

# Inteligencia Artificial

## Informe Final: Problema Green Vehicle Routing Problem (GVRP)

Lucio Fondón Rebolledo

24 de diciembre de 2021

### Evaluación

Mejoras 2da Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (15 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
<b>Nota Final (100):</b>	_____

### Resumen

En este trabajo se realizará un breve estudio del problema de optimización de Green Vehicle Routing Problem (GVRP), el cual consiste en minimizar la distancia recorrida por una flota de vehículos que deben realizar repartos a ciertos clientes predispuestos en posiciones fijas. Esta flota de vehículos debe iniciar su recorrido en un depósito y al finalizar deben volver a este. Durante su recorrido, los vehículos pueden detenerse a recargar combustible si es que éstos lo requieren. Se presentará un modelo matemático y sus respectivas técnicas utilizadas a lo largo de los años. Además, se presentarán, describirán e implementarán algoritmos basados en técnicas incompletas, utilizando una técnica Greedy y luego Simulated Annealing para la resolución del problema. Finalmente, se concluirá respecto a los puntos realizados en el documento.

## 1. Introducción

Durante las últimas décadas, se ha dado cada vez más importancia al cuidado del medio ambiente en general, donde los gobiernos y empresas se han dedicado a tomar medidas para poder contaminar lo menos posible y atenerse a lo que son las *políticas verdes*, ya que las logísticas que se utilizan hoy en día para la realización de ciertos procesos no son convenientes a largo plazo para el sostenimiento de la calidad del planeta ni el medio ambiente.

Existen una gran cantidad de problemas que se atienen a lo que son las políticas verdes. Uno de los problemas que se estudiará será el de Green Vehicle Routing Problem (GVRP), con el fin

de poder entender mejor como resolver esta clase de problemas y analizar la dificultad asociada a este. El GVRP consiste en minimizar la distancia recorrida por una flota de vehículos que deben realizar entregas a clientes, partiendo desde un depósito, para luego volver a éste. Durante el trayecto, pueden detenerse a recargar combustible, el cuál es un combustible alternativo que utilizan los vehículos, con el objetivo de minimizar el impacto ambiental producido por éstos. Esto puede parecer una tarea no muy difícil de realizar, pero la dificultad recae en el uso de combustible alternativo en los autos que se utilizarán en la flota, ya que, al ser combustible alternativo, también es menos común su uso, por lo que las estaciones de recarga de combustible serán más escasas.

Primero, se definirá el problema de una manera más coloquial, señalando las principales características del problema, sus restricciones y objetivos, además de mostrar también las principales variantes que existen del problema. Luego, se realizará el *Estado del Arte* del problema, en donde se hará hincapié principalmente en los métodos y algoritmos más utilizados para este tipo de problema, mostrando los distintos *approach* que se han utilizado por investigadores para tratar de resolver este problema.

Después, se presentará un modelo matemático para el problema en cuestión, con el fin de poder acercarse al problema y a su solución desde una perspectiva más matemática y computable, analizando más en profundidad las partes que componen al problema, tales como las variables, su función objetivo, sus restricciones y parámetros.

Luego, se detallará la representación de las soluciones del problema, indicando la forma en que se representarán las soluciones de las instancias predispuestas para el problema. También, se detallará la descripción de los algoritmos de Greedy y Simulated Annealing para poder implementar la resolución del problema.

Además, se presentarán las respectivas experimentaciones con las instancias dadas para la resolución del problema, para luego mostrar los resultados obtenidos de estas experimentaciones.

Finalmente, se mostrarán conclusiones con respecto a todo lo presentado anteriormente, especialmente respecto a ventajas y desventajas de utilizar o no ciertas técnicas para abordar el problema.

## 2. Definición del Problema

Uno de los problemas combinatoriales de optimización más famosos y estudiados es el Vehicle Routing Problem (VRP), el cual consiste en minimizar la distancia recorrida de una flota de vehículos que deben realizar entregas a un conjunto de clientes, con el objetivo de abaratar costos durante el trayecto. Los VRP caen en la clasificación de ser un problema *NP-Hard*, dado que es una generalización del TSP [7]. Antes de presentar las variantes del GVRP, se nombrarán y describirán brevemente las variantes más importantes que surgen del VRP, dado que están relacionadas también con las variantes de GVRP:

- ***Time-dependent VRP*** (TDVRP - 1966): Los primeros trabajos que consideraron esta variante del problema recae en los estudios realizados por Cooke y Halsey. La principal diferencia con VRP es que la distancia entre los puntos a repartir (ya sean clientes, estaciones de servicio o el depósito) varía dependiendo en qué momento del día se encuentre, ya sea por ejemplo, por condiciones climáticas, horas punta de tráfico, etc. La introducción del TDVRP fue muy relevante para estudios posteriores, en especial los problemas de optimización de redes de tráfico reales [6].

- **Multi-depot VRP** (MDVRP - 1969): Primeramente estudiado por Tillman en 1969, esta variante contempla que, en vez de que exista un sólo depósito inicial de donde salgan y tengan que volver los vehículos, existan múltiples depósitos de éstos. Este problema se origina principalmente debido a problemas de distribución de la vida real tales como el reparto de comidas, productos químicos, reparto de productos de una empresa, etc. [24]
- **Dynamic VRP** (DVRP - 1976): En las variantes anteriores, se lidia con información previa acerca de cómo y a donde se deben realizar las entregas, sin embargo, en la vida real no siempre es así. Speidel (1976) y Psaraftis (1980) estudiaron el problema de manera en que la información se va recibiendo en tiempo real, tales como localización de los vehículos y los pedidos de los clientes. Ejemplos de esto son los servicios de emergencia, servicios de rescate, etc. [22] [19].
- **VRP with Time windows** (VRPTW - 1977): Hasta ahora, no se han considerado los tiempos de servicio que se deban cumplir como restricciones. Russell (1977) presentó una heurística para este caso y propuso dos tipos de time windows [21]:
  - *Hard Time Windows*: El vehículo debe llegar antes o justo después del tiempo de entrega especificado, y no se permite llegar tarde. Además, si llega antes, el vehículo debe esperar hasta que pase la ventana de tiempo.
  - *Soft Time Windows*: Las violaciones a las ventanas de tiempo se permiten pero se penalizan a un cierto costo.

Teniendo lo anterior en cuenta, se da paso al GVRP (2012) [10]. Debido al constante deterioro del medioambiente en el planeta, los gobiernos y empresas comenzaron a adoptar las llamadas logísticas verdes dentro de sus operaciones, es por esto que nace el problema que estudiaremos acá, el Green Vehicle Routing Problem (GVRP), el cual es una variante del VRP en donde los vehículos predispuestos para realizar las entregas a los clientes utilizan un combustible alternativo, con el fin de dejar la menor cantidad de contaminación posible en la realización de sus tareas.

El GVRP consiste en realizar una ruta de entregas de una cierta flota de vehículos que parten desde un mismo depósito hacia un conjunto de clientes que se encuentran dispersos geográficamente en un área en particular, para luego volver al mismo depósito inicial, esto con el fin de poder minimizar la distancia recorrida por esta flota de vehículos y así poder también abaratar los costos asociados al trayecto en general. Estos vehículos pueden realizar paradas en caso de que requieran combustible. Notar que, al utilizar combustible alternativo por ser GVRP, las estaciones serán más limitadas. Además, debido a que GVRP es una variante de VRP, es considerado también un problema *NP-Hard*. Ahora, se presentarán variantes que han salido del GVRP en los últimos años:

- **Fuzzy GVRP** (F-GVRP - 2018): F-GVRP fue propuesto por primera vez por los autores Poonthalir y Nadarajan [18]. Las principales diferencias con GVRP recaen en que F-GVRP es multiobjetivo, ya que busca minimizar el costo de la ruta asociada y además el consumo de combustible, GVRP solo minimiza el costo de la ruta. Además, en GVRP, la velocidad y el consumo de combustible es constante, en F-GVRP no lo son.
- **Capacitated GVRP** (CGVRP - 2018): CGVRP fue introducido por Zhang [26]. La diferencia de esta variante con la original de GVRP recae principalmente en que considera restricciones que están asociadas a las capacidades de almacenamiento de inventario que puedan llevar los autos pertenecientes a la flota.
- **Pollution routing problem** (PRP - 2011): Introducida por primera vez por Bektas y Laporte [3]. Esta variante del GVRP se diferencia debido a que la función objetivo se concentra en minimizar el gasto de combustible de los autos y el costo de los conductores.

- **Energy-minimizing vehicle routing problem** (EMVRP - 2007): Presentado e investigado inicialmente por Kara et al. [11]. En esta variante, su función objetivo se enfoca principalmente en minimizar el trabajo (o energía) producida por los vehículos, basándose en un función de costos que se obtiene a partir de los pesos de los vehículos de la flota y las distancias recorridas.

Las principales variables a tener en cuenta dentro de este problema serán variables relacionadas a si cierto vehículo viaja de un punto a otro y variables relacionadas al tiempo de llegada de un vehículo a un punto y el combustible disponible en los vehículos. Además, se tienen que considerar ciertas restricciones que existen en el problema. Los vehículos están restringidos por una cantidad de distancia que pueden recorrer con la cantidad de combustible que almacenan. También, las visitas a un cliente tienen un tiempo de servicio asociado. Cuando un vehículo requiera llenarse de combustible, también tomará un tiempo de recarga de combustible. Además, los vehículos están restringidos a un tiempo de servicio que no pueden sobrepasar antes de volver al depósito. Por otro lado, cada cliente debe visitarse una sola vez por un solo vehículo. Por último, se tomará en consideración que no existe un límite de vehículos disponibles en la flota y que la velocidad de estos es constante.

### 3. Estado del Arte

Como se mencionó anteriormente, el GVRP es una variante del problema original VRP. El Vehicle Routing Problem fue introducido por primera vez por Dantzig y Ramser (1959), y mencionan que el problema viene directamente del conocido Traveling Salesman Problem (TSP) [7]. El problema lo describen como un *routing problem*, en donde camiones con gasolina debían repartir a ciertas estaciones de servicio geográficamente esparcidas en un espacio en particular. Propusieron un algoritmo basado en programación lineal entera, para poder obtener una solución cercana a la óptima. A esta variante inicial de VRP se le llamó *Capacitated VRP* [15].

Lo anterior es considerado un hincapié inicial para las tantas otras variantes que surgen del problema, hasta finalmente llegar a la variante del GVRP. Se muestran a continuación los métodos y algoritmos utilizados en general para la resolución de los GVRP.

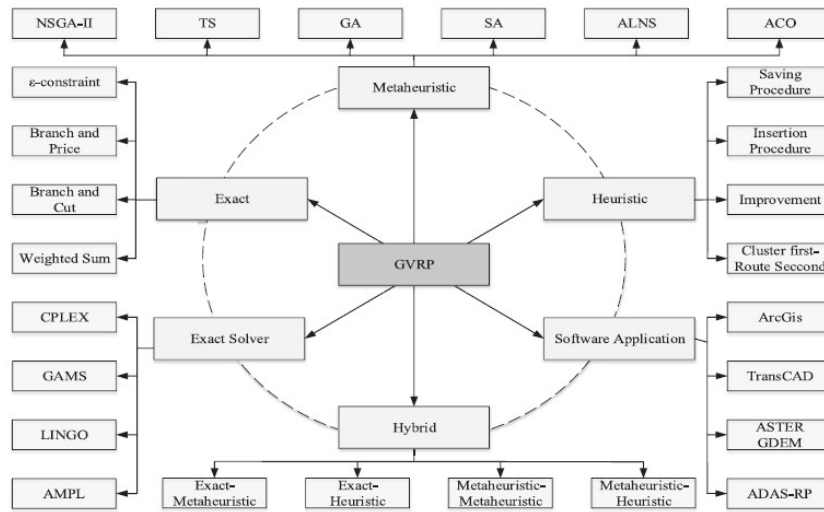


Figura 1: Métodos de resolución más utilizados para GVRP [16]

Se detallarán ahora brevemente algunos de los métodos utilizados más importantes para la resolución del GVRP:

### 3.1. Algoritmos Exactos

Dentro de los algoritmos exactos utilizados para poder resolver problemas de GVRP, podemos destacar:

- **Programación Lineal Entera:** Es sabido que la programación lineal entera ha sido uno de los primeros métodos en general para la resolución de problemas en donde se debían minimizar o maximizar una cierta variable para algún problema. En el caso de los problemas de GVRP no es distinto; Erdoğan y Miller-Hooks, quienes fueron las primeras autoras en proponer los GVRP como tal, propusieron el GVRP con un modelo de programación lineal entera mixta [10]. Lo proponen también Andelmin y Bartolini, donde formulan la resolución con un método de programación lineal entera mixta con *set partitioning* [1]. Por lo general, utilizan la modelación del problema como un programación lineal entera mixta y luego buscan distintos approach para resolver las instancias. Por ejemplo, en [4] se modela el problema como programación lineal entera mixta, pero se resuelve utilizando un approach basado en caminos. Por otra parte, también se han utilizado técnicas de *Branch and Price* [25] y *Branch and Cut* [5]. Sin embargo, las soluciones utilizando programación lineal entera, por lo general, no pueden escalar a problemas dentro de la vida real, específicamente por la gran cantidad de variables binarias que se generan, por lo que se hace muy difícil computacionalmente llegar a soluciones exactas óptimas [13].

### 3.2. Algoritmos Aproximados

En los algoritmos aproximados que se han utilizado a lo largo de la historia para resolver instancias del GVRP y sus variantes, podemos dividirlos en dos categorías: las *heurísticas clásicas* y las *metaheurísticas*. Dentro de las heurísticas clásicas se pueden destacar los siguientes métodos utilizados:

- **Algoritmos de Ahorro:** Clarke y Wriyth (1964) propusieron una heurística clásica para el VRP, y probablemente sea la más conocida para este tipo de problema, y generalmente se utiliza cuando el número de vehículos son una variable de decisión. Este método produjo soluciones cuyos valores se encuentran un 6,71 % por sobre las que se conocen [12]. En el contexto más contemporáneo con el problema del GVRP, se propuso en 2012 una utilización del método de Clarke y Wriyth modificado, logrando una optimización del 14,9 % en distancia recorrida [2].
- **Algoritmos de Clustering:** También, se han utilizado técnicas de agrupamientos de datos o *clustering* para poder resolver instancias del GVRP. Si bien Erdoğan y Miller-Hooks proponen el modelo para resolver el problema como uno de programación lineal entera mixta, utilizan una heurística *Density-Based Clustering Algorithm* (DBCA). Esta técnica aprovecha características que otras técnicas no consideran, tales como las propiedades espaciales del GVRP. La idea principal del algoritmo es tratar de agrupar en un círculo de radio  $\epsilon$  una cantidad mínima de clientes junto con las estaciones de recarga (*minPts*). Cada clúster representará una ruta factible que un vehículo de la flota puede recorrer. Este algoritmo es comparado con un algoritmo de ahorro de Clarke y Wright modificado, en donde se concluye que DBCA tiende a tener resultados un poco más óptimos [10].

Por otro lado, dentro de las metaheurísticas, se pueden dividir en dos ramas:

- **Técnicas de Búsqueda Local:** Estas técnicas se basan en explotar el espacio de soluciones al moverse iterativamente desde una solución a otra más prometedora dentro de su

vecindario [15]. Dentro del GVRP, está *Tabu Search* (TS), que Ehmke lo utiliza en sus estudios para modelar una resolución al problema de GVRP principalmente porque, para las variantes de VRP, Tabu Search ha dado buenos resultados [9]. También se ha utilizado *Simulated Annealing* (SA) en varias ocasiones [17], utilizándose también en algoritmos híbridos de SA con branch and cut (B&C), para poder diversificar e intensificar de manera correcta en el espacio de búsqueda de soluciones [27].

- **Técnicas basadas en Población:** Los métodos basados en población mantienen un *pool* de soluciones prometedoras, y se actualiza este pool a medida que encuentra otras mejores. En esta categoría se encuentran los *Genetic Algorithms* (GA) y los *Ant Colony Optimization* (ACO). En el caso de los GA, se ha demostrado que se obtiene un 17,91% de mejora en la solución que si se utilizan técnicas de modelado de TSP para resolver el problema de GVRP [8]. Para los ACO, se demuestra que para instancias de gran tamaño, tienen muy buenos resultados, por lo que utilizar ACO es útil para poder resolver instancias del mundo real de GVRP.[14].

Las publicaciones en los últimos años acerca de resoluciones y métodos nuevos para el GVRP y variantes del VRP siguen creciendo. Sin embargo, los estudios que existen de las variantes de GVRP son, limitados principalmente debido a que involucra muchos factores que son complicados hoy en día, tales como energía e impacto ambiental, polución, planificación urbano, etc [16].

En general, los problemas de GVRP que se analizan hoy en día tienen que ver con aquellos que tienen comportamiento *no determinista*, es decir, **GVRP con incertidumbres**. Algunos ejemplos son los GVRP *fuzzy chance-constrained mixed integer non-linear programming model* [23] y los de optimización estocástica [20].

## 4. Modelo Matemático

Ahora, se presentará un modelo matemático para el problema del GVRP, el cuál se define como un grafo completo no dirigido  $G = (V, E)$ . En este caso, se utilizará el modelo planteado por Erdoğan y Miller-Hooks [10]. Este estudio fue el primero en introducir lo que son los *Alternative Fuel-powered Vehicles* (AFV) en el problema del GVRP, por lo que toma en cuenta lo que necesitamos para poder resolver el problema utilizando políticas verdes y obtener una solución óptima en minimización de distancia recorrida y costos de entrega. Cabe destacar que este modelo es formulado en base a programación lineal entera mixta.

### 4.1. Parámetros

- $I$ : Conjunto de vértices de los clientes.
- $F$ : Conjunto de vértices de las estaciones de recarga.
- $v_0$ : Depósito inicial (punto de partida y llegada de los vehículos).
- $V$ : Conjunto de todos los vértices (Notar que  $V = \{v_0\} \cup I \cup F$ ).
- $d_{ij}$ : Distancia desde el vértice  $i$  hasta el vértice  $j$ .
- $p_i$ : Tiempo de servicio que toma en el vértice  $i$  (Si  $i \in I$ , entonces corresponde al tiempo de servicio al cliente. Si  $i \in F$ , entonces corresponde al tiempo de recarga de combustible en una estación).
- $Q$ : Capacidad del tanque de combustible del vehículo. (El valor del combustible restante de un vehículo toma el valor de  $Q$  cada vez que se recargue combustible).

- $r$ : Velocidad de consumo de combustible del vehículo.
- $T_{max}$ : Tiempo máximo de servicio del vehículo.

#### 4.2. Variables

- $x_{ij} = \begin{cases} 1 & \text{Si el vehículo viaja desde el vértice } i \text{ hasta el vértice } j \\ 0 & \text{Caso contrario} \end{cases}$
- $y_j$  = Nivel de combustible del vehículo restante al llegar al vértice  $j$ .
- $\tau_j$  = Tiempo de arribo de un vehículo al vértice  $j$ . Se inicializa en 0 partiendo desde  $v_0$ .

#### 4.3. Función objetivo

$$\text{mín } F : \sum_{\substack{(i,j) \in V \\ i \neq j}} d_{ij} x_{ij} \quad (1)$$

Ecuación 1: Función objetivo asociada al problema del GVRP propuesta por Erdoğan y Miller-Hooks. Busca minimizar la distancia total recorrida por un vehículo de la flota.

#### 4.4. Restricciones

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, \quad \forall i \in I \quad (2)$$

Ecuación 2: Maneja que cada vértice perteneciente al conjunto de los clientes tenga un solo sucesor, ya sea una estación de recarga, otro cliente o el depósito inicial  $v_0$ .

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} \leq 1, \quad \forall i \in F \quad (3)$$

Ecuación 3: Se asegura que cada vértice perteneciente al conjunto de las estaciones de recarga tenga a lo más un sucesor, ya sea una estación de recarga, otro cliente o el depósito inicial  $v_0$ .

$$\sum_{\substack{i \in V \\ j \neq i}} x_{ji} - \sum_{\substack{i \in V \\ j \neq i}} x_{ij} = 0, \quad \forall j \in V \quad (4)$$

Ecuación 4: Esta restricción se asegura que el número de llegadas a cada vértice perteneciente a  $V$  sea el mismo que el número de salidas.

$$\tau_j \geq \tau_i + (t_{ij} - p_j)x_{ij} - T_{\max}(1 - x_{ij}), \quad \forall i \in V, \forall j \in V \setminus \{0\}, i \neq j \quad (5)$$

Ecuación 5: Esta restricción es necesaria para tener registro del tiempo en el que cada vértice  $i$  es visitado y el tiempo en que le toma llegar al vértice  $j$ , para así poder prevenir subrutinas.

$$0 \leq \tau_0 \leq T_{\max} \quad (6)$$

Ecuación 6: Se asegura de que cada vehículo no exceda el tiempo máximo  $T_{\max}$  de tiempo de llegada de vuelta al depósito, especificando el tiempo con el que parte el vehículo desde el depósito  $v_0$ .

$$t_{0j} \leq \tau_j \leq T_{\max} - (t_{j0} + p_j), \quad \forall j \in V \setminus \{0\} \quad (7)$$

Ecuación 7: Se asegura que cada vehículo termine la ruta en el tiempo máximo  $T_{\max}$ .

$$y_j \leq y_i - r \cdot d_{ij}x_{ij} + Q(1 - x_{ij}), \quad \forall j \in I \text{ y } i \in V, i \neq j \quad (8)$$

Ecuación 8: Se encarga de reducir el nivel de carga de combustible del vehículo que llega a un vértice  $j$ , considerando la distancia recorrida desde  $i$  hasta  $j$  y la velocidad  $r$  a la que va el vehículo.

$$y_j = Q, \quad \forall j \in F \quad (9)$$

Ecuación 9: Encargada de setear nuevamente el nivel de combustible de un vehículo al visitar a una estación de recarga.

$$y_j \geq \min\{r \cdot d_{j0}, r(d_{jl} + d_{l0})\} \quad (10)$$

Ecuación 10: Garantiza que el vehículo cuente con el combustible necesario para poder volver al depósito o a una estación de recarga.

#### 4.5. Naturaleza de las Variables

- $x_{ij}$ : Variable de decisión booleana.  $x \in \{0, 1\}$
- $\tau_j, y_j$ : Variables tipo entero.  $\tau_j, y_j \in \mathbb{Z}$



## 5. Representación

Para efectos de este documento, la representación de la solución escogida viene dada por una lista enlazada simple, en donde esta lista estará compuesta de nodos secuencialmente unidos por un puntero. Estos nodos tendrán como dato (e identificación) al nodo que se visita durante la ruta. Una representación posible de una solución podría ser:



La anterior solución a una instancia ficticia del problema representa una ruta posible que puede tomar un vehículo en cuestión, es decir, el vehículo parte desde el depósito **d0**, para luego ir a visitar a 3 clientes, **c7-c4-c9**. Después va a visitar una estación de recarga **f1**, para luego visitar a un cliente más **c1**, y se devuelve al depósito **d0**.

Esta representación fue elegida más que nada por su fácil uso y secuencialidad intrínseca que conlleva, dado que es fácil observar cual es claramente la ruta que lleva un vehículo asociado a la flota. También, implementar esta representación no es difícil, ya que solo se tienen que representar los nodos correspondientes a cada tipo de punto de llegada o salida (ya sea depósito, estación o cliente a visitar) y conectarlos mediante un puntero, así hasta llegar al último nodo, que se conectará a un nodo nulo para poder representar que se llegó al final y se acabó el recorrido del vehículo.

En cuanto a la relación que tiene esta representación con el espacio de búsqueda asociado al problema de GVRP, esta no reduce el espacio de búsqueda en sí, ya que el espacio de búsqueda asociado consistiría en obtener todos los subconjuntos posibles de rutas que se puedan obtener a partir de una instancia provista para el problema, los cuales representarían las soluciones factibles e infactibles del problema, pero al fin y al cabo estas soluciones si podrían ser reproducidas por esta representación elegida. Por otra parte, el espacio de búsqueda en sí se ve reducido más por la utilización del algoritmo Greedy que se detallará más adelante, ya que solo obtendrán soluciones factibles del problema.

## 6. Descripción del algoritmo

El algoritmo que se utilizará en este documento será una mezcla de tipo *Greedy*, para poder obtener una solución factible inicial, y luego a esta solución mejorarla utilizando un approach algorítmico con *Simulated Annealing*.

El algoritmo greedy funcionará de la siguiente forma: se partirá desde un nodo inicial (depósito)  $d_0$ , que se nombrará como **nodoActual**. Luego, se realizará una visita de nodos utilizando la distancia mínima que haya desde el nodo actual  $d_0$  hasta el nodo a visitar (es decir, la función miope a utilizar en el problema sería escoger el nodo de distancia mínima desde el nodo actual al nodo a visitar y la función de evaluación la distancia total recorrida por los vehículos), llámese **nodoVisitado**. Si ese nodo no ha sido visitado y esa distancia no hace que se agote el combustible, entonces se visitará el nodo y se actualizará **nodoActual** a **nodoVisitado**. En caso contrario, se escogerá el siguiente nodo no visitado con la distancia mínima del nodo actual al nodo visitado, y así hasta que encuentre uno factible. El algoritmo tendrá como tope de término que el tiempo máximo de servicio del vehículo no se exceda. Un nodo factible a visitar se denotará como que no exceda el combustible ni el tiempo al visitarlo y para volver al depósito, entonces se asegurará si o si de tener tiempo siempre para volver al depósito. Esta factibilidad se manejará en la posterior implementación.

---

**Algorithm 1** GreedyGVRP( $C[1 \dots \#Customers]$ ,  $F[1 \dots \#Stations]$ , MaxTime, MaxDistance, Speed, ServiceTime, RefuelTime)

---

```

 $S \leftarrow \emptyset$ 
 $nodoActual \leftarrow d_0$ 
 $actualTime \leftarrow 0$ 
 $actualDistance \leftarrow 0$ 
 $actualEstanque \leftarrow MaxDistance$ 
 $Flag \leftarrow TRUE$ 
while  $Flag$  do
     $nodoVisitado \leftarrow$  select a minimum distance feasible node from array C
     $distance \leftarrow haversineDistance(nodoActual, nodoVisitado)$ 
    if  $actualEstanque - distance < 0$  then
         $estacionVisitada \leftarrow$  select a minimum distance feasible node from array F
         $distance \leftarrow haversineDistance(nodoActual, estacionVisitada)$ 
         $actualDistance \leftarrow actualDistance + distance$ 
         $actualEstanque \leftarrow MaxDistance$ 
         $actualTime \leftarrow actualTime + RefuelTime$ 
         $S \leftarrow S \cup \{estacionVisitada\}$ 
    else
         $actualDistance \leftarrow actualDistance + distance$ 
         $actualEstanque \leftarrow actualEstanque - distance$ 
         $actualTime \leftarrow actualTime + ServiceTime$ 
    end if
     $tripTime \leftarrow actualDistance / Speed$ 
     $actualTime \leftarrow actualTime + tripTime$ 
    if  $actualTime > MaxTime$  then
         $Flag \leftarrow FALSE$ 
    else
         $S \leftarrow S \cup \{nodoVisitado\}$ 
    end if
end while

```

---

Ahora, se presentará un algoritmo de SA para tratar de mejorar la solución entregada por el algoritmo greedy presentado anteriormente. Este algoritmo de SA funciona muy parecido a los algoritmos de SA por definición, los cuales *intensifican* al realizar un búsqueda local en el vecindario de la solución provista por el algoritmo greedy y *diversifica* al utilizar las técnicas de enfriamiento de la temperatura, para poder aceptar soluciones de mejor calidad. Es decir, se irán revisando los vecinos de la solución actual mediante movimientos, para luego enfriar la temperatura, para así haber menos probabilidades de aceptar una solución de peor calidad. Solo se deben definir correctamente los criterios de término y los movimientos para cambiar la solución actual. Para efectos de esta implementación, se utilizará como movimiento un intercambio entre las visitas que se hacen a los nodos que estén consecutivos. Para evidenciar un ejemplo, se usará el caso ficticio que se mostró anteriormente en la representación de las soluciones:

d0-c7-c4-c9-f1-c1-d0

Entonces, en este caso, se pueden realizar 4 posibles intercambios entre el orden de visita de los nodos: c7-c4, c4-c9, c9-f1 y f1-c1. Claramente, al ser una instancia ficticia del problema, se está asumiendo que estos intercambios hacen posible que la solución aún sea factible, por lo que es posible realizarlos (en el caso del código en sí, estos casos deben manejarse, ya que solo deben tomarse en cuenta los intercambios factibles). El algoritmo utilizará como criterio de parada que la temperatura se enfríe hasta cierto punto, que se deberá entregar como parámetro al algoritmo. Se presenta ahora un pseudocódigo de lo explicado anteriormente:

---

**Algorithm 2** SimulatedAnnealingGVRP( $T$ ,  $T_{min}$ , iter)

---

```

 $S_c \leftarrow GreedyGVRP()$ 
 $S_{best} \leftarrow S_c$ 
while  $T > T_{min}$  do
   $S_n \leftarrow$  select a new point in neighborhood of  $S_c$ 
  if  $f(S_n)$  is better than  $f(S_c)$  then
     $S_c \leftarrow S_n$ 
  else if  $random([0, 1]) < e^{\frac{\Delta_{eval}}{T}}$  then
     $S_c \leftarrow S_n$ 
  end if
  if  $f(S_c)$  is better than  $f(S_{best})$  then
     $S_{best} \leftarrow S_c$ 
  end if
   $T \leftarrow cooling(T)$ 
end while

```

---

## 7. Experimentos

Para dar comienzo al proceso de experimentación con las instancias provistas para la resolución del problema del GVRP, se deben tener en cuenta bien los objetivos de este proceso. En este caso, son dos los objetivos a considerar:

1. No excederse del tiempo total de recorrido por un vehículo en alguna ruta
2. Minimizar la distancia total recorrida por la flota de vehículos

Si bien la principal variable objetivo a minimizar en este caso es la distancia total recorrida por la flota de vehículos, también es importante tener en cuenta que no se debe exceder del

tiempo total de recorrido, ya que de lo contrario se estaría en presencia de una solución infactible de una instancia del problema de GVRP.

Para poder cumplir los objetivos anteriores, se experimentará y trabajará con varias instancias de las provistas para la resolución del GVRP. Durante el proceso, se utilizaron todas las instancias para poder experimentar, sin embargo, hubo en muchas de estas instancias que existían rutas en donde solo se visitaban 1 o 2 nodos, esto puede deberse principalmente porque los nodos están muy lejos entre sí y al utilizar en primera instancia un approach greedy, básicamente no alcanzaba a visitar más nodos y volver al depósito. Lo anterior sucede en todas las instancias provistas, pero en algunas se pueden encontrar rutas más largas. En este caso, se utilizarán 3 instancias distintas de las provistas para experimentar, AB101, AB108 y AB117. Se escogieron estas instancias debido a que cada una tiene distinta cantidad de clientes a satisfacer (50, 75 y 100 respectivamente) y existen clientes que están cercanos entre sí, por lo que será más propenso a generar rutas con varios clientes. Durante la experimentación se sintonizaron dos parámetros:

1. El factor multiplicativo de enfriamiento de temperatura para SA
2. El tope de término del algoritmo greedy, que en este caso se escogió un factor multiplicativo con la cantidad de clientes visitados

Para el primer parámetro, se fue variando el factor multiplicativo  $\alpha$  de la temperatura en un rango de  $\alpha \in \{0.8, 0.99\}$ , donde finalmente se optó por mantener un  $\alpha$  constante de 0.8, debido a que no se notó un cambio en los resultados durante la sintonización, entonces se optó por enfriar más rápidamente la temperatura. Cabe destacar que la temperatura mínima  $T_{min}$  se dejó en un valor constante de 10 y el valor inicial de la temperatura con un valor constante de 100, puesto que no afectaron en los resultados durante los experimentos. Para el segundo parámetro, se sintonizó en un rango de  $\{0.5, 0.95\}$ , donde se optó finalmente por utilizar 0.95, esto debido a que en cierto punto, se violaba la restricción de tiempo máximo de ruta debido a que visitaba nodos que se encontraban demasiado lejanos al depósito.

Los datos provistos por las instancias del problema contienen la información de todos los nodos (ya sea un cliente, depósito o estación), donde se entregan datos como la longitud y latitud de estos nodos, esto para poder calcular la distancia que exista entre cualquier nodo. Para el cálculo de esta distancia, se utilizó la *distancia de Haversine*, que permite calcular la distancia entre dos puntos cualesquiera de la tierra obteniendo sus respectivas latitudes y longitudes. Esta distancia se puede calcular con la siguiente fórmula:

$$d = 2r \sin^{-1} \left( \sqrt{\sin^2 \left( \frac{\Phi_2 - \Phi_1}{2} \right) + \cos(\Phi_1) \cos(\Phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (11)$$

Ecuación 11: Fórmula para la distancia de Haversine.

Donde  $\Phi_1, \Phi_2$  corresponde a las latitudes de los nodos y  $\lambda_1, \lambda_2$  a sus longitudes.  $r$  es el radio de la tierra, donde se utiliza un radio de aproximadamente  $r = 6729,56122$

Cabe destacar que el hardware utilizado en el proceso de experimentación contiene una CPU *Intel Core i5-8250U* y 8 GB de RAM. También, como se está haciendo uso de técnicas incompletas para la resolución de las instancias, se setean semillas aleatorias en el proceso del algoritmo de SA.

## 8. Resultados

Ahora, se mostrarán los resultados obtenidos a partir de las instancias AB101, AB108 y AB117. Los resultados se muestran en los Cuadros 1, 2 y 3 respectivamente, los cuales representan una tabla para cada resultado de la instancia. Estas tablas contienen 3 columnas distintas, en donde la primera denota las rutas recorridas por cada uno de los vehículos pertenecientes a la flota. La segunda y tercera columna representan la distancia y tiempo utilizado de las rutas asociadas a la primera columna.

Cuadro 1: Resultados Instancia AB101

Soluciones AB101		
Ruta	Distancia	Tiempo
d0-c37-c8-c9-c32-f7-c12-d0	280.981161	584
d0-f16-c6-c18-d0	300.000000	442
d0-c30-d0	233.823555	380
d0-c41-d0	236.184552	384
d0-c19-d0	245.342924	398
d0-c15-d0	283.193913	454
d0-c5-d0	286.682035	460
d0-c4-d0	289.816476	464
d0-c46-d0	298.636110	476
d0-c28-d0	299.440924	478
d0-f8-c50-d0	308.449428	506
d0-f8-c10-d0	311.638726	512
d0-f8-c22-d0	312.390574	512
d0-f8-c21-d0	315.309805	516
d0-f8-c26-d0	320.973457	526
d0-f8-c16-d0	323.537803	530
d0-f8-c45-d0	325.319337	532
d0-f8-c23-d0	327.380048	536
d0-f8-c17-d0	332.378687	542
d0-f8-c29-d0	338.256143	552
d0-f8-c24-d0	340.167810	554
d0-f8-c25-d0	341.239625	556
d0-f8-c44-d0	369.953386	598
d0-f8-c38-d0	374.281989	606
d0-f8-c49-d0	389.602059	628
d0-f8-c11-d0	398.080204	642
d0-f8-c39-d0	401.796071	646

Cuadro 2: Resultados Instancia AB108

Soluciones AB108		
Ruta	Distancia	Tiempo
d0-c45-c40-c41-c16-d0	204.280735	584
d0-c72-c71-c73-c34-c38-d0	198.886448	442
d0-c33-c53-c52-c39-f1-c56-d0	299.050652	380
d0-c51-c23-c22-d0	193.183486	384
d0-c35-c29-d0	215.296003	398
d0-c17-d0	214.455918	454
d0-c70-d0	236.184552	460
d0-c74-d0	241.346741	464
d0-c57-d0	245.342924	476
d0-c15-d0	249.059931	478
d0-c36-d0	249.372915	478
d0-c50-d0	276.144631	478
d0-c18-d0	283.058234	478
d0-c32-d0	286.604919	478
d0-c12-d0	288.997969	478
d0-c10-d0	289.816476	478
d0-c69-d0	298.701692	478
d0-c6-d0	299.104028	478
d0-c30-d0	299.312296	478
d0-f8-c58-d0	308.449428	506
d0-f8-c59-d0	308.449428	506
d0-f8-c60-d0	308.449428	506
d0-f8-c7-d0	309.380327	508
d0-f8-c5-d0	312.390574	512
d0-f8-c9-d0	313.932639	514
d0-f8-c47-d0	314.695356	516
d0-f8-c4-d0	315.309805	516
d0-f8-c11-d0	316.275457	518
d0-f8-c1-d0	324.617353	530
d0-f8-c2-d0	324.617353	530
d0-f8-c3-d0	325.319337	532
d0-f8-c8-d0	327.380048	536
d0-f8-c21-d0	335.866347	548
d0-f8-c55-d0	340.436816	554
d0-f8-c26-d0	343.735114	560
d0-f8-c37-d0	352.027579	572
d0-f8-c24-d0	356.817451	580
d0-f8-c44-d0	359.573339	584
d0-f8-c61-d0	374.281989	606
d0-f8-c48-d0	377.434821	610
d0-f8-c31-d0	387.464907	626
d0-f8-c25-d0	389.602059	628
d0-f8-c42-d0	397.978618	640
d0-f8-c62-d0	398.080204	642
d0-f8-c75-d0	401.308147	646
d0-f8-c63-d0	401.796071	646
d0-f8-c68-d0	402.778044	648

Cuadro 3: Resultados Instancia AB117

Soluciones AB108		
Ruta	Distancia	Tiempo
d0-c99-c9-c90-c21-d0	176.924360	384
d0-c81-c56-c24-c8-d0	192.644042	406
d0-c93-c34-c97-f17-c77-d0	290.350025	567
d0-c73-c11-c19-d0	194.630726	380
d0-c18-c10-c72-d0	199.753406	388
d0-c39-c14-c55-d0	199.478829	418
d0-c36-d0	150.104890	254
d0-f8-c12-c53-d0	300.000000	465
d0-c59-d0	207.384758	340
d0-c30-d0	220.826311	360
d0-c64-d0	225.156553	366
d0-c41-d0	233.823555	380
d0-c78-d0	237.195111	384
d0-c33-d0	237.619464	386
d0-c54-d0	245.498889	398
d0-c87-d0	249.885283	404
d0-c15-d0	254.285335	410
d0-c16-d0	261.182722	420
d0-c94-d0	262.015339	422
d0-c17-d0	274.372664	440
d0-c13-d0	276.144631	444
d0-c48-d0	283.058234	454
d0-c69-d0	283.193913	454
d0-c68-d0	286.682035	460
d0-c67-d0	299.104028	478
d0-f1-c1-d0	302.193272	497
d0-f1-c5-d0	304.671397	501
d0-f1-c22-d0	305.288744	501
d0-f1-c49-d0	309.986426	509
d0-f1-c66-d0	310.600876	509
d0-f1-c61-d0	315.369560	517
d0-f1-c6-d0	315.936831	517
d0-f1-c95-d0	320.610407	525
d0-f1-c44-d0	322.671118	529
d0-f1-c7-d0	331.157418	541
d0-f1-c40-d0	333.547214	545
d0-f1-c43-d0	334.820935	547
d0-f1-c91-d0	335.478871	547
d0-f1-c96-d0	336.530695	549
d0-f1-c47-d0	339.026185	553
d0-f1-c82-d0	345.959977	563
d0-f1-c98-d0	352.108522	573
d0-f1-c85-d0	354.101239	575
d0-f1-c38-d0	358.976990	583
d0-f1-c65-d0	365.244457	591
d0-f1-c4-d0	369.258475	597
d0-f1-c2-d0	369.573060	599

De los resultados, se pueden analizar varias cosas. Primero, notemos que el acercamiento greedy junto con su implementación hacen que prontamente se visiten a los nodos más cercanos entre sí, lo cual a la larga no es bueno en términos de resultados. Esto básicamente porque al visitarse entre sí los nodos más cercanos en las primeras rutas, luego los vehículos que restan tendrán que visitar a los más lejanos entre sí, por lo que tenderán a realizar rutas más cortas y estancarse en visitar a un solo cliente o a una sola estación y luego un cliente, para luego retornar al depósito sin violar las restricciones del problema.

También, cabe destacar que los resultados mostrados anteriormente son luego de aplicar Simulated Annealing a las soluciones obtenidas por el algoritmo greedy. Por lo explicado en el párrafo anterior, se puede decir también que Simulated Annealing no podrá mejorar (y hasta empeorar) la solución obtenida por greedy, esto básicamente porque al haber tan pocas rutas en donde existan varios nodos visitados, existirán menos rutas en las que se podrán realizar swaps para buscar soluciones mejores en su vecindario, por lo que se aplica poco SA basado en la implementación que se ha mostrado en este escrito. De hecho, en los 3 resultados obtenidos anteriormente, no se logró mejorar la solución obtenida por greedy. La siguiente tabla muestra un resumen con las distancias del recorrido total:

Cuadro 4: Distancias comparativas de cada algoritmo con las instancias

	AB101	AB107	AB117
Greedy	Distancia: 8584.856802	Distancia: 15178.652353	Distancia: 16129.914816
Greedy + SA	Distancia: 8591.764211	Distancia: 15199.592045	Distancia: 16163.875929

Finalmente, se puede notar que SA no mejoró sino que empeoró las soluciones que fueron entregadas por el algoritmo greedy, esto se puede explicar por lo dicho en el párrafo anterior y además por las características estocásticas que tiene SA al tener la probabilidad de poder aceptar soluciones de peor calidad en su proceso de selección de vecindario.

## 9. Conclusiones

Los tipos de problemas de optimización de VRP se han vuelto muy importantes para poder mejorar los procesos de reparto de suministros a clientes, en donde se busca minimizar los costos asociados y sacarle el máximo provecho a los vehículos que se utilizan. A lo largo del escrito, se vieron distintos tipos de métodos, algoritmos y técnicas que se utilizan para poder resolver estos problemas. Dependiendo del tipo de variante del GVRP que se esté tratando de resolver se utilizaron distintos métodos, pero por lo general, los algoritmos de programación lineal entera y programación lineal entera mixta son los que predominan principalmente por su simpleza en la formulación de éstos, no necesariamente siendo los que obtienen mejores resultados en el sentido de que, al ser algoritmos exactos, el tiempo de cómputo de las soluciones puede llegar a ser tremendamente alto en instancias donde se tenga un input grande. Por otro lado, los algoritmos aproximados suelen tener mejores resultados que los algoritmos exactos debido a que buscan soluciones cercanas las óptimas con un margen de error muy pequeño y con un tiempo de cómputo mucho menor que los algoritmos exactos, siendo las técnicas de búsqueda local como Simulated Annealing (SA) las que han llevado a obtener mejores resultados, especialmente la utilización híbrida de SA con B&C [27]. En este escrito, se propone una combinación de técnicas incompletas de búsqueda local, realizando una primera solución de instancias con un approach



greedy para obtener siempre la ruta más corta posible desde un nodo hacia otro, para luego intentar mejorar esta solución utilizando SA mediante movimientos de swaps entre los nodos de las rutas y posteriormente decidir estocásticamente esta nueva solución del vecindario.

Respecto a los análisis realizados en los resultados con la implementación de los algoritmos, se puede destacar que debido a la implementación y la característica intrínseca del algoritmo greedy de elegir la menor distancia posible entre nodos, deja un gran vacío para los posteriores vehículos que quieran visitar nodos, debido a que se visita en primera instancia a todos los nodos que estén más cercanos entre sí, por lo que para las siguientes iteraciones, existirán cada vez menos nodos cercanos entre sí y se comenzarán a generar rutas donde se visiten a un solo nodo. Por lo anterior, SA tampoco logra realizar completamente bien su trabajo, ya que, al haber pocas rutas con varios nodos visitados, entonces existirán menos soluciones que mejorar debido a que las rutas en donde exista un solo nodo no pueden mejorarse, y como toma decisiones estocásticamente, no es trivial tratar de monitorear el proceso de selección de una solución del vecindario. Si bien se comentó anteriormente que la utilización de SA ha dado excelentes resultados en estudios de problemas de GVRP, no fue el caso en este escrito, principalmente debido a la implementación que se realizó de la heurística greedy. Para poder mejorar esto, quizás la opción sería utilizar otra técnica para la solución inicial de las instancias, como por ejemplo la técnica B&C mencionada anteriormente, en donde se conoce que su utilización con SA han dado buenos resultados en estudios anteriores [27].

Aunque se hayan realizado bastantes estudios en los problemas de VRP en general, el problema de GVRP es relativamente nuevo debido a la preocupación de hoy en día por minimizar la polución y tratar de realizar procesos industriales utilizando políticas verdes. Si bien se efectúan continuamente estudios sobre como abordar mejor el GVRP, para las investigaciones futuras se está tornando interesante el estudio de problemas de GVRP de carácter estocástico, por lo que probablemente las investigaciones venideras harán un buen hincapié en ese lado para poder seguir profundizando más en distintos métodos y técnicas para la resolución del problema.

## 10. Bibliografía

### Referencias

- [1] Juho Andelmin and Enrico Bartolini. An exact algorithm for the green vehicle routing problem. *Transportation Science*, 51(4):1288–1303, 2017.
- [2] Alfonso Aranda-Usón, German Ferreira Ferreira, Ignacio Zabalza, and David Zambrana-Vasquez. Study of the environmental performance of end-of-life tyre recycling through a simplified mathematical approach. *Thermal Science*, 16, 01 2012.
- [3] Tolga Bektas and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45:1232–1250, 09 2011.
- [4] M. Bruglieri, S. Mancini, F. Pezzella, and O. Pisacane. A path-based solution approach for the green vehicle routing problem. *Computers Operations Research*, 103:109–122, 2019.
- [5] Chun Cheng, Peng Yang, Mingyao Qi, and Louis-Martin Rousseau. Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review*, 97:97–112, 2017.
- [6] Kenneth L. Cooke and Eric Halsey. The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications*, 14:493–498, 1966.

- [7] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.
- [8] Paulo Roberto de Oliveira da Costa, Stefano Mauceri, Paula Carroll, and Fabiano Pallonetto. A genetic algorithm for a green vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 64:65–74, 2018. 8th International Network Optimization Conference - INOC 2017.
- [9] Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W. Thomas. Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, 251(2):478–494, 2016.
- [10] Sevgi Erdogan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 109:100–114, 01 2012.
- [11] İmdat Kara, Bahar Y. Kara, and M. Kadri Yetis. Energy minimizing vehicle routing problem. In Andreas Dress, Yinfeng Xu, and Binhai Zhu, editors, *Combinatorial Optimization and Applications*, pages 62–71, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [12] G. Laporte, Michel Gendreau, Jean-Yves Potvin, and Frédéric Semet. Classical heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7:285 – 300, 09 2000.
- [13] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345 – 358, 1992.
- [14] Yongbo Li, Hamed Soleimani, and Mostafa Zohal. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *Journal of Cleaner Production*, 227:1161–1172, 2019.
- [15] Canhong Lin, K.L. Choy, G.T.S. Ho, S.H. Chung, and H.Y. Lam. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4, Part 1):1118–1138, 2014.
- [16] Reza Moghdani, Khodakaram Salimifard, Emrah Demir, and Abdelkader Benyettou. The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, 279:123691, 2021.
- [17] Aschkan Omidvar and Reza Tavakkoli-Moghaddam. Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. pages 1089–1094, 09 2012.
- [18] G. Poonthalir and R. Nadarajan. A fuel efficient green vehicle routing problem with varying speed constraint (f-gvrp). *Expert Systems with Applications*, 100:131–144, 2018.
- [19] Harilaos N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2):130–154, 1980.
- [20] Mohammad Rahimi, Armand Baboli, and Yacine Rekik. Multi-objective inventory routing problem: A stochastic model to consider profit, service level and green criteria. *Transportation Research Part E: Logistics and Transportation Review*, 101:59–83, 2017.
- [21] Robert A. Russell. Technical note—an effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, 25(3):517–524, 1977.
- [22] V. Speidel. Edp-assisted fleet scheduling in tramp and coastal shipping. 1976.

- [23] Yan Sun, Martin Hrusovský, Chen Zhang, and Maoxiang Lang. A time-dependent fuzzy programming approach for the green multimodal routing problem with rail service capacity uncertainty and road traffic congestion. *Complex.*, 2018:8645793:1–8645793:22, 2018.
- [24] Frank A. Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3):192–204, 1969.
- [25] Yang Yu, Sihan Wang, Junwei Wang, and Min Huang. A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transportation Research Part B: Methodological*, 122(C):511–527, 2019.
- [26] Shuai Zhang, Yuvraj Gajpal, and S. Appadoo. A meta-heuristic for capacitated green vehicle routing problem. *Annals of Operations Research*, 269, 10 2018.
- [27] Çağrı Koç and Ismail Karaoglan. The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39:154–164, 2016.