

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Desarrollo de un algoritmo de optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

Daniel Ferrer Valencia

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial.
Bogotá, Colombia
Año: 2023

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

Daniel Ferrer Valencia

Trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Magister en Ingeniería Industrial

Director (a):
Ph.D., Gustavo. A. Bula

Línea de Investigación: Gestión de Operaciones (Profundización)

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial.

Bogotá, Colombia
Año: 2023

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

A mis padres

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

Nombre

Fecha DD/MM/AAAA

Fecha

Desarrollo de un algoritmo de Optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos.

Agradecimientos

A mis amigos que tanto me han apoyado, al profesor Gustavo. A. Bula por los aprendizajes recibidos, y especialmente a mi mamá como soporte emocional y apoyo en cada momento.

Resumen

Desarrollo de un algoritmo de optimización basado en búsqueda local guiada para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos

El objetivo de este trabajo es la construcción de un algoritmo basado en búsqueda local guiada para la minimización de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible para una flota de vehículos heterogénea. Para lograr este objetivo se construye un conjunto de métricas o características que permitan distinguir entre soluciones “óptimas” de aquellas no óptimas. Considerando diferentes valores de parámetros, se crean diversas instancias para las soluciones “óptimas” y aquellas que no lo son. Luego, por medio de un árbol de decisiones se obtiene la caracterización de las métricas que reflejan la diferencia entre las soluciones.

Posteriormente se analiza el impacto de un grupo definido de operadores de búsqueda local en la mejora de las métricas de las soluciones, de acuerdo con la caracterización hecha por el árbol de decisiones y se propone una versión de algoritmo búsqueda local guiado por la ordenada secuencia de los operadores que mejores resultados obtuvieron en la caracterización de esas métricas. Por último, se evalúa las soluciones entregadas por estos algoritmos respecto a otras soluciones.

Palabras clave: Problema de ruteo de vehículos con flota heterogénea, búsqueda local guiada, búsqueda local, aprendizaje de máquina, metaheurísticas.

Abstract

Development of an Optimization algorithm based on Guided Local Search for the Carbon footprint minimization problem in Heterogeneous Vehicle Routing

The objective of this work is the construction of an algorithm based on guided local search for the minimization of CO₂ equivalent emissions through the minimization of fuel consumption for a heterogeneous fleet of vehicles. To achieve this goal, a set of metrics or features is constructed to distinguish between "optimal" solutions and non-optimal ones. Considering different parameter values, several instances are created for "optimal" and non-optimal solutions. Then, by means of a decision tree, the characterization of the metrics that reflect the difference between the solutions is obtained.

Subsequently, the impact of a defined group of local search operators on the improvement of the metrics of the solutions is analyzed, according to the characterization made by the decision tree, and a version of the local search algorithm is proposed, guided by the ordered sequence of the operators that obtained the best results in the characterization of those metrics. Finally, the solutions provided by these algorithms are evaluated with respect to other solutions.

Keywords: Fleet size and mix vehicle routing problem, guided local search, local search, machine learning, metaheuristics.

Este Trabajo Final de maestría fue calificado en octubre de 2023 por el siguiente evaluador:

Juan David Suárez Moreno Msc.
Profesor Facultad de Ingeniería
Universidad Nacional de Colombia

Contenido

Capítulo 1: Problema de minimización de las emisiones equivalentes de CO2 a través de la minimización del consumo de combustible en el ruteo de vehículos con flota heterogénea	1
1.1 Definición del Problema de VRP	1
1.2 Formulación del modelo matemático para la solución del VRP-Heterogéneo	2
1.3 Estimación de las emisiones de CO2 equivalentes en el ruteo de vehículos.	4
1.4 Formulación del Modelo de PRP (Pollution Routing Problem) para vehículos heterogéneos	7
1.5 Técnicas de Solución para el VRP y el HFVRP	10
1.5.1 Modelos exactos	10
1.5.2 Técnicas heurísticas:.....	11
1.5.3 Técnicas Metaheurísticas:	12
Capítulo 2: Heurísticas basadas en búsqueda local y la metaheurística de búsqueda local guiada	15
2.1 Algoritmos de Búsqueda Local	15
2.2 Operadores de búsqueda local en problemas VRP.	17
2.2.1 Operadores Intra-rutas	17
2.2.2 Operadores Inter-rutas	19
2.3 Búsqueda Local Guiada.....	25
2.4 Descripción del proceso de Búsqueda Local Guiada	27
Capítulo 3: Algoritmo de búsqueda local guiada.	31
3.1. Conocimiento específico del problema.....	31
3.2 Construcción del conjunto de datos.....	32
3.3 Herramientas de aprendizaje de máquina.....	33
3.3.1 CART- Árboles de Clasificación y Regresión	34
3.3.2 Desviación estándar.....	36
3.3.3 Covarianza	36
3.3.5 PCA – Análisis de Componentes principales	37
3.4 Definición de parámetros de las soluciones.	38
3.4.1 Definición de las métricas para las Soluciones.	38
3.4.2 Definición de las Instancias.....	40

3.5	Evaluación y resultados de las métricas por medio de los modelos de aprendizaje de máquina.....	41
3.6	Interpretación de los resultados.....	46
Capítulo 4: Desarrollo de Algoritmo de Búsqueda Local Guiada.....		53
4.1	Descripción del Algoritmo.	54
4.2	Definición de secuencia.	56
4.3	Evaluación de la solución propuesta.	60
Capítulo 5: Conclusiones y recomendaciones		64
5.1	Recomendaciones	67

Lista de figuras:

Figura 2-1	Movimiento 2-Opt.....	18
Figura 2-2	Movimiento Relocate. Fuente:	19
Figura 2-3	Movimiento Cross-Exchange.	20
Figura 2-4:	Movimiento Cross-Exchange	21
Figura 2-5:	Movimiento Cross-Exchange	22
Figura 2-6:	Movimiento Ejection Chain.	23
Figura 2-7:	Movimiento Relocation-Chain.	25
Figura 2-8:	Pseudo-Código.	28
Figura 2-9:	Interpretación de atributos de selección.....	29
Figura 3-1:	Correlación de Métricas.	42
Figura 3-2:	Explicación de la varianza acumulada por componentes.	43
Figura 3-4:	Árbol de decisiones.....	45
Figura 3-5:	Reglas ampliadas de árbol de decisión.....	45
Figura 3-6:	Matriz de Confusión.	46
Figura 3-7:	Comparación de las soluciones.....	47
Figura 3-8:	Detalle de VRP para solución “mala”.	48
Imagen 4-1.	Diagrama de Flujo del Algoritmo de Búsqueda Local Guiada	55
Imagen 4.2:	Soluciones obtenidas por el Modelo de programación lineal entera mixta en Gurobi.....	60
Imagen 4.2:	soluciones obtenidas por el algoritmo de búsqueda local guiada con los operadores “no escogidos”.....	61
Imagen 4.3:	soluciones obtenidas por el algoritmo de búsqueda local guiada con los operadores “escogidos”.....	61

Lista de tablas

Tabla.1-1: Parámetros y unidades para la formulación de modelo PRP.	8
Tabla 2-2: Algoritmo Cross-Exchange.	23
Tabla 3-1: Árboles de decisión.	34
Tabla 3-2: Ejemplo de arcos transformados sobre cada solución del VRP	49
Tabla 3-3: Rutas de las 3 soluciones respuesta del VRP.....	50
Tabla 3-4: Resumen de número de arcos transformados para 5 experimentos de la solución “mala” a la solución “no tan buena”.	51
Tabla 3-5: Resumen de número de arcos transformados para 5 experimentos de la solución etiquetada “no tan buena” a la solución “buena”.	51
Tabla 4.1 Valor Promedio, desviación y mapa de calor de métrica o característica 8 respecto a los operadores.	57
Tabla 4.2 Valor Promedio, desviación y mapa de calor de métrica o característica 7 respecto a los Operadores.	58
Tabla 4.3 Valor Promedio, desviación y mapa de calor de métrica o característica 2 respecto a los operadores.	59
Tabla 4.4 Valor Promedio, desviación y mapa de calor de las soluciones comparadas. .	62

Abreviaturas

Abreviatura	Término
• <i>RC</i>	Operador Relocation Chain
• <i>CE</i>	Operador Cross Exchange
• <i>2OPT</i>	Operador 2 Opt.
• <i>FSMF</i>	Fleet Size and Mix-VRP
• HPRVRP	Heterogeneous Pollution Routing Vehicle Routing Problem

Introducción

Este trabajo tiene como objetivo proponer un algoritmo de búsqueda local guiada para el problema de minimización de la huella de carbono en el ruteo de vehículos heterogéneos.

La reducción de las emisiones de CO₂ tienen impacto en la velocidad con la que el planeta se está calentando. El aumento de la temperatura en la tierra afecta los ecosistemas en el corto plazo y con ello el equilibrio ecológico de la tierra. El consenso científico llamado “Maintaining Humanity’s Life Support Systems in the 21st Century” expresa que “es extremadamente probable que los sistemas de soporte vital de la tierra, críticos para la prosperidad y existencia humana, sean irremediablemente dañados” (Grupo Intergubernamental de Expertos sobre el Cambio Climático, 2019). El Departamento Nacional de Asuntos Oceanográficos y Atmosféricos de Estados Unidos (NOAA) y la NASA (Huang et al., 2017) confirmaron que la época comprendida entre los años 2010 y 2019 ha sido la década más calurosa desde que se empezaron a tener registros de la temperatura hace 140 años.

Según lo convenido en el Acuerdo de París (Naciones Unidas, 2015), el objetivo de todas las naciones debe enfocarse en limitar el calentamiento global por debajo de los 1.5 grados en el presente siglo. Tanto empresas, como instituciones y los gobiernos deben incluir dentro de los objetivos de la planeación logística el desacelerar el ritmo del calentamiento global a través de medidas como la reducción de la emisión de gases contaminantes.

Las empresas deberían emprender acciones que permitan disminuir la huella de carbono como objetivo. Las actividades de los parques automotores que se realizan de manera cotidiana generan un 75% de las emisiones de CO₂, como es el caso de la distribución y recolección de mercancías. Las emisiones de CO₂ es el factor que más incide en el calentamiento global (Suarez-Chilma et al., 2018),

Dentro de los aportes que desde la gestión de la logística se han propuesto en la reducción de la emisión de gases contaminantes está la inclusión en el problema del ruteo de vehículos con el objetivo de la minimización de la huella de carbono, especialmente desde dos perspectivas: la gestión de vehículos que no usen combustibles, y el diseño adecuado de rutas para minimizar el uso de combustible fósil (Rezgui et al., 2019). En cuanto a la primera, no es sencillo pedir a las empresas que cambien las flotas debido a los costos que esto conlleva (Astarita, 2013). La segunda, resulta más viable puesto que sí es posible que las organizaciones estén en disposición de analizar y encontrar cómo combinar las flotas, de tal manera que se disminuyan la huella de carbono, siendo esta perspectiva la que responde a la planeación de rutas de vehículo con el objetivo de disminuir la huella ambiental que es el objetivo de este trabajo.

Frente a esta segunda opción, hay que tener en cuenta que los problemas de ruteo de vehículo cuentan con una naturaleza NP-Hard, es decir, no existe un algoritmo en tiempo polinomial para dar respuesta en optimalidad al problema (Sörensen et al., 2018), por lo que requieren de la utilización de técnicas de solución aproximadas o heurísticas.

Las técnicas heurísticas basadas en búsqueda local han demostrado ser exitosas y eficientes en tiempo de ejecución para encontrar soluciones cercanas a las óptimas. Las heurísticas basadas en búsqueda local se basan en un conjunto pequeño, pero diverso en número, de operadores de búsqueda, los cuales deben ser bien elegidos a fin de encontrar mejores soluciones mediante cambios o perturbaciones entre los nodos (clientes) de las rutas. Estos cambios pueden darse dentro de rutas y entre rutas en el caso específico de los problemas de ruteo de vehículos. Lo ideal es que los operadores de búsqueda local generen vecindarios suficientemente grandes a fin de tener un mayor potencial en encontrar una mejor solución a la de partida (ruta o solución inicial) pero sin la necesidad de explorar todo el vecindario generado de forma exhaustiva (Arnold & Sörensen, 2019).

La complejidad de la heurística de búsqueda local dentro del vecindario dependerá de la cantidad de soluciones que requieren ser generadas y evaluadas, lo que contribuye a una mayor probabilidad de que se seleccionen aquellas de mejor solución. Sin embargo, el consumo de tiempo en la exploración y evaluación de las soluciones candidatas se incrementa en función del tamaño del vecindario generado, es decir, operadores de búsqueda local con pequeños vecindarios como el intercambio entre dos nodos o clientes en una ruta o 2-opt, pueden ser fáciles y rápidos en implementación, más no

necesariamente permiten encontrar mejores soluciones que cuando se consideran operadores de mayor complejidad (en términos del vecindario generado).

En el artículo “Eficiently Solving Very Large Scale Routing Problems” , se describe que algunas empresas en las que hay muchos clientes como es el caso del servicio de distribución de paquetes o los camiones de basura, y sobrepasan los 10.000 clientes, el tiempo necesario para encontrar una solución “óptima” al problema de ruteo de vehículos es muy alto para tomar decisiones , puesto que heurísticas como la “heurística genética híbrida” (HGSDAC) por Vidal et al (2013) o la “búsqueda local iterativa” (ILS) de Subramanian et al (2013), requieren más de 500 minutos para lograr una solución para 1.000 clientes (Arnold et al., 2019,p.2). En palabras de los autores: “al usar una meta-heurística genética híbrida o una búsqueda local iterativa, puede requerir varios meses para encontrar una solución en situaciones donde haya más de 10.000 clientes” (Arnold et al., 2019,p.2), por lo que resulta necesario proponer un algoritmo que considere un criterio de guía que acompañe a diferentes operadores para una mayor eficiencia (Arnold et al., 2019).

Con base en el anterior panorama, desde este trabajo se busca analizar, proponer y desarrollar un algoritmo para la solución al problema de ruteo de vehículos considerando el objetivo la minimización de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible que use atributos que permitan una guía para una mayor eficiencia en tiempo de recorrido de los diferentes vecindarios.

En el primer capítulo 1 de este documento se presenta la descripción y formulación matemática del problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos, así como la revisión de algunos de sus antecedentes y al final la descripción de técnicas de solución para este problema. En el capítulo 2 se detallan algunos operadores de búsqueda local y se explica cómo funciona un algoritmo de búsqueda local guiada. En el capítulo 3 se generan métricas e instancias de las soluciones al problema y se implementa la técnica de árbol de decisión (CART) o árbol de regresión y clasificación de aprendizaje de máquina, para la definición de las características de una solución que indiquen la bondad de esta, de tal manera que las características de las métricas permitan guiar el algoritmo de búsqueda local guiada en el problema de VRP. Por último, en el capítulo 4 se desarrolla el algoritmo de búsqueda local que incorpora operadores complementarios entre rutas (inter-rutas) y dentro de rutas (intra-rutas). Y finalmente se

define la secuencia de operadores que guían este algoritmo basándose en el análisis expuesto en el capítulo 3, es decir, usando los operadores que mejor desempeño tengan en la respuesta de la caracterización de las métricas en el árbol de decisiones.

Capítulo 1: Problema de minimización de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible en el ruteo de vehículos con flota heterogénea

En este capítulo se presenta el modelo de optimización para el problema de ruteo de vehículos con flota heterogéneos (Heterogeneous Pollution Routing Vehicle Routing Problem- HPRVRP) en la minimización de la huella de carbono. En la sección 1.1 se detallan los enfoques en la literatura para el ruteo de vehículos; en la sección 1.2 se describe el problema para los vehículos heterogéneos, en la sección 1.3 se revisa la formulación de la función para la estimación de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible, y al final para la sección 1.4 se desarrolla el problema con la integración de las restricciones por heterogeneidad y la polución generada por el consumo de combustible como función objetivo; finalmente se cierra con una descripción de las soluciones que se han usado para este problema.

1.1 Definición del Problema de VRP

El problema de ruteo de vehículo (VRP Vehicle Routing Problem) es un problema de optimización combinatoria ampliamente estudiado, que consiste en asignar tareas (como entrega, recolección o servicios a un cliente) a un conjunto de vehículos de manera que se minimice una función de costo (Toth & Vigo, 2002), como resultado se obtienen las rutas (conjunto de nodos) y la secuencia que deben seguir cada uno de los vehículos, esta solución puede ser representada por medio de grafos. Varios aspectos deben ser tenidos en cuenta al momento de dar solución al problema, como, el tipo de ruta que se puede escoger, la capacidad limitada de carga en cada vehículo de la flota a rutear, la variedad de los vehículos con los que se cuenta, la función objetivo a considerar, entre otros. En

este estudio se considera un problema de ruteo de vehículos con flota heterogénea ilimitada en la minimización de la huella de carbono.

El modelo matemático del VRP data del año 1959 y fue introducido por Dantzing y Ramser, quienes propusieron una formulación matemática para hacer entrega de gasolina a diferentes estaciones de servicio. Clarke y Wright en 1964 definieron el primer algoritmo efectivo para resolver este problema (Daza et al., 2010).

Los métodos de solución para grandes instancias del VRP han consistido principalmente en metaheurísticas¹. Estas técnicas se empezaron a desarrollar entre 1970 y 1990 (Irnich et al., 2014) debido a la naturaleza del problema de tipo NP-Hard, es decir, no existe un algoritmo en tiempo polinomial que pueda resolver. Además, se debe considerar la complejidad² adicional que proviene de las diferentes variantes o características del VRP. Dentro de las meta-heurísticas que han sido utilizadas para resolver los problemas de VRP se encuentran: Algoritmos Genéticos (Wang et al., 2016) como el de Holland (1975) (Sörensen et al., 2018), Sistema Inmune Artificial de Labrador, Packard y Perelson (1986), Búsqueda Tabú de Glover (1989) (Sörensen et al., 2018), Colonia de Hormigas (Li et al., 2009) de Dorigo (1992), Recocido Simulado (Leung et al., 2010) creado por Černý (1985), Algoritmos meméticos (Arnold & Sörensen, 2019b), Variable Neighbourhood Search (Popović et al., 2012), Adaptive Large Neighbourhood (Ropke & Pisinger, 2006) y otros que se han usado especialmente en la última década (Koç et al., 2016). También se ha propuesto el diseño de hiper-heurísticas para unir varias de estas meta-heurísticas (Gendreau & Potvin, 2010).

1.2 Formulación del modelo matemático para la solución del VRP-Heterogéneo

El VRP con flotas heterogéneas, representa un modelo matemático de optimización que involucra vehículos con diferentes capacidades para la distribución de productos. Gheysens y col. (1984) citado por Baldacci, Battara y Vigo (2008) formulan la FSMF (Fleet

¹ Al final de este capítulo se aclara que es.

² La complejidad computacional es una medida de rendimiento de un algoritmo que se refiere al tiempo que toma el algoritmo para ejecutarse. Se representa mediante la notación "O" que describe el tiempo en relación con el tamaño de la entrada "n" (complejidad en tiempo polinómica), por ejemplo un algoritmo O(n^2) tiene una complejidad polinómica de segundo grado.

Size and MiX-vrp) que es un modelo de VRP heterogéneo para una flota limitada de vehículos usando variables binarias (X) de tres *índices* que toman valor 1 si un vehículo de tipo k viaja directamente desde cliente i al cliente j , y 0 en caso contrario. Además, proponen variables de flujo (Y) que especifica la cantidad de bienes que transporta un vehículo cuando deja al cliente i para atender al cliente j . (Baldacci et al., 2008a).

El modelo FSMF se representa como un grafo dirigido, con la siguiente formulación: $G=(V, A)$, donde $V=\{0, \dots, n\}$ es el *conjunto* de vértices y A es el conjunto de arcos $A=\{(i, j): 0 \leq i, j \leq n \forall i \neq j\}$, con vértices o nodos ($i=1, \dots, n$) que representan los clientes que tienen una demanda positiva y conocida con notación q_i , con el vértice 0 que corresponde al depósito. Cada arco (i, j) que pertenece a A tiene asociado un coste $c(i, j)$ positivo que representa el coste de ir de un nodo i a un nodo j . En el depósito hay disponibles un grupo de m vehículos de k tipos, cada uno con una capacidad Q_k (Koç et al., 2016).

Parámetros:

C_{ij} Costo de ir del nodo i al nodo j , generalmente asociado con la distancia

q_i Demanda del nodo i al nodo j

Q_k Capacidad máxima del vehículo tipo k

m_k Número de vehículos tipo k

F_k Costos fijos asociados a sacar el vehículo de la bodega

Variables de decisión:

x_{ij}^k variable binaria que indica si se va de un nodo i a un nodo j en un vehículo k

y_{ij} variable de flujo que indica la cantidad de bienes que transporta un vehículo de un nodo i a un nodo j

$$\text{Min } \sum_{k \in M} F_k \sum_{j \in V'} X_{0j}^k + \sum_{k \in M} \sum_{i, j \in V} C_{ij} x_{ij}^k \quad (1.1)$$

$$\sum_{k \in M} \sum_{i \in V} x_{ij}^k = 1, \forall j \in V' \quad (1.2)$$

$$\sum_{i \in V} x_{ip}^k - \sum_{j \in V} x_{pj}^k = 0, \forall p \in V', \forall k \in M \quad (1.3)$$

$$\sum_{j \in V'} x_{0j}^k \leq m_k, \forall k \in M \quad (1.4)$$

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = q_j, \forall j \in V' \quad (1.5)$$

$$q_i x_{ij}^k \leq y_{ij} \leq (Q_k - q_i) x_{ij}^k \forall i, j \in V, i \neq j, \forall k \in M \quad (1.6)$$

$$y_{ij} \geq 0 \quad (1.7)$$

$$x_{ij}^k \in \{0,1\} \quad (1.8)$$

En esta formulación la restricción 1.2 y 1.3 aseguran que el visitante sea visitado solo 1 vez, al igual que ese vehículo solo llegue una vez y también debe salir de este cliente o nodo. El número máximo de vehículos disponibles por cada tipo está impuesto por la restricción 1.4. La Restricción 1.5 especifica que la diferencia entre la cantidad de mercancías que transporta un vehículo antes y después de visitar a un cliente es igual a la demanda de ese cliente y finalmente, la restricción 1.6 asegura que la capacidad del vehículo nunca sea excedida. La ecuación 1.7 define los números del flujo en el conjunto de los enteros positivos y la ecuación 1.8 los define como dicotómicos (Baldacci et al., 2008b). El modelo mantiene la conservación de flujo, las restricciones de capacidad y el uso de un único vehículo.

La función objetivo tiene los costos de escoger un vehículo k e ir de i a j $C_{ij} x_{ij}^k$ y los costos fijos F_K que representan el costo asociado a la salida de ese tipo de vehículo desde el depósito para cada nodo.

1.3 Estimación de las emisiones de CO2 equivalentes en el ruteo de vehículos.

Minimizar la huella de carbono se relaciona con el consumo de combustible que depende de diferentes variables como son, el peso del vehículo, la carga del vehículo, la velocidad, la aceleración, las pendientes en la que se mueven los carros, o el tráfico (Eshtehadi et al., 2017).

Cooper et al (1994) introdujeron el concepto de Logística Verde y acuñaron el término de Green-VRP para soluciones que buscan reducir la huella ambiental en el consumo de carbono. Más adelante, algunos investigadores como Kara et al., (2007), Kuo (2010), Figliozzi (2010), Bektaş y Laporte (2011), Xiao et al., (2012) consideraron la disminución del combustible, y luego en la evolución de esta temática se analizó la huella ambiental generada especialmente por las emisiones de CO2 por parte de Kwon et al. (2013) y Pradenas et al. (2013). También autores como Suzuki (2011), Franceschetti et al. (2013), Ćirović et al., (2014) y Tajik et al., (2014) analizaron la disminución de combustible y su huella de carbono (Nikfarjam et al., 2019) Suzuki incluye en su estudio la pendiente de la carretera, en un problema que considera ventanas de tiempo para el análisis de dichas pendientes, usan una velocidad promedio para terrenos planos y otra velocidad para terrenos inclinados formulando un modelo no lineal (Suzuki, 2011) en Latinoamérica, Muñoz-Villamizar en el año 2017, evaluaron la implementación de una flota de vehículos eléctricos relacionados con la distribución de bienes en un entorno urbano, con el objetivo de minimizar los impactos ambientales para una flota de vehículos, manteniendo un nivel de servicio y analizando el impacto ambiental causado por la carretera (Muñoz-Villamizar et al., 2020). Bektaş y Laporte describieron el modelo PRP (Pollution Routing Problem), que se centró en analizar cómo afecta la carga del vehículo, la velocidad, el ángulo entre los vértices y los costos del viaje en las emisiones del CO2 equivalentes al consumo de combustible (Kramer et al., 2015).

De forma complementaria Figliozzi creó el modelo EVRP (Emissions Vehicle Routing Problem) analizó la congestión vehicular y la velocidad entre los nodos, dando especial atención a cómo afectaba la aceleración en los arranques generados por las condiciones de tráfico (Figliozzi, 2010). La formulación de su problema se encuentra en la ecuación (1.9):

$$V_{ij}(b_i) = \sum_{l=0}^{l=p} (a_0 + a_1 s_{ij}^l + a_2 (s_{ij}^l)^3 + a_3 \frac{1}{(s_{ij}^l)^2}) d_{ij}^l \quad (1.9)$$

La fórmula permite calcular el volumen de las emisiones generadas de un nodo i a un nodo j partiendo de un tiempo b_i , donde s indica la velocidad en un trayecto de i a j , considerando

un intervalo de tiempo l , y d representa la distancia entre los nodos con tiempo l , con parámetros a conocidos que varían de acuerdo con el vehículo.

El FCVRP (Fuel Consumption Vehicle Routing Problem) propuesto por Bektas y Laporte en el 2011 (Feng et al., 2017) es una adaptación del modelo CVRP (Consumption Vehicle Routing Problem) de Xiao et al.(2012), quienes le dieron atención a la forma cómo variaron las emisiones cuando se alteraba el peso de los vehículos. Los autores encontraron que el consumo de combustible se minimizaba al conseguir una secuencia óptima de descarga de pesos por parte de cada vehículo, por tanto, más allá de limitarse por la ventana de tiempo, concluyeron que se debe considerar entregar primero a los clientes con mayor demanda (peso) para minimizar la huella ambiental, incluso por encima de aquellos que tienen menor distancia. Su modelo se expresa como:

$$p_{ij} = p_0 + \frac{p^* - p^0}{Q} y_{ij} = p_0 + \alpha y_{ij} \quad (1.10)$$

El consumo de combustible de la ruta p_{ij} (función 1.10) depende del peso del vehículo, donde p_0 representa el vehículo al vacío más la carga del vehículo y_{ij} . Q representa la capacidad del vehículo y p^* es el peso del vehículo con carga.

Por último los previamente mencionados (Bektaş & Laporte, 2011) describieron el modelo PRP (Pollution Routing Problem), y se centraron en analizar cómo afecta la carga del vehículo, la velocidad, el ángulo entre los vértices y los costos del viaje. En cuanto a los costos, estos se calculan expresados como el costo unitario de combustible multiplicado por el consumo de combustible en cada arco y es la función que se pretende minimizar en el modelo. En la propuesta de dicho modelo se tienen en cuenta también los costos de conductores, combustibles y emisiones. Los cálculos del modelo de emisión que proponen Bektas y Laporte (2011) vienen de un estudio de campo realizado por Barth et al (2009). Los autores proponen calcular el consumo de combustible F_{ij} como:

$$F_{ij} \approx \left(KNV_d + \left[\frac{P_{ij}}{\eta_t} + P_\alpha \right] / \eta \right) U \quad (1.11)$$

$$P_{ij} = v * M(\alpha + g * \sin \theta_{ij}) + 0.5C_d A \rho v^2 + (M * g * C_r \cos \theta_{ij}) \quad (1.12)$$

La función (1.11) representa el consumo de combustible asociado al vehículo, considerando el consumo KNV_d del motor, junto al consumo asociado a las otras partes del vehículo.

Aquí tenemos a k como el factor de fricción, N es la velocidad del motor, V es el desplazamiento, P_{ij} Potencia para ir de i a j en watts, t es la eficiencia de la transmisión, P , potencia consumida por los accesorios del vehículo, η una media de eficiencia del motor para cierto combustible, U es un valor que depende de algunas constante incluyendo N , y el parámetro P_{ij} que se construye de la fórmula siguiente con valores como M que es la masa desplazada donde se incluye la carga y el peso del vehículo, α es la aceleración, g la gravedad, A es el área frontal del vehículo, ρ es la densidad del aire, C_d coeficiente aerodinámico y ij representa el ángulo entre los vértices. Desde esta lógica, diferentes autores han trabajado en el análisis del consumo de combustible y en la minimización de los problemas ambientales de manera separada, tratando de ampliar los factores y restricciones que inciden en mayores emisiones (Posada & González, 2013).

1.4 Formulación del Modelo de PRP (Pollution Routing Problem) para vehículos heterogéneos

El modelo PRP (Pollution Routing Problem) que se describe a continuación es el usado por Eshtehadi, et al. (2017), que a su vez se deriva del modelo propuesto de Bektas y Laporte (2011) con la función de Barth & Boriboonsomsin (2009). Este modelo busca la minimización de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible e incluye los factores de carga, velocidad, pendiente y se infiere como un modelo completo de las características que inciden en la variación del combustible, como se ve en la tabla 1.1.(Eshtehadi et al., 2017)

Variable	Definición	Uds
F_{ij}	Consumo de combustible entre los vertices i y j	ml
P_{ij}	Potencia requerida por el vehículo para ir entre los vertices (i,j)	W
m_{ij}	Masa desplazada entre los vertices (i,j) incluyendo el peso del vehículo y la carga	kg
V_{ij}	Velocidad de desplazamiento para ir de un nodo (i,j)	m/s
g	Aceleración debido a la gravedad	m/s^2
θ	Angulo de inclinación entre los vertices (i,j)	Adimensional
C_d	Coefficiente de rozamiento	Adimensional
A	Área frontal	m^2
ξ	Tasa de equivalencia aire-combustible	rpm
K	Fricción del motor	RPM
V_d	Volumen de desplazamiento del motor	L
η	Eficiencia del motor	Adimensional
η_t	Eficiencia de la transmisión	Adimensional
κ	Poder calorífico bajo el combustible	J/g
P_a	Potencia consumida por los accesorios del vehículo	W
t_{ij}	Tiempo que tarde tarda un vehículo en desplazarse de i a j	s
N	la velocidad del motor	m/s
α	Aceleración de i a j	m/s^2

Tabla.1-1: Parámetros y unidades para la formulación de modelo PRP.

Fuente: Demir et al., (2017).

$$F_{ij} = \frac{\xi \left[\kappa N V_d + \frac{P}{\eta} \right]}{\kappa} \quad (1.13)$$

La función (1.13) estima el consumo de combustible, se parte de la ecuación (1.11), a excepción que la constante U cambia por ξ/κ . El parámetro P de la ecuación (1.13) será calculado como:

$$P = \frac{P_{ij}}{\eta_t} + P_a \quad (1.14)$$

Con la Potencia P_{ij} de la ecuación (1.14) requerida por el vehículo para ir de i a j calculada expresada como:

$$P_{ij} = v(M * \alpha + Mg \sin \theta_{ij}) + 0.5C_d A p v^2 + (M * g * C_r \cos \theta_{ij})v/1000 \quad (1.15)$$

Para un arco dado (i,j) de longitud de d y tiempo t_{ij} , entonces la velocidad del vehículo atravesando este arco la llamaremos v. Para simplificar el modelo se toma λ de la función (1.16) y γ de la función (1.17) como constantes.

$$\lambda = \frac{\xi}{k\Psi} \quad (1.16)$$

$$\gamma = \frac{1}{1000\eta_t f \eta} \quad (1.17)$$

Con Ψ que es el factor de conversión de gasolina de gramos/segundos a litros/segundo. Además, que M sea el portador de carga entre el nodo i y el nodo j. Más específicamente $M=w+f$ donde w es el peso en vacío del vehículo y f es la carga del vehículo. Entonces $\alpha = (\tau + g * \sin \theta) + g * C_r \cos \theta$. $\beta = 0.5C_d\rho A_f$ sea la constante de un vehículo específico, si se omiten las variables (i,j) se puede reescribir la ecuación como:

$$F(v) = \lambda(KNV_d + w\gamma\alpha v + \gamma\alpha f v + \beta\gamma v^3)d/v \quad (1.18)$$

Para la formulación que sigue tenemos que f_{ij} representa la cantidad total de flujo para cada arco (i,j) $\in A$. Z_{ij}^r indica si el arco (i,j) pertenece a A, que es recorrido a una velocidad r. Por tanto, la formulación más general para Heterogeneous Pollution Routing Vehicle Routing Problem (HVRPFD) es la siguiente:

$$\min \sum_{(i,j) \in A} KNV\lambda d_{ij} \sum_{r=1}^R \frac{Z_{ij}^r}{v} + \sum_{(i,j) \in A} w\gamma\lambda\alpha_{ij}x_{ij}^h d_{ij} + \sum_{(i,j) \in A} \gamma\lambda\alpha_{ij}f_{ij}d_{ij} + \sum_{(i,j) \in A} \beta\gamma\lambda d_{ij} \sum_{r=1}^R Z_{ij}^r v^2 \quad (1.19)$$

$$\sum_{i=1}^n x_{ij}^h = 1 \quad \forall i = 2, \dots, n \quad (1.20)$$

$$\sum_{j=1}^n x_{ji}^h = 1 \quad \forall i = 2, \dots, n \quad (1.21)$$

$$\sum_{j=1}^n x_{0j}^k \leq m_k \quad (1.21), \quad (1.22)$$

$$\sum_{j=1}^n x_{ji}^h = \sum_{i=1}^n x_{ji}^h \quad \forall i = 2, \dots, n \quad (1.23)$$

$$\sum_{j=1}^n f_{ji} - \sum_{j=1}^n f_{ij} = q_i \quad \forall i = 2, \dots, n \quad (1.24)$$

$$\sum_{j=1}^n f_{ji} - \sum_{j=1}^n f_{ij} = q_i \quad \forall i = 2, \dots, n \quad (1.25)$$

$$q_i x_{ij}^k \leq f_{ij} \leq (Q_k - q_i)x_{ij}^k \quad \forall i, j = 1, \dots, n \in A \quad (1.26)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i, j = 1, \dots, n \quad (1.27)$$

$$f_{ij} \geq 0 \quad (1.28)$$

Modelo propuesto por el autor de acuerdo con lo planeado por Demir, Bektas y Laporte (2012)

La ecuación 1.18 representa la función de consumo de combustible previamente expuesta. Las restricciones de este modelo de la 1.20 a la 1.28 son las mismas restricciones para la formulación de una flota limitada de vehículos heterogéneos propuesto por Gheysens y col. (1984).

Para la evaluación de este problema de optimización se usa el método de programación entera y mixta, el cual da respuestas enteras positivas en una función objetivo, pudiendo tener el resto de las variables de tipo continuo.

1.5 Técnicas de Solución para el VRP y el HFVRP

En la literatura se identifican tres tipos enfoques o aproximaciones para la solución de problemas de optimización combinatoria como es el caso de los problemas VRP, estos tipos de solución son: modelos exactos, técnicas heurísticas y metaheurísticas (Molina, 2012).

1.5.1 Modelos exactos

Los modelos exactos son aquellos que usan técnicas de programación matemática y búsqueda en espacio completo para una solución óptima, sin embargo, son eficientes en tiempo solo para problemas pequeños, no más de 50 nodos y se pueden clasificar en tres categorías (Molina, 2012):

1. **Programación dinámica:** Es un modelo que divide el problema en sub-problemas más pequeños y manejable y se guardan las soluciones a estos sub-problemas para evitar tener que volver a resolverlos cada vez que se necesiten. Esta actividad se le llama "memorización". Un algoritmo que aplica esta solución es "Clarke y Wright". El algoritmo combina técnicas de programación dinámica y heurísticas para encontrar una solución sub-óptima para el VRP construyendo una matriz de ahorro que evalúa los clientes de manera progresiva. Se recomienda ver el algoritmo en Taha (2012) (TAHA, 2012)

2. **Programación entera mixta** : Que combina técnicas de programación lineal con programación entera y algoritmos de ramificación (Wolsey, 2008)

3. **Búsqueda directa de árbol**: El cual busca la solución, para cada uno de los nodos de un árbol. Este algoritmo divide el campo de solución en nodos y se va orientando entre las ramas de los nodos de mejor solución. Un nodo se elimina con su rama si no representa una solución factible, si existe una solución factible este nodo se vuelve una cota inferior. El algoritmo finaliza cuando cada uno de los nodos hayan sido evaluado y la solución será la de mayor cota inferior (Medina et al., 2011)
Uno de los métodos exactos conocidos para resolver este tipo de problemas de VRP es el algoritmo de ramificación y acotamiento o Branch and Bound, el cual divide el problema en subproblemas iniciando por el problema relajado (sin restricciones de variables enteras), escogiendo el mayor o igual al valor entero próximo, o el menor o igual al valor entero próximo en la solución actual. De esta manera se va ramificando cada vez hasta encontrar la mejor solución en medio de esas ramificaciones (Hillier & Lieberman, 2010). Westerlund, Göthe-Lundgren y Larsson (2003) ofrecen una formulación ampliada y nuevos cortes válidos. Yaman (2006) propuso nuevos cortes válidos respecto a los anteriores. Choi y Tcha (2007) adherieron límites inferiores utilizando la generación de columnas. Baldacci, Battarra, et al. (2008) propusieron desigualdades válidas, y una formulación de programación entera mixta (MIP)(Pessoa et al., 2018).

1.5.2 Técnicas heurísticas:

Las técnicas heurísticas son métodos de resolución de problemas que se basan en la experiencia y conocimiento previo, en lugar de algoritmos precisos o una búsqueda exhaustiva de todas las soluciones (Medina et al., 2011). Son técnicas rápidas y con respuestas aceptables, aunque sean distantes al óptimo. Algunos de estos tipos de técnicas son:

2. **Heurística de construcción**: Es una técnica en la que se construye una solución paso a paso, es decir, usando una estrategia incremental. Estas técnicas se basan en construir una solución parcial, evaluarla e ir añadiendo elementos o cambios para ser mejoradas (Medina et al., 2011).

3. **Heurística de dos fases:** La cuál es una variante de la de construcción, como su nombre indica esta divide el problema en dos fases: La primera fase: Se genera una solución parcial por un agrupamiento que puede violar las restricciones del modelo, esta solución parcial luego es evaluada y si cumple con las restricciones y el objetivo deseado se continua con ella, de lo contrario se busca una nueva solución. La segunda fase es una fase de mejora, donde se realizan cambios para mejorar la solución actual y luego se evalúa la solución mejorada, para aceptarla como solución final o continúa mejorando (Medina et al., 2011).

4. **Heurísticas de mejora iterativa:** Estas técnicas son algoritmos que realizan cambios a una solución inicial hasta alcanzar una solución que cumpla con un criterio de parada. Se definen en un espacio de soluciones candidatas o vecindarios dentro de una estructura de espacio de búsqueda. Los métodos que hacen parte de la heurística de mejora iterativa representan los algoritmos de búsqueda local los cuales se enfocan en encontrar la mejor solución en el espacio de soluciones de una vecindad definida. (Esta heurística se describirá a mayor profundidad en el siguiente capítulo)

1.5.3 Técnicas Metaheurísticas:

El término meta-heurística fue usado por Glover en 1996 y conceptualizado con mayor precisión por Sorensen, Sevaux y el mismo Glover (2013), como: "... es un marco algorítmico independiente de problemas de alto nivel que proporciona un conjunto de pautas o estrategias para desarrollar algoritmos de optimización heurística" (Daza et al., 2010). Es decir, se puede ver como un espacio donde se aplica un algoritmo para solucionar problemas complejos por medio de actividades que usan heurísticas que entregan soluciones aceptables y lo más cercanas al óptimo absoluto.

También es posible conceptualizarlo desde su nombre teniendo en cuenta que el prefijo griego "meta" quiere decir más allá de la heurística, por tanto, busca ser más precisa que una heurística convencional, es decir una solución metaheurística trata de identificar y combinar las heurísticas que mejor se aproximen a la solución óptima, explorando el

espacio de soluciones. Se usan generalmente cuando no existe una heurística específica (No hay un procedimiento definido para solucionar un problema) Según Blum (2003) las Meta-heurísticas deben cumplir con las siguientes propiedades “:1) Ser estrategias que “guían” el proceso de búsqueda, 2) Su meta es explorar de manera eficiente el espacio de búsqueda para encontrar un óptimo global, 3) Sus técnicas varían desde métodos de búsqueda local simple a complejos métodos de aprendizaje,4) Pueden usar mecanismos para no estancarse en óptimos locales, 5) Permiten un nivel de descripción abstracto y no específico del problema, 6) Las meta-heurísticas avanzadas usan experiencia para guiar la búsqueda, y 7) Pueden hacer uso de un conocimiento específico para controlar sus estrategias desde un nivel superior”(Blum, 2003)

En este capítulo se presenta la formulación matemática del problema de minimización de las emisiones equivalentes de CO₂ a través de la minimización del consumo de combustible en el ruteo de vehículos con flota heterogénea en el transporte de mercancías, para realizar dicha formulación se usó el modelo de Demir, Bektas y Laporte (2017), que usa las funciones de Barth y Boriboonsomsin (2009) (Eshtehadi et al., 2017) para el PRP, con modificaciones para que considere la restricción de una flota heterogénea limitada del modelo de Gheysens y col. (1984)

La formulación matemática de este problema entrega tres soluciones que se diferencian en su calidad respecto al óptimo global usando el optimizador de Gurobi, y estas soluciones junto con una serie de métricas asociadas a cada solución servirán como suministro para poder obtener las reglas que sirvan en la construcción del algoritmo de búsqueda local guiada que se usa como metaheurística para la obtención de una mejor solución.

Capítulo 2: Heurísticas basadas en búsqueda local y la metaheurística de búsqueda local guiada

Un algoritmo de búsqueda local está compuesto por una combinación de operadores de búsqueda local que perturban una solución inicial, la selección y orden de aplicación impactan el desempeño del algoritmo, por lo que se necesita escoger adecuadamente los operadores que van a ser usados y su orden de implementación en la metaheurística de búsqueda local guiada.

En este capítulo se presenta una descripción de la búsqueda local y la metaheurística de búsqueda local guiada, sección (2.1), se describen algunos operadores de búsqueda local que realizan movimientos de intercambio de nodos entre la misma ruta (intra-rutas) o intercambian nodos entre diferentes rutas inter-rutas, sección (2.2), se explican en qué consisten sus movimientos y se describe el algoritmo de búsqueda local guiada de Voudouris et al., (2003), que se usará como metaheurística para guiar a los operadores, sección (2.3).

2.1 Algoritmos de Búsqueda Local

El concepto búsqueda local parte de la premisa de encontrar soluciones mejoradas a partir de pequeñas modificaciones de una solución inicial de búsqueda (Arnold & Gendreau, 2017), es decir, se pretende minimizar una función $f(x)$, por medio de un desplazamiento x' que al evaluar la función $f(x')$ se cumpla que: $f(x') < f(x)$ (en el caso de minimización), indica si el desplazamiento de x' ha mejorado respecto a aquella función evaluada en x .

La búsqueda local puede ser vista como un enfoque heurístico para resolver problemas combinatorios como son los tipos VRP, realizando movimientos que puedan mejorar una

solución basados en una función de costo. Sin embargo, la búsqueda local tiene la desventaja de poder estancarse en óptimos locales, es decir encontrar una solución óptima para un vecindario.

Las investigaciones de Hansen & Mladenovic (1995-1997) fueron las primeras en las que se propuso una solución para salir de los óptimos locales y poder realizar una búsqueda global, para eso trabajaron en escoger un conjunto de estructuras de entorno o vecindario que, con cambios sistemáticos e iterativos, por medio de movimientos o perturbaciones, pueda buscar un óptimo global para todos los entornos o vecindarios previamente definidos a esta metaheurística (Hansen et al., 2003).

En palabras de Arnold y Sorensen (2019), todas las metaheurísticas que desean solucionar un VRP tienen en común la búsqueda local, que se ha usado como una forma de ajustar una heurística sobre la marcha (Arnold et al., 2019). Las metaheurísticas dedicadas a los problemas de VRP giran en torno a una búsqueda local efectiva, generalmente en un marco de múltiples vecindarios, o vecindarios variables (Arnold & Gendreau, 2017), realizando movimientos operacionales llamados operadores sobre una solución para dichos vecindarios, en un proceso de búsqueda global. En el caso de los *Algoritmos Genéticos* estos operadores se pueden ver en el proceso de selección, cruce y mutación, pues en estas actividades realizan una operación que cambia los estados de la solución del problema. Por ejemplo, en el *operador de cruce* se intercambian dos valores con el fin de que este intercambio mejore la función objetivo. Por otro lado, el operador Cross-over trata de realizar intercambios de segmentos y mutación es un operador que permite no estancarse en óptimo local.

Un operador de búsqueda local sencillo de entender para la solución de un VRP es λ -opt, el cual trata de decidir el nuevo orden o secuencia para una ruta por medio de un criterio de costo, que está mayoritariamente asociado a la distancia. La idea radica en que cada iteración se intercambia λ nodos entre sí de tal manera en que se mejore la solución. Así, dentro de estos algoritmos λ -opt, el 2-opt es aquel que intercambia 2 arcos de una ruta con otros dos arcos adyacentes de la misma u otra ruta entre sí, y calcula el nuevo costo de viaje haciendo este proceso para una ruta.

La manera en que se puede usar cada uno de los operadores varía entre los algoritmos metaheurísticos.

2.2 Operadores de búsqueda local en problemas VRP.

Muchos son los operadores de búsqueda local para problemas de VRP que han sido definidos (Vidal et al., 2014), sin embargo, algunos resaltan por su simplicidad, como aquellos que menciona Kenneth Sörensen en el artículo “Efficient Heuristic for Routing and Integrated logistics” (Gendreau & Sörensen, 2017), entre ellos están:

- Shift (1,0): Un cliente k es transferido de una ruta a otra
- Swap (1,1): Intercambia dos nodos
- 2-Opt: Intercambia dos arcos
- Cross-over: Cambia sub-rutas de las rutas
- Insert/Relocate: Reubica un cliente
- 3-Opt: Intercambia tres arcos
- Cross Exchange: Cambia dos secuencias de rutas

Los primeros 5 operadores cuenta con complejidad $O(n)^2$, el siguiente con complejidad $O(n)^3$ y el último con complejidad $O(n)^4$. Los dos primeros operadores son Intra rutas (intercambio de nodos para una ruta) y los siguiente entre rutas (intercambian nodos entre 2 o más rutas). Siendo los operadores intra-rutas aquellos que se aplican en una ruta, modificando la secuencia de nodos para mejorar o refinar la solución específica, este proceso se conoce comúnmente como explotación de vecindarios, mientras que los operadores inter-rutas actúan entre dos o más rutas de diferentes vehículos intercambiando nodos entre las rutas para obtener una mayor cantidad de vecindarios y ampliar la estructura global del conjunto de soluciones. Este proceso es comúnmente conocido como exploración de vecindarios, ya que como su nombre indica explora nuevos vecindarios.

2.2.1 Operadores Intra-rutas

Como previamente se mencionó los operadores de búsqueda local intra-rutas son aquellos que tratan de obtener la mejor solución para una ruta específica, sus movimientos consisten en mejorar el orden de la ruta de tal manera en que se pueda minimizar la función objetivo para la solución en esa ruta.

2-Opt: Dos arcos son removidos y otros dos arcos se añaden de tal manera que se crea una nueva ruta, es decir, cambia la posición de los nodos dentro de una ruta. La figura 2-1 se muestra cómo se hace este movimiento (perturbación).

Análogamente a los operadores k-opt o λ -opt que como ya se mencionó en la introducción, eliminan k arcos y los vuelven a unir en diferente orden, conservando la premisa de optimizar la función objetivo, así, además de 2-opt existe el 1-Opt, 3-Opt o el 4-Opt.

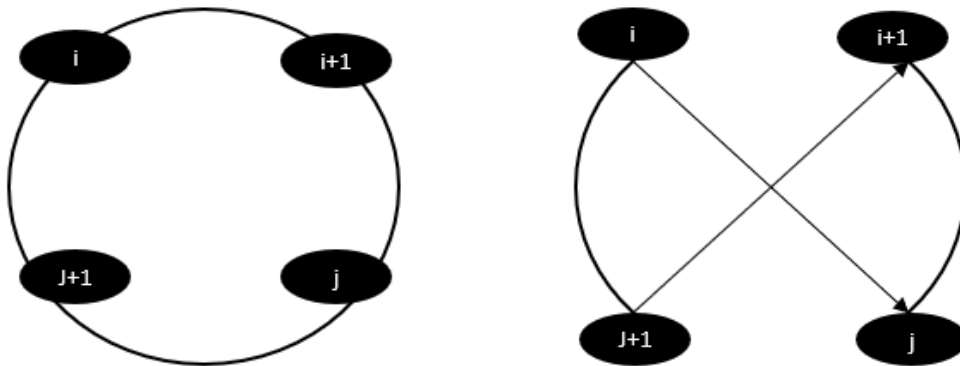


Figura 2-1 Movimiento 2-Opt.

Lin-Kernighan (LK): Este operador modifica una solución mediante la realización de un conjunto de movimientos del tipo λ -opt, generalmente con movimientos 2-opt, o 3-opt, hasta conseguir soluciones λ -óptimas. El proceso consiste en realizar movimientos λ -opt y evaluarlos, si el movimiento realizado mejora la solución, ésta pasa a ser la solución inicial y se inicia un nuevo proceso de búsqueda; en caso contrario, se rechaza esa solución y se procura no repetir la secuencia de movimientos. Se realizan estas iteraciones hasta que no exista un intercambio que mejore la solución entregando el orden que optimiza la ruta evaluada (Bautista-Valhondo, 2003).

Para realizar movimientos eficientes en tiempo dando como resultado soluciones factibles, comienza con el arco más largo en la ruta considerada, los arcos se eliminan y agregan de forma iterativa, de manera que los arcos eliminados y agregados comparten un punto final; en este caso si cumple el criterio de ganancia parcial y da como resultado un recorrido factible, el recorrido se cerrará. La heurística se detiene, si para cada nodo inicialmente

eliminado no se puede encontrar ningún movimiento de mejora. Se propone ver Helsgaun (2000) (Arnold & Sörensen, 2019a)

2.2.2 Operadores Inter-rutas

Los operadores inter-rutas se pueden resumir como aquellos que buscan mejorar una solución de un problema VRP perturbando (modificando) varias rutas de forma simultánea.

Relocate Operators: Reubica un nodo de una ruta en otra o en la misma, pero en una posición diferente (Arnold & Gendreau, 2017). Formalmente se define como un sub-cadena r_i de una ruta r_i , la cual es insertada en otra sub-cadena r_j . Donde la sub-cadena solo contiene un elemento $r_i=1$. En la figura 2-2 se ve cómo un nodo es entregado de una ruta r_i a otra r_j . (Huemer, 2011)

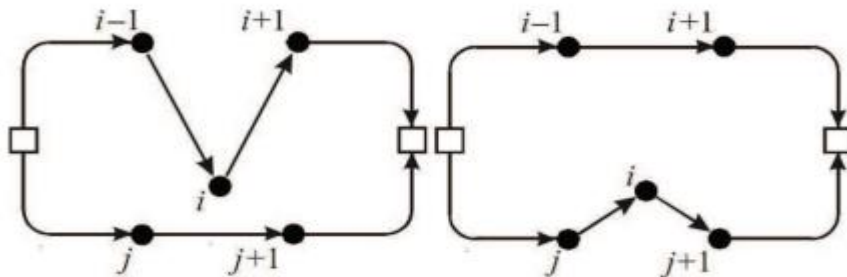


Figura 2-2 Movimiento Relocate. Fuente: Huemer (2011)

Swap Operators: Un intercambio de nodos donde un nodo se pueda ubicar en la posición de otro. Si tenemos que: $r_i=r_j=1$, este proceso es un doble relocate. En el caso de que $r_i \geq 1$ Se llama el operador Or-exchange.

Crossover: Resulta una generalización del swap, donde un subconjunto de nodos de una ruta es intercambiado con otro subconjunto de nodos. Un intercambio de operadores tal que sub-cadena $r_i=r_j$. Ambas sub-cadenas intercambiadas están conectadas al depósito, por lo cual su movimiento posiblemente es invertir las cadenas. Por tanto, su movimiento consiste en intercambiar sub-cadenas de manera de iterativas entre distintas rutas, comienza con una sub-cadena de dos nodos entre rutas, luego una sub-cadena de 3 y así hasta que logre intercambiar la mayor cantidad de sub-cadenas posibles.

Cross- Exchange/Exchange: Cruza dos sub-cadenas de dos diferentes rutas con cualquier tamaño. Este algoritmo se construye en dos partes: 1. La identificación del inicio de dos sub-cadenas y 2. La determinación de una longitud adecuada en ambas sub-cadenas. Para realizar el primer punto, se parte por definir qué nodo se eliminará y hallar porqué vecino potencial de otra ruta se intercambiará (para que sea potencial debe ser cercano) (Arnold & Sörensen, 2019a).

El ejemplo de la figura 2-3, se hace respecto a dos rutas en los intercambios de sus respectivas cadenas. Sean los arcos de los nodos $(i-1, i)$ y $(k, k+1)$ removidos de una ruta uno y dos arcos $(j-1, j)$ y $(l, l+1)$ removidos de una segunda ruta, de esta manera los segmentos (i, k) y (j, l) son simultáneamente reinsertados de la forma: $(i-1, j)$, $(l, k+1)$, $(j-1, i)$ y $(k, l+1)$ (Huemer, 2011)

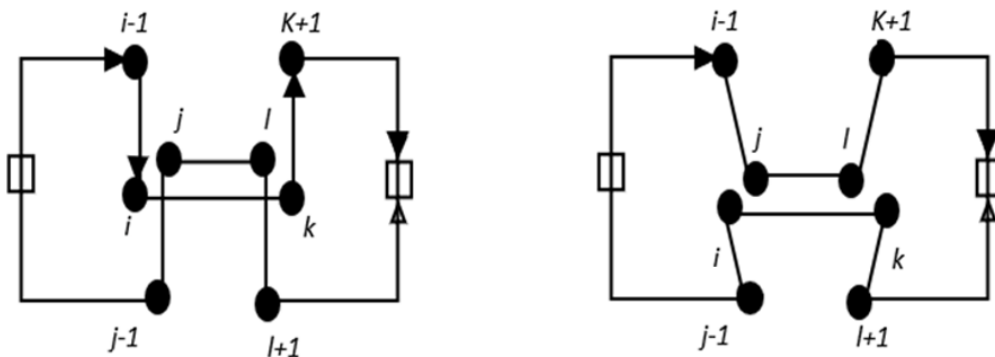


Figura 2-3 Movimiento Cross-Exchange. Fuente: Huemer (2011).

Para verificar la factibilidad de la capacidad, debe asegurarse en las dos rutas seleccionadas que:

$$\text{TotalSale} + \text{Totalentra} \leq 0$$

TotalSale es la suma de las demandas de todos los nodos salientes de la ruta y el Totalentra es la suma de los nodos que entran a la ruta. Este candidato se moverá si se cumple que los costos de los dos movimientos en las dos rutas cumplen con un costo

menor a 0, $c_1+c_2 \leq 0$. Estos movimientos se siguen haciendo hasta alcanzar el depósito o que no se cumpla la restricción (Olivera, 2004).

Otro caso se ve, en (Arnold & Sörensen, 2019a) Figura2-4 en el que se quitá el arco (I_k, I_{k+1}) de la ruta r_i .

Teniendo en cuenta los vecinos potenciales de la ruta r_j se debe analizar (I_k, J_l) como una opción y de acuerdo a lo anterior se debe remover uno de los arcos que se une a J_l , ya se (J_l, J_{l-1}) o (J_l, J_{l+1}) , puesto que sólo podemos considerar dos nodos adyacentes.

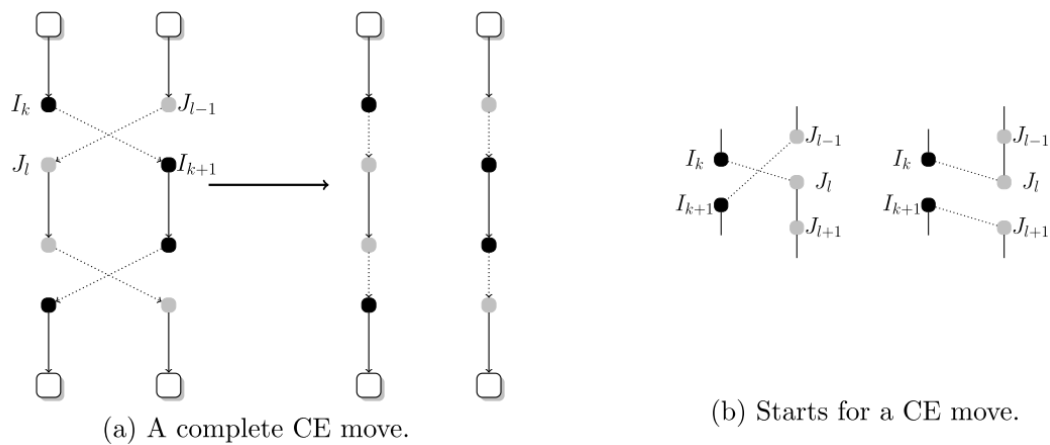


Figura 2-4: Movimiento Cross-Exchange Fuente: Arnold & Sörensen, 2019b.

Si se cambia el nodo (I_k, I_{k+1}) por (I_k, J_l) y en su orden (J_l, J_{l-1}) por (J_{l-1}, I_{k+1}) entonces, la determinación del costo será: $c_1 = c(I_k, J_l) + c(J_{l-1}, I_{k+1}) - c(I_k, I_{k+1}) - c(J_l, J_{l-1}) \leq 0$ En el caso de que se cumpla esto entonces se realizan los respectivos cruces entre rutas y nodos consecutivos, como se ve en la figura2-5.

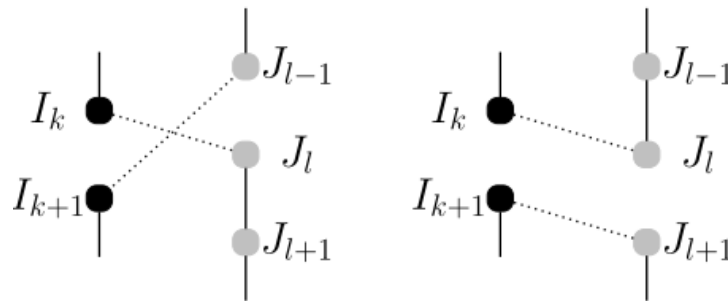


Figura 2-5: Movimiento Cross-Exchange 3.
 Nombre de la fuente: Arnold & Sörensen, 2019^a.

Si escogemos el segundo nodo como la sub-cadena $r_i = \{I_{k+1}\}$ el nodo que vamos a insertar en la sub-cadena $r_j = \{J_l\}$ la sub-cadena que se va insertar en r_i , y de manera análoga a los cambios realizados en $r_i = \{I_k\}$ and $r_j = \{J_{l-1}\}$, entonces el costo para el segundo cruce será:

$$c_2 = c(I_{k+1}, J_{l+1}) + c(J_l, I_{k+2}) - c(I_{k+1}, J_{l-1}) - c(J_{l+1}, J_l) \leq 0$$

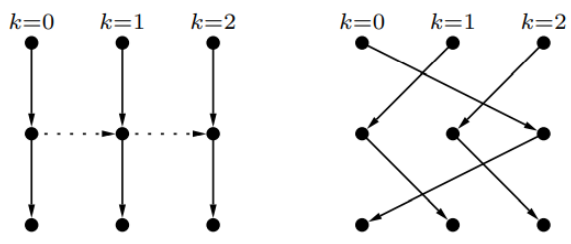
Y este candidato se moverá (swap) si se cumple que $c_1 + c_2 \leq 0$. Estos movimientos se repiten para la sub-cadena teniendo fijos aquellos movimientos que fueron posibles hasta alcanzar el depósito o que no se cumpla la restricción. Es decir, su movimiento consiste en el intercambio de arcos o pares de nodos entre cadenas. Para mayor claridad se deja el siguiente pseudocódigo (tabla 2-2):

Algoritmo Cross-Exchange
1. Determinar el nodos más de e que esta en una ruta diferente a a. 2. Para cada inicio restante, evalúe y amplie iterativamente las subcadenas. Si el intercambio da como resultado una ruta factible almacene el movimineto como movimiento candidato 3. Ejecutar el movimiento que mejor optimiza la función objetivo

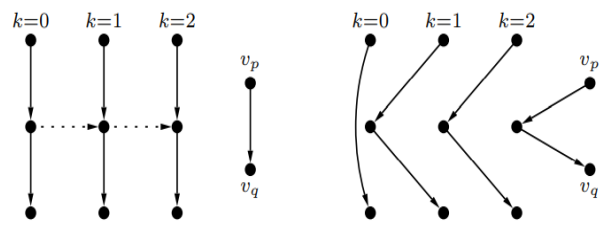
Tabla 2-2: Algoritmo Cross-Exchange.

Ejection-Chain de Rego y Roucarior: También nombradas cadenas de expulsión, su movimiento consiste en “seleccionar algunos elementos que cambiarán su estado o valor y como resultado otros elementos serán expulsados de su estado actual”, para el ruteo de vehículos se cambian las posiciones de algunos nodos en las rutas (Olivera, 2004).

Dado una cadena de expulsión del niveles de una ruta v_i con triplas de rutas, es decir, nodos secuenciales de v_i de la forma $(v_{i-1k}, v_{ik}, v_{i+1k})$ con $k=0, \dots, l$, en la que cada v_{i+1k} reemplaza a v_{ik} luego de una movida. Los nodos v_{ik} se llaman nodos centrales. “Una movida se interpreta como que el nodo central de un nivel expulsa al nodo central del nivel siguiente de su ubicación; este último debe ubicarse en otro lugar y expulsará al siguiente nodo central, ocurriendo una cadena de expulsiones. El último nodo v_{il} está obligado a cerrar la ruta por lo que se presentan dos opciones”. Movidas tipo 1 ubicada entre v_{i-10} y v_{i+10} . En la movida tipo 2 se crea un arco (v_{i-10}, v_{i+10}) y v_{il} se coloca entre dos nodos cualesquiera v_p y v_q que no han sido nodos centrales en ninguna de las triplas (Olivera, 2004). Sin embargo, no todas las triplas son posibles, para eso deben cumplirse las restricciones de legitimidad. Una mayor claridad del algoritmo lo da la figura 2-6.



1. Movida Tipo 1 para tres niveles k



2. Movida Tipo 2 para tres niveles k

Figura 2-6: Movimiento Ejection Chain. Fuente: Olivera (2004).

Relocation-Chain: Los operadores pueden ser complementarios en un operador que afecte más de dos rutas de manera simultánea. Este operador se construye a través de un vecindario embebido, que se compone de muchos movimientos simples dentro de un movimiento compuesto de relocalización entre rutas. Este movimiento es propuesto por Arnold & Sörensen (2019b) y su propuesta es muy parecida a ejection-chain, pero se enfoca en considerar movimientos inter-rutas con tal que los movimientos dentro de rutas sean desarrollados por Lin-Kerningham.

Un RC comienza con la relocalización de un nodo de cliente de ruta r_i a r_j , y esta relocalización traslada el nodo r_j a un nodo r_k : esta dinámica sigue hasta que un límite de relocalización es alcanzado, alcanzando una complejidad que crece de manera exponencial ya que un nodo puede ser relocalizado en todos los nodos de la siguiente ruta y así sucesivamente, por lo cual se requiere que su complejidad sea reducida con búsqueda secuencial, pre-procesamiento y poda.

El costo de relocalizar l_k al lado de J_l será la diferencia entre el costo mínimo de insertar l_k al lado J_l y el desvío por l_k al entrar en la ruta actual. Para mayor claridad se puede ver la figura 2-7.

$$c_l = \min_{J^*} |J_l - 1, J_l + 1| c_{l, J^*} - c_{l, J^*}$$

Por otra parte, el costo de la desviación es:

$$c_D = c(l_{k-1}, l_{k+1}) + c(l_k, l_{k+1}) - c(l_k, l_{k-1})$$

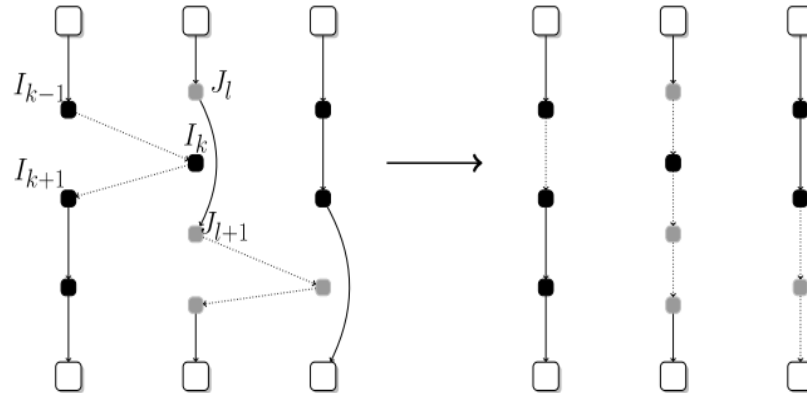


Figura 2-7: Movimiento Relocation-Chain. Fuente: Arnold & Sörensen, 2019^a.

Si no se incrementan los costos de la solución $cl+cD \leq 0$, entonces veremos esto como el comienzo de RC. Así, secuencialmente relocalizan nodos de tal manera que el agregado de los costos esté por debajo de 0 y todas, excepto la ruta de destino de la última reubicación, son factibles. Para disminuir los costos computacionales no se permite la inserción en la posición de un nodo de cliente previamente expulsado en el mismo RC, solo se consideran reinsertiones cercanas, una relocalización para ruta de destino. La idea consiste en reubicar nodos cercanos entre diferentes rutas (más de dos).

2.3 Búsqueda Local Guiada

La búsqueda local guiada es una metaheurística con la cual se busca mejorar la eficiencia en lo referente a tiempos de ejecución, guiando el tipo y orden de los operadores que se utilizan para llegar cerca de un óptimo global introduciendo una forma inteligente de evaluar los vecindarios.

En ese sentido, el interés de hacer uso de esta metaheurística reside en proponer un orden para cada uno de los operadores de acuerdo con la penalización en la función objetivo para los arcos “malos”.

La Búsqueda local guiada fue propuesta por Voudouris (1995)(Voudouris et al., 2003) y surge como una extensión de GENET, una red neuronal para la satisfacción de restricciones; adaptada para que trate de hallar óptimos de acuerdo a una función dada. Según Arnold & Gendreau, (2017), la introducción formal del algoritmo se debe a Applegate

et al. (2003), quienes proponen la guía de este algoritmo comenzando con el arco más largo (distancia entre nodos o puntos). Los arcos son removidos y añadidos iterativamente, validando si se cumple el criterio parcial de ganancia al modificar los arcos.

Luego, Voudris y Tang (2003) aplican este concepto efectivamente en un TSP alternando movimientos de operadores de diferente complejidad (operadores para explorar y explotar) para obtener soluciones más precisas a medida que se pasa de un algoritmo más complejo a uno menos, usando así: Algoritmos genéticos, Búsqueda tabú granular e intercambios 2-opt (Fisher, 1995). Algunas mejoras para el VRP las realizan Mester y Braysy (2007), guiando los algoritmos con una combinación de los siguientes métodos: Relocate, swap, 2-opt y Or-exchange citado por Arnold y Sorensen (2019), con lo cual se logran obtener óptimos eficientes en el vecindario. Por otra parte, consideran, unos movimientos de operadores entre rutas que trata de cambiar la asignación de clientes (nodos), involucrando varias rutas entre sí, y para minimizar los resultados computacionales tratan de limitar el vecindario, realizando pruning o poda, al dejar solo opciones prometedoras en términos de distancia.

Arnold y Sorensen en el año 2017, proponen una heurística mejorada que no solo elimina arcos largos, como lo hacen Mester y Braysy (2007), sino también arcos anchos, entendiendo que así se pueden obtener mejores resultados para optimizar la forma en que guían la heurística hacia soluciones óptimas de manera más rápida. Es decir, cada vez que se encuentra un arco ancho se penaliza aumentando su valor de costo y se utiliza luego un operador entre rutas como “Ejection Chain (EC)” (Usado para más de dos rutas) o “Cross Exchange (CE)” (Usado para dos rutas); se finaliza con algoritmo tipo “Linkinghan (LK)” que lo definen como la heurística top para el TSP. Estos algoritmos de búsqueda local se complementan usando la idea de “embeded neighborhoods” con el algoritmo de “Relocation Chain (RC)” que son Operadores entre rutas más amplios, de tal manera que el algoritmo LK solo se encargaría de los movimientos dentro de las rutas o intra-routes, mientras los otros se enfocan en los movimientos entre ruta o inter rutas (en inglés *inter-routes*). Este método logró encontrar soluciones de alta calidad con una brecha de 0.70% o menos frente al óptimo global para instancias de hasta 1,000 clientes (lo cual ha dado respuestas efectivas en el 90% de las veces y varía 0,1% respecto al óptimo global el 10% de las veces), obteniendo 40 veces más rapidez que otras heurísticas de

vanguardia, incluso que la de Vidal et al (2013) quienes previamente habían guiado los algoritmos: Relocate, 2-opt, Or-exchange y Cross Exchange.

A pesar de que la Búsqueda local guiada es un concepto reciente como se menciona, se relaciona con la manera en que cada metaheurística guía el proceso de explorar y explotar los vecindarios, y el concepto de Pruning o Poda, el cual antecede a la idea de Guía Inteligente, se ha elaborado para el criterio de “sequential search” propio de LK o CE que usualmente considera solo cierto tipo de longitud en la sub-ruta. En mayor medida fue mejorado esta lógica para el algoritmo de “Granular Tabu Search” como indican Toth y Vigo (Toth & Vigo, 2002), es decir la Búsqueda Local Guiada, es solo una parte de la evolución de los algoritmos de Búsqueda por mejorar los ejercicios combinatorios en cuanto ya se había empleado anteriormente criterios de búsqueda guiada por medio de la extracción y explotación de características como en el caso de “sequential search” el cuál define un criterio de longitud para realizar un intercambio entre rutas para operadores complejos limitando el número de posibles intercambios . La descripción de su proceso se menciona a continuación:

2.4 Descripción del proceso de Búsqueda Local Guiada

Cada operador previamente visto en algún punto se estancará en un óptimo local, para una función $g(s)$ en la que se encontrará una solución óptima sólo para dicho vecindario. Las metaheurísticas crean entonces una perturbación que permite salirse de ese óptimo local. Como ya se ha mencionado la idea de GLS consiste en cambiar la evaluación de costo (s^*) y penalizar las características “malas”

Se tiene una función objetivo g evaluada en s (distancia o en este caso consumo entre i y j) que se modifica con λ que es un parámetro, i varia sobre todos los atributos, p_{ij} es la penalización para el i -ésimo atributo (Voudoriorous,2010).

$$h(s) = g(s) + \lambda \times \sum (p_i \times I_i(s)) \quad (2.1)$$

$s = \text{solución candidata}$ (arco entre i y j)

$\lambda =$ Parametro de GLS que controla penalización

$i =$ los atributos a evaluar

$p_i =$ numero de penalizaciones del atributo (ruta i)

$$I_i = \begin{cases} 1 & \text{Si } s \text{ tiene el } i\text{esimo atributo} \\ 0 & \text{de lo contrario} \end{cases}$$

Una vez penalizado se trata de cambiar el arco por uno nuevo usando búsqueda local. Debido a que la búsqueda local solo se concentra en cierta área, la penalización del arco puede que sirva de guía en la ejecución del algoritmo.

```

    k ← 0

    s0 ← genera una solución inicial

    Para i=1 hasta M hacer

        pij=0

        h(s) = g(s) + λ × ∑ (pi × Iij(s))

    Mientras criterio de parada hacer

        sk+1 = búsqueda local (sk, h)

        Para i=1 hasta M hacer:

            utilij (s*)

            Para cada i tal que útil es máxima hacer:

                pi ← pi + 1

                k ← k+1

    Regrese mejor solución
    
```

Figura 2-8: Pseudo-Código.

La utilidad está dada por la función:

$$util_{ij} = \frac{(c_{ij}+w_{ij})(1+0.1I_{ij})}{1+p_i} \quad (2.2)$$

El pseudocódigo (Figura 2-8) explica que el proceso comienza con una solución inicial s_0 , e inicializando un valor k . Luego se evalúan la penalización para cada uno de los arcos (i,j) sobre la función objetivo gs en los atributos que se desean penalizar. Se comienza a iterar perturbando la solución actual para cada operador k hasta un criterio de parada, esto se hace teniendo en cuenta una función de utilidad la cuál divide sobre el número de penalizaciones recibidas para cada arco evaluado, de forma tal que esta función de utilidad mejore la diversificación de la penalización.

Este ejemplo de $util_{ij}$ viene de Arnold & Sörensen (2019), quienes evalúan una función de utilidad, siendo c el costo de la función de ir de un nodo i a un nodo j , con $w(i,j)$ que denota la anchura en términos de la compactación por ruta $l(i,j)$, cuenta el número de intersecciones entre bordes. Desde esta perspectiva el arco en la solución actual con la función de utilidad más alta es penalizado. El denominador asegura que la penalización no sea repetitiva. En la figura se aclara a qué se refieren con c y w .

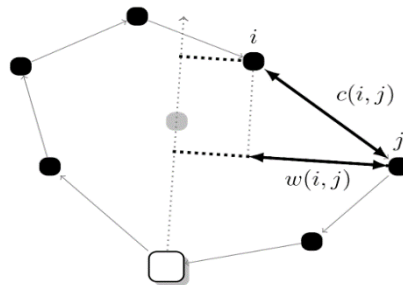


Figura 2-9: Interpretación de atributos de selección. Fuente: Arnold & Sörensen, 2019^a.

La razón de escoger un criterio de penalización como w (promedio de compactación por ruta, medido por ancho) radica en que Arnold y Sorensen (2019) encontraron en un ejercicio de aprendizaje de máquina que el criterio de anchura respecto al centro de

gravidad es un atributo que puede servir de guía para el algoritmo de búsqueda local guiada.

Debido a que los algoritmos de búsqueda local intra-rutas se centran en la optimización de una solución dada, mientras que los algoritmos inter-rutas exploran diferentes soluciones, su selección debe ser complementaria. En este trabajo se selecciona los algoritmos intra-rutas k-opt (1-Opt, 2-Opt y 3-Opt) así como lo hacen Arnold y Sorensen por medio del algoritmo Lin-Kerningham (2019), y tres algoritmos inter-rutas: Cross, Exchange y 2-Opt para posteriormente definir una secuencia entre estos operadores. Los algoritmos escogidos son complementarios entre ellos, ya que el Cross permite intercambiar sub-rutas, exchange permite intercambiar arcos o nodos adyacentes entre rutas y 2-Opt permitirá intercambiar dos nodos de cada ruta.

Una vez definido y seleccionados estos operadores de búsqueda local, se deben determinar las características de penalización que servirá como guía para el diseño del algoritmo de búsqueda local guiada, que es el objetivo principal de este trabajo. Para este trabajo la forma de guiar el algoritmo no se hará por medio de penalizaciones y utilidad de los atributos, sino por medio de la caracterización que ofrezca la herramienta de machine learning escogida. Esto se presenta en los siguientes dos capítulos.

Capítulo 3: Algoritmo de búsqueda local guiada.

En este capítulo se presenta la formulación matemática del problema de optimización y la evaluación de las métricas (características) de las soluciones, también la definición de los parámetros de las instancias del problema, que servirán como insumo para la construcción del algoritmo de búsqueda local guiada.

Se utilizan herramientas de máquina de aprendizaje para establecer que características de las soluciones servirán de guía para el algoritmo propuesto. Para la preparación y preprocesamiento de los datos se realiza un análisis de correlación y de componentes principales. Para la definición de las reglas se construye un árbol de decisión y posteriormente se evalúa a través de una matriz de confusión. Este proceso finaliza con la extracción de las reglas del árbol de decisión para caracterizar las soluciones que etiquetaremos como “buena”, “no tan buena” o “mala” de acuerdo con el valor de las métricas propuestas.

las métricas en los tres tipos de soluciones al modelo de VRP de nuestro estudio.

Este capítulo inicia con la definición de conocimiento del problema específico desarrollado por Arnold y Sorensen (2019) para el caso de los problemas VRP, sección 3.1. La sección 3.2 Continúa con la definición del proceso de aprendizaje de máquina. En la sección 3.3 se describen las herramientas de máquina de aprendizaje que se van a usar en el modelo de búsqueda local guiada. Se finaliza este capítulo con la definición de las reglas de guía que serán insumo para el modelo de búsqueda local guiada y una interpretación del proceso de guía para ir de una solución “mala” a una solución “buena” de acuerdo con las reglas generadas por la herramienta de aprendizaje de máquina.

3.1. Conocimiento específico del problema

Para poder disminuir el tiempo que le toma alcanzar una solución a una metaheurística Arnold y Sorensen (2019) hablan de la necesidad de obtener un “conocimiento” sobre

aquello que distingue una buena de una mala solución, que vaya más allá del valor de la función objetivo (Sörensen et al., 2018).

Arnold y Sorensen (2019) definen dos tipos de problemas que se han estudiado en los problemas de optimización combinatoria: el problema de conocimiento con “clase específica” que se aplica a algunos parámetros de las instancias específicas (número de clientes o la posición del depósito), propio de las metaheurísticas actuales, y el otro es el conocimiento de un problema específico. Este conocimiento es independiente de las propiedades de instancias definidas y es aquel que es posible generalizar de acuerdo con las instancias” (Sörensen et al., 2018).

La metodología propuesta usa herramientas de aprendizaje de máquina para identificar características y/o patrones relevantes en las métricas proporcionadas que permitan predecir la categoría de solución entre las etiquetas de “buena”, “no tan buena” y “mala” se obtendrá con el mínimo error. Esta herramienta al usar un conjunto de datos a partir de métricas propuestas con diferentes tipos de parámetros puede ayudar en la obtención de conocimiento del problema específico que posteriormente permita guiar la búsqueda de una buena solución para el problema de minimización en la huella de carbono en el ruteo de vehículos heterogéneos en el menor tiempo posible.

3.2 Construcción del conjunto de datos.

En este estudio, se utilizan métricas (características) derivadas de tres tipos de soluciones del problema VRP-heterogéneo para la minimización de la huella de carbono que se utilizan como insumo (variables predictoras) en las herramientas de aprendizaje de máquina. Estas métricas se evalúan a partir de los parámetros que caracterizan la representación de las soluciones de los problemas VRP.

La calidad de las soluciones al problema VRP se establece en función de la diferencia del valor de la función objetivo evaluada con respecto al valor del óptimo global, de la siguiente manera: $MIPGap = |Obj.Bound - Obj.Val| / |Obj.Val|$, donde *Obj.Bound* representa el valor de la mejor solución factible encontrada hasta el momento u óptimo global calculado al momento, mientras que *Obj.Val* es el valor de la función objetivo de la mejor solución encontrada hasta ese momento. Dado lo anterior, se considera y etiqueta una solución "mala" cuando la diferencia con el óptimo global encontrado al momento es inferior a 0.4, o de 40% de diferencia, una solución "no tan buena" cuando la diferencia es inferior a 0.2

o del 20% de diferencia respecto a ese óptimo global, y una solución "buena" cuando la diferencia es inferior a 0.01 o del 1% respecto a dicho óptimo global. Estas etiquetas sobre una solución constituyen la variable de respuesta para poder identificar qué características posee una solución "mala", basado en las métricas definidas. El objetivo es construir una base de conocimientos amplia y específica, independiente de las instancias establecidas que ofrezca información sobre el problema específico usando herramientas de aprendizaje de máquina.

3.3 Herramientas de aprendizaje de máquina

Entre los modelos que se han usado para poder predecir observaciones futuras se destacan dos modelos supervisados principalmente: los modelos de "regresión" y los modelos de "clasificación" (Alpaydin, 2014). Los modelos de clasificación buscan patrones que permitan diferenciar entre dos o más clases o categorías, mientras que los modelos de regresión tratan de modelar la relación entre un conjunto de variables explicativas y una variable respuesta.

Entre los modelos de clasificación, los árboles de decisión cuentan con la ventaja de ser de fácil interpretación por el uso de nodos y ramas que permiten identificar reglas entre las distintas características. Sin embargo, los árboles de decisión no tienen tan buen poder predictivo frente a otros modelos de clasificación, debido a que los nodos se sesgan hacia el modelo de entrenamiento. En los últimos 20 años se ha trabajado en disminuir esta desventaja de los modelos de árboles de decisión construyendo nuevos modelos que agregan a la lógica de los árboles otro tipo de herramientas, como los ensambles, con tal de minimizar el error entre la predicción realizada por el modelo y los valores reales de los datos. Pero esto ha hecho que los modelos pierdan poder de interpretación, como es el caso de los últimos dos modelos realizados por L. Breiman Bagging y Random Forest, o Ada Boost de Freund & Schpaire. En Tabla 3-1 se muestra la fecha de creación de algunos modelos de árbol de decisión en los últimos años y los nombres de ellos.

Algoritmos	Autores	Año
Chaid	G.Kass	1980
Cart/Rpart	L.Breiman	1984
Bagging	L.Breiman	1986
M5	R.Quinaln	1992
C4.5/J.4.8	R.Quinaln	1993
Quest	Loh & Shih	1997
AdaBoost	Freund & Schpaire	1999
RandomForest	L.Breiman	2001
Ctree	Hothorn et al	2006
MOB	Zeileis et al	2008
C5.0	Kuhn & Johnson	2013
EvTree	Grubinger et al	2014

Tabla 3-1: Árboles de decisión. Fuente: Hothorn & Zeileis (2017).

Por su parte los modelos de “regresión” se aplican cuando la variable de respuesta es continua. Si bien una herramienta como la regresión logística ha ofrecido buenos resultados en la predicción y descripción de características usando variables de respuesta dicotómicas, como es el caso de los puntos por atributos por riesgo o “scorecard” en el otorgamiento de créditos, esta herramienta no explora como se relacionan las características de los atributos (Ha, 2010). Considerando lo anterior un árbol de decisiones como CART, brinda la oportunidad de entender e interpretar las relaciones entre las características de las variables usando una variable de respuesta que puede ser vista como continua, dicotómica o categórica.

Además se debe tener en cuenta que de acuerdo con Ge et al., (2017) el proceso de analítica de datos se puede dividir en: preparación, pre- proceso, selección, entrenamiento-modelado y por último el proceso de análisis de datos (Ge et al., 2017). Para poder hacer el pre-proceso y la selección de variables antes del entrenamiento se usan principalmente herramientas como las de correlación de Pearson y Análisis de Componentes principales (Olson & Moore, 2016) .

3.3.1 CART- Árboles de Clasificación y Regresión

Esta herramienta consiste en un árbol de decisiones que se divide por medio de un único nodo y este se separa de forma dicotómica en ramas que se convierten en nodos hijos y

de esta manera se crea el proceso de partición recursiva que finaliza en una de tres posibilidades: sólo hay una observación en cada hoja, las observaciones tienen la misma probabilidad en los nodos hojas o alcanza un límite de profundidad (Wu et al., 2008).

En un árbol de decisiones CART, se selecciona una medida de pureza para determinar la mejor variable a utilizar en la división. La variable que logra la mayor pureza se elige como la primera en ser utilizada, y este proceso continúa sucesivamente (Trujillano et al., 2008) El objetivo de la clasificación es maximizar la pureza mediante una división óptima. La función de criterio utilizada para la división, conocida como medida de impureza, se describe en la ecuación 3.1 (Larose & Larose, 2014) :

$$\Phi(s, t) = 2P_L P_R \sum_{j=1}^{\# \text{ de clases binario}} |P(t_L) - P(t_R)| \quad (3.1)$$

$$\Phi(s, t) = \max_{s \in S} \Phi(s, t) \quad (3.2)$$

t_L → nodo hijo de la izquierda en el nodo t

t_R → nodo hijo de la derecha en el nodo t

P_L → Porcentaje de los que van a la izquierda respecto al total = $p = \frac{n_L}{n}$

P_R → Porcentaje de los que van a la derecha respecto al total = $p = \frac{n_R}{n}$

$$P(t_L) \rightarrow \frac{\text{Numero de registros en } t_L \text{ de clase } j}{\text{Numero de registros en } t} \quad (3.3)$$

$$P(t_R) \rightarrow \frac{\text{Numero de registros en } t_R \text{ de clase } j}{\text{Numero de registros en } t} \quad (3.4)$$

Esta herramienta entrega entonces probabilidades para segmentos de las características de las variables y una ramificación que supone unas reglas entre las características de cada variable. La fórmula (3-1) indica el criterio de pureza, la formula o ecuación (3-2)

indica que se escoge la mayor medida de impureza para cada nodo. El cálculo de dicha impureza se realiza con las ecuaciones (3.3 y 3.4)

3.3.2 Desviación estándar

Es una medida de la dispersión de los datos (distancia promedio de cada una de las variables respecto al promedio de la variable) (Hines & Montgomery, 1993) su formulación se presenta en la ecuación (3.5) y (3.6) para una variable x y otra y.

$$S_x = \sqrt{\frac{(x_i - \bar{x})^2}{n}} \quad (3.5)$$

3.3.3 Covarianza

Se define la covarianza como una medida que representa la variación de dos variables aleatorias respecto a sus medias, se originan usando el producto cruzado entre las dos varianzas (Hines & Montgomery, 1993)

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y}) \quad (3.6)$$

3.3.4 Correlación de Pearson:

Es una medida de la relación lineal entre dos variables continuas. Entregando dos posibles respuestas, existe dependencia o independencia lineal. Tomando valores entre -1 y 1, siendo uno perfectamente correlacionado, 0 incorrelacionado y -1 perfectamente correlacionado con direcciones opuestas. Es decir, a medida que una variable crece en una dirección, la otra también lo hace, en el caso de ser los valores positivos; mientras que si sus valores son negativos, indica que mientras una crece la otra decrece de forma lineal (Hines & Montgomery, 1993).

Su interpretación geométrica resulta ser el coseno del ángulo del producto escalar (punto o interno) entre las desviaciones de los datos (en forma vectorial) respecto a sus medias; lo que implica la linealidad entre los dos vectores, por eso ofrece la fuerza y dirección y proporcionalidad de esa relación (Corzo, 2019).

Su fórmula es:

$$r = \frac{S_{xy}}{S_x S_y} \quad (3.7)$$

La covarianza sobre el producto de las desviaciones típicas de cada variable.

3.3.5 PCA – Análisis de Componentes principales

El profesor Jimmy A. Corzo define el PCA como una representación óptima de una matriz de datos $Y_{n \times p}$ con $r < p$ variables independientes (componentes principales) (Corzo, 2019).

Lo “*optimo se refiere a que las r componentes utilizadas para la representación contienen la mayor cantidad posible de información sobre la matriz de datos $Y_{n \times p}$, en términos de la cantidad de varianza del conjunto de variables que acumulan las r componentes. En este sentido, las componentes principales son una forma de reducir la dimensionalidad de con $r < p$ variables que además facilitan representaciones gráficas de fácil comprensión e interpretación*”. Su proceso consiste en realizar la matriz de correlaciones o de covarianzas, y por medio del teorema de descomposición espectral hallar los valores y vectores propios de las variables a analizar dentro de una matriz ortogonal, este método aplica los métodos de multiplicadores de LaGrange para garantizar que la varianza entregada por la variable es la máxima que se puede ofrecer y tiene la forma de un vector que se puede explicar en dos dimensiones (Corzo, 2019).

Los componentes principales entonces se pueden definir como combinaciones lineales, que crean una proyección en las dimensiones significativas que pueden explicar la mayor parte de la varianza. Entre más larga sea una flecha mejor explica un espacio vectorial su varianza, y entre más pequeño es el ángulo entre dos flechas significa que mayor es su correlación (Corzo, 2019).

Este proceso entrega entonces una idea de que variables descritas como vectores cuentan con mayor variabilidad y cuales tienen una correlación al estar muy cercanos entre sí, hasta por fin alcanzar las mejores características de las variables en cuanto a aquellas que puedan ser significativas y la mejor forma de segmentar.

3.4 Definición de parámetros de las soluciones.

Para el diseño de este trabajo se genera un conjunto de datos con las métricas que permitan diferenciar una solución “buena” una solución “no tan buena” y una solución “mala” para el problema formulado de minimización en la huella de carbono en el ruteo de vehículos heterogéneos. Se parte de cuatro parámetros o atributos principales para la generación de instancias: 1) Número de clientes, 2) Ubicación del depósito, 3) Distribución de la demanda homogénea o heterogénea, 4) El tamaño promedio de la ruta definido por las características de cada vehículo en cuanto a capacidad, costo y peso.

Se define un conjunto de instancias del problema con nodos de demanda con un número entre 14 y 17, dichos nodos se localizan de manera aleatoria con coordenadas que están entre los 12 a 15 grados de latitud y entre 100 a 101 grados de longitud. La ubicación del depósito puede ser en el centro o en un vértice escogido de manera aleatoria. La demanda de los clientes puede ser heterogénea u homogénea con valores entre 1 y 15 unidades demandadas. Por último, cada vehículo tiene una capacidad previamente definida, en algunos casos todos los vehículos tienen la misma capacidad y en otros su capacidad variaba de manera aleatoria con entre 20 y 50 unidades.

3.4.1 Definición de las métricas para las Soluciones.

Es necesario crear medidas (características) que permitan diferenciar una “buena”, “no tan buena” o “mala” soluciones caracterizadas por una herramienta de aprendizaje de máquina, esto es, que dicha herramienta nos permita caracterizar las medidas para cada tipo de solución. Este proceso implica un enfoque exploratorio, pero para refinarlo aún más, se recurre a las métricas propuestas por Arnold & Sorensen (2018), las cuales han sido analizadas en profundidad por Lucas Falvien, Roman Billot y Marc Sevaux (2019) (Lucas et al., 2019).

Para lograr una definición clara de estas métricas, se utilizan las siguientes notaciones: el conjunto de todas las rutas del VRP se denota como "R", y cada ruta se identifica con "r". El número de nodos se representa por "n", y la distancia se denota como "d" (distancia geoestadística). "Ang" denota el ángulo entre dos nodos adyacentes con respecto al centro de gravedad. "D" representa el depósito y "I" es el número de intersecciones entre rutas. Por último, "G" se refiere al centro de gravedad de una ruta. Estas métricas o características se etiquetan como "Rpta" y se calcula de la siguiente manera:

Rpta1-Número de Intersecciones por cliente promedio.

$$\frac{\sum_{i=1}^{|R|-1} \sum_{j=i+1}^{|R|} I(r_i, r_j)}{N} \quad (3.8)$$

Rpta2-Mayor distancia entre dos clientes conectados por ruta.

$$\frac{\sum_{r \in R} \max_{i \in \{1, \dots, |r|-1\}} d(n_i^r, n_{i+1}^r)}{|R|} \quad (3.9)$$

Rpta3- Distancia promedio entre el depósito a clientes conectados.

$$\frac{\sum_{r \in R} (d(D, n_1^r) + d(n_{|r|}^r, D))}{2|R|} \quad (3.10)$$

Rpta4-Distancia promedio entre rutas

$$\frac{\sum_{r_1 \in R} \sum_{r_2 \in R \setminus r_1} d(G_{r_1}, G_{r_2})}{|R| \cdot (|R|-1)} \quad (3.11)$$

Rpta5-Ancho promedio entre rutas

$$\frac{\sum_{r \in R} (\max_{i \in \{1, \dots, |r|\}} d(L_{G_r}, n_i) - \min_{i \in \{1, \dots, |r|\}} d(L_{G_r}, n_i))}{|R|} \quad (3.12)$$

Rpta6-Intervalo promedio de radianes por ruta.

$$\frac{\sum_{r \in R} \max_{i \in \{1, \dots, |r|-1\}} \text{rad}(n_i^r, n_{i+1}^r)}{|R|} \quad (3.13)$$

Rpta7-Promedio de compactación por ruta, medido por ancho.

$$\frac{\sum_{r \in R} \sum_{i=1}^{|r|} d(L_{G_r}, n_i)^+}{N} \quad (3.14)$$

Rpta8-Promedio de compactación por ruta, medido por ángulo.

$$\frac{\sum_{r \in R} \sum_{i=1}^{|r|} \text{ang}(G_r, n_i)}{N} \quad (3.15)$$

Rpta9-Promedio de profundidad por ruta.

$$\frac{\sum_{r \in R} \max_{i \in \{1, \dots, |r|\}} d(n_i^r, D)}{|R|} \quad (3.16)$$

Rpta10-Desviación estándar de número de clientes por ruta.

$$\sqrt{\frac{\sum_{r \in R} (|r| - \frac{N}{|R|})^2}{|R|}} \quad (3.17)$$

Las primeras dos métricas o Rptas reflejan observaciones en el campo de la investigación de operaciones, dichas observaciones son debidas a que en la solución al problema del agente viajero no se esperan intersecciones en la ruta, o nodos con mucha distancia, se asume que lo mismo puede pasar en un VRP. En palabras de Arnold & Sorensen (2018), “se espera para el VRP que las buenas soluciones tienden a tener menos intersecciones, y arcos menos largos”. La Rpta 3 está basada en la hipótesis de que en una buena solución los arcos conectados entre el depósito y el cliente deben ser pequeños. Intuitivamente se espera encontrar una buena separación entre cada ruta (Rpta 4) y que no se superpongan (Rpta 5), ni tener profundidades parecidas (Rpta 9). También se espera poca varianza entre los nodos o clientes por ruta, es decir rutas homogéneas en número de clientes (Rpta 10). También se espera que las rutas no sean anchas respecto al ángulo entre los clientes y el depósito (Rpta 6). Por último, se espera que las rutas sean bastante compactas (casi como una línea y no como una rueda), para esto se mide el ancho promedio en distancia respecto al centro de gravedad (Rpta 7) o el ancho promedio del ángulo respecto al centro de gravedad (Rpta 8). En general se pretende que haya una separación visible entre rutas, que las rutas sean bastante compactas entre sí mismas, con pocas intersecciones y los nodos de esas rutas sean bastante cercanos entre ellos mismos, además de tener rutas con números de clientes parecidos.

3.4.2 Definición de las Instancias

Para determinar las principales características que hacen diferentes a una solución “buena” de una “mala” se diseñaron las métricas previamente presentadas; sin embargo, los autores Arnold y Sorensen (2019) proponen esta hipótesis: dos problemas con las mismas métricas, pero para diferentes instancias (parámetros exclusivos de un modelo en particular) pueden tener diferente calidad, por lo cual se proponen ocho instancias que permitan caracterizar la solución etiquetada como “buena” de la solución “mala”, estas instancias permiten verificar que haya diversidad en cuanto a los parámetros escogidos para alimentar la base que se usará para obtener las reglas del modelo del árbol de decisiones.

Características de las Instancias

- I1. Número de clientes
- I2. Número de Rutas
- I3. Promedio de demanda entre los clientes por cada ruta
- I4. Consumo promedio entre clientes
- I5. Desviación estándar entre consumo promedio entre clientes
- I6. Distancia promedio de cliente a depósito.
- I7. Desviación estándar del consumo entre cliente al depósito

Se excluye la característica 8 para las instancias propuestas, la desviación estándar de radianes de los clientes hacía el depósito y se analiza la desviación estándar de radianes de los clientes entre ellos mismos. Las razones de la ausencia de esta característica se deben al análisis de Lucas et al (2019), indicando que tanto en la predicción media junto como un índice Gini fue bastante pobre en la interpretación de las soluciones, por lo que se va a analizar la distancia respecto al ángulo y no solo entre el nodo del depósito y el resto, sino entre cada uno de los nodos.

3.5 Evaluación y resultados de las métricas por medio de los modelos de aprendizaje de máquina

Lo primero que se hizo fue analizar la correlación de Pearson sobre las métricas mencionadas (Figura 3-1) El color rojo intenso indica una alta correlación positiva, el color verde intenso una alta correlación negativa, como determina la escala (rectángulo de la derecha). Podemos observar es la alta correlación (mayor a 0.8) entre algunas métricas como son:

Rpta2, Rpta3 y Rpta 9: es decir hay una alta correlación entre la mayor distancia de dos clientes conectados por ruta y la distancia promedio entre clientes y depósito, así como la profundidad de cada ruta está altamente correlacionado con los nodos o clientes con largas

distancias. También podemos observar que la Rpta o métrica 7 y la Rpta o métrica 5, están correlacionadas, pero de una menor manera, aunque sigue estando por encima del 0.75 de correlación: es decir, a medida que aumenta la anchura entre las rutas, son más compactas las rutas.

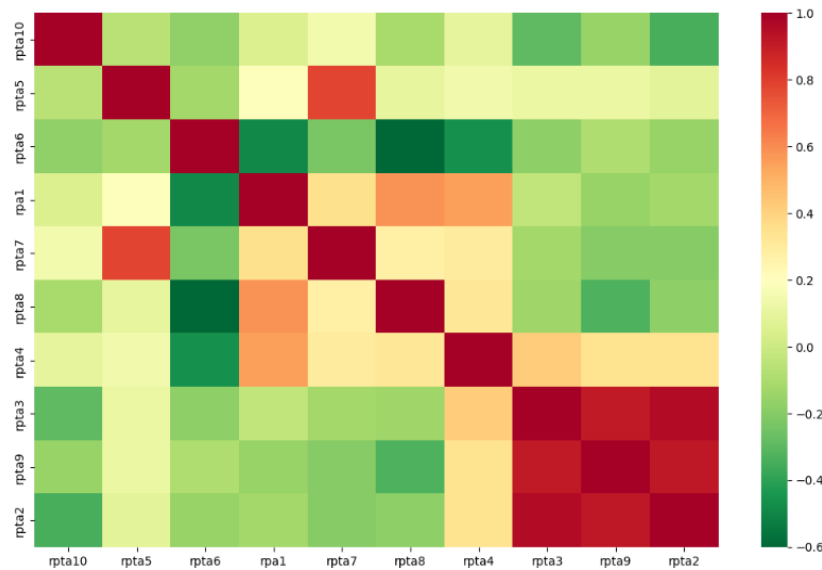


Figura 3-1: Correlación de Métricas.

- 3.4.2 Resultados del análisis de Componentes principales

De acuerdo con la figura 3-2, 4 componentes son capaces de explicar la mayor parte de la varianza, el 98.5 % de ella. Las dos primeras son capaces de explicar en más de 76%, con la tercera en casi 92%.

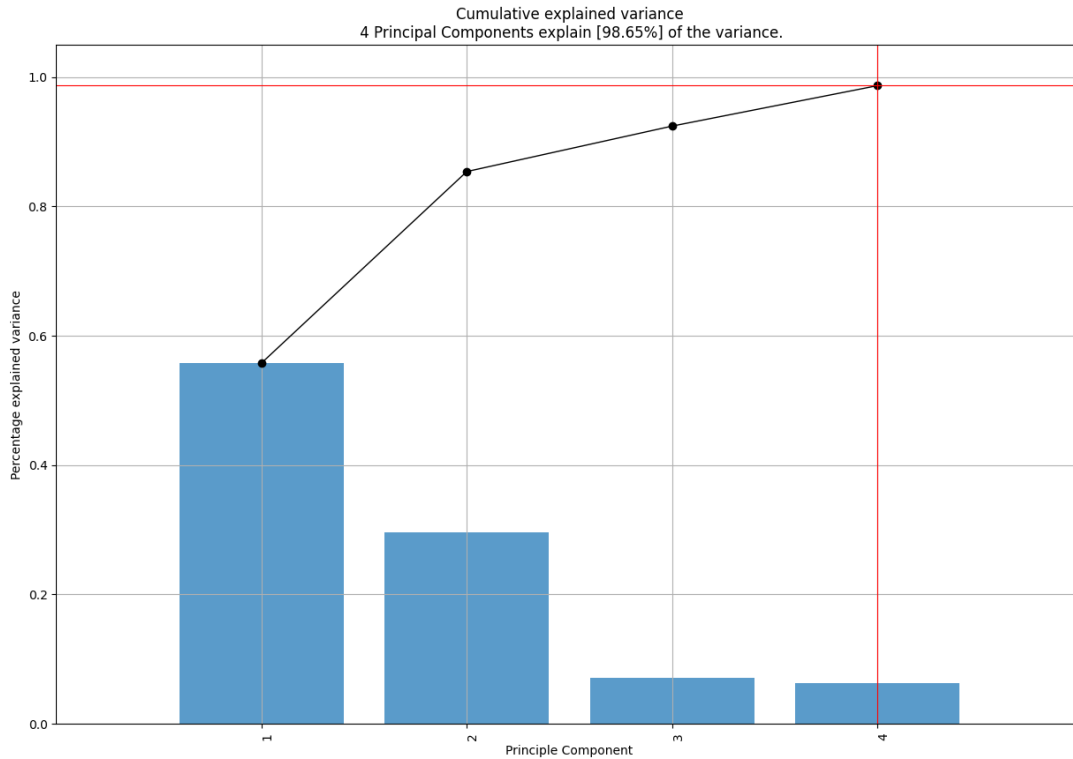


Figura 3-2: Explicación de la varianza acumulada por componentes.

Es decir, que se puede describir la mayor parte de las variables y su varianza por medio de pocos componentes. La figura 3-3, usa las 2 primeras componentes para indicar que las variables están correlacionadas entre ellas como previamente se había visto en el análisis de correlación y cuentan con tamaños de flechas similares, indicando que explican casi la misma cantidad de varianza para esas dos componentes. Por tanto, se puede reafirmar que las características de las soluciones en las métricas (Rptas) 3, 2 y 9 están altamente correlacionadas y cuentan con casi la misma cantidad de varianza que explica el espacio vectorial para esas dos componentes (figura 3-3). Por otra parte, no es fácil analizar diferencias entre los puntos morado, azul y amarillo que representan las categorías de la variable respuesta (color morado solución “buena”, color verde solución no tan “buena”, color amarillo solución “mala”), por lo que resaltamos la importancia de realizar el modelo usando un árbol de decisión.

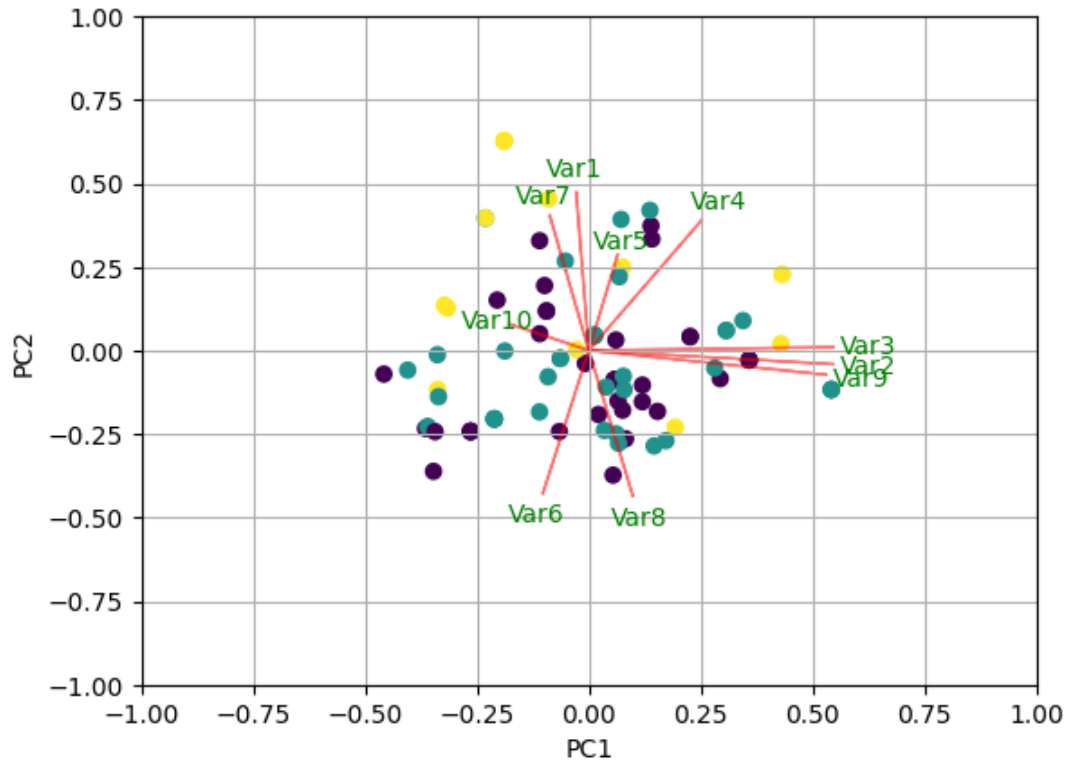


Figura 3-3: Análisis de los 2 primeros Componentes Principales.

El árbol de decisión ayuda a comprender las características de las buenas y malas respuestas para la solución, para poder definir los criterios sobre los cuales se va a guiar el algoritmo de búsqueda local. El árbol se hace con un máximo de 3 ramificaciones.

El árbol de decisión se lee de la siguiente manera: Dentro del primer cuadro o nodo primero aparece el atributo, por ejemplo, la métrica o $Rpta\ 8 \leq 78.649$, es decir si se cumple dicho valor se debe dirigir a la izquierda (Rpta 4). Debajo de ese número en el primer nodo aparece un 100% que significa que esta muestra tiene el 100% de los registros. Y los valores de abajo [0.379, 0.424, 0.197], indican que si eso se cumple hay un 0.379 de probabilidad de que sea 0 (solución buena), un 0.424 de que sea 1 (solución no tan buena) y un 0.197 de que sea la categoría 2 (solución mala del modelo). El nombre final de este nodo indica que de acuerdo con dicha probabilidad la etiqueta a la que pertenece este nodo Solución es: “No tan buena”, “buena” o “Mala”.

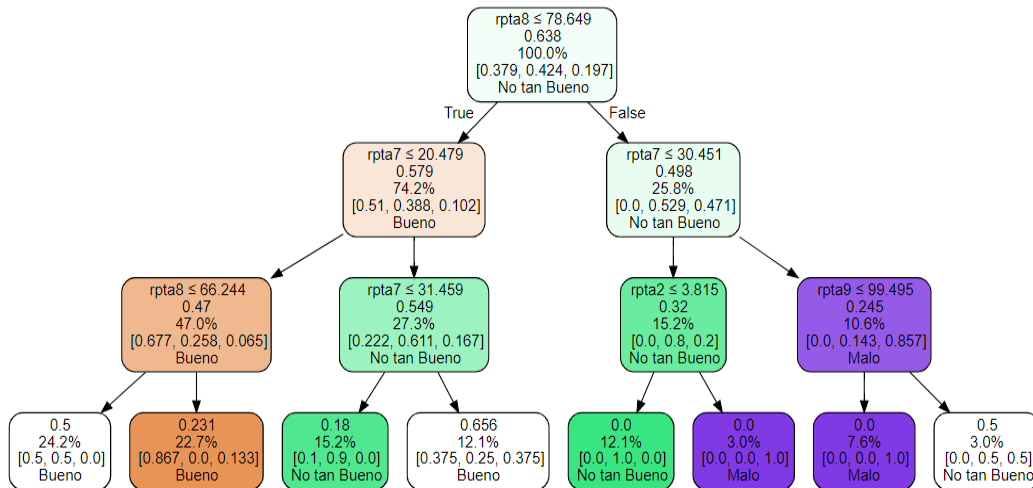


Figura 3-4: Árbol de decisiones.

```

--- rpta8 <= 78.65
|--- rpta7 <= 20.48
| |--- rpta8 <= 66.24
| | |--- class: 0
| |--- rpta8 > 66.24
| | |--- class: 0
|--- rpta7 > 20.48
| |--- rpta7 <= 31.46
| | |--- class: 1
| |--- rpta7 > 31.46
| | |--- class: 0
--- rpta8 > 78.65
|--- rpta7 <= 30.45
| |--- rpta2 <= 3.81
| | |--- class: 1
| |--- rpta2 > 3.81
| | |--- class: 2
|--- rpta7 > 30.45
| |--- rpta9 <= 99.50
| | |--- class: 2
| |--- rpta9 > 99.50
| | |--- class: 1
  
```

Figura 3-5: Reglas ampliadas de árbol de decisión.

De acuerdo con lo anterior se entiende que una solución buena deben tener una Rpta 8 inferior a 78.64, mejor si está por debajo de 66.24, ya que en ese caso se garantiza que son soluciones “buenas” o si es mayor a 66.24 debe tener una Rpta 7 mayor a 31.459, mientras que una solución mala debe tener Rpta 8 superior de 78.649 básicamente.

- 3.4.4. Matriz de confusión de las soluciones al modelo.

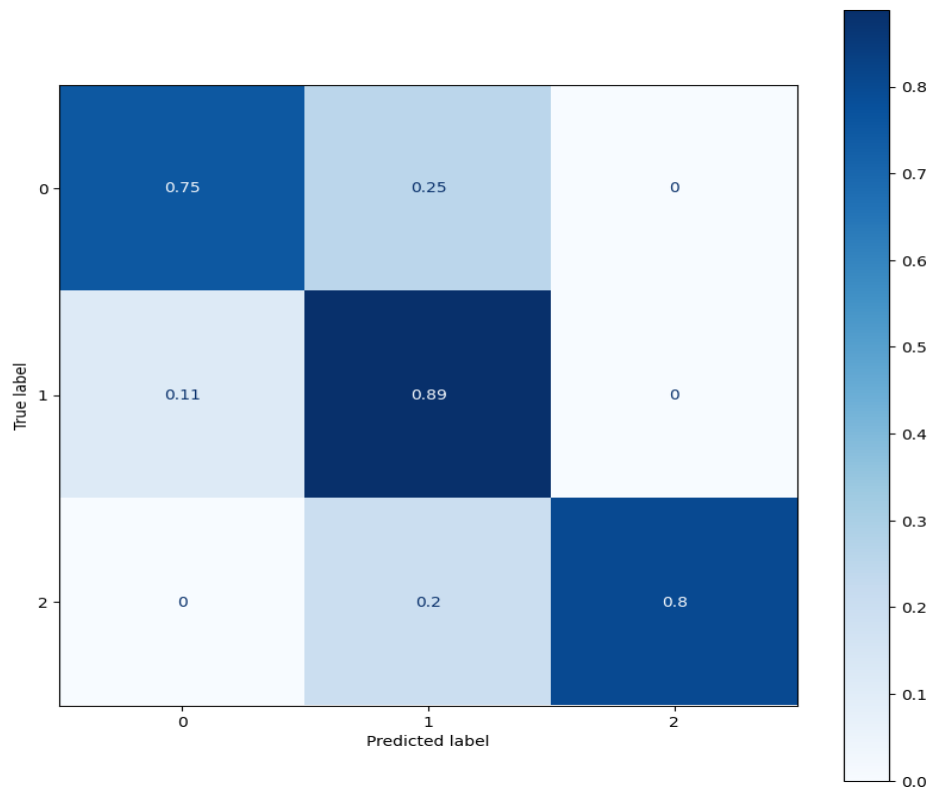


Figura 3-6: Matriz de Confusión.

La matriz de confusión se realizó con 35 registros de la base y los resultados indican que de la solución “buena” se puede predecir que el 75% se comportara con estas reglas, de la solución “no tan buena” se predice en un 89% y de la solución mala alcanza a predecir el 80%. Es decir, el modelo puede predecir las 3 soluciones con una precisión mayor al 75 % lo cual nos indica que las métricas seleccionadas si aportan en la generación de las reglas del modelo de guía para el modelo de búsqueda local guiada.

3.6 Interpretación de los resultados

Para poder interpretar los resultados anteriormente descritos (figura 3-5) se comienza con ejemplificar cómo estas características pueden constituirse en reglas que pueden guiar la búsqueda de una solución y así pasar de ser una solución “mala” a una “buena”. Se usa el ejemplo de la figura 3-7 en el que muestran tres tipos de soluciones usando el árbol de

decisiones aplicado al problema VRP formulado en los capítulos anteriores. (solución “mala”, solución “no tan buena” y solución “buena”)

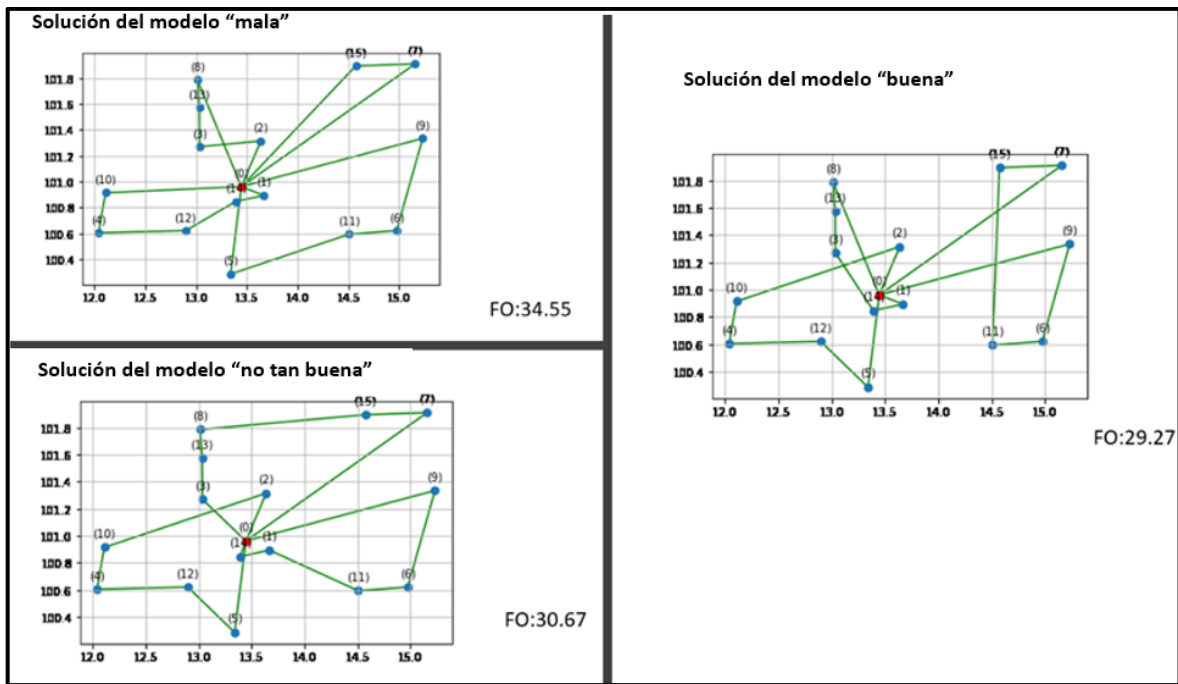
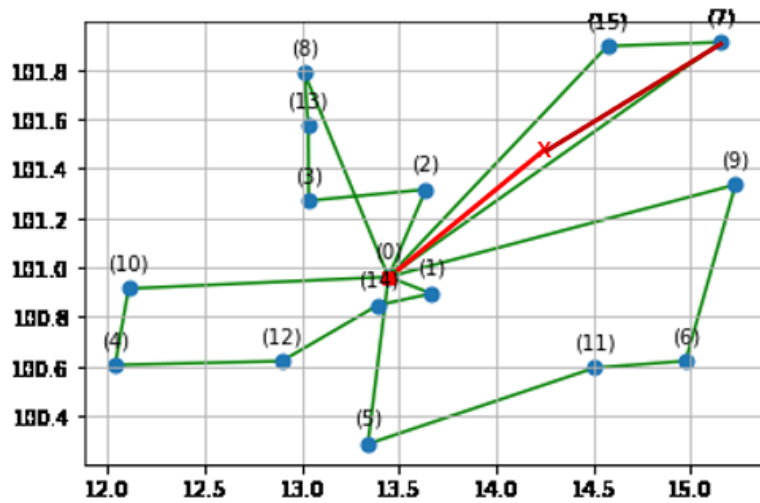


Figura 3-7: Comparación de las soluciones.

Para el cálculo de la distancia se utilizan métricas geodésicas en kilómetros, con la latitud y longitud, y para calcular el consumo de combustible se calculó una velocidad óptima y se asumió que el terreno es plano y los carros no tienen interrupciones de tráfico.

Estas soluciones al problema VRP que se visualizan en la Figura 3-7 cuenta con la siguiente respuesta de la función objetivo en litros de consumo de combustible en emisiones de CO2 equivalentes:

1. 34.55 para la solución “mala”
2. 30.67 para la solución “no tan buena”
3. 29.27 para la solución “buena”



FO:34.55

Figura 3-8: Detalle de VRP para solución “mala”.

Las líneas rojas y el círculo de la figura 3-8 indican cómo es medida la compactación del arco, medido por ángulo suponiendo que la x es el centro de gravedad de dicha ruta. Entendiendo lo anterior se despliega la siguiente tabla 3-2, con la que se analizan los 4 arcos que mayor ángulo poseen respecto al centro de gravedad de todas las rutas para cada solución:

	Transforma	Angulo arco al centro de gravedad de la ruta	Vertices	Rutas
Solución al modelo mala		171.9120291	[0,7]	Ruta 3
	x	156.2722824	[2,3]	Ruta 2
		142.7077078	[0,9]	Ruta 4
	x	106.2479776	[11,5]	Ruta 4
Solución al modelo no tan buena		155.0936504	[0,9]	Ruta 3
	x	141.2190657	[8, 15]	Ruta 1
		140.6739941	[10,2]	Ruta 2
	x	104.1869631	[11,1]	Ruta 3
Solución al modelo buena		176.8030727	[15,11]	Ruta 3
		149.470284	[0,7]	Ruta 3
		140.6739941	[2,10]	Ruta 2
		118.2026576	[14,3]	Ruta 1

Tabla 3-2: Ejemplo de arcos transformados sobre cada solución del VRP

La tabla 3-2 está compuesta de 4 columnas:

1. Transforma, la cual indica si ese arco tendrá algún cambio respecto a la siguiente solución. Es decir, la solución “mala” transforma los arcos (2,3) y (11,5) para llegar a la solución “no tan buena”, ya que estos arcos dejan de existir en la siguiente solución, bien sea porque se unen a un nuevo nodo de su ruta, o bien de una nueva ruta.
2. Ángulo de arco al centro de gravedad de la ruta
3. Vértices que son los nodos o clientes que forma el arco que se desea analizar
4. Rutas se refiere justamente a la ruta a la que pertenece.

Como la penalización del arco está dada por el ángulo descrito en la figura 3-8, será mucho mayor la penalización en los ángulos grandes que se encuentran en la tabla y están por encima de 78.96 que es la regla del árbol analizado en el capítulo anterior en cuanto, se espera minimizar el promedio de compactación por ruta, medido por ángulo.

Para entender mejor la transición de la solución “mala”, a la solución “no tan buena” y la transición de la solución “no tan buena” a “buena” se puede visualizar la siguiente Tabla 3-3

<i>Mala</i>	<i>No tan buena</i>	<i>Buena</i>
Ruta 1	Ruta 1	Ruta 1
(0, 1)	(0, 3)	(0, 1)
(1, 14)	(3, 13)	(1, 14)
(14, 12)	(13, 8)	(14, 3)
(12, 4)	(8, 15)	(3, 13)
(4, 10)	(15, 7)	(13, 8)
(10, 0)	(7, 0)	(8, 0)
Ruta 2	Ruta 2	Ruta 2
(0, 2)	(0, 5)	(0, 2)
(2, 3)	(5, 12)	(2, 10)
(3, 13)	(12, 4)	(10, 4)
(13, 8)	(4, 10)	(4, 12)
(8, 0)	(10, 2)	(12, 5)
Ruta 3	Ruta 3	Ruta 3
(0, 7)	(0, 9)	(0, 7)
(7, 15)	(9, 6)	(7, 15)
(15, 0)	(6, 11)	(15, 11)
Ruta 4	Ruta 4	Ruta 4
(0, 9)	(11, 1)	(11, 6)
(9, 6)	(1, 14)	(6, 9)
(6, 11)	(14, 0)	(9, 0)
(11, 5)		
(5, 0)		

Tabla 3-3: Rutas de las 3 soluciones respuesta del VRP.

En la Tabla 3-3 se puede observar que en el arco (2,3) de la Ruta 2 de la solución “mala” se transforma, yendo el vértice 3 a la Ruta 1 de la solución “no tan buena”, mientras el vértice 2 va a la ruta 2 de la solución de la solución “no tan buena”. La Ruta 2 y 3 de la solución de la solución “mala” será la unión de los vértices que dan como resultado la ruta 1 de la solución “no tan bueno”, una vez ha salido el vértice 2.

El arco (11,5) de la ruta 4 de la solución “mala” el cual se transforma, llevando el vértice 11 a la ruta 3 y el vértice 5 a la ruta 2 de la solución “no tan buena”. Este ejemplo permite

entender cómo se usan las penalizaciones del algoritmo de búsqueda local guiada, y pueden “guiar” en la transición de ser una solución “mala” a una solución “no tan buena”.

El mismo análisis se hizo sobre la Rpta 7, donde fueron escogidos los arcos (4,10), (10,0), y (12,4) y (0,8) de las rutas de la solución “mala”, como potenciales arcos para ser penalizados. Sin embargo, de estos cuatro arcos se penaliza sobre los arcos (10,0) y (0,8)

La Tabla 3-4 resume 5 experimentos de muestra que se diseñaron para explicar por qué primero se penaliza la Rpta 8 (Promedio de compactación por ruta, medido por ángulo). Para este experimento, se escogen los 4 arcos con mayor probabilidad de ser penalizados, de acuerdo con los criterios árbol de decisión, se alcanzan a transformar entre 2 y 3 arcos de las rutas que componen la solución “mala”.

	Arcos criticos Rpta 8	Arcos criticos Rpta 7
Experimento 1	2	2
Experimento 2	3	1
Experimento 3	2	1
Experimento 4	3	0
Experimento 5	3	1

Tabla 3-4: Resumen de número de arcos transformados para 5 experimentos de la solución “mala” a la solución “no tan buena”.

Es decir, en la tabla anterior se indica que, para el experimento 1 de los 4 arcos críticos para la Rpta 8 (Promedio de compactación por ruta, medido por ángulo), 2 de ellos son transformados para mejorar la solución (pasar de solución “mala” a “no tan buena”), y de los 4 arcos críticos de la Rpta 7, dos de ellos serán transformados para mejorar la solución.

Un caso similar se tiene en el análisis de transformación de una solución “no tan buena” a una solución “buena” (tabla 3-5)

	Arcos criticos Rpta 8	Arcos criticos Rpta 7
Experimento 1	2	1
Experimento 2	3	0
Experimento 3	2	1
Experimento 4	2	0
Experimento 5	1	1

Tabla 3-5: Resumen de número de arcos transformados para 5 experimentos de la solución etiquetada “no tan buena” a la solución “buena”.

Este caso reafirma la importancia de la Rpta 8, sobre la Rpta 7 a la hora de escoger arcos que deban ser transformados para mejorar la solución.

Por lo tanto, es posible inferir que el proceso de guía debe comenzar por la Rpta 8 (compresión respecto al ángulo del centro de gravedad) y seguir con la Rpta 7 (Promedio de compactación por ruta, medido por ancho)

En el árbol de decisiones CART la variable (comprensión de compactación por ruta medido por ancho) resulta ser la mejor división en términos de la homogeneidad de la respuesta y en la anterior explicación se puede comprender la razón de su importancia.

De acuerdo con esta herramienta al igual que en Arnold & Sorensen (2019), o Lucas et al (2019), indican que la característica más importante resulta ser la compactación del vehículo dadas por la Rpta 8 (promedio de compactación por ruta, medido por ángulo) y Rpta 7 (Promedio de compactación por ruta, medido del ancho). En otras palabras, en el caso de algoritmos de búsqueda local movimientos que reduzcan el ancho de la ruta respecto al centro de gravedad, o aumenten su compactación indican mejores soluciones acorde con la calidad de la respuesta, por tanto, están más cerca de una solución “óptima”. También se destacan la Rpta 2 (mayor distancia entre dos nodos por ruta) y la Rpta 9 (Promedio de profundidad por ruta), y debido a que estas dos métricas están altamente correlacionadas a la Rpta 3 (distancia promedio entre el depósito a clientes conectados) podemos inferir que una buena solución suele ser poco profundo y no muy distante al depósito y con arcos cortos entre los nodos, al igual que lo mencionado por los autores previamente citados.

Capítulo 4: Desarrollo de Algoritmo de Búsqueda Local Guiada

A partir de los resultados obtenidos a través de la herramienta de aprendizaje de maquina (CART), fue posible definir cuales de las métricas propuestas servirán de guía del algoritmo de búsqueda local. Se busca construir un algoritmo de búsqueda local guiada que obtenga respuestas aceptables en la minimización de la huella de carbono en el ruteo de vehículos heterogéneos de forma eficiente.

En este capítulo se presenta y se describe el diagrama de flujo del algoritmo de búsqueda local guiada (4.1), se realizan una serie de experimentos para determinar que secuencia de operadores para guiar el algoritmo considerando la caracterización hecha en el capítulo anterior (4.2), por último, se evalúan las diferencias del algoritmo propuesto respecto a las soluciones etiquetadas previamente como “buena”, “no tan buena” y “mala” y el mismo algoritmo, pero con otra secuencia de operadores.

4.1 Descripción del Algoritmo.

Un algoritmo de Búsqueda Local Guiada explora iterativamente el espacio de soluciones, comenzando con una solución inicial. Esta solución inicial es perturbada por un operador escogido. En el caso de problemas de vehículos con flota heterogénea debe tenerse en cuenta los límites de las capacidades de cada vehículo (factibilidad). Es decir, una vez realizada la perturbación de la solución se evalúa la demanda acumulada para cada nueva ruta, así como qué vehículo puede aceptar esa demanda acumulada, considerando su capacidad y si esto responde a la minimización de la función del consumo de combustible se guarda como nueva solución.

Teniendo en cuenta que para el problema en estudio de minimización de la huella de carbono en problemas VRP con flota heterogénea, la función objetivo es el consumo de combustible, una posible mejora o perturbación que disminuya el consumo de combustible respecto a la anterior solución será entonces la guardada como nueva solución y se comparará con la siguiente solución propuesta. Este criterio se mantiene hasta que se logra un número de iteraciones, para posteriormente comparar la mejor solución de ese operador con la solución que ofrece el siguiente operador con el mismo ciclo de iteraciones.

Este algoritmo parte de la solución etiquetada como solución “mala” en cuanto a calidad implementada usando programación lineal entera mixta por medio del solver de Gurobi.

El número de iteraciones definido será constante para cada operador (10 iteraciones) y el conjunto de operadores para este algoritmo serán aquellos mencionados en el final del capítulo 2: Cross, Exchange y 2-Opt como operadores inter-rutas y 1-opt, 2-Opt y 3-Opt para los operadores intra-rutas. Se ejecuta también un proceso de perturbación aleatoria N_s ($s = 1, 2, \dots, s_{\max}$) con el fin de ampliar el espacio de búsqueda y minimizar la posibilidad de estancarse en un mínimo local, para los operadores de búsqueda local N_k ($k = 1, 2, \dots, k_{\max}$).

Se obtiene a la salida del algoritmo la solución que minimiza la función objetivo en el consumo de combustible. La solución está compuesta por las rutas con la secuencia de nodos y los vehículos para cada ruta asociada.

El algoritmo se describe en el siguiente diagrama de flujo.

X- solución mejor total
 X'- solución mejor en la búsqueda local
 K es movimiento, donde $k=1$ no es inicial
 S- movimiento de búsqueda local

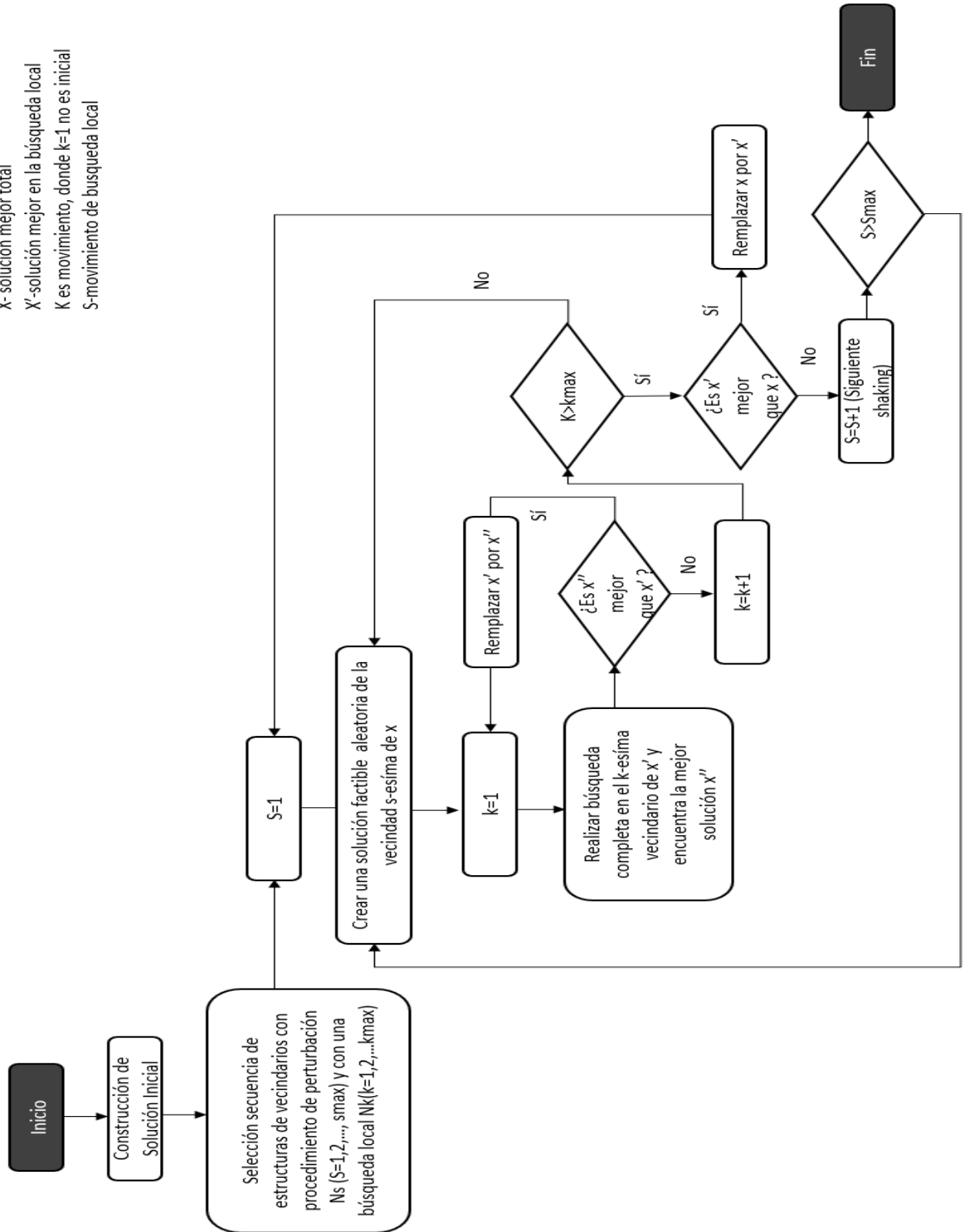


Imagen 4-1. Diagrama de Flujo del Algoritmo de Búsqueda Local Guiada

4.2 Definición de secuencia.

Para definir la secuencia de guía nos basaremos en los resultados obtenidos a partir del análisis realizado usando el árbol de decisiones; es decir, primero se observa qué operadores disminuyen la métrica 8 (Promedio de compactación por ruta, medido por ángulo). Luego, qué operadores disminuyen la métrica 7 (Promedio de compactación por ruta, medido por ancho) y considerando que los valores de la métrica 2 (Mayor distancia entre dos clientes conectados por ruta) y la métrica 9 (Promedio de profundidad por ruta) están correlacionados. Se analizarán solo los valores más pequeños de la métrica 2 (Mayor distancia entre dos clientes conectados por ruta). Al final se escogen 4 de los 6 operadores en los que se obtienen los mejores resultados. Se escoge un operador por cada métrica entregada en el árbol de decisiones analizado.

Los resultados de los experimentos se presentan en tablas que ofrecen el valor de respuesta escogida promedio para cada operador seleccionado, junto a su desviación; además se analizan usando un mapa de calor por cada fila o experimento hecho.

En las tablas 4.1, 4.2 y 4.3 se expresan los resultados de 20 experimentos sobre los operadores y métricas a analizar, con instancias de problemas que tienen entre los 14 y los 17 nodos; las demandas de los nodos se escogen de manera aleatoria entre 1 y 12 unidades, y definidas también de manera aleatoria las coordenadas del depósito y los nodos con entre los 12 a 14 grados de latitud y entre 100 a 101 grados de longitud; por último, la capacidad de los vehículos va de 20 a 50 unidades.

	CROSS	2-OPT	1-OPT	3-OPT	EXCHANGE	2OPT INTER
rpta8	49,08	49,57	50,86	49,08	49,08	49,08
rpta8	55,13	64,83	59,05	59,05	59,05	59,05
rpta8	52,03	51,46	56,58	52,03	52,03	52,03
rpta8	46,96	57,81	56,31	48,80	48,80	48,80
rpta8	64,04	71,20	75,51	75,51	70,40	66,49
rpta8	50,24	49,66	52,03	52,03	57,87	61,72
rpta8	61,46	48,84	51,75	79,22	78,56	83,02
rpta8	75,64	43,17	68,35	75,64	75,64	75,64
rpta8	42,18	47,97	45,73	66,00	66,00	66,00
rpta8	51,08	53,05	64,46	50,08	50,08	50,08
rpta8	54,27	56,53	47,98	61,09	61,09	61,09
rpta8	58,66	71,29	49,27	58,22	62,24	47,41
rpta8	63,98	42,70	70,58	76,63	76,63	64,58
rpta8	71,17	63,97	53,73	70,51	71,17	71,17
rpta8	48,84	72,40	52,88	65,90	65,90	65,90
rpta8	46,60	69,50	62,32	69,54	76,12	63,97
rpta8	61,10	45,54	60,51	45,54	45,54	45,54
rpta8	43,92	68,80	68,42	68,80	68,80	68,80
rpta8	69,01	46,12	47,86	49,69	49,69	49,69
rpta8	51,32	70,02	53,29	68,90	68,90	68,90
Promedio	56,07	56,55	57,59	61,76	62,35	60,53
Desv	9,41	10,68	8,48	10,91	10,70	10,45

Tabla 4.1 Valor Promedio, desviación y mapa de calor de métrica o característica 8 respecto a los operadores.

De acuerdo con los resultados de los experimentos previos y el mapa de calor por cada fila (tabla 4.1) se puede analizar que el operador que más disminuye la característica, Rpta, o métrica 8 (Promedio de compactación por ruta, medido por ángulo) es el Operador: 1) Cross, en el que se puede observar un valor promedio de ángulo de: 56,07 grados, menor respecto a los otros operadores; después del operador Cross está el operador: 2) 2-Opt con un valor promedio de la métrica 8 de 56,55 grados, seguido por el operador: 3) 1-Opt con valor promedio de 57,59 grados. Por tanto, la secuencia de los 3 principales operadores, de acuerdo con este experimento, queda como: 1) Cross, 2) 2-Opt (Intarutas), 3) 1-Opt.

En la siguiente tabla 4.2 es posible analizar los valores de la métrica 7 (Promedio de compactación por ruta, medido por ancho en kilómetros)

	CROSS	2-OPT	1-OPT	3-OPT	EXCHANGE	2OPT INTER
rpta7	49,38	37,27	36,91	34,88	34,88	34,88
rpta7	33,81	42,97	43,05	43,05	43,05	43,05
rpta7	42,42	42,42	42,26	42,42	42,42	42,42
rpta7	35,24	36,53	35,24	35,72	35,72	35,72
rpta7	27,87	27,85	27,87	27,87	27,87	25,87
rpta7	33,64	29,11	20,50	30,48	44,12	31,45
rpta7	30,08	38,41	30,52	44,97	27,56	36,67
rpta7	36,32	27,76	25,03	36,32	36,32	36,32
rpta7	41,41	31,04	37,10	46,04	46,04	46,04
rpta7	38,78	37,22	42,12	30,41	30,41	30,41
rpta7	35,26	32,12	23,15	30,25	30,25	30,25
rpta7	44,71	39,96	38,70	40,48	40,48	37,75
rpta7	47,11	43,28	37,81	44,46	44,46	43,46
rpta7	27,90	30,59	26,93	37,06	27,90	27,90
rpta7	35,03	32,77	22,11	31,71	31,71	31,71
rpta7	46,52	33,13	33,32	46,00	46,00	46,00
rpta7	46,06	40,18	40,44	29,77	32,51	34,07
rpta7	28,39	42,31	42,42	39,46	39,46	39,46
rpta7	50,08	45,20	44,16	39,85	39,85	39,85
rpta7	46,08	29,11	20,50	41,23	41,23	41,23
Promedio	38,80	35,96	33,51	37,62	37,11	36,73
Desv	7,43	5,74	8,23	6,01	6,40	5,94

Tabla 4.2 Valor Promedio, desviación y mapa de calor de métrica o característica 7 respecto a los Operadores.

En cuanto a la secuencia de operadores que disminuye más la métrica 7 (Rpta 7) se tiene: 1) 1-OPT con valor promedio de 33.51 km, 2) 2-OPT (Intra-rutas) con valor promedio de 35.96 km y 3) 2-Opt (Inter-rutas) con valor promedio de 36.73 km, tal como se puede observar en la tabla 4.2.

Por último, tenemos que la secuencia de operadores para la métrica 2 (Mayor distancia entre dos clientes conectados por ruta, medido en kilómetros) es: 1) 1-Opt con valor promedio de 4.21 kms, 2) 2-Opt (Intra-rutas), con valor promedio de 4.46 kms y 3) Cross con valor promedio de 5.20 kms para el conjunto de experimentos de la métrica 2, tal como se ve a continuación en sus valores promedios (tabla 4.3)

	CROSS	2-OPT	1-OPT	3-OPT	EXCHANGE	2OPT INTER
rpta2	5,49	4,31	4,11	5,07	5,07	5,07
rpta2	3,81	3,41	3,43	3,72	3,72	3,72
rpta2	3,07	3,07	3,35	3,07	3,07	3,07
rpta2	4,74	3,70	4,01	3,70	3,70	3,70
rpta2	4,27	4,27	4,27	4,27	4,48	4,17
rpta2	4,93	4,68	3,15	4,13	5,88	3,80
rpta2	6,57	4,89	4,76	4,21	4,21	4,21
rpta2	6,97	4,40	4,83	6,97	6,97	6,97
rpta2	6,94	4,55	5,40	6,95	6,95	6,95
rpta2	5,77	3,55	4,50	6,35	6,35	6,35
rpta2	6,37	4,16	4,00	4,72	4,72	4,72
rpta2	4,26	5,53	4,27	7,88	7,45	7,79
rpta2	4,94	4,59	4,14	4,89	7,50	5,39
rpta2	7,73	4,60	3,83	4,71	7,73	7,18
rpta2	4,04	3,50	4,35	6,25	4,14	6,13
rpta2	4,38	4,98	5,43	6,97	4,30	6,48
rpta2	6,23	5,27	4,55	6,83	7,55	6,69
rpta2	5,40	5,76	4,25	7,31	7,57	4,87
rpta2	4,50	5,15	4,28	4,97	6,52	6,39
rpta2	3,64	4,90	3,27	5,50	4,32	4,60
Promedio	5,20	4,46	4,21	5,42	5,61	5,41
Desv	1,10	0,75	0,64	1,45	1,62	1,43

Tabla 4.3 Valor Promedio, desviación y mapa de calor de métrica o característica 2 respecto a los operadores.

Por tanto, es posible pensar en una secuencia: 1) Cross, 2) 1-Opt, 3) 2-Opt (Intra-rutas) y 4) 2-Opt (Inter-rutas) sea un indicativo de guía para mejorar la calidad de la solución de la función objetivo. Debido a que Cross es el operador que más disminuye el promedio de compactación por ruta, medido por ángulo. 1-Opt y 2-Opt (Intra-rutas) son consecutivamente los operadores que mejor disminuyen el promedio de compactación por ruta, al igual que más disminuyen el valor de mayor distancia entre dos clientes conectados por ruta. Por último, el operador 2-Opt (Inter-rutas) se escogió por estar en la secuencia de operadores que disminuyen el promedio de compactación por ruta.

4.3 Evaluación de la solución propuesta.

Para analizar el efecto de la secuencia de operadores seleccionadas, se lleva a cabo una comparación entre los resultados de las soluciones denominadas “mala”, “no tan buena” y “buena”, en relación con la solución que utiliza la siguiente secuencia de operadores: 1). Cross, 2).1-Opt, 3).2-Opt (Intra-rutas) y 4). 2-Opt (Inter-rutas), que fueron aquellos escogidos de acuerdo con los resultados de los experimentos previos, comparado con los Operadores: 1) Exchange, 2) 1-Opt, 3) 3-Opt y 4) 2-Opt (inter-rutas). Lo que se hace es intercambiar dos operadores como son Cross (Inter-rutas) y 2-Opt (Intra-rutas) que dieron los mejores resultados en la disminución de la métrica 8 (Promedio de compactación por ruta, medido por ángulo), por otros dos operadores: 3-Opt (intra-ruta) y Exchange (inter-rutas)

De ahora en adelante al conjunto de operadores escogidos de acuerdo con los experimentos lo llamaremos operadores “escogidos” y al otro conjunto de operadores que excluyen Cross y 2-Opt lo llamaremos operadores “no escogidos”.

Ejemplo: Sean las soluciones obtenidas usando el modelo matemático por medio de programación entera-mixta en el solver de Gurobi, se obtienen los resultados de la evaluación de la función objetivo de cada solución calculado en litros, seguido de las rutas que representa esa evaluación (Imagen 4.2):

```
Solucion Mala: 9.187753787610255
[[0, 11, 13, 0], [0, 8, 6, 10, 15, 5, 0], [0, 9, 7, 2, 3, 4, 0], [0, 12, 1, 14, 0]]
Solucion no tan buena: 9.102091461106433
[[0, 11, 13, 0], [0, 8, 6, 10, 15, 5, 0], [0, 9, 7, 14, 0], [0, 12, 1, 2, 3, 4, 0]]
Solucion buena: 4.47622885660531
[[0, 8, 6, 2, 3, 4, 0], [0, 9, 11, 13, 12, 1, 0], [0, 10, 15, 5, 7, 14, 0]]
```

Imagen 4.2: Soluciones obtenidas por el Modelo de programación lineal entera mixta en Gurobi

Ahora, para este ejemplo se analizan las soluciones obtenidas usando el algoritmo de búsqueda local guiada por medio del software de Python con los operadores “no escogidos” obteniendo como salida del algoritmo: Las rutas de la solución final que se obtiene después de las 10 iteraciones para esos operadores, los vehículos que se seleccionan para dichas rutas, y el valor de la evaluación de la función objetivo en litros para esa operación, comparado con el resultado de las rutas de la solución inicial (solución “mala”) que se ve al final de la siguiente imagen:

```
Mejor solución encontrada: [[0, 11, 13, 0], [0, 8, 6, 15, 10, 5, 0], [0, 12, 1, 14, 0]]
Vehiculos asignados: [1, 4, 4]
Consumo mínimo: 7.602866085500336
Solución original: 9.187753787610255
Lista original: [[0, 11, 13, 0], [0, 8, 6, 10, 15, 5, 0], [0, 9, 7, 2, 3, 4, 0], [0, 12, 1, 14, 0]]
```

Imagen 4.2: soluciones obtenidas por el algoritmo de búsqueda local guiada con los operadores “no escogidos”.

El experimento con las mismas instancias y ahora con los operadores “escogidos” entrega las mismas respuestas de la imagen anterior, pero con los valores de los operadores “escogidos” en la siguiente imagen:

```
Mejor solución encontrada: [[0, 11, 13, 0], [0, 8, 6, 15, 10, 5, 0], [0, 9, 7, 2, 3, 4, 0]]
Vehiculos asignados: [1, 4, 4]
Consumo mínimo: 8.50237700031447
Solución original: 9.187753787610255
Lista original: [[0, 11, 13, 0], [0, 8, 6, 10, 15, 5, 0], [0, 9, 7, 2, 3, 4, 0], [0, 12, 1, 14, 0]]
```

Imagen 4.3: soluciones obtenidas por el algoritmo de búsqueda local guiada con los operadores “escogidos”.

De acuerdo con lo anterior se obtienen 4 valores respuestas de la función objetivo:

Solución mala es: 9.18 litros

Solución no tan buena es: 9.10 litros

Solución buena es: 4.47 litros

Solución con los operadores no escogidos es: 7.60 litros

Solución con los operadores escogidos es: 8.50 litros

Se mirará el porcentaje de diferencia entre los valores de la función objetivo para la solución “mala”, respecto a las soluciones “buena” y “no tan buena” usando el solver de Gurobi, al igual que respecto a las soluciones de los operadores “escogidos” y “no escogidos” usando el algoritmo de búsqueda local.

Considerando lo anterior se obtiene que el valor evaluado de la función objetivo en la solución etiquetada como “no tan buena” es 0.87 % mejor que la solución etiquetada como “mala”; el valor de la función objetivo de la solución con los operadores “escogidos” es 20.78 % mejor que la solución etiquetada como “mala”, el valor de la función objetivo de la solución con los operadores “no escogidos” es 8 % mejor que el valor de la solución etiquetada como “mala” y el valor de la solución etiquetada como “buena” es 105.8 % mejor que el valor de la función objetivo de la solución mala.

Una tabla resumen de 20 experimentos ofrece los siguientes resultados al usar el mencionado algoritmo de búsqueda local guiada con los operadores “escogidos” y “no escogidos” así como las soluciones ofrecidas usando programación entera y mixta por medio del solver de Gurobi y etiquetadas de acuerdo con la calidad en “buena”, “no tan buena” y “mala”.

Solucion No Tan Buena	Solución Buena	Solución Op. No Escogidos	Solución Op. Escogidos
0%	42%	7%	28%
0%	11%	0%	0%
0%	67%	1%	23%
0%	42%	0%	0%
78%	137%	18%	96%
158%	158%	6%	61%
15%	43%	15%	34%
50%	77%	33%	35%
10%	121%	49%	70%
12%	27%	0%	16%
46%	181%	10%	100%
0%	43%	15%	25%
0%	2%	0%	0%
0%	5%	0%	6%
33%	144%	33%	94%
28%	51%	5%	51%
0%	14%	0%	1%
20%	140%	20%	63%
24%	39%	0%	39%
0%	8%	0%	0%

Promedio	24%	68%	11%	37%
Desviación	38%	58%	14%	34%

Tabla 4.4 Valor Promedio, desviación y mapa de calor de las soluciones comparadas.

De acuerdo con lo presentado se ve que los valores de la función objetivo de la solución etiquetada como “buena” logra tener en promedio soluciones 68% mejores respecto a los valores de la solución etiquetada como “mala”; por otra parte, el valor de las soluciones

con los 4 operadores escogidos obtiene un valor promedio de 37 % mejor respecto a los valores de la solución etiquetada como “mala”, mientras que el valor de la solución para aquellos operadores “no escogidos” ofrece un valor promedio de 11 % mejor respecto a la solución etiquetada como “mala”, siendo los operadores “escogidos” mejores que los operadores “No Escogidos”

Analizando lo anterior, es posible inferir que la secuencia de operadores definidas por el análisis de las características del árbol de decisiones sirve como guía para la obtención de mejores soluciones; sin embargo, es cierto que siguen siendo valores distantes a la solución “buena”, puesto que la solución “buena” mejoraba en promedio un 68% de las otras soluciones, mientras que la solución con los operadores “escogidos” mejoraba en un 37% de las soluciones, siendo más cercana aún a la solución “no tan buena” que mejoraba la solución en un promedio de 24 %; por lo que es necesario pensar en un análisis con un mayor número de operadores o una mayor cantidad de iteraciones para acercarse a los valores de la solución “buena”.

Capítulo 5: Conclusiones y recomendaciones

Este trabajo tuvo como principal objetivo desarrollar un algoritmo de búsqueda local guiada para dar solución al problema de minimización en la huella de carbono. Desarrollando el enfoque desde una visión más realista, se considera una flota heterogénea.

Para lograr el objetivo del estudio, se partió de la definición y construcción del modelo matemático para el problema de optimización de ruteo de vehículo de las flotas heterogéneas (Fleet Size and Mix-VRP) propuestos por Gheysens y col. (1984). El problema parte de una variable dicotómica con tres índices, considerando si un vehículo k viaja de un arco i a un arco j , junto a una variable de flujo que especifica la cantidad de bienes que transporta un vehículo cuando va de un arco i a un arco j , para un número limitado de vehículos.

En cuanto a la propuesta de la función objetivo se ahondó en los estudios sobre el Green-VRP y qué factores ha tenido en cuenta la literatura para analizar la huella de carbono, seguido por la selección de la formulación llamada Pollution Routing Problem (PRP) del modelo usado por Eshtehadi, et al. (2017), con la función de Barth & Boriboonsomsin (2009). Esta función objetivo recoge factores a analizar como son: carga del vehículo, velocidad del vehículo y pendiente entre los diferentes arcos. Para determinar estos valores se estimó una velocidad constante del vehículo, un peso inicial aleatorio para cada uno de los diferentes vehículos y un terreno sin pendientes.

El modelo matemático construido a partir de la fusión de la función objetivo del modelo de Eshtehadi, et al. (2017) y del VRP en su variante con vehículos heterogéneos Fleet-size Mix-Vrp de Gheysens y col. (1984) no es lineal, por lo cual se resolvieron instancias para un número pequeño de nodos con el uso de un solver comercial para problemas de optimización lineal entera mixta "Gurobi", con el objetivo de conseguir tres soluciones etiquetadas como "mala", "no tan buena" y "buena" acorde con la calidad, que

Capítulo 5: Conclusiones y recomendaciones

65

posteriormente sirvió de insumo para definir las características que diferencian una “buena” solución de las otras dos soluciones.

Se exploró desde la literatura la implementación de diferentes heurísticas basadas en búsqueda local, para perturbaciones entre rutas y dentro de una ruta. Es decir, se conceptualiza un conjunto de movimientos u operadores de búsqueda local que permitan perturbar una solución para mejorarla en un espacio de búsqueda definida, considerando una posible complementariedad. De esta forma, se seleccionaron una serie de operadores como: Cross, Exchange, 2-Opt para intercambiar nodos entre diferentes rutas, y 1-Opt, 2-Opt y 3-opt para intercambiar nodos en la misma ruta. Un movimiento como Cross permite el intercambio de un subconjunto de rutas, un operador como Exchange permite el intercambio de arcos entre rutas, mientras un operador como 2-Opt permite el intercambio de dos nodos por cada ruta, lo que se entiende como un buen complemento de operadores, mientras que los operadores 1-Opt, 2-Opt y 3-Opt entre rutas operan sobre conjunto de intercambios de 1, 2 y 3 nodos en la misma ruta.

El siguiente propósito fue fijar una serie de métricas que permitan la caracterización de las soluciones anteriormente mencionadas del problema acorde con su calidad.

Las métricas implementadas son las mismas diez definidas por Arnold & Sorensen (2019) para solucionar un algoritmo de búsqueda local guiada en el artículo “*Efficiently Solving Very Large Scale Routing Problems*”; estas métricas tratan de definir qué características pueden tener en común las buenas soluciones y para esto se asume que una buena solución debe tener características como arcos pequeños entre el cliente y el depósito, las rutas deben estar separadas entre ellas, pero bastante unidas entre sí mismas, el número de clientes se espera que sea homogéneo entre rutas y cada ruta debe ser compacta y con pocas intersecciones.

Para lograr dicho propósito de caracterizar las soluciones del problema acorde con la calidad considerando tales métricas se genera un conjunto de datos con los valores de cada métrica y la solución que representa para un conjunto de instancias. Una vez hecho eso se hace un preprocesamiento de los datos y se escoge como mejor herramienta en la caracterización de las métricas un árbol de decisiones CART.

Capítulo 5: Conclusiones y recomendaciones

66

Al desplegar un árbol de decisiones con variable de respuesta (soluciones al modelo) y variables predictoras (métricas construidas) se obtiene como respuesta que cuatro métricas permitían diferenciar las tres soluciones con una precisión mayor al 75%. Tales métricas fueron: La compactación de la ruta medida por el ángulo, la compactación de la ruta medida por el ancho, la distancia máxima entre dos clientes conectados por una ruta y la profundidad de la ruta. Lo que se encontró es coherente con los análisis realizados por Arndol & Sorensen (2019), así como Lucas et, al (2019), a pesar de que la función de minimización es diferente, ya que las métricas que mejor permiten guiar el modelo son las métricas asociadas a la compactación de la ruta. Estas métricas luego se usaron como guía para mejorar la solución del problema de VRP en el algoritmo de Búsqueda Local Guiada para la obtención de una solución aceptable al modelo de minimización de las emisiones equivalentes de CO2 a través de la minimización del consumo de combustible en el transporte de mercancías para flotas de vehículos heterogéneos.

Posteriormente se trabajó en la construcción del algoritmo de búsqueda local guiada, el cuál siguió las reglas de las respuestas del árbol de decisiones en cuanto a las características asociadas con una “buena” solución, definiendo la secuencia de operadores como 1). Cross, 2).1-Opt, 3).2-Opt (Intra-rutas) y 4). 2-Opt (Inter-rutas), sobre la que se va a guiar el algoritmo, siendo estos los operadores que mejor minimizaban las métricas de: la compactación de la ruta medida por el ángulo, la compactación de la ruta medida por el ancho, la distancia máxima entre dos clientes conectados por una ruta y la profundidad de la ruta.

Por último, se evaluó esta propuesta de Algoritmo de Búsqueda Local Guiado usando los operadores con la secuencia 1). Cross, 2).1-Opt, 3).2-Opt (Intra-rutas) y 4). 2-Opt (Inter-rutas) que mejor respuesta ofrecen en la caracterización de las métricas entregadas por el árbol de decisiones respecto a las soluciones calculadas a partir del solver de Gurobi, etiquetadas como “mala”, “no tan buena” y “buena”, así como frente a un algoritmo de búsqueda local guiada con la secuencia 1) Exchange, 2) 1-Opt, 3) 3-Opt y 4) 2-Opt (inter-rutas) alterando los operadores Cross y 2-Opt (Intra-rutas) por Exchange y 3-Opt (intra-rutas) en la mejor secuencia encontrada.

Los resultados son prometedores en cuanto a que, comparada la secuencia de estos operadores que mejor desempeño tenían respecto a la caracterización del árbol frente a la otra secuencia de operadores, arrojó que la secuencia de operadores definida dio

mejores resultados (valores 37% mejores respecto a la solución “mala”), un valor mayor a la secuencia de operadores “no escogidos” que obtuvo valores un 11% mejores respecto a la solución “mala”; sin embargo, aún está lejos de la solución “buena” que tuvo mejoras con un valor promedio de un 67% respecto a la solución “mala”, y está más cercana a la solución “no tan buena” con valores promedio de mejora de un 24% respecto a la solución “mala”.

5.1 Recomendaciones

Este trabajo puede ser ampliado, incrementando nuevas métricas que permitan analizar el consumo de combustible.

También hay que tener en cuenta que el modelo queda limitado frente a las suposiciones que realiza, como son: considerar una velocidad óptima o carreteras planas y sin tráfico, situaciones que no se acercan a la realidad colombiana, ya que debido a la geografía del país, las características de sus carreteras están lejos de ser planas en todos los sectores y el tráfico suele ser denso; además no se considera el tiempo que es un factor importante en la toma de decisiones.

Además, se debe analizar los modelos con un mayor número de clientes pues al observar la compactación de la ruta medida por el ángulo, se ve mucha diferencia frente a las respuestas del mismo cálculo de Arnold & Sorensen (2019).

Requiere ser ampliado el estudio de los operadores para escoger ya que se ve diferencia respecto a la solución óptima, así como repensar la forma en que se define la guía del algoritmo, puesto que un algoritmo de guía con utilidad y penalizaciones puede resultar más eficiente en la búsqueda de una solución óptima; igualmente, experimentar con diferente número de iteraciones que pueden arrojar mejores soluciones.

Bibliografía

Alpaydin, E. (2014). Introduction to Machine Learning Second Edition. In *Methods in molecular biology (Clifton, N.J.)* (Vol. 1107).

<http://www.ncbi.nlm.nih.gov/pubmed/24272434>

Arnold, F., & Gendreau, M. (2017). *Efficiently Solving Very Large Scale Routing Problems Efficiently Solving Very Large Scale Routing Problems. CIRRELT-2017-75.*

Arnold, F., Gendreau, M., & Sörensen, K. (2019). Efficiently solving very large-scale routing problems. *Computers and Operations Research*, *107*, 32–42.

<https://doi.org/10.1016/j.cor.2019.03.006>

Arnold, F., & Sörensen, K. (2019). What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. *Computers and Operations Research*,

106, 280–288. <https://doi.org/10.1016/j.cor.2018.02.007>

Astarita, C. (2013). El Gran Juego de las Tierras Raras. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>

Baldacci, R., Battarra, M., & Vigo, D. (2008a). Routing a heterogeneous fleet of vehicles. In *Operations Research/ Computer Science Interfaces Series* (Vol. 43, pp. 3–27).

https://doi.org/10.1007/978-0-387-77778-8_1

- Baldacci, R., Battarra, M., & Vigo, D. (2008b). The Vehicle Routing Problem: Latest Advances and New Challenges. In *Operations Research/ Computer Science Interfaces Series* (Vol. 43).
- Bautista-Valhondo, J. (2003). *Adaptación de la heurística LIN-KERNINGHAM para la resolución de problemas de diseño de itinerarios : Secuenciación de unidades con restricciones múltiples* View project. October.
<https://doi.org/10.13140/RG.2.2.17660.33929>
- Bektaş, T., & Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B: Methodological*, 45(8), 1232–1250.
<https://doi.org/10.1016/j.trb.2011.02.004>
- Blum, C. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison Metaheuristics in Combinatorial Optimization. In *ACM Computing Surveys* (Vol. 35, Issue 3).
- Climatico, G. I. de E. sobre el C. (2019). *Maintaining Humanity's Life Support Systems in the 21st Century Scientific Consensus on*. http://consensusforaction.stanford.edu/scientific-consensus/consensus_english.pdf
- Corzo, J. A. (2019). *Análisis Multivariado Aplicado con aplicaciones en R Notas de clase*.
- Daza, J. M., Montoya, J. R., & Narducci, F. (2010). *RESOLUCIÓN DEL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS CON LIMITACIONES DE CAPACIDAD UTILIZANDO UN PROCEDIMIENTO METAHEURÍSTICO DE DOS FASES*.
- Eshtehadi, R., Fathian, M., & Demir, E. (2017). Robust solutions to the pollution-routing

- problem with demand and travel time uncertainty. *Transportation Research Part D: Transport and Environment*, 51, 351–363. <https://doi.org/10.1016/j.trd.2017.01.003>
- Feng, Y., Zhang, R. Q., & Jia, G. (2017). Vehicle Routing Problems with Fuel Consumption and Stochastic Travel Speeds. *Mathematical Problems in Engineering*, 2017. <https://doi.org/10.1155/2017/6329203>
- Figliozzi, M. (2010). Vehicle routing problem for emissions minimization. *Transportation Research Record*, 2197, 1–7. <https://doi.org/10.3141/2197-01>
- Fisher, M. (1995). Vehicle routing. *Handbooks in Operations Research and Management Science*, 8(C), 1–33. [https://doi.org/10.1016/S0927-0507\(05\)80105-7](https://doi.org/10.1016/S0927-0507(05)80105-7)
- Ge, Z., Song, Z., Ding, S. X., & Huang, B. (2017). Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*, 5, 20590–20616. <https://doi.org/10.1109/ACCESS.2017.2756872>
- Gendreau, M., & Potvin, J.-Y. (2010). Variable Neighborhood search (chapter). *Handbook of Metaheuristics*, 146(September 2010), 648. <https://doi.org/10.1007/978-1-4419-1665-5>
- Ha, S. H. (2010). Behavioral assessment of recoverable credit of retailer's customers. *Information Sciences*, 180(19), 3703–3717. <https://doi.org/10.1016/j.ins.2010.06.012>
- Hansen, P., Mladenovic, N., Andrés Moreno Pérez, J., Sánchez, F., & Cruz de Tenerife, S. (2003). Variable Neighbourhood Search. *Revista Iberoamericana de Inteligencia Artificial*. No, 19, 77–92. <http://www.aepia.org/revista>
- Hillier, F. S., & Lieberman, G. J. (2010). *Introducción a la Investigación de Operaciones*

(M. G. Hill (ed.); Novena Edi).

Hines, W. W., & Montgomery, D. C. (1993). Probabilidad y estadística para ingeniería y administración. In *Applied Statistics* (Vol. 22, Issue 1).

Huang, B., Thorne, P. W., Banzon, V. F., Boyer, T., Chepurin, G., Lawrimore, J. H., Menne, M. J., Smith, T. M., Vose, R. S., & Zhang, H. M. (2017). Extended reconstructed Sea surface temperature, Version 5 (ERSSTv5): Upgrades, validations, and intercomparisons. *Journal of Climate*, *30*(20), 8179–8205.
<https://doi.org/10.1175/JCLI-D-16-0836.1>

Huemer, M. (2011). *Heuristics for the vehicle routing problem with multiple deliverymen*. August.

Irnich, S., Toth, P., & Vigo, D. (2014). Chapter 1: The Family of Vehicle Routing Problems. In *Vehicle Routing* (pp. 1–33). Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611973594.ch1>

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. In *European Journal of Operational Research* (Vol. 249, Issue 1, pp. 1–21). Elsevier. <https://doi.org/10.1016/j.ejor.2015.07.020>

Kramer, R., Subramanian, A., Vidal, T., & Cabral, L. D. A. F. (2015). A matheuristic approach for the Pollution-Routing Problem. *European Journal of Operational Research*, *243*(2), 523–539. <https://doi.org/10.1016/j.ejor.2014.12.009>

Larose, D. T., & Larose, C. D. (2014). Discovering Knowledge in Data. In *Discovering Knowledge in Data*. <https://doi.org/10.1002/9781118874059>

- Leung, S. C. H., Zheng, J., Zhang, D., & Zhou, X. (2010). Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services and Manufacturing Journal*, 22(1–2), 61–82. <https://doi.org/10.1007/s10696-010-9061-4>
- Li, X. Y., Tian, P., & Leung, S. C. H. (2009). An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems. *Journal of the Operational Research Society*, 60(7), 1012–1025. <https://doi.org/10.1057/palgrave.jors.2602644>
- Lucas, F., Billot, R., & Sevaux, M. (2019). A comment on “What makes a VRP solution good? The generation of problem-specific knowledge for heuristics.” *Computers and Operations Research*, 110, 130–134. <https://doi.org/10.1016/j.cor.2019.05.025>
- Medina, L. B. R., Rotta, E. C. G. La, & Castro, J. A. O. (2011). Una Revisión al Estado del Arte del Problema de Ruteo de Vehículos: Evolución Histórica Y Métodos De Solución. *Ingeniería*, 16(2), 35–55. <http://revistas.udistrital.edu.co/ojs/index.php/reving/article/view/3832>
- Molina, J. C. (2012). *Problema VRP con flota heterogénea*.
- Muñoz-Villamizar, Montoya-Torres, & Caceres Cruz, J. (2020). A SIMULATION-BASED ALGORITHM FOR THE INTEGRATED LOCATION AND ROUTING PROBLEM IN URBAN LOGISTICS. *Suparyanto Dan Rosad (2015, 5(3), 248–253*.
- Nikfarjam, A., Moosavi, A., Rezaei, N., & Ebrahimnejad, S. (2019). A green vehicle routing problem with time windows considering the heterogeneous fleet of vehicles:

two metaheuristic algorithms. *European J. of Industrial Engineering*, 13(4), 507.
<https://doi.org/10.1504/ejie.2019.10022249>

Olivera, A. (2004). *Heurísticas para Problemas de Ruteo de Vehículos*.

Olson, R. S., & Moore, J. H. (2016). *Olson and Moore - TPOT A Tree-based Pipeline Optimization Tool for .pdf*. 163–173.

Pessoa, A., Sadykov, R., & Uchoa, E. (2018). Enhanced Branch-Cut-and-Price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270(2), 530–543. <https://doi.org/10.1016/j.ejor.2018.04.009>

Posada, J., & González, C. (2013). Fuel Consumption in Vehicles for Highway Transportation - Predictive Models. *Revista Ingenierías Universidad de Medellín*, 12(23), 35–46. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-33242013000200004&lng=en&nrm=iso&tlng=es

Rezgui, D., Chaouachi Siala, J., Aggoune-Mtalaa, W., & Bouziri, H. (2019). Application of a variable neighborhood search algorithm to a fleet size and mix vehicle routing problem with electric modular vehicles. *Computers and Industrial Engineering*, 130(January 2018), 537–550. <https://doi.org/10.1016/j.cie.2019.03.001>

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472. <https://doi.org/10.1287/trsc.1050.0135>

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. *Handbook of Heuristics*, 2–2(January), 791–808. https://doi.org/10.1007/978-3-319-07124-4_4

- Suarez-Chilma, V. F., Sarache, W. A., & Costa, Y. J. (2018). Una Solución al Enrutamiento de Vehículos en Ciudades Montañas Considerando Aspectos Ambientales y Económicos. *Información Tecnológica*, 29(3), 3–14. <https://doi.org/10.4067/s0718-07642018000300003>
- Suzuki, Y. (2011). A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, 16(1), 73–77. <https://doi.org/10.1016/j.trd.2010.08.003>
- TAHA, H. A. (2012). *I Nvestigación*.
- Toth, P., & Vigo, D. (2002). 1. An Overview of Vehicle Routing Problems. In *The Vehicle Routing Problem* (pp. 1–26). Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898718515.ch1>
- Trujillano, J., Sarria, A., Esquerda, A., Mariona, B., Matilde, P., & Jaume, M. (2008). Aproximación a la metodología basada en árboles de decisión (CART). Mortalidad hospitalaria del infarto agudo de miocardio. *Gaceta Sanitaria*, 22(SUPPL. 1), 65–70. <https://doi.org/10.1157/13115113>
- Voudouris, C., Tsang, E. P. K., & Alsheddy, A. (2003). *Guided Local Search*. <https://doi.org/10.1007/978-1-4419-1665-5>
- Wang, S., Lu, Z., Wei, L., Ji, G., & Yang, J. (2016). Fitness-scaling adaptive genetic algorithm with local search for solving the Multiple Depot Vehicle Routing Problem. *SIMULATION*, 92(7), 601–616. <https://doi.org/10.1177/0037549715603481>
- Wolsey, L. A. (2008). Mixed Integer Programming. *Wiley Encyclopedia of Computer*

Science and Engineering, 1, 1–10. <https://doi.org/10.1002/9780470050118.ecse244>

Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., & Steinberg, D. (2008). Top 10 algorithms in data mining. In *Knowledge and Information Systems* (Vol. 14, Issue 1). <https://doi.org/10.1007/s10115-007-0114-2>