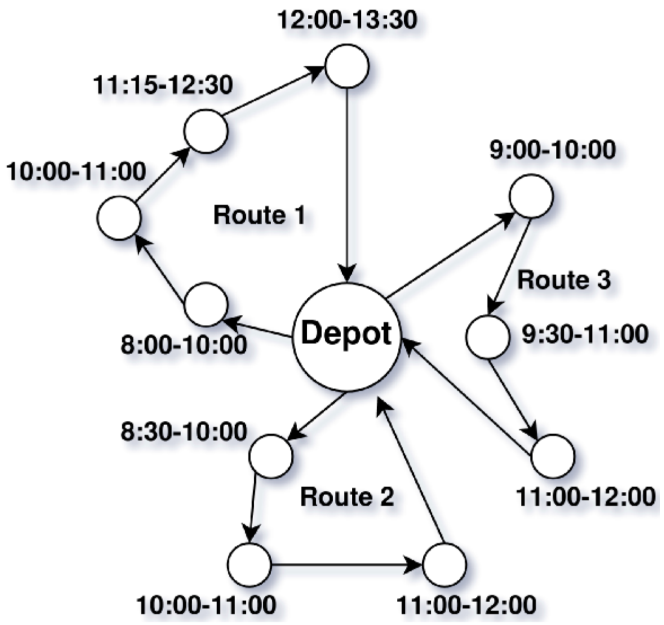

Figura 2.1: Representación del problema de ruta de vehículos con restricción temporal

2.2. Importancia del VRP con restricciones de capacidad y días consistentes.

Este problema es muy relevante en el ámbito de la logística y distribución de mercancías. Optimizar las rutas de transporte en estas condiciones puede generar beneficios bastante significativos, tanto en términos de eficiencia económica como satisfacción general del cliente.

Tener una correcta gestión de capacidad de carga de los vehículos disponibles evita situaciones de sobrecarga o de mala utilización del espacio, lo que influye altamente en los costos de las operaciones. Además, cumplir con la consistencia temporal en los días establecidos asegura que las entregas siempre sean esperadas y recibidas correctamente por los clientes, mejorando la satisfacción y evitando los viajes innecesarios, además teniendo un impacto menor en las emisiones contaminantes y mejorando la sostenibilidad de la empresa.

2.3. Relevancia en la actualidad

Existen varios factores que hacen que el problema de enrutamiento de vehículos con restricción de carga y consistencia de días sea relevante hoy en día.

1. *Eficiencia en la distribución*. Las empresas buscan optimizar sus operaciones logísticas, las restricciones de capacidad y consistencia de días permiten mejorar la eficiencia de la entrega de mercancía. La planificación de rutas permite reducir los costos de transporte, minimizar el tiempo de entrega y maximizar la capacidad de los vehículos disponibles.
2. *Satisfacción del cliente*. Cumplir con los horarios y tener consistencia en las entregas en los días determinados es altamente importante para garantizar la distribución a los clientes de manera satisfactoria ya que siempre se hará en el horario preferido del cliente, mejorando así la experiencia del comprador.
3. *Crecimiento del e-commerce*. El e-commerce o comercio electrónico está en pleno esplendor, la entrega a domicilio o empresas se ha vuelto crítica. Las empresas que ofrecen entrega se enfrentan a grandes retos logísticos y el VRP con restricciones de capacidad y días consistentes proporciona una solución específica que garantiza la entrega en el tiempo esperado por los clientes.

2.4. Visión matemática del problema

Al ver este problema en matemática algebraica, este tiene los siguientes parámetros:

- Conjunto de ubicaciones (incluyendo el depósito): U
  - Número total de ubicaciones, este incluye todos los clientes y el depósito.
- Demanda del cliente: K_{id}
  - Representa la demanda máxima o el máximo de productos que cada cliente tiene o puede requerir.
- Capacidad de Vehículos: Q
  - La cantidad de capacidad máxima que tiene cada vehículo para transportar.
- Conjunto de Días: D
  - Cada día se refiere a un día del calendario, por ejemplo Lunes, Martes o Viernes. Permite modelar situaciones en las que las restricciones hacen variar si existe un servicio o no a un cliente en el día.
- Clientes: C
  - Es el conjunto de clientes, excluyendo el depósito.
- Valor grande para la restricción de carga: Md
- Valor grande para la restricción de tiempos de llegada: Mt
- Probabilidad de que el cliente tenga una demanda: Pd
- Longitud máxima entre las ubicaciones: R
- Tiempo máximo que un cliente puede esperar: L
- Coordenadas: $c_i = (x_i, y_i)$
  - Un conjunto de pares de coordenadas que representan la localización de cada cliente.
- Distancias: f_{ij}
  - Una matriz que almacena las distancias entre cada par de ubicaciones i y j.

Las variables de decisión en el contexto del problema de enrutamiento de vehículos con restricciones de capacidad y días consistentes son aquellas que determinan las características de las soluciones óptimas, como las rutas de los vehículos, los tiempos de llegada y las cargas asociadas.

- $x_{ijd} = \{1, 0\}$
  - X es una variable binaria que toma valor 1 si el vehículo realiza el viaje desde la ubicación *i* a la ubicación *j* en el día *d*, y el 0 en el caso contrario.
- t_{id}
  - El tiempo de llegada es una variable que representa el tiempo de llegada al cliente *i* en el día *d*.
- q_{id}
  - Es una variable que indica la capacidad del vehículo al llegar al cliente *i* en el día *d*.

La función objetivo del modelo matemático del problema de enrutamiento de vehículos con restricciones de capacidad y días consistentes tiene como objetivo minimizar la distancia total recorrida por los vehículos. La función objetivo se define de la siguiente manera:

$$\text{Minimizar} \sum_{i} \sum_{j} \sum^{d} K_{ij}^{d} * x_{ij}^{d}$$

El objetivo de minimizar la distancia total recorrida es importante ya que se busca encontrar las rutas más eficientes y con el menor costo posible. Se optimizan aspectos como el consumo de combustible, el tiempo de entrega y los recursos utilizados.

Las restricciones que tenemos en cuenta en este problema garantizan que se cumplan las limitaciones de asignación de clientes, capacidad del vehículo, tiempos de llegada, tiempo de espera y salida desde el depósito.

1. Restricción de asignación de clientes:
    a. Se garantiza que cada cliente *i* sea visitado por exactamente un vehículo en cada dia *d*:

$$\sum_{j} x_{ij}^{d} = 1$$

    b. Se garantiza que cada cliente *i* sea visitado por exactamente un vehículo que sale de él en cada dia *d*:

$$\sum_{j} x_{ji}^{d} = 1$$

2. Restricción de capacidad del vehículo:

La carga acumulada en el cliente *j* no puede exceder la capacidad máxima del vehículo para cada cliente *i* y dia *d*:

$$q_{j}^{d} \geq q_{i}^{d} + K_{i}^{d} * x_{ij}^{d} - Md * (1 - x_{ij}^{d})$$

Donde q_{i}^{d} representa la carga acumulada en el cliente *i* en el día *d* y *Md* es un valor grande que actúa como un límite superior para la carga.

3. Restriccion de tiempo de llegada:

El tiempo de llegada al cliente *j* debe ser mayor o igual al tiempo de llegada a la ubicacion *i* mas el tiempo de viaje entre *i* y *j* para cada ubicacion *i*, cliente *j* y dia *d*:

$$t_{j}^{d} \geq t_{i}^{d} + f_{ij} * x_{ij}^{d} - Mt * (1 - x_{ij}^{d})$$

Donde t_i^d representa el tiempo de llegada a la ubicación *i* en el día *d* y *Mt* es un valor grande que actúa como un límite superior para el tiempo de viaje.

4. Restricción de tiempo:

La diferencia de tiempo de espera entre el cliente *i* en el día *d* y el dia anterior no puede exceder un límite predefinido *L* para cada cliente *i*, dia *d* y dia *e* diferente de *d*:

$$t_i^d - t_i^e \leq L$$

5. Restricción de inicio y fin en el depósito:

Cada vehículo debe comenzar su ruta en el depósito cada día d.

$$\sum_j x_{1j}^d = 1$$

Capitulo 3.
Desarrollo de la aplicación

3.1. Descripción de la aplicación

La aplicación tiene como objetivo encontrar rutas óptimas para un conjunto de vehículos que deben dar servicio a varios clientes en diferentes días determinados, considerando datos relevantes como la capacidad de los vehículos y las demandas de los clientes. Para implementar esta funcionalidad, se utilizó el lenguaje de programación de alto nivel Julia, que es un lenguaje sobresaliente en la eficiencia al hablar de computación científica y problemas de optimización en el área matemática.

Figura 3.1:Logo del lenguaje de programación Julia

3.2. Componentes de la aplicación

3.2.1. Modelado del problema

Se utilizó el paquete JuMP o Julia for Mathematical Programming para construir el problema. Esta es una librería de optimización matemática que permite formular problemas de manera elegante y eficiente. Proporciona la definición de variables, restricciones y función de objetivo del modelo de forma intuitiva.

Figura 3.2: Logo oficial de JuMP

3.2.2. Solucionador del problema

Para resolver el problema definido anteriormente con la librería JuMP, se utilizó el solucionador Gurobi que es un software comercial de optimización matemática, el cual es altamente útil y potente para resolver problemas como el que se presenta.

Figura 3.3: Logo oficial de Gurobi

Gurobi ofrece algoritmos sofisticados y su integración junto a JuMP permite aprovechar todas sus funcionalidades de manera eficiente y de alta calidad. Además, se distingue por su capacidad de resolver problemas de optimización lineal, entera mixta, cuadrática y no lineal, entre otros; Gracias a esto permite la formulación de modelos matemáticos con restricciones complejas, como restricciones de igualdad y desigualdad o variables binarias, enteras y continuas.

3.2.3. Manipulación de Datos y Visualización

Para generar los datos de entrada y crear el gráfico para visualizar la información de las rutas, se utilizaron las siguientes librerías:

- *Random*: Se utilizó el paquete Random de Julia para generar datos aleatorios, como las demandas de los clientes en cada día que se realiza un servicio.
- *Distances*: Esta librería brinda funciones para calcular distancias entre puntos utilizando diversas métricas. En la solución de mi aplicación, se utilizó la Euclidiana la cual es una medida utilizada para calcular la distancia entre dos puntos en un espacio de cualquier dimensión.
- *Plots*: El paquete Plots se utiliza para crear gráficos y soluciones gráficas de los resultados del problema.

- *Colors*: Esta librería proporciona una variedad de paletas de colores que se utilizaron para que hubiera una paleta de colores homogénea y se resaltaron los diferentes elementos del gráfico.

3.3. Elección de librerías y lenguaje

A partir del criterio de eficiencia y rendimiento, el lenguaje de programación *Julia* es conocido por su alto cómputo científico, lo cual ha sido fundamental para encontrar rutas con diferentes datos y situaciones en un tiempo razonable. Además, *Julia* ofrece librerías como *JuMP*, que como se ha mencionado anteriormente, proporciona una interfaz de alto nivel para la formulación del problema, lo cual facilita la comprensión del mismo, también el uso de *Random, Distances, Plots* y *Colors* ofrecen funcionalidades para manipular datos y generar visualizaciones lo que mejora la comprensión de los resultados obtenidos por el solucionador *Gurobi*.

El uso de *Gurobi* es ampliamente utilizado en el campo de la optimización y la investigación de nuevas soluciones de *VRP* debido a su gran capacidad de resolución de problemas matemáticos de forma eficiente.

3.4. Profundización del desarrollo

Hablando a fondo del desarrollo de la aplicación, en primer lugar se implementan las librerías *JuMP, Gurobi, Plots, Random, Distances* y *Colors* que proporcionan funcionalidades y capacidades necesarias para resolver el problema presentado.

```
using JuMP, Gurobi, Plots, Random, Distances, Colors
```

Figura 3.4: Definición de librerías

A continuación, se definen las variables y parámetros, estos incluyen el número de coordenadas, la demanda máxima por cliente, la capacidad del vehículo y los días en los que se ofrecerá el servicio.

```
# Definimos las variables de los datos
U = 10                  # número de coordenadas
client_demand = 5       # Demanda maxima por cliente
vehicle_capacity = 50 # Capacidad maxima del vehiculo
D = 3                   # número de días
```

Figura 3.5: Definición de variables y parámetros.

La generación de los datos de entrada es también algo fundamental para la resolución del problema ya que se generan aleatoriamente las coordenadas y a partir de estas la distancia entre los clientes. Además, como existe una demanda por cada cliente en cada día, agrega alta complejidad a la resolución.

```
# Generamos datos de entrada
coords = [(rand(1:R), rand(1:R)) for i in clients] # Coordenadas de los clientes
pushfirst!(coords, (R/2,R/2))                      # Coordenadas del deposito

distances = [Euclidean()(coords[i], coords[j]) for i in locations, j in locations] # Distancias entre clientes

# Demanda de los clientes (Empieza en 2)
demands = [rand(0:client_demand) for i in locations, j in days] # Demanda de cada cliente en cada dia
```

Figura 3.6: Generación de datos de entrada

Para definir el modelo con el solucionador Gurobi, lo primero es definir el modelo, y a continuación escribir las variables de decisión y definir la función objetivo, la cual tiene como objetivo minimizar la distancia total entre clientes y ha de tener en cuenta las distancias entre clientes, el tiempo de llegada y la carga disponible en los vehículos.

```
# Variables de decisión

@variable(model, x[locations, locations, days], Bin)

@variable(model, time[locations, days] >= 0)

@variable(model, load[clients, days] >= 0)

@variable(model, y[days], Bin)

# Función objetivo

@objective(model, Min, sum(distances[i,j]*x[i,j,d] for i in locations, j in locations,
d in days))

# Minimizar la distancia total recorrida
```

Figura 3.7: definición del modelo y función objetivo.

Como vemos en la figura, contamos con las variables de decisión *x*, *time* y *load*; *x[i j,d]* es una variable con valores 0 o 1 que se desplaza del cliente *i* al cliente *j* en el dia *d*, esta variable sirve para determinar si la ruta es seleccionada o no, con lo cual es una variable binaria. La variable de tiempo de llegada, o *time[i,d]* es una variable continua que representa el tiempo en el que el vehículo llega al cliente *i* en el día *d* y se utiliza para determinar los tiempos en los que se realiza el servicio a cada cliente, garantizando que se cumplan las restricciones de consistencia en la visita a cada cliente. La variable que sigue es *load[i, d]* o carga del vehículo, esta también se trata de una variable continua que representa la carga al llegar al cliente *i* en el día *d*, se utiliza para determinar que no se supere la capacidad máxima que tenga el vehículo determinado. Por último, se define la variable *y[days].* La variable y se utiliza para controlar el inicio de rutas nuevas en cada día. Si *y[d]* es igual a 1, entonces significa que se ha empezado una nueva ruta. La inclusión de esta variable permite que el modelo determine de forma automática la cantidad de rutas óptimas que se deben crear.

Posteriormente, se define la función objetivo. Esta busca encontrar la ruta óptima de los vehículos que minimice la distancia recorrida en total. Se define teniendo en cuenta las distancias generadas anteriormente *distances[i,j]* que representa el recorrido entre el cliente *i* y el cliente *j*, también el conjunto de destinos o *locations* que incluye tanto a los clientes como al depósito y el conjunto de días que se ofrece el servicio.

Las restricciones del problema son el desarrollo crucial para que se cumplan las condiciones del problema de enrutamiento de vehículos con restricciones de capacidad y consistencia de días. Estas aseguran que cada cliente sea visitado al menos una vez, que la capacidad de los vehículos nunca sea superior al valor máximo, que se cumplan los tiempos de visita a cada cliente y que se limite el tiempo de espera que tienen los clientes dentro del margen en el que existe la consistencia de las visitas. Las restricciones se expresan mediante operaciones algebraicas y desigualdades lineales.

A continuación se hará una explicación a detalle de cada una de estas restricciones:

- *Restricción de asignación de clientes*. Al principio vemos que para cliente *i* y el día *d*, la suma de los vehículos que llegan a dar servicio a un cliente debe ser igual a 1. Esto quiere decir que todos los clientes que tengan demanda serán atendidos exactamente una vez por día. A continuación, nos aseguramos de que no se visiten los clientes con demanda 0 incluyendo la restricción segunda en la condición.

```
## Clients must be visited only once per day
for i in clients, d in days
    if demands[i, d] > 0  # Considerar solo los clientes con demanda mayor a cero
        @constraint(model, sum(x[i, j, d] for j in locations) == 1)
        @constraint(model, sum(x[j, i, d] for j in locations) == 1)
    else
        @constraint(model, sum(x[i, j, d] for j in locations) == 0)  # Restricción para evitar la visita a clientes sin demanda
        @constraint(model, sum(x[j, i, d] for j in locations) == 0)
    end
end
```

Figura 3.8: Restriccion de asignacion de cliente

- *Restricción de carga*. Cada vehículo tiene una capacidad determinada para cada viaje, la carga de estos vehículos al llegar al cliente *j* en el dia *d* debe ser mayor o igual a la carga que tenía al llegar al cliente *i* en el dia *d* que se ofrece el servicio, además de tener en cuenta la demanda del cliente *i* en el dia *d*, también teniendo en cuenta un valor grande *Md* utilizado para penalizar aquellas rutas que excedan la capacidad de carga que está determinada.

```
## Restricción de carga

@constraint(model, [i in clients, j in clients, d in days],

load[j, d] >= load[i, d] + demands[i, d] * x[i, j, d] - Md *(1 - x[i, j, d]))
```

Figura 3.9: Restricción de carga

- *Restricciones de tiempo de llegada.* El tiempo de llegada al cliente *j* en el dia *d* debe ser mayor o igual que el tiempo de llegada al cliente *i* en el dia *d*, añadiendo el tiempo de llegada del cliente *i* en el dia *d*, y también teniendo en cuenta la distancia entre los clientes *i* y *j*. Además también existe un valor grande *Mt* de penalización el cual no permite que existan días en los que los clientes son visitados en otros momentos diferentes.

```
## Restricción de tiempo de llegada

@constraint(model, [i in locations, j in clients, d in days],
time[j,d] >= time[i,d] + distances[i,j] * x[i,j,d] - Mt * (1 - x[i,j,d]))
```

Figura 3.10: Restricción de tiempo de llegada

- *Restricción de tiempo máximo de espera*. Para cada cliente i en el dia d y dia e, donde d es diferente de e, se asegura que la diferencia de tiempo entre un cliente en el dia anterior no supere el límite L, el cual limita el tiempo de espera para que exista consistencia y evitar esperas prolongadas así como asegurar que se realice la entrega en un horario en el que el cliente esté abierto o suela estar, aumentando así las probabilidades de éxito en las rutas.

```
## Time constraint

for i in clients, d in days, e in days

    if d != e && demands[i, d] > 0

        @constraint(model, time[i, d] - time[i, e] <= L)

    end

end
```

Figura 3.11: Restricción de tiempo de espera

- *Restricción que asegura que se empieza y termina en el depósito.* Estas restricciones se hacen cargo de que en cada día, exactamente un solo vehículo termine y comience su ruta en el depósito, el cual es el índice 1.

```
# Constraints to ensure each vehicle starts and ends at the depot

@constraint(model, [d in days], sum(x[1, j, d] for j in clients) == 1)

@constraint(model, [d in days], sum(x[i, 1, d] for i in clients) == 1)
```

Figura 3.12: Restricción que se asegura que se empieza y termina en el depósito

- Restricción para separar las rutas. Se usa la variable *y*, ya que esta indica si se ha de crear una ruta en el día *d*. La restricción impone que la suma de *x* para todo *j* en el dia *d* debe ser mayor o igual a la ruta y en el dia *d* multiplicado por el número de vehículos con los que contamos. Esto significa que si *y[d]* es 0, entonces no deben haber vehículos que comiencen ese día, y si es 1 entonces la suma debe ser menor o igual a *K* para que no haya existencia de más vehículos que los que propone *K*.

```
# Additional constraint to separate new_routes

@constraint(model, [d in days], sum(x[1, j, d] for j in clients) <= y[d] * K)
```

Figura 3.13: Restricción para separar las rutas

Para encontrar la solución del modelo, se utiliza la función ***optimize!,*** la cual proporciona la solución óptima. Esta revela las rutas de los vehículos, así como los tiempos de llegada y la carga de los vehículos en cada etapa. Finalmente, la visualización de los resultados se realiza utilizando la librería *Plots*. Las rutas se ven en un plano gráfico, mostrando nodos que representan a los clientes y líneas que representan las rutas en orden conectadas.

Capitulo 4.
Experimentación y Análisis de Resultados

4.1. Introducción y objetivo

En este capítulo, se harán distintos cambios a los datos con el objetivo de evaluar la eficacia del algoritmo implementado para resolver el VRP con días consistentes. Se llevaron a cabo diferentes casos de estudio en los que se alteraron los datos de entrada para ver los resultados obtenidos.

Se busca determinar cómo el algoritmo responde a los cambios de datos, especialmente en los parámetros relacionados con la novedad de este estudio, las restricciones temporales y las consistencias.

4.2. Casos de estudio y parámetros utilizados

Se realizaron tres casos de estudio, cada uno con diferentes valores. Estos casos se crearon para abarcar la variedad de situaciones y evaluar el impacto de los cambios en el comportamiento del algoritmo, sin embargo, el dato que será siempre igual seran los dias en los que se realiza el servicio, 3, ya que podemos ver las diferencias entre los días sin tener demasiado volumen de resultados en la experimentación. A continuación se presentan dichos casos y resultados:

- Caso 1: En este caso, se generan 10 ubicaciones aleatoriamente que incluyen tanto a los clientes como el depósito inicial, la demanda máxima por cliente ha de ser 7, la capacidad máxima de los vehículos es de 20 y se consideraron 3 días en los que se ofrecen servicios. Además la probabilidad de demanda que cada cliente tiene es del 50%, la longitud máxima de recorrido es de 20 unidades y el tiempo máximo que un cliente espera el servicio es de 15 unidades de tiempo, además tenemos en cuenta que contamos con 3 vehículos.

Coordenadas (U)	Demanda máxima	Capacidad máxima	Probabilidad demanda (Pd)	Distancia máxima	Tiempo máximo (L)
10	7	16	0.5	20	15

Tabla 4.1: Datos del caso 1

Este caso dio los siguientes resultados:

- En el día 1, las rutas que se realizaron fueron: [1, 10, 9, 2, 1] y [1, 7, 1], siendo 1 el depósito. A continuación, se muestra cada cliente y el tiempo de llegada en los que se visitó.
  - Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 10	8.9442	6
Cliente 9	33.0372	1
Cliente 2	25.9662	5
		Total(10) <= 16

Tabla 4.1.1: Caso 1, dia 1, Ruta 1

  - Ruta 2

Clientes	Tiempo de llegada	Demanda
Cliente 7	22.3606	4
		Total(4) <= 16

Tabla 4.1.2: Caso 1, dia 1, Ruta 2

Como se puede observar, el cliente 2 es el más alejado al depósito y el cliente 10 el más cercano. Aquellos clientes que no se tuvieron en cuenta simplemente no tuvieron demanda ese día.

Es importante destacar que el algoritmo tiene en cuenta las restricciones de capacidad y tiempo, por lo que la generación de los resultados está sujeta a estas limitaciones. En algunos casos puede ser necesario realizar un mayor recorrido en las rutas para cumplir las restricciones lo que puede concluir en un mayor tiempo de espera para los clientes que se encuentran más alejados, pero lo importante es mantener la consistencia de los momentos de llegada.

También podemos ver en la siguiente figura las rutas gráficas que se han realizado en este caso. Donde los nodos con números representan los clientes, el nodo central representa el depósito y las líneas representan la ruta.

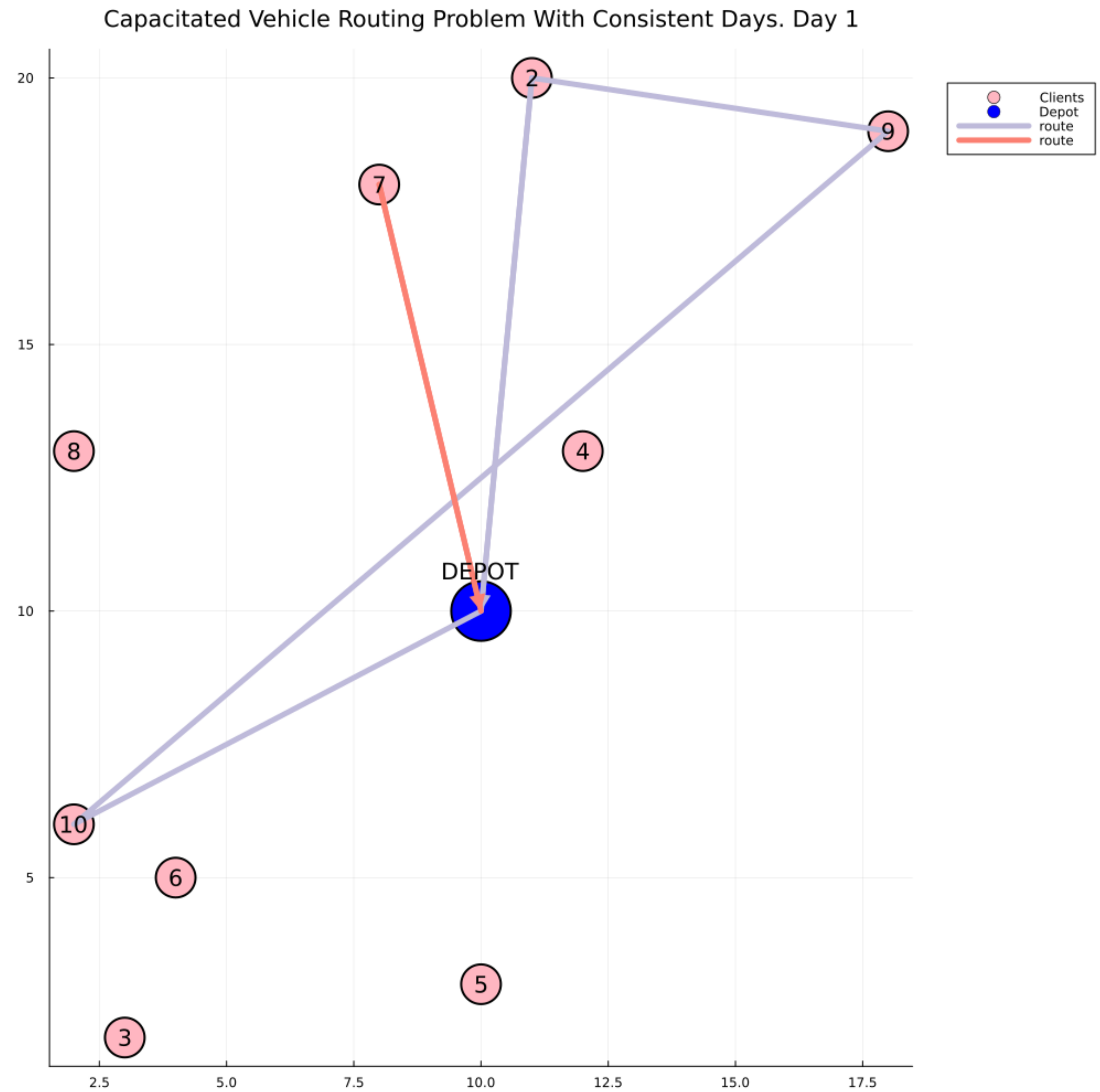

Figura 4.1: Caso 1, dia 1

- En el día 2, se realizó la siguiente ruta: [1, 3, 5, 9, 1]. Siendo 1 el depósito, y el resto los clientes.

Clientes	Tiempo de llegada	Demanda
Cliente 3	10.6301	2
Cliente 5	18.8540	7
Cliente 9	36.7426	4
		Total(13) <= 16

Tabla 4.1.3: Caso 1, dia 2, Ruta 1

En el día 1, se realizaron dos rutas, lo que significa que los clientes han sido visitados en un orden diferente, lo que quiere decir que el algoritmo encuentra soluciones óptimas para cada día de forma diferente. En el día 1 se visitaron más clientes que el día 2.

Además se observó que las demandas de los clientes también han cambiado, siendo mayores en general en el día 1, la capacidad de la aplicación para adaptarse a demandas cambiantes y encontrar soluciones es obvia en la comparación de ambos días. Sin embargo, el día 2 cuenta con un cliente nuevo que no se visitó el día 1. Este es el cliente 5 el cual tiene la demanda más alta de este día siendo 7 unidades.

En la siguiente figura se muestra gráficamente la ruta que se realizó el día 2:

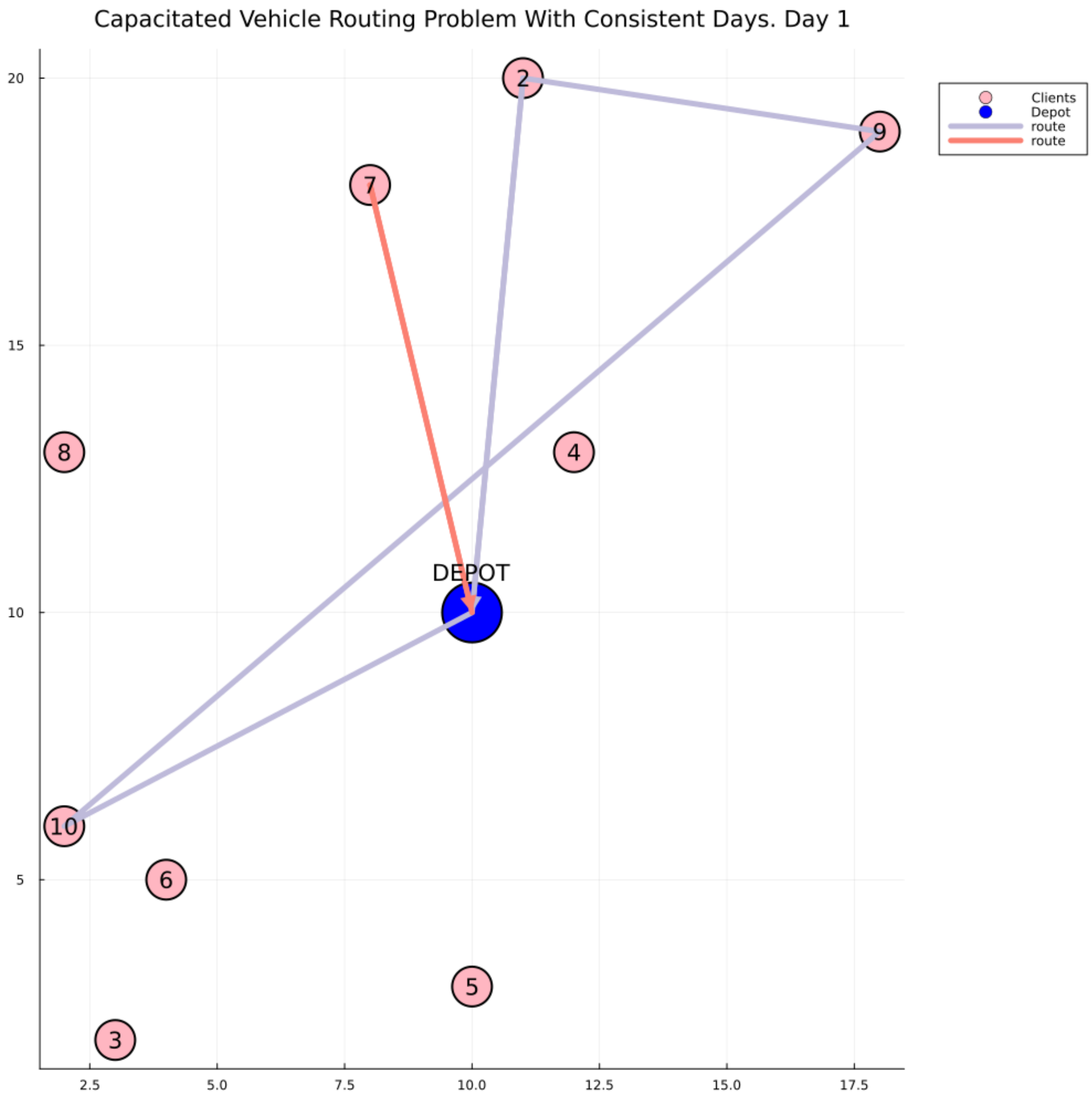

Figura 4.2: Caso 1, dia 2

En el día 3, las rutas que se generaron fueron las siguientes: [1, 2, 9, 1] y [1, 4, 1]. Siendo 1 el depósito y el resto los clientes visitados.

Clientes	Tiempo de llegada	Demanda
Cliente 2	10.9662	7
Cliente 9	21.7426	5
		Total(12) < 16

Tabla 4.1.4: Caso 1, dia 3, Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 4	36.7426	4
		Total(4) < 16

Tabla 4.1.5: Caso 1, dia 3, Ruta 2

Las observaciones que podemos realizar a partir de esta información es que en el día 2, se visitaron los clientes 3, 5 y 9, mientras que en el día 3 se visitaron los clientes 2, 9 y 4, ambos tienen en común el cliente 9. Además, poniendo también en contraste el dia 1, se puede ver que también se visitó el día 9, también coincidiendo el cliente 2.

Además se observa que existe variación en la demanda de los clientes, por ejemplo, el día 1 el cliente 9 tiene una demanda de solo 1 unidad, avanzando al día 2 este mismo cliente tiene una demanda de 4 y finalmente en el día 3 el cliente 9 tiene una demanda de 5 unidades.

Seguidamente podemos observar la ruta tomada en el día 3 de forma gráfica.

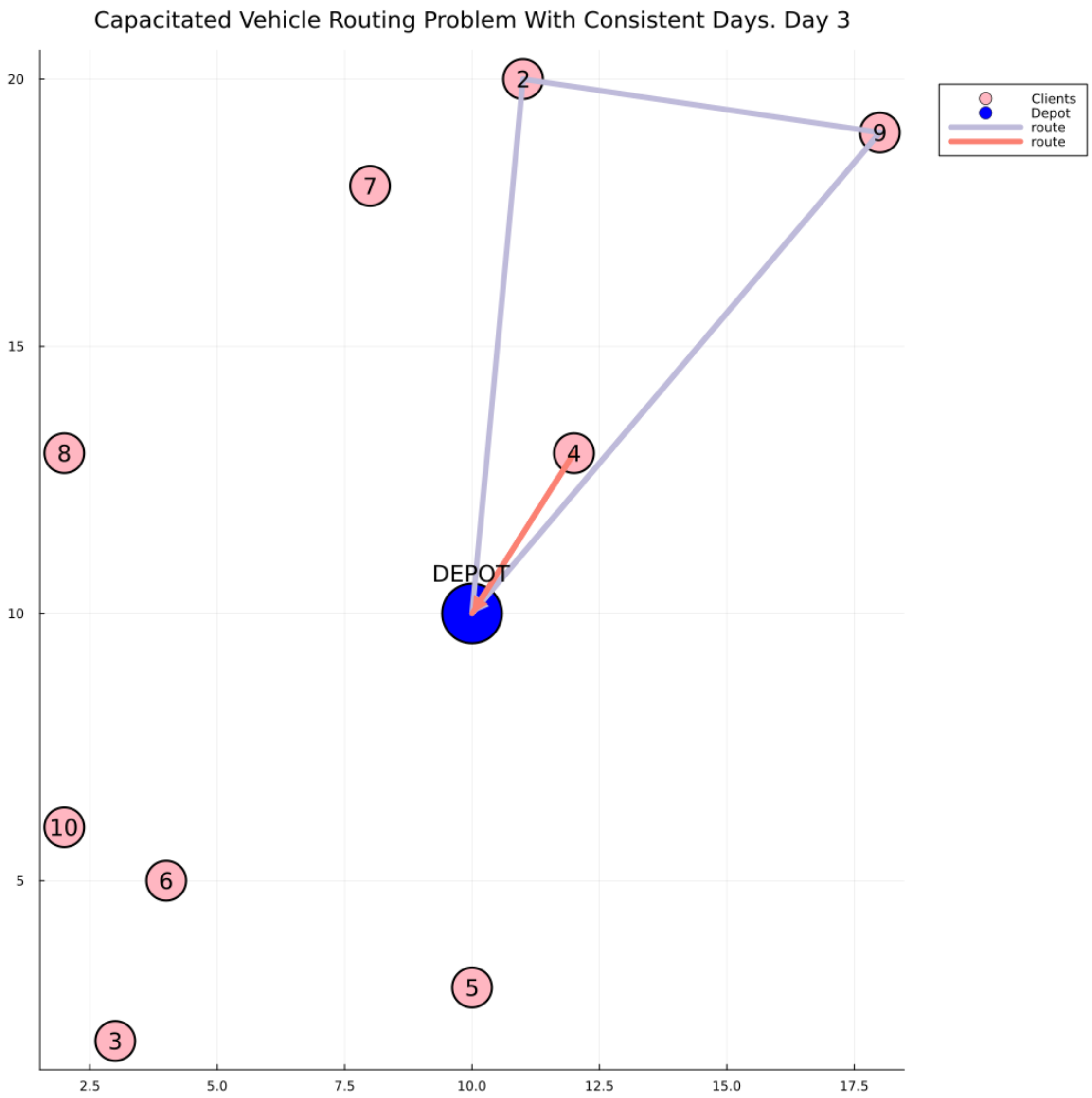

Figura 4.3: Caso 1, dia 3

- Caso 2: En este caso de estado, existen 8 ubicaciones en total que incluye tanto a los clientes como al depósito. La demanda máxima de los clientes cambia a 5 en este caso y la capacidad máxima del vehículo también, ahora siendo 15. Como he mencionado anteriormente el número de días que se ofrece el servicio permanecerá como 3. La probabilidad de que un cliente tenga demanda aumentó al 90% y la longitud máxima de un recorrido entre clientes es de 28 unidades, siendo, además, el tiempo máximo de espera de cada cliente 30 unidades.

Coordenadas (U)	Demanda máxima	Capacidad máxima	Probabilidad demanda (Pd)	Distancia máxima	Tiempo máximo (L)
8	5	15	0.9	28	30

Tabla 4.2: Datos del caso 2

Los resultados de este caso son los siguientes en cada dia:

- En el día 1, se generaron las siguientes rutas:

  [1, 4, 5, 7, 1] y [1, 8, 3, 1]. Siendo 1 el depósito y el resto los clientes.

Clientes	Tiempo de llegada	Demanda
Cliente 4	4.2426	4
Cliente 5	100.0002	4
Cliente 7	13.6766	2
		Total(10) < 15

Tabla 4.2.1: Caso 2, día 1. Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 8	75.8121	2
Cliente 3	90.9448	2
		Total(4) < 15

Tabla 4.2.2: Caso 2, día 1. Ruta 2

Los tiempos de llegada a los clientes varían en función de la distancia, por ejemplo, el cliente 3 es visitado después del cliente 8. Además se puede observar que los clientes 4 y 5 tienen la demanda más alta con 4 unidades, mientras que el resto de clientes tienen demanda de 2 unidades

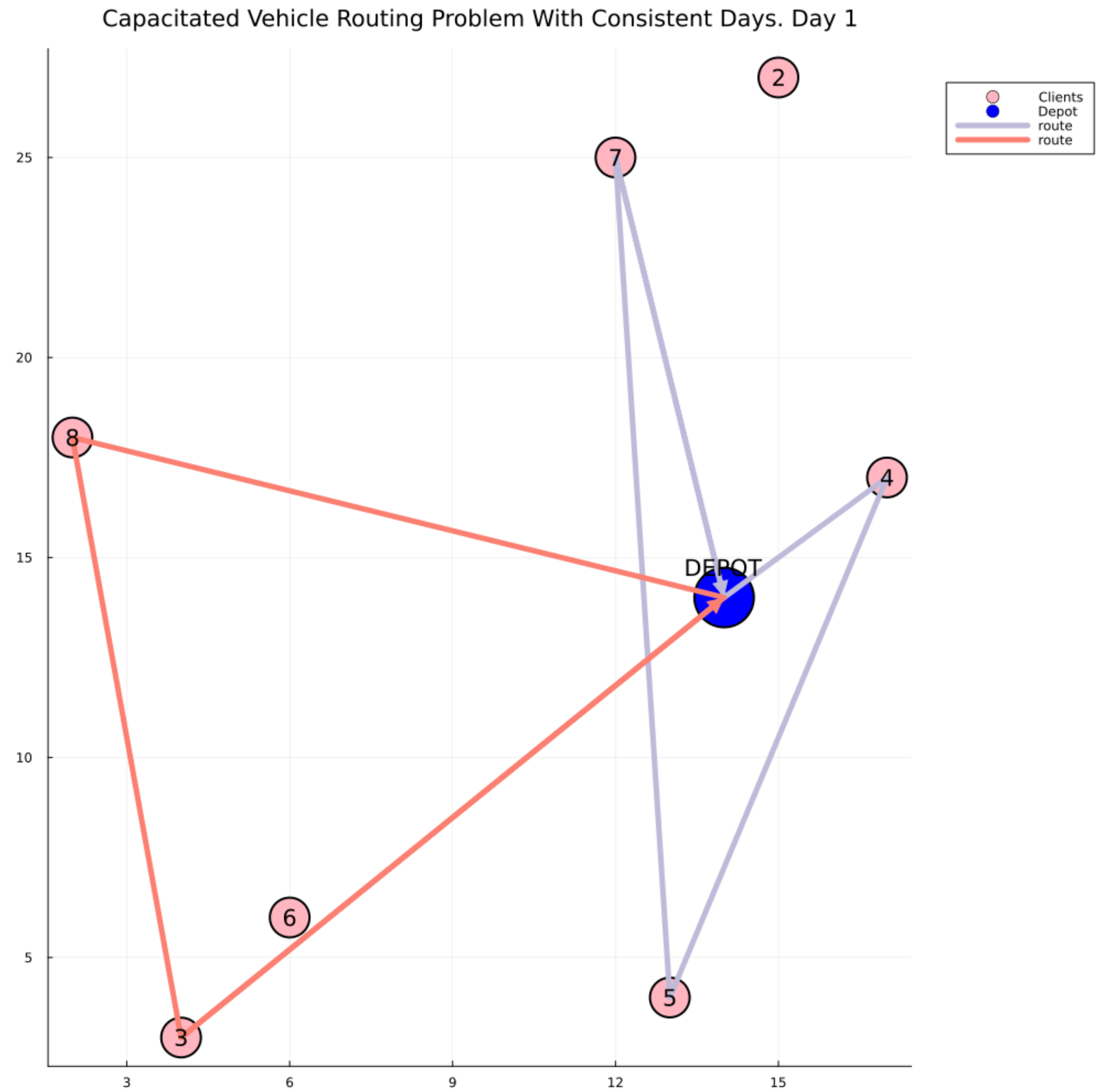

Figura 4.4: Caso 2, dia 1

- ❖ El día 2, se generaron las rutas de entrega siguientes:

  [1, 4, 7, 5, 2, 1] y [1, 8, 3, 1]. Siendo 1 el depósito.

Clientes	Tiempo de llegada	Demanda
Cliente 4	4.2426	4
Cliente 7	18.0462	1
Cliente 5	100.0002	4
Cliente 2	14.4406	2
		Total(11) <= 15

Tabla 4.2.3: Caso 2, dia 2, Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 8	75.8121	4
Cliente 3	90.9448	1
		Total(5) <= 15

Tabla 4.2.4: Caso 2, dia 2, Ruta 2

Poniendo el contraste el día 1 y el día 2, se puede observar que en ambos días se hicieron dos rutas. Vemos que los clientes se visitan en orden distinto, por ejemplo el día 1 se visita al cliente 5 antes que al cliente 7 y el día 2 es al contrario. También se visita al cliente 2 el día 2, cuando en el día 1 no ha sido así.

Se puede observar también que han habido cambios en las demandas de los clientes, por ejemplo en el día 1 los clientes 7 y 8 tenían una demanda de 2 unidades. Sin embargo, el día 2 estos clientes tenían demandas de 1 y 4 respectivamente, lo que indica que el cliente 7 disminuyó la demanda y el cliente 8 la aumentó.

A continuación se muestra la ruta del dia 2 gráficamente:

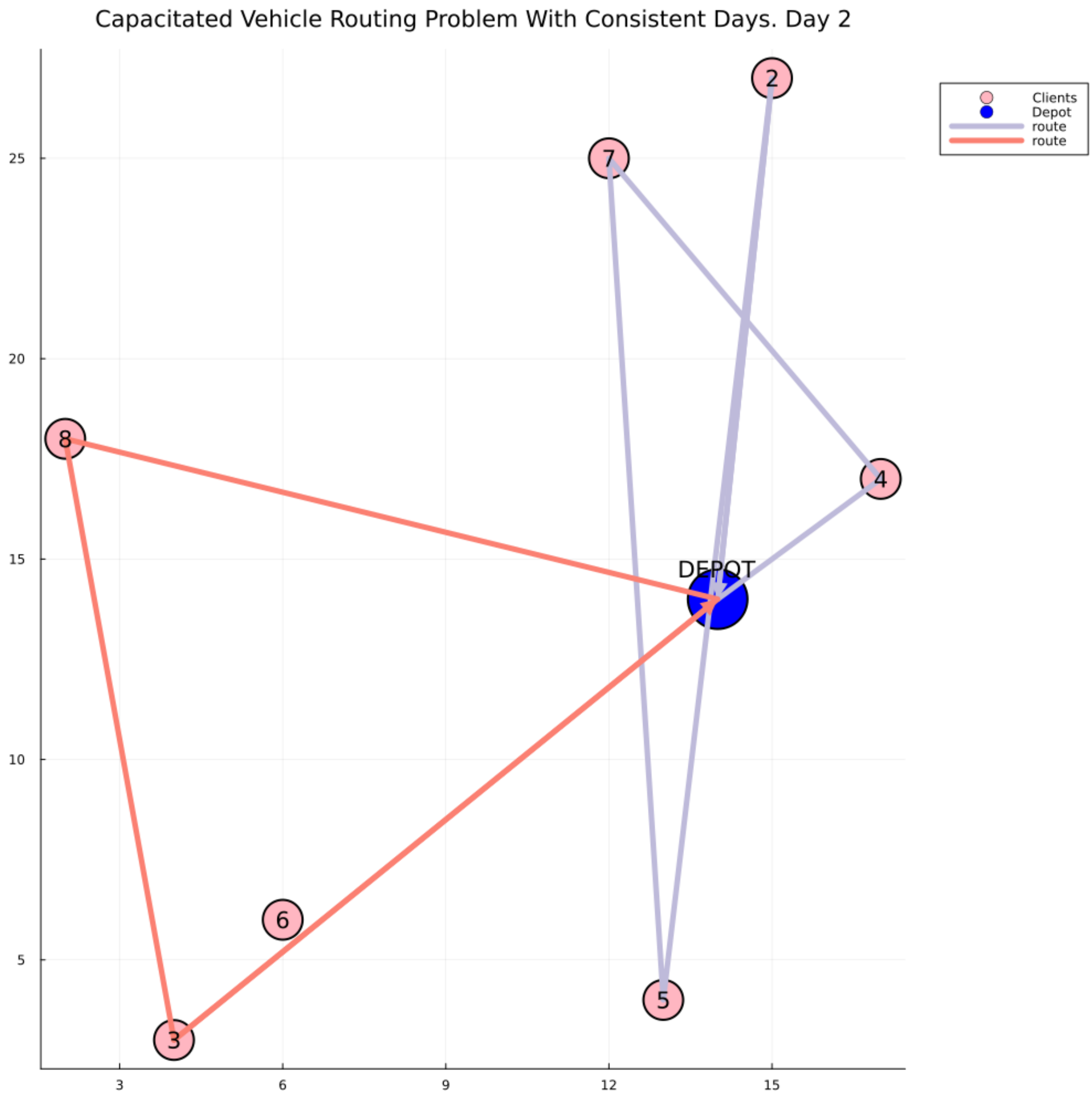

Figura 4.5: Caso 2, dia 2

- ❖ En el tercer día, las rutas tomada por los vehículos fue la siguiente:

  [1, 4, 7, 2, 1] y [1, 8, 5, 1]. Siendo 1 el depósito y el resto los clientes

Clientes	Tiempo de llegada	Demanda
Cliente 4	4.2426	3
Cliente 7	33.6055	3
Cliente 2	30.0	4
		Total(10) < 15

Tabla 4.2.5: Caso 2, dia 3, Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 8	45.8121	5
Cliente 5	70.0002	3
		Total(8) < 15

Tabla 4.2.6: Caso 2, dia 3, Ruta 2

Al comparar los días 2 y 3, se observa que en ambos días se realizaron 2 rutas. En el día 2 se visita al cliente 5 en la ruta 1 pero en la el día 3, este cliente es visitado en la ruta 2 y se deja de visitar al cliente 3.

Los resultados del día 3 también muestran que las demandas han cambiado. En el día 2 se requiere menos carga que en el día 3 lo que indica que alguna demanda ha de haber aumentado, esta demanda es la del cliente 8 y la del cliente 7, las cuales eran 1 y 4 respectivamente y aumentaron a 3 y 5.

A continuación se encuentra la representación gráfica de este dia:

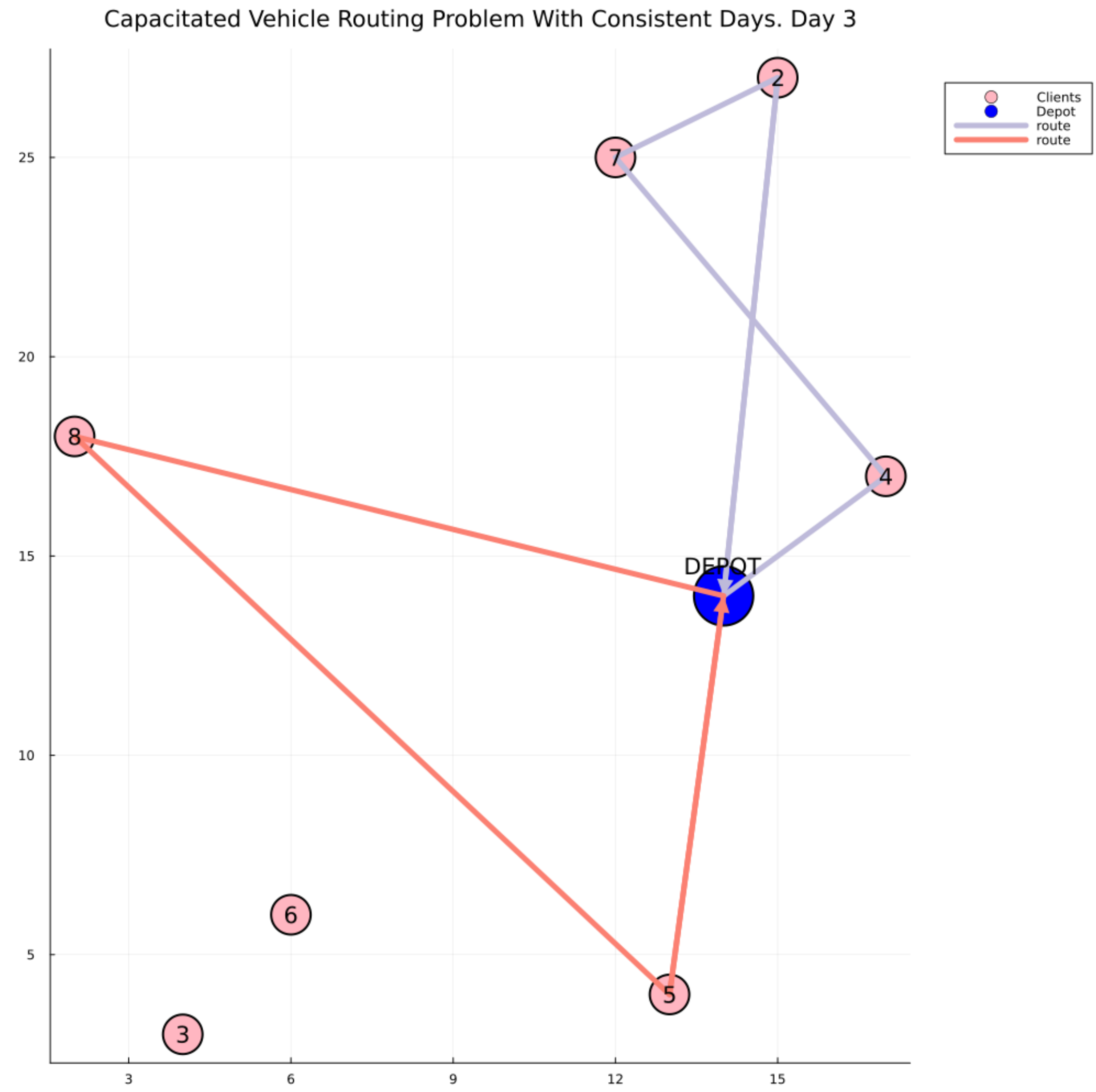

Figura 4.6: Caso 2, dia 3

- En el caso 3, el número de ubicaciones (las cuales incluyen a los clientes y el depósito) es 5, la demanda máxima del cliente es 18 y la capacidad máxima de los vehículos pasa a ser 35. Además la probabilidad de que los clientes tengan demanda también aumenta a un 70% y la distancia máxima sube a 30 unidades, además siendo el tiempo de espera máxima de los clientes 20.

Coordenadas (U)	Demanda máxima	Capacidad máxima	Probabilidad demanda (Pd)	Distancia máxima	Tiempo máximo (L)
9	18	48	0.7	30	20

Tabla 4.3: Datos del caso 3

Los resultados que han dado lugar en este caso han sido:

❖ En el dia 1, las rutas generadas son:

[1, 2, 5, 6, 3, 4, 1], siendo 1 el depósito y el resto los clientes.

Clientes	Tiempo de llegada	Demanda
Cliente 2	11.1803	7
Cliente 5	14.3426	5
Cliente 6	82.8121	11
Cliente 3	71.9954	10
Cliente 4	60.9954	14
		Total(47) < 48

Tabla 4.3.1: Caso 3, dia 1, Ruta 1

Al analizar los resultados del día 1, podemos ver que los clientes que se encuentran más cercanas del depósito son los clientes 2 y 5. El cliente más lejano es el cliente 6. Además podemos ver que el cliente que más demanda tiene es el cliente 4

La representación gráfica se muestra de la siguiente manera:

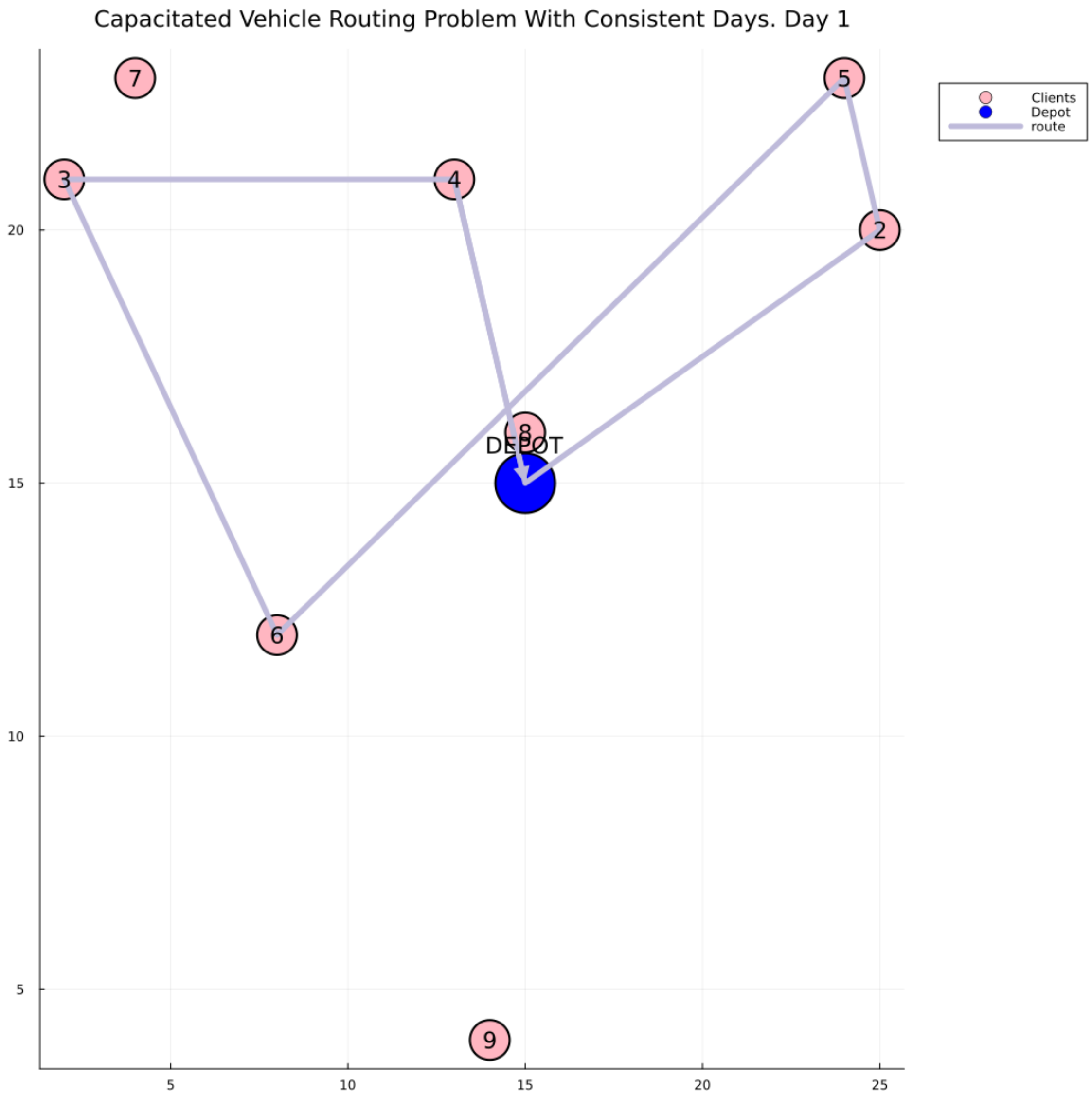

Figura 4.7: Caso 3, dia 1

❖ En el dia 2, las rutas fueron la siguientes:

[1, 8, 9, 7, 1] y [1, 4, 2, 3, 1], siendo 1 el depósito y el resto los clientes.

Clientes	Tiempo de llegada	Demanda
Cliente 8	1.0	10
Cliente 9	31.8238	8
Cliente 7	54.8238	8
		Total(26) < 48

Tabla 4.3.2: Caso 3, dia 2, Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 4	64.0434	17
Cliente 2	11.7703	15
Cliente 3	51.9954	12
		Total(44) < 48

Tabla 4.3.3: Caso 3, dia 2, Ruta 2

Al analizar los datos del día 2, vemos que los clientes más cercanos son el cliente 8 y el cliente 2, estos tienen respectivamente una demanda de 15 y 10 unidades. En comparación con el día 1, el cliente 8 es un cliente que no se había visitado anteriormente, y además, la demanda del cliente 2 ha aumentado considerablemente.

Poniendo otros datos en contraste, en el día uno tan solo hubo una ruta, en cambio el día 2 hubo dos rutas, lo que sugiere que el día 2 la demanda fue mucho más alta que en el día 1. Por ejemplo, el cliente 4 aumentó su demanda a 17, mientras que en el día 1 era 14.

Este gráfico muestra la ruta proporcionada por los datos anteriores.

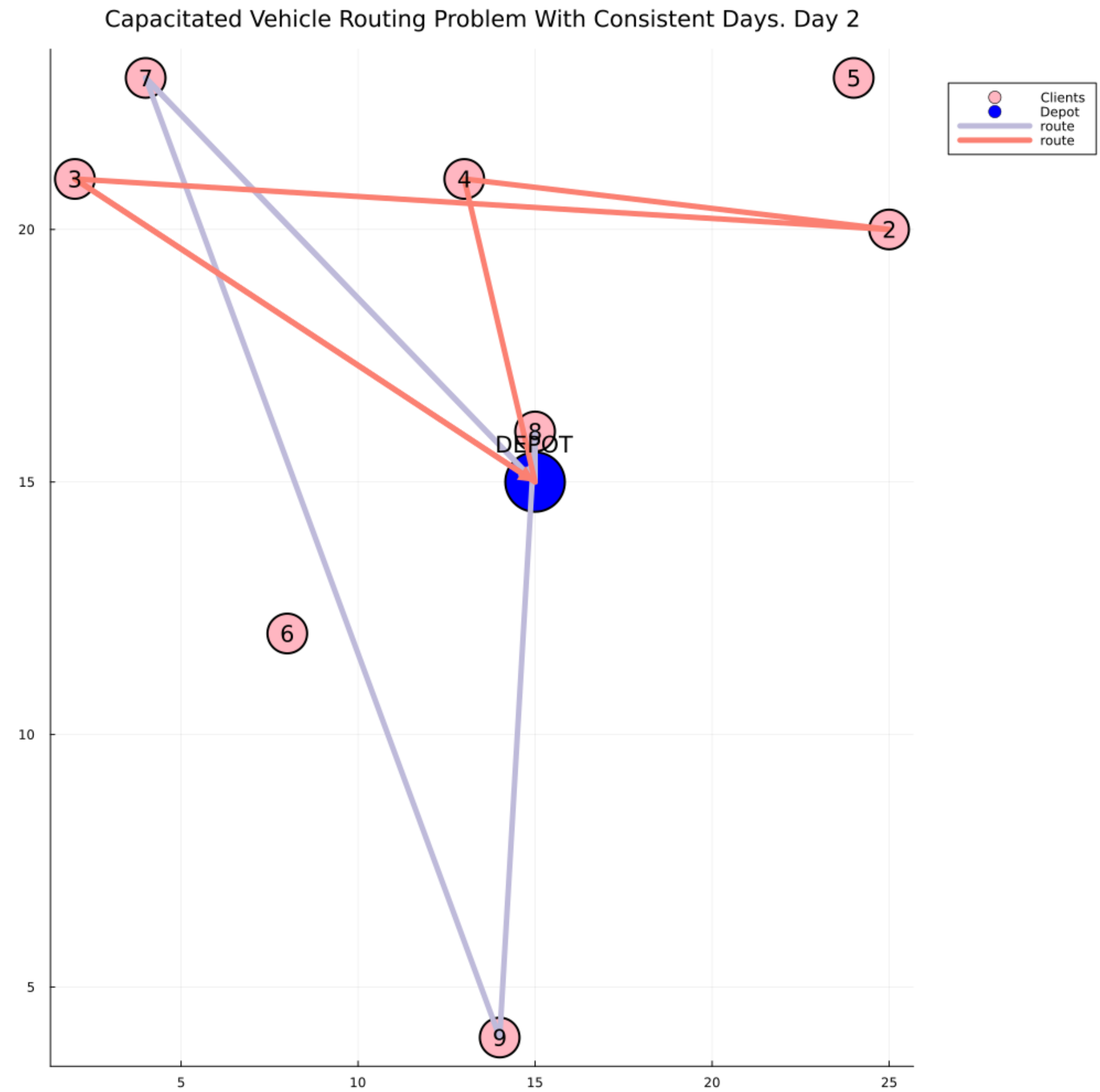

Figura 4.8: Caso 3, dia 2

❖ En el dia 3 las rutas que se tomaron fueron:

[1, 8, 6, 3, 1] y [1, 5, 1]siendo 1 el depósito.

Clientes	Tiempo de llegada	Demanda
Cliente 8	1.0	16
Cliente 6	82.8121	4
Cliente 3	51.9954	3
		Total(23) < 48

Tabla 4.3.4: Caso 3, dia 3, Ruta 1

Clientes	Tiempo de llegada	Demanda
Cliente 5	12.4017	5
		Total(5) < 48

Tabla 4.3.5: Caso 3, dia 3, Ruta 2

Algunos clientes están cerca del depósito estos son el cliente 6 y el cliente 5, y los otros están más alejados, los cuales corresponden al cliente 3 y 6. Además, el cliente 8 tiene una demanda de 16 unidades, la cual es la más alta, siendo la del cliente 3 la más baja con 3 unidades. Además en total se ve que el día 3 es el que menos demanda tuvo en total, siendo el día 2 el que más tuvo.

Seguidamente, se muestra la ruta en formato gráfico:

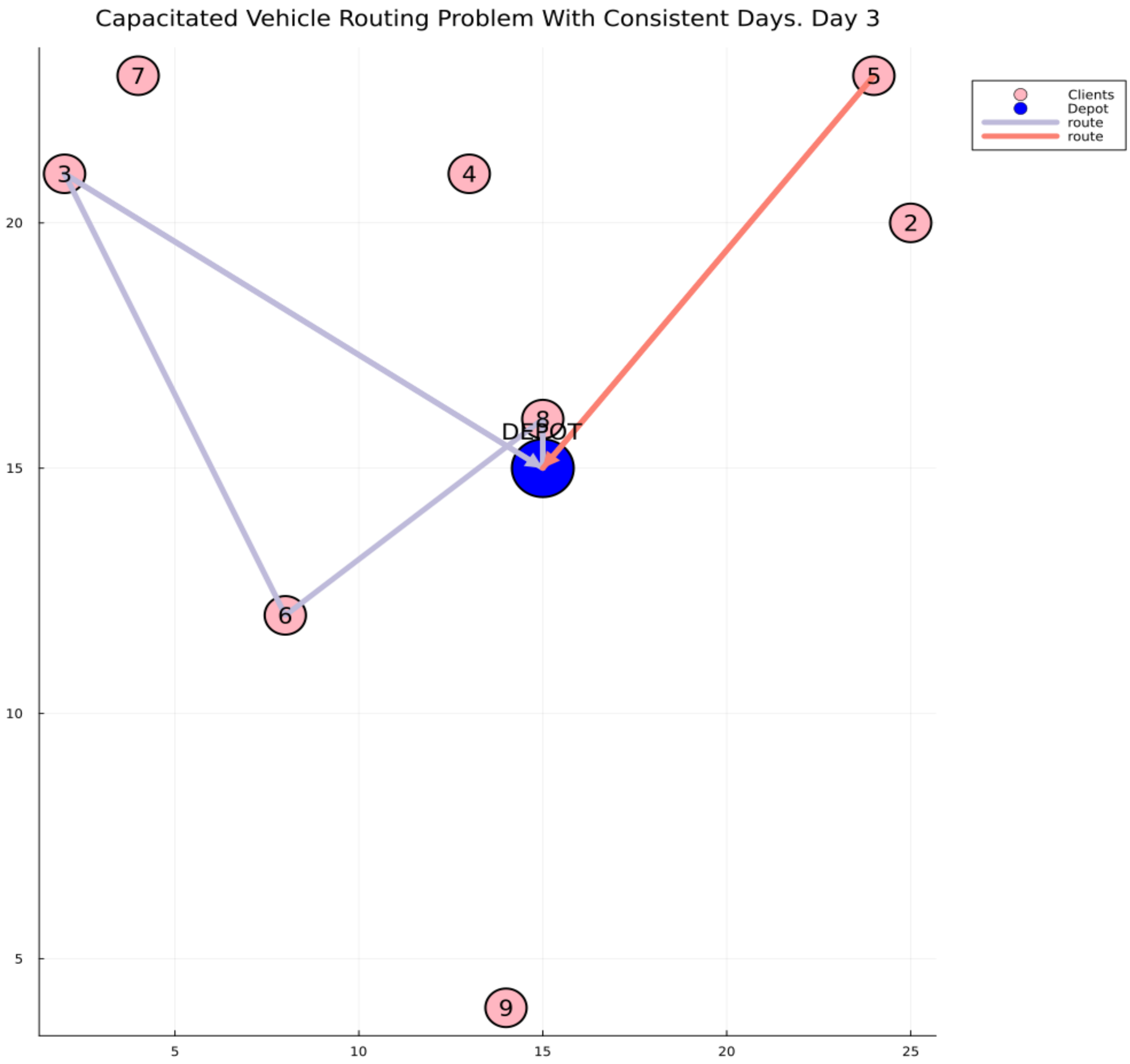

Figura 4.9: Caso 3, dia 3

Capitulo 5.
Conclusiones y líneas de investigación futuras

5.1. Conclusiones

En este proyecto de optimización de enrutamiento de vehículos con restricciones de tiempo consistentes, se aborda un problema complejo con el objetivo de minimizar las rutas logísticas recorridas por vehículos.

A través del uso de la programación, se ha realizado un modelo matemático implementado en el lenguaje de programación Julia. Para lograr el desarrollo del algoritmo, se definieron las variables de decisión, la función objetivo y las restricciones del problema. Durante la experimentación con diferentes conjuntos de datos y parámetros, se ha observado que el algoritmos propuesto es capaz de generar las rutas eficientes que minimizan la distancia total recorrida y que cumplen con las restricciones de capacidad y tiempo.

También se compararon estos resultados de diferentes escenarios y se encontró que el rendimiento del modelo variaba en función de estos parámetros, lo que demostró la flexibilidad y adaptabilidad del modelo.

5.2. Posibles investigaciones del futuro

Este proyecto proporciona una base sólida para otras investigaciones en el área de la optimización y sobre todo en los modelos de VRP. Puede haber algunas líneas de investigación a explorar como por ejemplo, la consideración de restricciones adicionales, como restricciones de prioridad de servicio para adaptarse a problemas más realistas.

Se puede también investigar el comportamiento del algoritmo en escenarios en los que las demandas y otros parámetros cambian a lo largo del tiempo de forma dinámica, esto implicaría desarrollar elementos de predictibilidad y adaptabilidad a cambios en tiempo real.

Es posible también explorar la optimización de varios objetivos, como la minimización de costos o de emisiones de CO_2 o la maximización de la satisfacción de los clientes.

Capitulo 6.
Conclusions and future lines of research

6.1. Conclusions

In this project, we board the vehicle routing problem with consistent time constraints, it is a complex problem to address with the objective of minimizing the logistics routes traveled by vehicles.

Through the use of programming, a mathematical model was implemented in the Julia programming language. To achieve the development of the algorithm, there were defined decision variables, an objective function and the restrictions of the problem. During experimentation with different data sets and parameters, it has been observed that the proposed algorithm is capable of generating efficient routes that minimize the total distance traveled that comply with capacity and time constraints.

These results from different scenarios were compared and found that the performance of the model varied based on these parameters, demonstrating that the flexibility and adaptability of the model was capable.

6.2. Possible future research

This project provides a solid base for further research in the area of optimization and especially in VRP models. There may be some lines of research to explore such as consideration of additional constraints such as service priority constraints to accommodate more realistic problems.

It is also possible to investigate the behavior of the algorithm in scenarios in which the demands and other constraints change dynamically over time, this would imply developing elements of predictability and adaptability to changes in real time.

Also, there's the possibility of exploring the optimization of various objectives different to the distance, such as minimizing costs or CO_2 emissions or maximizing customer satisfaction.

Bibliografía

[1] P. Toth and D. Vigo. *The vehicle routing problem*. Philadelphia : SIAM, cop., 2002.

[2] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*, volume 18. Siam, 2a edition, 2014

[3] Antonello Lobianco. *Julia Language: A Concise Tutorial.* 2022.

[4] A. N. Letchford and J.J. Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2-3):251-274, 2006.

[5] “Vehicle routing Problem” https://en.wikipedia.org/wiki/Vehicle_routing_problem. [Online; Accedido 24-Mayo-2023].

[6] Gilbert Laporte. Fifty years of vehicle routing. Transportation Science, 43(4):408– 416, 2009.

[7] J. Tang, Z. Pan, R. Y. K. Fung, and H. Lau, *Vehicle routing problem with fuzzy time windows,* Fuzzy Sets Syst., vol. 160, no. 5, pp. 683–695, 2009.

[8] R. Eglese and T. Bektas, *Green Vehicle Routing, in Vehicle routing: Problems, Methods, And Applications*, 2nd ed., P. Toth Vigo, D., Ed. SIAM, 2014, p. 463.

[9] A. V Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli, and L. M. Gambardella, *Time dependent vehicle routing problem with a multi ant colony system*, Eur. J. Oper. Res., vol. 185, no. 3, pp. 1174–1191, 2008