

Solución al Problema de Ruteo de Vehículos de Flota Fija Heterogénea con Restricciones de Carga Bidimensional (2L-HFFVRP) Mediante la Metaheurística de Algoritmo Genético.

Ingrith Marcela Mendoza Mendoza

Carlos Ernesto Mendoza Giménez

Trabajo de Grado para Optar el título de Ingeniero(a) Industrial

Director

Julio César Camacho Pinto

Magíster en Ingeniería Industrial

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2023

Dedicatoria

A mis padres, Irma Teresa y Hermes; y a mis hermanos, Andrea, Daniela, Natalia, Liseth, Leandro, Patricia, Yurany y Jenny, por alegrarse conmigo con este logro y apoyarme incondicionalmente.

~ Ingrith M

A mi madre, Elena Giménez.

~ Carlos.

Agradecimientos

A Dios por iluminarnos con su Santo Espíritu.

Al profesor Julio por creer en nosotros y guiarnos durante el desarrollo del proyecto.

A la Cooperativa de Impresores del Oriente por brindarnos la oportunidad de colaborar con ellos.

A los profesores del Grupo OPALO por su disposición para responder nuestras preguntas.

A los autores por el excelente equipo conformado y por demostrarse que el éxito es la suma de pequeños esfuerzos repetidos un día sí y al otro también.

Tabla de contenido

Introducción	16
1. Objetivos.....	20
1.1. Objetivo General	20
1.2. Objetivos Específicos	20
2. Generalidades de la Investigación	21
2.1. Planteamiento del Problema.....	21
2.2. Metodología	23
2.2.1. <i>Metodología para el Diagnostico</i>	23
2.2.2. <i>Fase I. Definición del Problema de Interés y Recolección de Datos Relevantes ...</i>	25
2.2.3. <i>Fase II. Formulación del Modelo Matemático</i>	26
2.2.4. <i>Fase III. Desarrollo un Procedimiento de Computación para Derivar una Solución del Problema a Partir del Modelo Propuesto</i>	27
2.2.5. <i>Fase IV. Prueba del Modelo</i>	27
2.2.6. <i>Fase V. Síntesis de Resultados</i>	27
3. Revisión de Literatura y Análisis Bibliométrico	28
3.1. Análisis Bibliométrico.....	28
3.1.1. <i>Año de Publicación</i>	29
3.1.2. <i>Países</i>	30
3.1.3. <i>Autores</i>	30

3.1.4.	<i>Áreas de Investigación</i>	31
3.2.	Revisión de Literatura	32
4.	Marco de Referencia.....	33
4.1.	Marco de Antecedentes	33
4.2.	Marco Teórico	35
4.2.1.	<i>Optimización Matemática</i>	35
4.2.2.	<i>Modelos Matemáticos</i>	36
4.2.3.	<i>Modelos de Programación Lineal.</i>	37
4.2.4.	<i>Modelos de Programación No Lineal.</i>	37
4.2.5.	<i>Optimización Combinatoria</i>	37
4.2.6.	<i>Complejidad Computacional</i>	38
4.2.7.	<i>Problema de Ruteo de Vehículos (VRP)</i>	39
4.2.8.	<i>Problema de Ruteo de Vehículos con Restricción de Capacidad (CVRP)</i>	39
4.2.9.	<i>Métodos de Solución para Problemas de Optimización Combinatoria</i>	40
4.2.10.	<i>Problema con Restricción de Carga Bidimensional (2L)</i>	46
4.2.11.	<i>Métodos de Solución para el Problema 2L</i>	47
4.2.12.	<i>APIs de Google Maps Platform</i>	48
4.2.13.	<i>OpenStreetMap (OSM)</i>	50
4.2.14.	<i>OpenRouteService (ORS)</i>	51
4.2.15.	<i>Proceso de Descubrimiento de Conocimiento en Bases de Datos</i>	52

5.	Estrategia KDD para la Matriz de Distancia	56
5.1.	Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD).....	56
5.1.1.	Selección de los Datos	56
5.1.2.	<i>Preprocesamiento</i>	57
5.1.3.	<i>Transformación</i>	58
5.1.4.	<i>Minería de Datos</i>	58
5.1.5.	<i>Interpretación</i>	59
5.2.	Clientes Clave	62
5.3.	Matriz	64
5.3.1.	<i>Software de Servicio de Mapas</i>	65
5.3.2.	<i>Geolocalización</i>	66
5.3.3.	<i>Matriz</i>	67
6.	Modelo de Optimización para el Problema Ruteo de Vehículos de Flota Fija Heterogénea con Restricciones de Carga Bidimensional (2L-HFFVRP).....	67
6.1.	Definición Formal del Problema.....	69
6.2.	Modelo Matemático	70
7.	Descripción la Metaheurística Algoritmo Genético	76
7.1.	Generación y Codificación de la Población Inicial.....	77
7.2.	Evaluación de Aptitud (Fitness).....	77
7.3.	Selección	78

7.3.1.	<i>Selección Elitista</i>	79
7.3.2.	<i>Selección por Ruleta.</i>	79
7.3.3.	<i>Selección por Torneo.</i>	80
7.3.4.	<i>Muestreo Estocástico Universal</i>	81
7.4.	Cruce o Recombinación	81
7.4.1.	<i>Cruce de 1 Punto</i>	81
7.4.2.	<i>Cruce de 2 Puntos</i>	82
7.4.3.	<i>Cruce Uniforme</i>	82
7.5.	Mutación.....	83
8.	Diseño del Algoritmo Genético para la Solución del 2L-HFFVRP	84
8.1.	Etapa 1. Programa de Ingreso de Datos	84
8.1.1.	<i>Inicialización y Configuración</i>	86
8.1.2.	<i>Carga de la Lista de Clientes</i>	86
8.1.3.	<i>Añadir un Nuevo Cliente</i>	87
8.1.4.	<i>Selección y Configuración de Clientes</i>	88
8.1.5.	<i>Finalización de la Selección y Ruteo</i>	89
8.2.	Etapa 2. Matriz de Distancia para Clientes Nuevos	90
8.3.	Etapa 3. Algoritmo Genético para el Problema de Ruteo de Vehículos (HFFVRP)	91
8.3.1.	<i>Parámetros de Entrada</i>	92
8.3.2.	<i>Parámetros de Instancias</i>	92

8.3.3.	<i>Representación de los Cromosomas</i>	93
8.3.4.	<i>Generación de la Población Inicial</i>	94
8.3.5.	<i>Evaluación de Aptitud</i>	95
8.3.6.	<i>Creación de Nuevas Generaciones</i>	95
8.3.7.	<i>Criterio de Parada</i>	97
8.4.	Fase 4. Algoritmo de Primera Posición para el Problema de Carga Bidimensional	98
8.4.1.	<i>Inicialización</i>	¡Error! Marcador no definido.
8.4.2.	<i>Empaquetamiento</i>	100
8.4.3.	<i>Iteración</i>	100
8.4.4.	<i>Solución</i>	101
9.	Validación del Algoritmo	101
9.1.	Validación del Algoritmo Genético del HFFVRP	101
9.2.	Validación del Algoritmo de Primera Posición del Problema 2L	102
10.	Resultados de la Validación.....	104
10.1.	Resultados de la Validación del Algoritmo Genético	104
10.1.1.	<i>Escenario 1: Visita el Depósito en la Mitad del Recorrido</i>	105
10.1.2.	<i>Escenario 2: Visita al Depósito al Inicio o al Final del Recorrido</i>	107
10.2.	Resultados de la Validación del Algoritmo de Primera Posición 2L	111
10.2.1.	<i>Instancia 1</i>	111
10.2.2.	<i>Instancia 2</i>	115

11. Conclusiones	119
12. Recomendaciones	121
Referencias Bibliográficas	123

Lista de Tablas

Tabla 1	Cumplimiento de objetivos.....	19
Tabla 2	Consideraciones tácticas presentadas por Bischoff y Ratcliff (1995).	46
Tabla 3	Ejemplo de repetición de tuplas.....	54
Tabla 4	Errores más frecuentes en el ingreso de datos.	58
Tabla 5	Características de Google Maps y OSM.....	65
Tabla 6	Clientes utilizados para el ejercicio.	65
Tabla 7	Diferencia de las distancias entre Google Maps y OSM.	66
Tabla 8	Conjunto de índices, parámetros y variables.	70
Tabla 9	Ejemplo de selección por ruleta.....	80
Tabla 10	Parámetros de entrada para la ejecución del algoritmo genético.....	92
Tabla 11	Parámetros de entrada adicionales para la ejecución del algoritmo genético.....	93
Tabla 12	Dimensiones de la flota de Coimpresores.....	98
Tabla 13	Dimensiones de las estibas de Coimpresores.....	100
Tabla 14	Instancias para la validación del algoritmo genético del HFFVRP.	103
Tabla 15	Instancias para la validación del algoritmo de primera posición del 2L.....	104
Tabla 16	Resultado de las instancias 1-26 para el escenario 1.	105
Tabla 17	Resultados de las instancias 27-44 para el escenario 1.....	106
Tabla 18	Resultado de las instancias 1-7 para el escenario 2.	107
Tabla 19	Resultado de las instancias 9-32 para el escenario 2.	108
Tabla 21	Ahorros potenciales en pesos colombianos de los escenarios propuestos.....	111
Tabla 22	Consolidado de las cantidades máximas de estibas para cada vehículo.	114

Lista de Figuras

Figura 3	Mapa mental de los modelos de interés.....	29
Figura 4	Cantidad de artículos por año.....	29
Figura 5	Países.....	30
Figura 6	Autores.	31
Figura 7	Áreas de investigación.....	31
Figura 8	Proceso de Descubrimiento de Conocimiento en Bases de Datos.	53
Figura 9	Promedio de entregas por recorrido al mes.....	60
Figura 10	Cantidad de facturas por recorrido y por bodega.	61
Figura 11	Porcentaje de incumplimiento por recorrido.....	62
Figura 12	Ubicación de clientes fijos identificados.....	64
Figura 13	Interfaz de Google Maps y OSM.	66
Figura 14	Ejemplo del problema 2L-HFFVRP con carga secuencial.	68
Figura 15	Posiciones relativas de los artículos de dos clientes.....	76
Figura 16	Diagrama de flujo del algoritmo genético.....	78
Figura 17	Ejemplo de cruce de 1 punto.....	82
Figura 18	Ejemplo de cruce de 2 puntos.....	82
Figura 19	Ejemplo de cruce uniforme.	83
Figura 20	Diagrama de flujo del prototipo propuesto para la solución del 2L-HFFVRP.	85
Figura 21	Ventana principal del prototipo de la aplicación.....	86
Figura 22	Lista de clientes cargada en el prototipo de la aplicación.	87
Figura 23	Ventana para añadir clientes nuevos al prototipo de la aplicación.....	88
Figura 24	Ventana de ingreso de datos de los pedidos al prototipo de la aplicación.	89

Figura 25 Representación de un cromosoma.....	94
Figura 26 Diagrama de flujo del algoritmo de primera posición.....	99
Figura 27 Resultado del prototipo propuesto para el problema de ruteo de vehículos.....	107
Figura 28 Error relativo de las distancias del escenario 2 con respecto a la distancia real	109
Figura 29 Comparación entre las distancias propuesta respecto a la distancia real.	110
Figura 30 Cantidad máxima de estibas para el vehículo NKR II.	113
Figura 31 Resultados del prototipo de programa propuesto.	115
Figura 32 Priorización en el orden de carga de las estibas.	117
Figura 33 Resultados del prototipo de programa propuesto.	118
Figura 34 Cantidad máxima de estibas priorizando las estibas pequeñas	119

Lista de Apéndices

Los apéndices están adjuntos y puede visualizarlos en base de datos de la biblioteca UIS.

Apéndice A. Diagnóstico del Proceso de Distribución de la Empresa Cooperativa de Impresores y Papeleros del Oriente.

Apéndice B. Matriz de distancia de los clientes clave

Apéndice C. Código del prototipo de programa propuesto

Apéndice D. Artículo de carácter publicable

Resumen

Título del proyecto: Solución al problema de Ruteo de Vehículos de Flota Fija Heterogénea con restricción de Carga Bidimensional (2L-HFFVRP) mediante la metaheurística de Algoritmo Genético*

Autores: Ingrith Marcela Mendoza Mendoza, Carlos Ernesto Mendoza Giménez**

Palabras claves: Ruteo de vehículos, flota fija heterogénea, carga bidimensional, algoritmo genético.

Descripción:

En el presente trabajo de investigación se plantea la metaheurística de algoritmo genético para abordar el problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional (2L-HFFVRP). La particularidad de esta restricción de carga bidimensional radica en la capacidad de rotación de los artículos en un ángulo de 90° y la carga de manera secuencial en el vehículo, lo que representa una variante que no ha sido estudiada del problema clásico de ruteo y tiene una amplia gama de aplicaciones en el campo de la logística. En este problema, cada cliente demanda un conjunto de artículos bidimensionales rectangulares y el objetivo es encontrar las rutas de entrega que minimicen la distancia total recorrida por una flota de vehículos con diferentes capacidades y superficies de carga bidimensional. Para dar solución al problema, se desarrolla un algoritmo genético para abordar el HFFVRP y un algoritmo de primera posición (First-fit) para abordar las restricciones de carga 2L. Como resultado, se crea un prototipo de programa con interfaz de usuario. Para la validación del modelo y algoritmo propuesto, se realizan escenarios e instancias, utilizando los datos reales proporcionados por la Cooperativa de Impresores del Oriente, una empresa del sector papelerero de la ciudad de Bucaramanga, que se encarga de atender al sector de la industria de la comunicación gráfica en la región oriente del país, suministrando materias primas e insumos a sus asociados. Los resultados obtenidos respaldan la efectividad de este enfoque: el modelo propuesto describe correctamente el problema, el algoritmo obtiene buenas soluciones factibles y demuestra su capacidad para resolver problemas a pequeña escala como a gran escala. Además, la heurística de empaquetamiento es una estrategia eficiente en la solución de este tipo de problemas de carga.

* Trabajo de Grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Estudio Industriales y Empresariales. Director: MA. Julio César Camacho Pinto.

Abstract

Title: Solution to the Heterogeneous Fixed Fleet Vehicle Routing problem with two-dimensional load restrictions (2L-HFFVRP) using the Genetic Algorithm metaheuristic *

Authors: Ingrith Marcela Mendoza Mendoza, Carlos Ernesto Mendoza Giménez**

Keywords: Vehicle routing, heterogeneous fixed fleet, two-dimensional load, genetic algorithm.

Description:

In this research work, a genetic algorithm is proposed to address the problem of routing heterogeneous fixed fleet vehicles with two-dimensional loading constraint (2L-HFFVRP). The particularity of this two-dimensional loading constraint resides in the ability to rotate the items at an angle of 90° and load them sequentially on the vehicle, which represents an unstudied variant of the classic routing problem and has a wide range of applications in logistics. In this problem, each customer demands a set of two-dimensional rectangular items and the objective is to find delivery routes that minimize the total distance traveled by a fleet of vehicles with different capacities and two-dimensional loading surfaces. To solve this problem, a genetic algorithm is developed to address the HFFVRP and a first-fit algorithm to address the bidimensional loading constraints. As a result, a program prototype with a graphical user interface is created. For the validation of the proposed model and algorithm, scenarios and instances are conducted, using the real data provided by the Cooperativa de Impresores del Oriente, a company in the paper sector of the city of Bucaramanga, which is responsible for serving the graphic communication industry in the eastern region of the country, supplying raw materials and inputs to its associates. The results obtained support the effectiveness of this approach: the proposed model correctly describes the problem; the algorithm obtains good feasible solutions and demonstrates its ability to solve small-scale and large-scale problems. Furthermore, the packing heuristic is an efficient strategy in solving this type of loading problems.

* Degree Project

** Faculty of Physic-Mechanical Engineering. School of Industrial and Business Studies. Director: MA. Julio César Camacho Pinto.

Introducción

En muchas organizaciones, la logística es un elemento fundamental a la hora de tomar decisiones. Cada sistema productivo necesita una manera eficiente de suministrar sus productos a sus clientes. La mayoría de las empresas requieren vehículos para realizar sus entregas, lo que significa que el uso eficiente de una flota de vehículos es la principal característica de casi todos los problemas de logística, ya que los costos de transporte y los tiempos de entrega tienen un impacto directo en la gestión de la cadena de suministro (Dey et al., 2021).

El problema de ruteo de vehículos (VRP) se considera uno de los factores más importantes tanto en la logística como en los sistemas de transporte de carga. Consiste en proponer la forma más rentable de entregar un conjunto de artículos a los clientes utilizando una flota de vehículos. Los costos más comunes asociados con este problema se encuentran relacionados con la distancia de recorrido (Braekers et al., 2016). En cuanto a lo que la composición de la flota se refiere, el problema de ruteo clásico, denominado como el problema de ruteo de vehículos capacitados (CVRP), considera vehículos con una misma capacidad limitada, mientras que el problema de ruteo de vehículos de flota heterogénea (HFVRP) considera vehículos con diferente capacidad (Koç et al., 2016). Se ha estudiado diferentes variantes de estos VRP de flota heterogénea. Por ejemplo, el problema de ruteo de vehículos de flota heterogénea (HFVRP) trata con un número ilimitado de vehículos, mientras que el problema de ruteo de flota fija heterogénea (HFFVRP) considera un número limitado de vehículos.

En las variantes de VRP mencionadas anteriormente, las demandas de los clientes se expresan en peso o volumen. Para estos casos, comprobar la factibilidad de una solución simplemente requiere asegurarse que la suma de las demandas de los clientes no supere la capacidad de carga del vehículo. No obstante, existe una variante derivada del contexto cotidiano

de las empresas que involucra el transporte de artículos con geometría rectangular, es decir, artículos que no pueden ser apilados ya sea debido a su peso o su fragilidad, como lo pueden llegar a ser los electrodomésticos. En todos estos casos, es necesario considerar restricciones adicionales para reflejar la carga bidimensional del problema, porque la forma en que estos artículos se cargan a los vehículos puede tener una influencia significativa sobre los costos de distribución (Côté et al., 2017). Dado que el problema de ruteo y problema de carga son ambos problemas NP-Hard, combinarlos lleva a un problema de optimización de extrema dificultad.

El problema de ruteo de vehículos con restricciones de carga bidimensional (2L-CVRP) es uno de los primeros que trata de integrar los problemas de carga y el ruteo de vehículos. En este problema, los clientes piden un conjunto de artículos bidimensionales que se asume cuentan con una configuración rectangular y los vehículos, homogéneos, cuentan con una superficie de carga bidimensional. En el contexto cotidiano, a diferencia del 2L-CVRP, las empresas cuentan con una flota heterogénea de vehículos, lo que les ofrece la flexibilidad necesaria para diseñar un plan de distribución más rentable. El problema de ruteo de vehículos de flota heterogénea con restricciones de carga bidimensional (2L-HFVRP) trata el 2L-CVRP con una flota de vehículos heterogénea e ilimitada. No obstante, dado que la mayoría de las empresas que tienen que entregar mercancías poseen una flota limitada de vehículos debido a las limitaciones de sus recursos, es crucial estudiar los problemas de ruteo que involucran flotas fijas heterogéneas y restricciones de carga. Hasta donde se sabe, no se conoce de investigaciones donde se haya abordado dicha variación del VRP, aunque sea un problema práctico en las empresas de transporte y logística del mundo real.

Así pues, esta investigación se centra en una variante específica del problema de ruteo de vehículos conocido como el problema de ruteo de vehículos de flota fija heterogénea considerando, además, las restricciones que se pueden presentar a la hora de cargar los vehículos por las

dimensiones de largo y ancho de los productos (2L-HFFVRP). Dicho estudio va a ser probado con datos reales proporcionados por la empresa Cooperativa de Impresores del Oriente (Coimpresores). Coimpresores es una Empresa del Sector de la Industria de la comunicación gráfica con carácter Cooperativo supervisada por la Superintendencia de Economía solidaria.

Actualmente la Cooperativa de Impresores del Oriente cuenta con 22 asociados, que son personas naturales, propietarias de empresas dedicadas a la comercialización y producción de elementos de papelería. Al considerarse su mercado como un sector transversal a todos los sectores de la economía, su crecimiento se está dando en niveles considerables, pero su estructura de distribución continúa siendo manejada por un recurso humano sin el dominio en temas de logística y distribución, generando problemas de suministro e insatisfacción de los clientes. A raíz de esto, se decidió realizar un estudio sobre el tema de la distribución y encontrar una herramienta que les permita hacer sus planes de entrega de modo más técnico y seguro, generando mayores oportunidades y mayor satisfacción de los clientes.

En el capítulo 2, se presenta el planteamiento del problema, el cual implica un diagnóstico exhaustivo de la situación actual del proceso de distribución de la empresa Coimpresores. Se contextualiza acerca de los avances y evolución que ha tenido este problema de ruteo hasta el presente planteamiento, mediante la revisión de literatura en el capítulo 3. Por medio del capítulo 5, se realiza el proceso de descubrimiento de conocimiento en bases de datos (estrategia KDD) de la información histórica de la empresa para la obtención de la matriz de distancia. Por otro lado, se propone el planteamiento del modelo matemático para la variación propuesta del problema de ruteo, en el capítulo 6, y se desarrolla la metaheurística de algoritmo genético en el capítulo 7 y 8. La metaheurística se valida numéricamente con los datos proporcionados por la empresa en el capítulo 9. Finalmente se presentan los resultados, las conclusiones y las recomendaciones de la

investigación en los capítulos 10, 11 y 12 respectivamente. A continuación, se presenta el cumplimiento de los objetivos propuestos en la investigación:

Tabla 1

Cumplimiento de objetivos.

Objetivo específico	Cumplimiento
Realizar una revisión de literatura contextualizando el problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional.	Capítulo 4
Aplicar las etapas del proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD) a los documentos proporcionados por la empresa Cooperativa de Impresores del Oriente y consolidar una base de datos de clientes con el objetivo de poder obtener una matriz de distancias.	Capítulo 6 Apéndice B
Formular un modelo de optimización matemática para el 2L-HFFVRP.	Capítulo 7
Diseñar un algoritmo genético para la solución del 2L-HFFVRP en el lenguaje de programación Python.	Capítulo 8 Capítulo 9 Apéndice C
Validar el modelo propuesto a través de diferentes instancias de prueba ajustadas a datos reales de la empresa.	Capítulo 10 Capítulo 11
Elaborar un artículo académico publicable en el que se documente la investigación y los resultados más relevantes obtenidos de ésta.	Apéndice D

1. Objetivos

1.1. Objetivo General

Solucionar el problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional (2L-HFFVRP) a partir de un modelo de optimización matemática utilizando la metaheurística de algoritmo genético, que permita mejorar la eficiencia en la programación de rutas de entrega de producto terminado.

1.2. Objetivos Específicos

- Realizar una revisión de literatura contextualizando el problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional.
- Aplicar las etapas del proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD) a los documentos proporcionados por la empresa Cooperativa de Impresores del Oriente y consolidar una base de datos de clientes con el objetivo de poder obtener una matriz de distancias.
- Formular un modelo de optimización matemática para el 2L-HFFVRP.
- Diseñar un algoritmo genético para la solución del 2L-HFFVRP en el lenguaje de programación Python.
- Validar el algoritmo propuesto a través de diferentes instancias de prueba ajustadas a datos reales de la empresa.
- Elaborar un artículo académico publicable en el que se documente la investigación y los resultados más relevantes obtenidos de ésta.

2. Generalidades de la Investigación

2.1. Planteamiento del Problema

En el entorno competitivo actual, las empresas deben tomar decisiones estratégicas en el diseño de sus redes de distribución; una correcta gestión de la cadena de suministro en cualquier sector industrial es trascendental, teniendo en cuenta que allí se encuentran asociados los costos logísticos de producción y distribución. La cadena de suministro comienza en la adquisición y procesamiento de las materias primas y finaliza con la distribución a los clientes. Este último es uno de los componentes más importantes para fortalecer la relación con el cliente y mejorar la competitividad de las empresas.

En Colombia, los costos logísticos tuvieron una disminución de 0,9 puntos porcentuales, es decir que, de cada \$100 que facturaron las empresas, destinaban en promedio \$13,50 en 2018 y \$12,60 en 2020, según la última actualización de la Encuesta Nacional Logística (ENL) del Departamento Nacional de Planeación. De igual forma, la encuesta indica que el componente de transporte es el de mayor participación dentro del costo logístico total (Departamento Nacional de Planeación, 2020). En este sentido, se pueden evidenciar los esfuerzos por reducir estos costos asociados. Sin embargo, siguen siendo altos en comparación con otras economías mucho más desarrolladas, en las cuales su reducción ha sido de hasta 4%. Ahora bien, teniendo en cuenta que el sector productivo del país está mayormente centrado en las pequeñas y medianas empresas, cabe resaltar que el 56% cuentan con flota propia de diferente tipología vehicular y limitada (fija), entre ellas el 14,1% utiliza vehículos con capacidades de hasta 4,5 toneladas (Departamento Nacional de Planeación, 2020), resultando ser una oportunidad para utilizar eficientemente esos recursos propios y enriquecer la cadena de suministro.

Optimizar eficientemente las redes logísticas teniendo en cuenta la variable de costo, a menudo, se asocia con la distancia de conducción, la capacidad del vehículo y la forma de empaquetar los productos para alcanzar altos niveles de utilización del vehículo. La flexibilidad que brinda disponer de vehículos con características diferentes, lo que comúnmente se conoce como flota heterogénea, permite dar solución a problemas reales teniendo en cuenta restricciones por espacio insuficiente debido al tráfico, clientes ubicados en calles estrechas o las limitaciones del peso del vehículo en el núcleo urbano. El problema de ruteo de flota fija heterogénea (HFFVRP) se reduce al clásico problema de ruteo capacitado (CVRP) cuando los vehículos tienen la misma capacidad. En consecuencia, este tipo de problema es un NP-hard.

Por otro lado, ha surgido gran interés por problemas que están presente en muchos escenarios de la vida real relacionados, en este caso, con el transporte de artículos voluminosos y/o delicados que no pueden apilarse ni superponerse, en donde se debe considerar restricciones de embalaje bidimensional (2L); tener en cuenta este tipo de restricciones lo convierte en un problema NP-hard (Dominguez et al., 2016). Además, está directamente relacionado con la utilización eficiente del espacio del vehículo y la cantidad de clientes que se pueden atender. El 2L-HFFVRP es una variante del problema de ruteo de vehículos que tiene en cuenta tanto la flota de vehículos, como la restricción de carga; poco estudiada en la literatura.

Considerando que el problema a abordar es un problema de optimización combinatoria del tipo NP-hard, con dos problemas NP-hard, es necesaria la adopción de una metaheurística para obtener soluciones factibles en una cantidad razonable de tiempo de computación como una forma viable de resolver instancias del 2L-HFFVRP a gran escala (Subramanian et al., 2012).

Esta investigación se plantea a través del Algoritmo Genético como propuesta para dar solución al problema y trabajarlo con datos reales de la Cooperativa de Impresores del Oriente

(Coimpresores), una empresa del sector papelerero de la ciudad de Bucaramanga, que se encarga de atender al sector de la industria de la comunicación gráfica en la región oriente del país, suministrando materias primas e insumos a sus asociados. Coimpresores tiene que realizar entregas de acuerdo con los pedidos diarios, para los cuales utilizan dos tipos de estibas en forma rectangular, pero con medidas diferentes y deben ser empaquetados de forma eficiente sobre la superficie del camión para alcanzar altos niveles de utilización del vehículo. En definitiva, se pretende realizar la formulación de un modelo matemático para abordar la problemática planteada; posteriormente, se desarrolla el algoritmo genético en el lenguaje de programación *Python* para finalmente, validar lo propuesto con instancias de prueba a partir de datos reales.

Por último, al tratarse de un trabajo en modalidad de investigación, se aportará al proceso de generación de conocimiento para la Universidad Industrial de Santander, en pro de dar cumplimiento a una de sus funciones misionales.

2.2. Metodología

La presente investigación se aborda mediante una metodología de diagnóstico para el proceso de distribución de Coimpresores y la metodología propuesta por Hillier y Lieberman (2010) para la investigación de operaciones. A continuación, se detalla lo anteriormente mencionado:

2.2.1. Metodología para el Diagnostico

La Cooperativa de Impresores del Oriente cuenta con un proceso de distribución de mercancías para la ciudad de Bucaramanga y su área metropolitana. Este proceso será exhaustivamente analizado en el diagnóstico. Inicialmente, se obtuvo información general de la empresa, incluyendo aspectos como el número de personal, la ubicación de las instalaciones, el número de vehículos, así como los métodos empleados para el monitoreo y la planeación de las

rutas, entre otros. Gracias a este primer acercamiento, se evidenció la ausencia de un modelo de distribución sólido; en su lugar, el proceso se sustenta principalmente en la experiencia del encargado. Sin embargo, la empresa manejaba ciertas políticas de despacho. Por ejemplo, establece que cada pedido debe ser entregado en un plazo máximo de cuatro horas a partir del momento de su facturación. Además, administra el proceso de facturación y entrega en función de jornadas específicas: si un pedido se genera en la jornada de la mañana, se procede a su facturación y entrega en la tarde del mismo día; en cambio, si se hacía en horas de la tarde, se programaba la facturación y entrega para la mañana del día siguiente (ver Apéndice A).

En este diagnóstico se empleó una combinación de enfoques cualitativos y cuantitativos para obtener conclusiones concretas que servirán de base para el establecimiento del modelo. Entre las metodologías utilizadas se incluyen:

- Visita general a la planta: Este procedimiento se consideró crucial para comprender el proceso desde el principio. Durante la visita, se realizaron entrevistas tanto a la gerente, como al responsable del proceso de distribución, con el objetivo de conocer y entender a profundidad el proceso que siguen los pedidos desde que se emite la orden de pedido por parte de un vendedor hasta que es entregado al cliente.
- Recopilación de información a través de registros históricos: Se facilitó la obtención de datos mediante los históricos que llevaba el jefe del proceso de distribución. Esto se realizó con la previa autorización de la gerente de la empresa, lo que contribuyó a la comprensión del proceso.
- Análisis exploratorio: Se realizó un análisis de los registros históricos, explorando patrones, tendencias y posibles áreas de mejora.

- Entrevista con conductores de vehículos: Esta práctica proporcionó información más específica sobre la operatividad real del proceso.

2.2.2. Fase I. Definición del Problema de Interés y Recolección de Datos Relevantes

Para esta investigación, esta fase se va a dividir en dos etapas, las cuales permiten dar cumplimiento a dos de los objetivos específicos propuestos.

2.2.2.1. Etapa 1.1. Definición del problema de interés. Se aborda una revisión de literatura preliminar en bases de datos científicas con el objeto de comprender los diferentes modelos de ruteo de vehículos, definir los componentes y factores relevantes para el estudio.

Actividad 1: Identificar palabras clave que sean relevantes para la investigación.

Actividad 2: Seleccionar las bases de datos científicas, disponibles en la Universidad Industrial de Santander, en donde se hará la revisión de literatura.

Actividad 3: Construir ecuación de búsqueda que abarque los aspectos relevantes al tema de investigación.

Actividad 4: Realizar el respectivo análisis bibliométrico de la información.

Actividad 5: Seleccionar artículos que tengan afinidad con la investigación a realizar, depurándolos por medio del resumen y la justificación.

Actividad 6: Revisar la literatura obtenida acerca del problema 2L-HFFVRP, sus características, su aplicabilidad y los métodos de solución o algoritmos propuestos para el modelo.

2.2.2.2. Etapa 1.2. Recolección de datos relevantes. Cumpliendo con el segundo objetivo específico, se realiza la limpieza y depuración de los documentos proporcionados por la empresa Cooperativa de Impresores del Oriente, aplicando las etapas del proceso KDD, la cual cuenta con las siguientes actividades:

Actividad 1: Seleccionar los datos que se encuentren convenientes de utilizar.

Actividad 2: Realizar un pre-procesamiento de los datos para remover datos vacíos y unificar datos ruidosos.

Actividad 3: Reducir las dimensiones de las tablas para disminuir el número de variables a tener bajo consideración.

Actividad 4: Identificar patrones de interés mediante modelos descriptivos.

Actividad 5: Interpretar y documentar los patrones descubiertos.

Finalmente, se calcula la matriz de distancias con las siguientes actividades:

Actividad 6: Hallar los clientes más representativos para la empresa mediante la Ley de Pareto.

Actividad 7: Geolocalizar las plantas de producción de los clientes hallados en la actividad anterior.

2.2.3. Fase II. Formulación del Modelo Matemático

Esta fase abarca la formulación del modelo matemático del problema 2L-HFFVRP, el cual busca representar la realidad de la empresa Cooperativa de Impresores del Oriente a través de un modelo analítico, preservando la naturaleza de este para que pueda ser generalizado. Se presentan las siguientes actividades para el cumplimiento del tercer objetivo específico:

Actividad 1: Comprender las bases teóricas que involucran el problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional aplicado a la empresa Cooperativa de Impresores del Oriente.

Actividad 2: Establecer el propósito del modelo adaptado a los intereses de la empresa.

Actividad 3: Determinar los supuestos a considerar para relajar el modelo matemático.

Actividad 4: Definir las variables de decisión para el 2L-HFFVRP.

Actividad 5: Determinar los parámetros o datos de entrada a tener en cuenta en el 2L-HFFVRP.

Actividad 6: Plantear la función objetivo y las restricciones que mejor representen la realidad.

Actividad 7: Verificar la consistencia del modelo propuesto.

2.2.4. Fase III. Desarrollo un Procedimiento de Computación para Derivar una Solución del Problema a Partir del Modelo Propuesto

Para el cumplimiento del cuarto objetivo específico, se llevan a cabo las siguientes actividades:

Actividad 1: Identificar los elementos necesarios para implementar la metaheurística de algoritmo genético.

Actividad 2: Especificar los pasos a seguir para permitir dar solución al problema 2L-HFFVRP.

Actividad 3: Programar el algoritmo genético planteado para la solución del 2L-HFFVRP en el lenguaje de programación Python.

2.2.5. Fase IV. Prueba del Modelo

Se valida el modelo, en cumplimiento del quinto objetivo específico planteado, explorando posibles fallas y/o errores, mediante las siguientes actividades:

Actividad 1: Determinar la validez de los resultados del modelo propuesto y la eficiencia del algoritmo.

Actividad 2: Desarrollar instancias de prueba para el modelo propuesto.

Actividad 3: Validar nuevamente el modelo con las instancias desarrolladas en la actividad anterior.

2.2.6. Fase V. Síntesis de Resultados

En esta fase se concentra la información obtenida como resultado de la presente investigación, registrando el cumplimiento del último objetivo específico mediante las siguientes actividades:

Actividad 1: Consolidar la base de datos de los clientes depurada.

Actividad 2: Escribir un artículo académico publicable en el que se muestre el 2L-HFFVRP planteado para la empresa y los resultados obtenidos con el método de solución propuesto a manera de caso de estudio en una revista indexada

Actividad 3: Elaborar el libro de proyecto de grado.

3. Revisión de Literatura y Análisis Bibliométrico

3.1. Análisis Bibliométrico

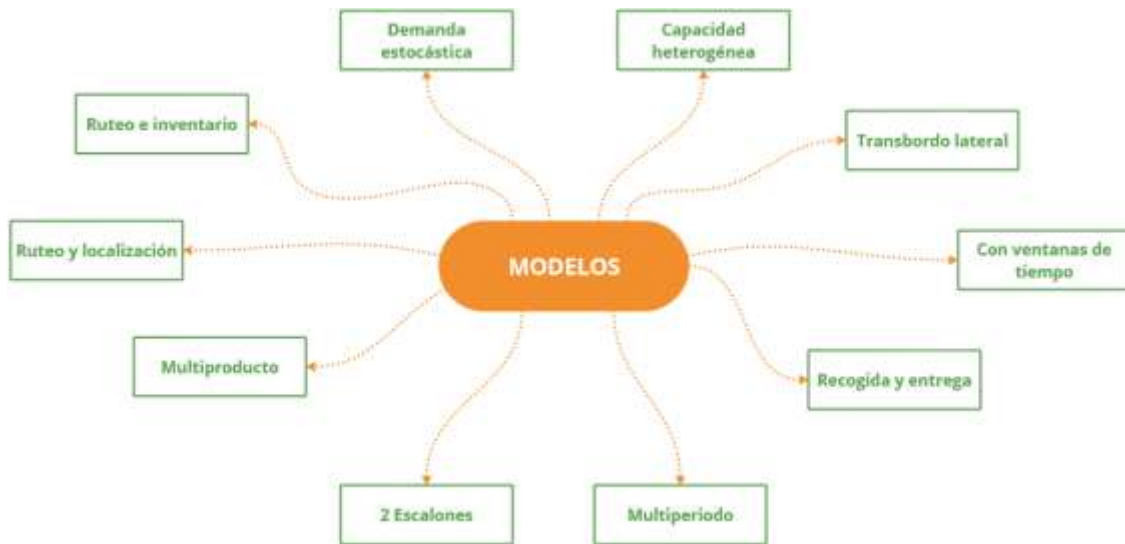
El análisis de las publicaciones científicas es una etapa fundamental dentro del proceso de investigación, volviéndose una herramienta para comprender tres preguntas esenciales: ¿qué se ha investigado?, ¿cuánto se ha investigado? y ¿dónde se ha investigado?, y calificar la calidad y el impacto de la generación de conocimiento. Para el presente trabajo de investigación, dicho análisis se desarrolla a través de la herramienta *Clarivate Analytics*. La búsqueda se realizó en la base de datos *Web of Science* suministrada por la Universidad Industrial de Santander, con la siguiente ecuación de búsqueda:

(Vehicle routing problem) **AND** [(capacity restriction) **OR** (heterogeneous fleet)] **NOT** (emission*) **NOT** (electric*) **NOT** (drones).

A continuación, se delimitó la búsqueda a artículos científicos de los últimos 10 años (2012-2022). Seguidamente, se filtró por artículos y artículos de revisión, y en el idioma inglés, obteniendo en total 628 resultados. Esta investigación inició con la búsqueda de artículos que ampliaran la perspectiva de los modelos de ruteo de vehículos (VRP) y la creación de un mapa mental con los modelos de interés (**Figura 1**). Siguiendo la investigación en el problema de ruteo de vehículos que involucra a la capacidad de los vehículos.

Figura 1

Mapa mental de los modelos de interés.

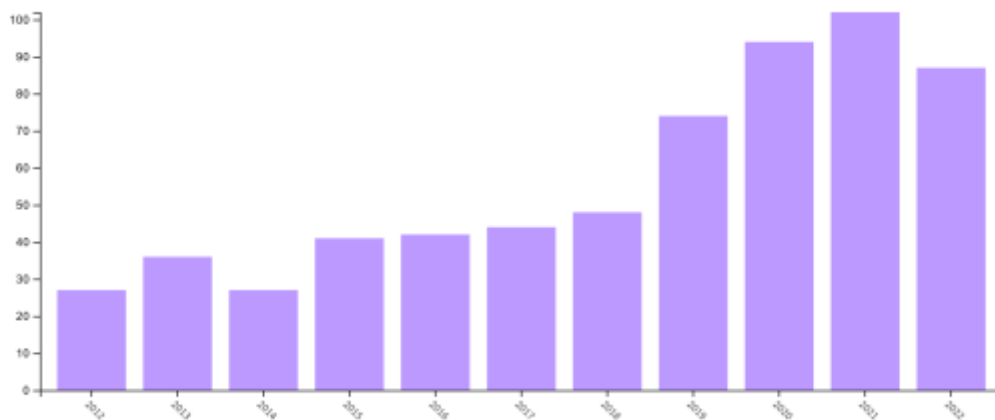


3.1.1. Año de Publicación

La siguiente figura muestra la cantidad de artículos publicados por año, en donde se evidencia que el problema de ruteo de vehículos y sus variantes ha aumentado su interés de estudio a través de los años.

Figura 2

Cantidad de artículos por año.



Nota. Tomado de Clarivate Analytics.

3.1.2. Países

En la **Figura 3**, se muestra el número de publicaciones de acuerdo con el país de procedencia del autor en los años de estudio. Se puede observar que, en el tema de ruteo, gran parte de las investigaciones han sido realizadas por autores provenientes de China y Estados Unidos. Sus aportes contribuyen aproximadamente a un cuarto del total de los artículos encontrados. Cabe señalar que, Colombia se encuentra dentro de los quince (15) países con mayor cantidad de investigadores en el tema.

Figura 3

Países.



Nota. Tomado de *Clarivate Analytics*

3.1.3. Autores

Diversos autores fueron identificados en este análisis, pero son Marrielle Christiansen y Gilbert Laporte los que se destacan como los autores con mayor aporte en el estudio de los modelos VRP con un total de doce (12) artículos cada uno. A continuación, se presenta la figura de las contribuciones de los principales autores encontrados.

Figura 4

Autores.



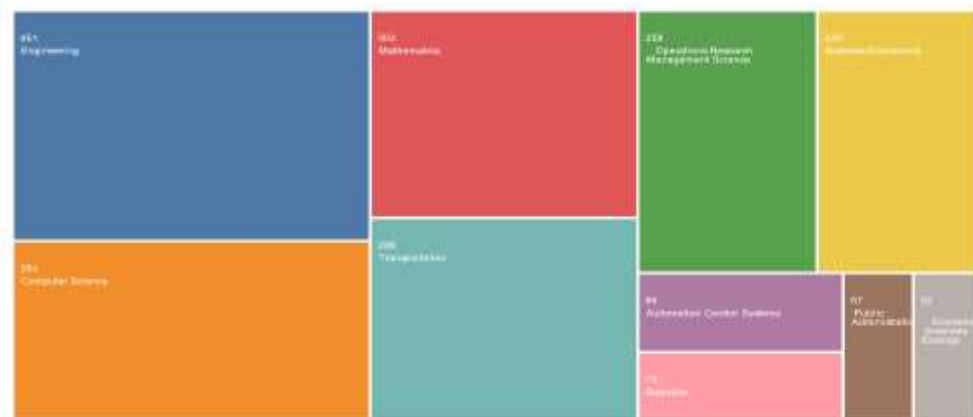
Nota. Tomado de *Clarivate Analytics*

3.1.4. Áreas de Investigación

De acuerdo con el área de investigación, alrededor del 73% de los artículos encontrados fueron sobre temas relacionados con la ingeniería; sin embargo, dentro de los resultados se destacan otras áreas del conocimiento distintas, como las ciencias de la computación, las matemáticas, el transporte, la investigación de operaciones y la economía empresarial.

Figura 5

Áreas de investigación.



Nota. Tomado de *Clarivate Analytics*

Del análisis precedente, se afinó la búsqueda en primer lugar por área de investigación seleccionando la ingeniería, el transporte y la investigación de operaciones; en segundo lugar, se filtró por resumen y justificación.

3.2. Revisión de Literatura

En los últimos años, los problemas donde se integra el ruteo de vehículos con los problemas de carga están despertando un interés cada vez mayor, esto debido a su relevancia para los sistemas logísticos de las organizaciones (Côté et al., 2017). Hasta donde se sabe, no existen estudios sobre el 2L-HFFVRP hasta la fecha. Debido a esto, esta revisión de literatura se centrará en las investigaciones previas realizadas acerca del 2L-CVRP y el 2L-HFVRP. Ambos problemas combinan el problema de ruteo de vehículos con restricciones de carga bidimensional, pero se asume una flota de vehículos homogénea y heterogénea ilimitada, respectivamente.

El 2L-CVRP fue introducido por primera vez por Iori et al., 2007. Trataron el problema de ruteo empleando un algoritmo de ramificación y corte (Branch-and-cut) para tratar los aspectos de ruteo del problema anidado con un procedimiento de ramificación y acotamiento (Branch-and-Bound) para garantizar que las cargas a llevar dentro de los vehículos fuesen factibles. Posteriormente, Gendreau et al., 2008 propusieron una metaheurística de búsqueda tabú para el problema de ruteo, y realizaron la verificación de la factibilidad de las cargas a través de heurísticas y de un procedimiento de Branch-and-Bound, considerando la carga como secuencial y sin restricciones en su orientación. Fuellerer et al., 2009, desarrollaron un algoritmo de colonia de hormigas para solucionar el problema de ruteo de vehículos combinado con una heurística para el problema de la carga. Estos autores fueron los primeros que permitieron la rotación de los artículos. Duhamel et al., 2011, propusieron un método que combinaba una metaheurística GRASP (Greedy Adaptive Random Search Procedure) con una búsqueda local evolutiva transformando el problema

de cargas en un problema de programación de proyectos. Wei et al., 2018, propusieron un algoritmo de recocido simulado con un mecanismo de aumento y disminución de la temperatura que utilizan en una heurística basada en espacio abierto para tratar con las restricciones de carga que mejoró la solución de todas las versiones del problema abordadas anteriormente.

Con respecto al 2L-HFVRP, fueron Leung et al., 2013 los primeros en abordar el problema a través de un algoritmo híbrido de recocido simulado con una búsqueda local para dar solución al problema de ruteo y una serie de seis heurísticas para resolver las restricciones de la carga. Dominguez et al., 2016, desarrollaron un algoritmo que combinaba versiones aleatorizadas sesgadas de algunas heurísticas para resolver el problema de carga con orientación y sin orientación de los objetos. La aleatorización sesgada de heurísticas hace referencia al uso de distribuciones de probabilidad sesgadas para inducir un comportamiento aleatorio no simétrico en una heurística (Grasas et al., 2017). Zhang et al., 2018, propusieron un algoritmo de enjambre de abejas con sistema inmunológico artificial para tratar problemas con cargas secuenciales y no restringidas.

Esta revisión de literatura acerca del problema de ruteo de vehículos de flota heterogénea con restricciones de carga bidimensional evidencia que otros investigadores sólo han resuelto estos problemas considerando una flota ilimitada de vehículos. No obstante, en la práctica, no es un escenario real, ya que las empresas por lo general poseen una flota limitada de vehículos debido a las limitaciones en sus recursos, generalmente económicos.

4. Marco de Referencia

4.1. Marco de Antecedentes

En la Universidad Industrial de Santander han sido escasos los estudios realizados sobre el problema de ruteo de vehículos de flota heterogénea y no existen investigaciones que involucren

restricciones de carga bidimensional. Debido a esto, y teniendo en cuenta que la flota heterogénea es el clásico problema ruteo capacitado, pero de diferente capacidad, se decide también tomar en cuenta proyectos que busquen dar solución a problemas con restricción de capacidad. A continuación, se abordan grosso modo proyectos que contribuyen a la presente investigación con características del problema de flota heterogénea y capacitada, algoritmos de solución y literatura.

Pabon & Picón, 2019 presentaron una solución al problema de distribución de ayuda humanitaria con tiempos de viaje estocásticos utilizando una flota heterogénea de vehículos para un escenario de desastre en la ciudad de Bucaramanga presentando un algoritmo genético como método de solución como trabajo de investigación de pregrado. Propusieron su modelo mediante una distribución de probabilidad para la velocidad de desplazamiento por cada uno de los arcos que componen la red de la malla vial de la ciudad. Este modelo tenía como objetivo la minimización del costo operativo, el costo por demanda insatisfecha y el costo por demora en entrega. Posteriormente, implementan una heurística del vecino más cercano para generar un conjunto de soluciones iniciales las cuales posteriormente fueron mejoradas por los operadores del algoritmo genético. Por último, para validar este modelo, hacen uso de instancias encontradas en la literatura. Esta investigación se relaciona con la que está en curso en el uso de una flota heterogénea de vehículos presente en sus respectivos problemas, evidenciando que el uso de una flota de estas características es más común de lo que puede llegar a parecer, a pesar del poco o nulo estudio que se ha implementado acerca del tema.

Tolosa A, 2017 en su estudio de pregrado presentó un modelo de ruteo que se ajusta a la operación de la empresa Espumas Santander S.A.S., de manera similar a lo planteado en la presente investigación. Para lograr esto, el autor realizó una revisión de literatura en búsqueda de un modelo base hasta llegar a definir el funcionamiento logístico de la empresa como un problema de ruteo

de vehículos de flota heterogénea con ventanas de tiempo (HFVRPTW). En su proyecto buscó disminuir la distancia total del recorrido por vehículo desarrollando algoritmos de solución basados en la heurística de Clark & Wright y en la metaheurística de Búsqueda Local Adaptativa (ALNS). La solución obtenida es alimentada a una interfaz gráfica escrita en diversos lenguajes evaluando su funcionamiento y eficacia contrastándola con el ruteo programado previamente por la empresa mostrando una disminución en la distancia recorrida por los vehículos.

Córdoba, 2020 abordó un problema de ruteo de vehículos de dos escalones con gestión de inventario considerando transbordos laterales formulando el problema como un modelo de programación lineal entera mixta donde se buscaba minimizar los costos logísticos considerando restricciones de capacidad, una demanda determinística y transbordos ejecutados solo entre distribuidores. Para dar solución a este problema, desarrolló una metaheurística híbrida compuesta por un algoritmo genético y una búsqueda tabú. Para la validación del algoritmo, el autor utilizó un diseño factorial con nueve instancias donde evaluó el comportamiento de los costos totales y el recurso computacional consumido. La metaheurística híbrida obtuvo buenos resultados, reduciendo el tiempo computacional comparado con soluciones exactas. Este trabajo va a ser de utilidad como base en el uso de la metaheurística de algoritmo genético para el presente estudio.

4.2. Marco Teórico

4.2.1. Optimización Matemática

La optimización matemática consiste en la selección de la mejor alternativa de solución a un problema determinado, dentro de un conjunto de alternativas posibles. Por su parte, una solución óptima, es la mejor solución matemáticamente hablando. Para optimizar, se comprenden los siguientes elementos:

- **Función objetivo:** Es la función matemática que representa la medida de desempeño del sistema a optimizar.
- **Variables:** Simboliza las alternativas de solución o decisiones que pueden tomarse para afectar el resultado de la función objetivo.
- **Restricciones:** Representan el conjunto de limitaciones que se pueden imponer sobre las variables y que se están obligadas a satisfacer. Naturalmente, se expresan en forma de ecuación o de desigualdad.
- **Parámetros:** Son los coeficientes o constantes del lado derecho de las restricciones y de la función objetivo, propias del problema de estudio.

En definitiva, para resolver un problema de optimización matemática se deben encontrar el valor de las variables de decisión de manera que se maximice o minimice la función objetivo, sujeta a unas restricciones dadas (Linares et al., 2001).

4.2.2. Modelos Matemáticos

Un modelo matemático es una representación abstracta del mundo real, centralizada en variables que controlan el comportamiento de ese sistema (Taha, n.d.). Esta representación se realiza mediante la relación entre símbolos que representan dichas variables, sistemas de ecuaciones y expresiones matemáticas. Describir un problema por medio de un modelo muestra relaciones causa-efecto implícitas y simplifica la situación de estudio, volviéndola mucho más comprensible. Por otro lado, crea una relación directa con la forma de analizar el problema, a través del empleo de técnicas matemáticas y software de computadora con alto nivel computacional (Hillier & Lieberman, 2010).

Teniendo en cuenta que lo que se busca es que el problema sea resuelto por medio del modelo, se requieren diversas aproximaciones y supuestos que sinteticen, pero no abandonen la

realidad y validez del problema real. Finalmente, los modelos matemáticos se clasifican de acuerdo con la naturaleza de la ecuación, de la siguiente manera:

4.2.3. Modelos de Programación Lineal.

Son modelos en los cuales la función objetivo y las restricciones son estrictamente lineales, involucrando la planeación de actividades para obtener el resultado que mejor alcance el objetivo de optimización especificado (Hillier & Lieberman, 2010). Las expresiones matemáticas generales se plantean de la siguiente manera (Linares et al., 2001):

$$\text{mín o máx } c^T x$$

$$Ax = b$$

$$x \geq 0$$

$$x \in R^n, c \in R^n, A \in R^{m \times n}, b \in R^m$$

4.2.4. Modelos de Programación No Lineal.

Hace referencia a los modelos en los que la función objetivo y/o restricciones son no lineales. Depende de las características de las funciones $f(x)$ y $g_i(x)$ (Hillier & Lieberman, 2010) y de manera general, se plantea así (Linares et al., 2001):

$$\text{mín o máx } f(x)$$

$$g_i(x) \leq 0$$

$$l \leq x \leq u$$

$$f: R^n \rightarrow R$$

$$g_i: R^n \rightarrow R^m$$

4.2.5. Optimización Combinatoria

Es una de las áreas más extensas de la optimización en las matemáticas aplicadas, la investigación de operaciones y la ciencia de la computación. Estudia los problemas que se

caracterizan por encontrar el valor que maximice o minimice una función determinada, en la cual el espacio de soluciones posibles es discreto.

Un problema de optimización combinatoria $P = (S, f)$ se define mediante:

- Un conjunto de variables $X = \{x_0, x_1, \dots, x_n\}$
- Dominios de variables D_0, D_1, \dots, D_n
- Restricciones entre variables
- Función objetivo $f: D_0 \times D_1 \times \dots \times D_n \rightarrow R^+$

Se denomina S al conjunto de posibles soluciones en donde se satisfacen todas las restricciones y se define como:

$$S = \{s = \{(x_0, v_0), (x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisface todas las restricciones}\}$$

Para resolver este tipo de problemas se requiere encontrar una solución $s^* \in S$ tal que $f(s^*) \leq f(s)$ para todo $s \in S$ (Blum & Roli, 2003).

Debido a la importancia de los problemas de optimización combinatoria, se han desarrollado algoritmos para solucionarlos de acuerdo con su complejidad computacional que, además, permite validar la optimalidad de la solución obtenida.

4.2.6. Complejidad Computacional

Estudia la cantidad de recursos informáticos que requiere para dar solución de un problema, como lo son el tiempo, definido como los pasos que requiere el problema para ser resuelto; y el espacio, que hace referencia a la cantidad de memoria que utiliza (Cortez, 2004). Además, analiza todos los posibles algoritmos para resolver el mismo problema.

Existen dos clases principales de complejidad a partir de máquinas Turing deterministas y no deterministas (Cortez, 2004):

- Clase P, son problemas de decisión que pueden ser resueltos por una máquina determinista estableciendo una relación polinómica entre el tamaño del problema y el tiempo de ejecución.
- Clase NP, es el conjunto de problemas de decisión que pueden resolverse por una máquina no determinista en tiempo no polinómico debido a que son inherentemente difíciles.

Sin embargo, existen subclases caracterizadas como problemas más difíciles de cada clase denominados P-completo, NP-completo y NP-hard (Cortez, 2004).

4.2.7. Problema de Ruteo de Vehículos (VRP)

De acuerdo con la literatura, el primer problema planteado del VRP fue el agente viajero o TSP (Travelling Salesman Problem), data del año 1954 en un artículo por Dantzig, Fulkerson y Johnson. El problema de ruteo de vehículos (VRP) consiste en diseñar un conjunto de rutas de entrega que genere el menor costo a través de un conjunto de clientes dispersos geográficamente, sujeto a una serie de restricciones secundarias (Laporte et al., 2013).

4.2.8. Problema de Ruteo de Vehículos con Restricción de Capacidad (CVRP)

Consiste en encontrar un conjunto de rutas para visitar un conjunto de clientes utilizando una flota de vehículos capacitados idénticos entre sí. Cada cliente debe ser visitado exactamente una vez, por exactamente un vehículo. La carga del viaje no puede exceder la capacidad de carga del vehículo y el objetivo es determinar un conjunto de rutas que minimicen el costo total del recorrido. Es decir, determinar una secuencia para visitar a cada uno de los clientes con el fin de minimizar el total de la distancia del recorrido (Akpınar, 2016).

4.2.9. Métodos de Solución para Problemas de Optimización Combinatoria

Para abordar los distintos problemas y modelos de optimización se han desarrollado algoritmos dependientes del enfoque de optimización, que pueden clasificarse de acuerdo con exactitud en cuanto a la solución óptima global.

4.2.9.1. Métodos exactos. Este tipo de métodos se formulan a partir de la programación lineal entera y son capaces de encontrar la solución óptima global. Se usa eficientemente para problemas pequeños dado que, al acrecentar los parámetros y restricciones, empiezan a adquirir mayor complejidad computacional. Los métodos más utilizados en problemas de optimización combinatoria son:

- Método simplex, es un procedimiento algebraico iterativo conceptualizado a partir de conceptos geométricos. Parte de una solución inicial, el vértice (0,0), para empezar a examinar soluciones adyacentes que den mejores resultados en la función objetivo, desarrollando de forma matricial y gráfica hasta alcanzar el óptimo (Hillier & Lieberman, 2010).
- *Ramificación y acotamiento*, también conocido como algoritmo *Branch and Bound*. Este tipo de algoritmo, también iterativo, intenta eliminar de la búsqueda las ramas en las que no se va a obtener una solución mejor que la ya existente. Esta eliminación se realiza a base de cotas calculadas a partir de la exploración, que dan elementos suficientes para descartar o no una rama y seguir la búsqueda (Clausen, 1999).

4.2.9.2. Métodos aproximados. Dada la complejidad computacional NP-hard o “difícil de resolver” de muchos de los problemas de optimización que aparecen en la vida cotidiana, se han desarrollado diversos algoritmos y métodos de solución, en los que se encuentra una solución factible muy buena; esto debido a que en este tipo de problemas no es posible encontrar la solución óptima global en un tiempo razonable. Se clasifican en heurísticas y metaheurísticas.

4.2.9.2.1. Heurísticas. Este tipo de algoritmos se han desarrollado y diseñado para problemas específicos por lo que no es posible generalizar o aplicarlos a problemas que se consideren similares. Por otro lado, no suelen utilizar ningún tipo de memoria para guiar la búsqueda de soluciones, buscando una alta calidad en su solución a un costo computacional razonable. A continuación, se presentan algunas categorías, en forma amplia y no excluyente (De Antonio Suárez, 2011):

- *Métodos de descomposición*, como su nombre lo indica busca descomponer el problema inicial, en problemas más sencillos de resolver.
- *Métodos inductivos*, en donde se generalizan las propiedades o técnicas identificadas más fáciles de analizar para aplicar al problema completo.
- *Métodos de reducción*, en los cuales se restringe el espacio de solución, identificando las propiedades que se cumplen para las buenas soluciones. Existe el riesgo de descartar soluciones óptimas del problema original.
- *Métodos constructivos*, como su nombre lo indica consiste en construir el paso a paso de la solución al problema teniendo en cuenta los elementos de la mejor solución que se encuentra en cada iteración.

- *Métodos de búsqueda local*, parten de una solución inicial y la mejoran progresivamente, encontrando en cada paso una solución con mejor valor y finaliza cuando, para una solución, no existe otra que la mejores.

4.2.9.2.2. Metaheurísticas. Son métodos de solución iterativa que proporcionan un marco general de guía para el diseño de métodos heurísticos ajustados a problemas específicos, indicando cómo y dónde aplicarlos. Por otro lado, las metaheurísticas son capaces de realizar unas búsquedas fuertes en la región factible, evadiendo óptimos locales (Hillier & Lieberman, 2010), son estratégicas para el proceso de búsqueda dado el uso inteligente de la aleatoriedad, partiendo de procedimientos de búsqueda local hasta llegar a procesos de aprendizaje mucho más complejos. Generalmente se basan en decisiones probabilísticas realizadas durante la búsqueda (Blum & Roli, 2003).

Las metaheurísticas tienen diferentes formas de clasificación que van desde la inspiración con procesos naturales como la genética, hasta la dependencia de su historial de búsqueda, teniendo en cuenta el uso de memoria. Sin embargo, la clasificación más común en la literatura depende del número de soluciones sobre las que puede trabajar al mismo tiempo, algoritmos basados en población y algoritmos basados en trayectoria (Blum & Roli, 2003):

Algoritmos basados en población, trabajan en cada iteración con un conjunto de soluciones, manteniendo la diversidad en la población y evitando que las soluciones se queden en los óptimos locales, como lo son los algoritmos evolutivos y los de inteligencia de enjambre. Algunos algoritmos son:

- **Algoritmo Genético (GA)**, es un algoritmo de optimización inspirado en la teoría darwiniana de la supervivencia del más apto. Este método, transforma una población de cadenas cromosómicas, cada una con un valor de aptitud asociado que se mide mediante

la función objetivo, en una nueva población denominada descendencia, utilizando operadores de inspiración biológica como la selección, la mutación y el cruce. La población se define como un conjunto de N soluciones factibles que se van mejorando de forma iterativa hasta que se cumpla el criterio de parada, el cual depende de las características de la solución encontrada o del número máximo de iteraciones establecido (Katoch et al., 2021).

— **Algoritmos de Optimización basada en Colonia de Hormigas (ACO)**, es un modelo probabilístico inspirado en el comportamiento real de búsqueda de alimento de las colonias de hormigas, de uso práctico para problemas de optimización combinatoria y, caracterizado por su versatilidad y robustez. En esta metodología, las hormigas artificiales construyen soluciones al problema de optimización a medida que se mueven, depositando rastros de feromonas para dirigir la búsqueda de soluciones en futuras iteraciones del algoritmo. Dicha sustancia se evapora con el tiempo, por lo que, en el camino más corto entre la fuente de alimento y el nido, habrá mayor concentración de feromona; este corresponderá a la mejor solución encontrada y por el que toda la población de hormigas transitará al final (Glover & Gary, 2003).

— **Algoritmos de Optimización por Enjambre de Partículas (PSE)**, es un metaheurística inspirada en el comportamiento social de grupos de animales como bancos, parvada, manadas, etc., para la búsqueda de alimento y evasión de depredadores; desarrollada por James Kennedy y Russell Eberhart en 1995. Este procedimiento iterativo inicia con una población de soluciones aleatorias, llamadas partículas pertenecientes a una población denominada enjambre. Cada partícula evalúa el espacio de búsqueda en su posición actual. Luego, determinan su siguiente movimiento integrando la posición actual,

la mejor posición anterior (mejor aptitud) y la velocidad individual o de más miembros del enjambre, tendiendo hacia el óptimo de la función objetivo (Poli et al., 2006).

Algoritmos basados en trayectoria, se caracterizan porque la solución sucesora puede o no pertenecer a la vecindad de la solución actual, tratan con una única solución. A continuación, se describen algunos algoritmos básicos:

— **Recocido Simulado (SA)**, es una de primeras metaheurísticas estudiadas, presentada por primera vez por Kirkpatrick, Gelatt y Vecchi en 1983. Este proceso es análogo al comportamiento de algunos sistemas físicos sometidos a un descenso paulatino y lento en su temperatura, alcanzando así una configuración de energía de red mínima. El recocido simulado establece la conexión entre este tipo de comportamiento termodinámico y la búsqueda de mínimos globales. Esta metodología, se caracteriza por permitir movimientos de escalada, que empeoran el valor de la función objetivo, para escapar de los óptimos locales. La probabilidad de aceptar estos valores depende de un parámetro de temperatura y a medida que se reduce a cero, los movimientos recurren con menor frecuencia. Un elemento crucial para el rendimiento del algoritmo es la caracterización de las condiciones para considerar *frio* al sistema, en función de la temperatura y el número de iteraciones (Glover & Gary, 2003).

— **Búsqueda Tabú (TS)**, fue introducida en 1986 por Fred Glover, es una extensión de la heurística clásica de búsqueda local, integrando la memoria adaptativa. Esta memoria funciona teniendo en cuenta cuatro dimensiones: lo reciente definido como memoria a corto plazo; y la frecuencia, la calidad y la influencia relacionado con la memoria a largo plazo, permitiendo implementar procesos de búsquedas más eficientes. Con la memoria a corto plazo, se contribuye a la creación de listas tabú, las cuales evitan

ciclos y evadir mínimos locales, por medio del seguimiento a las soluciones visitadas y prohibiendo los movimientos hacía ellas. Por otro lado, la memoria a largo plazo proporciona información acerca de las soluciones con óptimos locales de alta calidad, identificando elementos comunes y la calidad y estructura de las mismas, encontradas durante todo el proceso de búsqueda. La sinergia entre la memoria a corto y largo plazo, logran un balance entre las estrategias de intensificación y de diversificación para explorar soluciones no visitadas antes. Por último, esta metaheurística incorpora la exploración responsiva como guía hacía zonas prometedoras integrando estrategias de búsqueda inteligente (Glover & Batista, 2006).

— **Búsqueda de Local Guiada (GLS)**, La búsqueda local guiada (GLS), propuesta por Voudouris y Tsang en 1996, es una metaheurística que fue diseñada para problemas de optimización combinatoria (Glover & Gary, 2003). En esta metaheurística se penaliza, usando una función de utilidad, algunas características de las soluciones no deseadas durante cada iteración de la búsqueda local. Esta penalización aumenta el valor de la función objetivo en cada iteración. De esta manera, el algoritmo puede escapar de los óptimos locales y permitir continuar con la búsqueda (Vansteenwegen et al., 2009). En general, la búsqueda local iterada consta de dos etapas. La primera consiste en una búsqueda local (LS) sencilla, cuyo principal objetivo es encontrar un mínimo local de la función a optimizar. En la segunda etapa, la función se modifica con el fin de escapar del óptimo local encontrado al final de la primera etapa. Estas dos etapas se repiten reiteradamente, hasta que una solución subóptima satisfactoria es obtenida o se agota el tiempo de ejecución dedicado a resolver el problema (Naanaa & Belghith, 2018).

4.2.10. Problema con Restricción de Carga Bidimensional (2L)

Los problemas con restricciones de carga bidimensional consisten en minimizar los espacios vacíos resultantes del empaquetamiento de artículos rectangulares ortogonalmente dentro de un contenedor o vehículo. Esto implica evitar cualquier tipo de solapamiento entre los artículos al interior del mismo.

Por consiguiente, los problemas con restricciones de carga bidimensional deben satisfacer dos consideraciones básicas:

- Los artículos deben ser cargados completamente dentro del contenedor o vehículo.
- Los artículos no deben superponerse, es decir, no deben ocupar un mismo espacio dentro del contenedor o vehículo.

Estas dos consideraciones definen lo que se conoce como restricciones dimensionales (Junqueira, 2013). Los primeros trabajos que trataron problemas con restricciones de carga contemplaban solo estas dos consideraciones. No obstante, a medida que avanzan las investigaciones en el tema, otras consideraciones pasaron a cobrar mayor importancia a la hora de abordar problemas con restricciones de carga. Bischoff & Ratcliff, 1995, presentaron nueve consideraciones prácticas, mostradas en la **Tabla 2**, que pueden tenerse en cuenta cuando se desea modelar problemas con restricciones de carga más realistas.

Tabla 2

Consideraciones tácticas presentadas por Bischoff y Ratcliff (1995).

Consideración	Descripción
Agrupación de artículos	Los artículos con un destino en común o de un mismo tipo deben ser colocados cerca, unos de los otros, dentro del contenedor o vehículo.
Distribución de peso	El centro de masa de un contenedor o vehículo cargado debe estar localizado cerca del centro geométrico de su base.

Estabilidad	Algunos artículos deben tener su superficie inferior soportada sobre la superficie superior de otro artículo y/o su superficie lateral apoyada en la superficie lateral de otro artículo.
Límite de peso	Los artículos deben ser cargados dentro del contenedor o vehículo sin exceder el peso máximo que este puede soportar.
Manipulación	Algunos artículos debido a sus dimensiones o a su peso deben ser colocados en un determinado espacio dentro del contenedor o vehículo.
Múltiples destinos	Los artículos que van a ser entregados en diferentes destinos deben ser cargados considerando la ruta que va a recorrer el contenedor o vehículo y el orden en el que serán descargados.
Orientación	Algunos artículos deben ser cargados dentro del contenedor o vehículo con orientaciones preestablecidas.
Prioridades	Los artículos con una fecha de entrega o fecha de caducidad próximas, pueden tener una mayor prioridad para ser cargadas, en detrimento de otros artículos con prioridades más bajas.
Separación de artículos	Los artículos que no pueden estar en contacto, entre sí, deben estar separados dentro del contenedor o vehículo.

Nota. Tomado de Bischoff & Ratcliff, 1995.

4.2.11. Métodos de Solución para el Problema 2L

Ante la complejidad computacional NP-hard presente en los problemas con restricción de carga bidimensional, se han desarrollado diversas heurísticas de solución. Estas heurísticas se adaptan a diversas configuraciones de embalaje, con el propósito de mejorar las probabilidades de encontrar una distribución de carga factible en un tiempo computacional razonable.

4.2.11.1. Heurística de colocación inferior izquierda. Esta heurística implica encontrar, entre las ubicaciones de carga factibles para los artículos, la posición con la mínima coordenada de un eje especificado (correspondiente al ancho o al largo del vehículo). En caso de empate, en la coordenada mínima del eje especificado, se rompe el empate utilizando la coordenada mínima del otro eje. Al aplicar esta heurística, la carga tiende a formar franjas de artículos paralelas al eje donde se ejecutó (Zachariadis et al., 2009).

4.2.11.2. Heurística del perímetro del contacto máximo. La posición elegida para colocar un nuevo artículo va ser aquella que presenta la suma más alta de bordes en compartidos entre el artículo a insertar, los artículos ya cargados en el vehículo y la superficie de carga del vehículo. Esta heurística tiende a distribuir los elementos hacia los bordes de la superficie de carga y posteriormente se llena el espacio interior (Candido & De Souza, 2022).

4.2.11.3. Heurística del área mínima. En esta heurística, se establece que la posición de colocación un artículo va ser aquella que presente el área rectangular más pequeña dentro de las posiciones factibles disponibles (Zachariadis et al., 2009).

4.2.11.4. Heurística de primera posición. Este algoritmo evalúa los elementos de manera secuencial. Cada elemento se asigna al contenedor, ubicándolo en las coordenadas más bajas disponibles que permitan su colocación (Dósa & Epstein, 2018).

4.2.12. APIs de Google Maps Plataform

Google Maps Plataform engloba un conjunto de Interfaces de Programación de Aplicaciones (API) y Kits de Desarrollo de Software (SDK) diseñados para que los desarrolladores puedan integrar Google Maps en aplicaciones tanto móviles como web, así como para acceder a datos de Google Maps. Entre las diversas API disponibles, se incluyen la API de Maps JavaScript, las versiones destinadas a Android y iOS, así como las API de Places y Address Validation, entre otras (*Google Maps Platform*, n.d.). A continuación, se presentan algunas de las APIs de particular interés en los problemas de Ruteo de Vehículos.

4.2.12.1. API de Geocoding. La API de geocoding de Google Maps es una herramienta proporcionada por Google que permite a los desarrolladores y empresas integrar capacidades, tanto la geocodificación, como la geocodificación inversa, en sus aplicaciones, sitios web y servicios. La geocodificación es el proceso de convertir una dirección en coordenadas de latitud y longitud, o de convertir las coordenadas en una dirección de lenguaje natural. La API proporciona información detallada sobre la dirección encontrada, como componentes individuales (calle, ciudad, estado, país, etc.), nivel de precisión y más. Además, ofrece una prueba gratuita de 90 días con un crédito de \$300 y por USD\$5 al mes, brinda la capacidad de realizar hasta 1.000 consultas (Descripción General de La API de Geocoding, n.d.).

4.2.12.2. API de Directions. Esta API de Google, constituye una herramienta para obtener las instrucciones sobre cómo llegar de un lugar a otro, mediante la configuración del modo de viaje sobre cómo llegar, ya sea en automóvil, bicicleta, transporte público o a pie. La API muestra tanto la ruta óptima como el tiempo estimado para la ruta. Al igual que la API anterior, ofrece una prueba gratuita de 90 días con un crédito de \$300, permitiendo a los usuarios explorar sin incurrir en costos iniciales. Para aquellos que requieren un uso más continuo, Google proporciona acceso desde USD\$5 para realizar hasta 1.000 consultas al mes (Documentación de Google Maps Platform | Directions API, 2023).

4.2.12.3. API de Distance Matrix. El API de Distance Matrix de Google se presenta como una herramienta para acceder a información sobre la distancia y el tiempo de viaje, mediante el análisis de las condiciones de tráfico en tiempo real. Esta API permite calcular de manera precisa la matriz de orígenes y destinos. Esta API brinda la oportunidad de experimentar la prueba gratuita de 90 días y desde USD\$5 efectuar hasta 1.000 consultas mensuales (Documentación de Google Maps Platform | Distance Matrix API, 2023).

4.2.13. OpenStreetMap (OSM)

El software OpenStreetMap, creado en 2004, es un proyecto colaborativo de mapeo en línea y gratuito, que permite crear y mantener una base de datos cartográficos del mundo, de modo que los usuarios finales no se ven obligados a comprar geodatos, liberando a los usuarios finales de la necesidad de adquirir geodatos y, por ende, de incurrir en compromisos restrictivos vinculados a derechos de autor y licencias. Su éxito radica en el amplio crecimiento de la comunidad global, dispuesta a participar en la recopilación y mantenimiento de geodatos, con el desarrollo de código abierto. Las investigaciones sobre OSM han demostrado que, en algunas partes del mundo, sus geodatos son más completos y más precisos (Jokar Arsanjani et al., 2015).

La información geográfica de libre acceso ha suscitado un gran interés entre investigadores y profesionales por la utilidad de estos datos, tanto por sus limitaciones como por su potencial. Algunos de los usos y aplicaciones más comunes de los geodatos son: enrutamiento, respuesta de emergencias y simulación de evaluación, logística, desarrollo de servicios basados en localización (LBS), entre otras (Jokar Arsanjani et al., 2015).

Este proyecto de OSM consta de su propio mapa interactivo en línea, disponible en: <https://www.openstreetmap.org/#map=5/4.632/-74.299> (OpenStreetMap, n.d.), así como su propio sitio web oficial llamado Wiki OpenStreetMap en

https://wiki.openstreetmap.org/wiki/ES:P%C3%A1gina_principal (*OpenStreetMap Wiki*, n.d.).

En este sitio web, los miembros de la comunidad contribuyen con información, tutoriales, guías y documentación relacionada con el proyecto. Por ejemplo, se proporciona información sobre cómo mapear diferentes tipos de características geográficas, información sobre etiquetas y etiquetado, recursos para desarrolladores que deseen utilizar los datos en sus aplicaciones, entre otros aspectos.

4.2.14. *OpenRouteService (ORS)*

Fue creado en 2008 como un sitio de enrutamiento en línea y de código abierto, que utiliza datos geospaciales para proporcionar rutas y direcciones para diferentes modalidades de transporte, como coches, peatones y bicicletas basado en estándares abiertos y geodatos abiertos (*ORS Maps*, n.d.). Este servicio basa sus datos a partir de los datos de OpenStreetMap (OSM). Actualmente, ORS tiene los siguientes servicios:

- Directorio de servicios: proporciona acceso a un directorio en línea para encontrar la ubicación de un lugar, producto o servicio específico.
- Servicio de localización útil: provee geocodificación inversa/geocodificación, transforma una descripción de un lugar, como un nombre de lugar, dirección o código postal, en una descripción normalizada de la ubicación con una geometría point.
- Servicio de rutas: determina las rutas de viaje e información de navegación de acuerdo a diversos criterios. Por ejemplo, se ha realizado para coches, varias opciones para evitar túneles, enrutamiento de sillas de ruedas, uso del tráfico en tiempo real para Alemania, bicicletas, etc.
- Servicio de análisis de accesibilidad (AAS) calcula un polígono que representa el área accesible dentro de una distancia de tiempo determinada en función de una red de calles alrededor de una ubicación determinada.

- Servicio de ruta de emergencia: se pueden especificar áreas que serán evitadas por la ruta.

La utilización de los servicios de ORS, se realiza por medio de una API, una interfaz de programación que permite acceder a las funcionalidades de ORS. Además del enrutamiento multimodal, proporciona características, como el cálculo de rutas alternativas entre ubicaciones, personalización de los vehículos, isoyetas y zonas de acceso, acceso a datos de elevación (importantes para el ciclismo), entre otras opciones (*Openrouteservice*, n.d.).

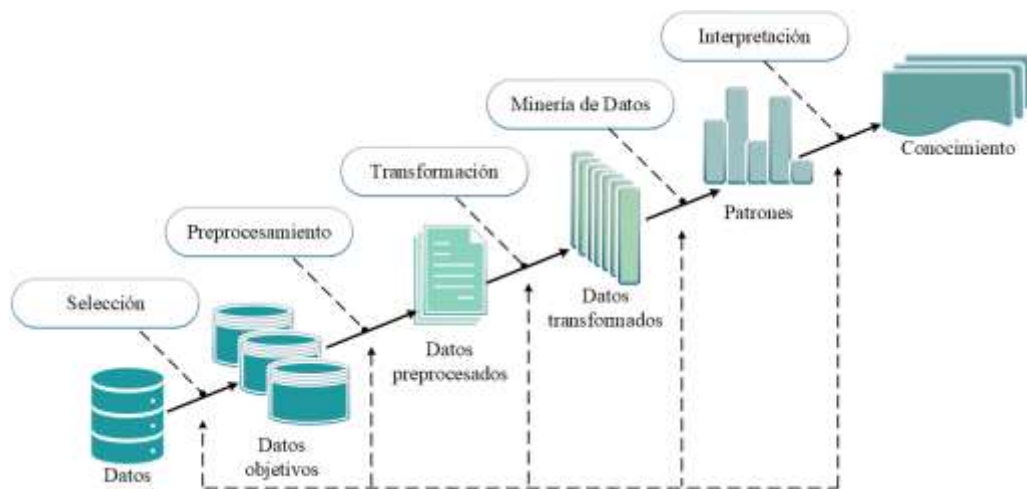
4.2.15. Proceso de Descubrimiento de Conocimiento en Bases de Datos

En 1989, Piatetsky-Shapiro introdujo por primera vez el concepto de Descubrimiento de Conocimiento en Bases de Datos (KDD sus siglas en inglés) como un proceso destinado a descubrir y extraer información y patrones válidos, potencialmente útiles, dentro de un conjunto de datos, los cuales serán utilizados en análisis posteriores. Este descubrimiento de nuevo conocimiento tiene una amplia aplicación en diversas áreas, como la detección de tendencias del mercado, la identificación de relaciones implícitas existentes entre síntomas y enfermedades, la determinación de perfiles de clientes fraudulentos, entre otros (Timarán Pereira et al., 2016).

Este proceso contempla un conjunto de etapas definidas para el tratamiento de los datos antes de aplicar las diferentes técnicas de minería de datos que son: selección de datos, preprocesamiento de los datos, transformación de los datos, minería de datos e interpretación de los datos. A continuación, se muestra el diagrama del proceso KDD y se definen cada una de las etapas de éste (Timarán Pereira et al., 2016):

Figura 6

Proceso de Descubrimiento de Conocimiento en Bases de Datos.



Nota. Tomado de *Timarán Pereira et al., 2016*

4.2.15.1. Selección. En esta primera etapa, se seleccionan, ya sea todo el conjunto de datos o una muestra representativa, sobre la cual se va a realizar el proceso de descubrimiento. Esto implica la comprensión de la información que se pretende extraer, las preguntas se buscan responder o los problemas que se intentan abortar (Timarán Pereira et al., 2016).

4.2.15.2. Preprocesamiento. Una vez seleccionadas las fuentes de datos, se procede a analizar la calidad de los mismos mediante la aplicación de operaciones básicas como la remoción de datos ruidosos, el tratamiento para el manejo de datos desconocidos, datos nulos, datos duplicados y datos inconsistentes. Los datos ruidosos son aquellos valores que se encuentran fuera del rango de valores esperados. Estos errores suelen originarse principalmente a errores humanos, información no disponible a tiempo y a fuentes heterogéneas de datos.

Durante el proceso de limpieza, estos valores se omiten, se reemplazan por un valor predeterminado o por el valor más cercano. También se aborda la corrección de errores de transcripción, como problemas de mayúsculas/minúsculas (Timarán Pereira et al., 2016) (Sang,

n.d.). Es importante destacar que esta fase, junto con la de selección de datos, consume aproximadamente el 70% del esfuerzo en los proyectos de Data Mining (Sang, n.d.).

4.2.15.3. Transformación. En esta etapa, se busca la identificación de características valiosas para representar los datos dependiendo del objetivo del proceso. Para lograrlo, se emplean métodos de reducción de dimensiones o de transformación para disminuir eficazmente el número efectivo de variables bajo consideración, con el fin de trabajar con menos datos y obtener los mismo resultados (Timarán Pereira et al., 2016) (Sang, n.d.).

Los métodos de reducción de dimensiones pueden simplificar la estructura de datos de dos maneras distintas. Por un lado, pueden actuar horizontalmente, eliminando tuplas idénticas en una tabla de base de datos, como se ilustra en la **Tabla 3**. Por otro lado, pueden operar verticalmente, eliminando atributos que resulten insignificantes o redundantes para el problema en cuestión (Timarán Pereira et al., 2016). Asimismo, se utilizan otras técnicas de reducción como la normalización de datos, la segmentación de datos, la compresión de datos, el muestreo, entre otras.

Tabla 3

Ejemplo de repetición de tuplas.

Identificador	Nombre	Dirección	Centro
24.567	Poch	Roca, 33-1	1
32.456	Martínez	Travesera, 222	2
24.567	Poch	Roca, 33-1	1
33.440	Sala	Diagonal 556	1

Nota: Tomado de Sang, n.d.

4.2.15.4. Minería de Datos. Esta etapa implica la aplicación de una serie de algoritmos y técnicas de minería de datos con el objetivo de explorar y descubrir patrones, tendencias y relaciones dentro de los datos. Esto se logra mediante la aplicación de técnicas como la clasificación, la clusterización, la identificación de asociaciones, entre otras (Timarán Pereira et al., 2016).

Las técnicas de minería de datos generan modelos que pueden ser de dos tipos: predictivos o descriptivos. Los modelos predictivos buscan estimar valores futuros o desconocidos, mientras que los modelos descriptivos identifican patrones que explican o resumen los datos existentes (Timarán Pereira et al., 2016).

En consecuencia, el proceso de búsqueda y elección de un algoritmo de minería de datos es crucial porque involucra la toma de decisiones sobre las técnicas que mejor se ajusten al análisis, así como la configuración de los modelos y la calibración de los parámetros relevantes. Esta elección puede ser influenciada por factores como la complejidad de los datos, el tamaño, los recursos computacionales disponibles y la naturaleza de las preguntas que se intentan responder (Sang, n.d.).

4.2.15.5. Interpretación. En la etapa final de la estrategia KDD, se procede a la interpretación de los patrones descubiertos, y se considera la posibilidad de volver a las etapas anteriores para realizar iteraciones y mejoras adicionales. Esta fase es crucial para transformar los resultados de la minería de datos en conocimientos valiosos y accesibles para los usuarios (Timarán Pereira et al., 2016).

La interpretación de los patrones se lleva a cabo mediante la creación de visualizaciones, la generación de insights o percepciones clave. Un aspecto fundamental en esta etapa es la traducción de los patrones significativos en un lenguaje comprensible para los usuarios (Timarán Pereira et al., 2016).

5. Estrategia KDD para la Matriz de Distancia

En esta sección, se describe exhaustivamente el proceso de descubrimiento de conocimiento en bases de datos aplicado a los documentos proporcionados por la empresa Cooperativa de Impresores del Oriente. A continuación, se presenta la identificación de los clientes más representativos para la empresa. Por último, se procede a la generación de la matriz de distancia a través de la geolocalización, necesaria para abordar el problema de ruteo de vehículos.

5.1. Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD)

La Gerencia de la Cooperativa de Impresores del Oriente facilitó los datos de la asignación de rutas y vehículos de su actividad. En una revisión preliminar de esta información, se evidenciaron errores en la digitalización. Estos errores incluyen nombres de clientes escritos en mayúscula y minúscula, direcciones de entrega inexistentes, datos irrelevantes, entre otros. Debido a estas irregularidades, se procede a realizar el proceso de descubrimiento de conocimiento a estas bases de datos.

5.1.1. Selección de los Datos

Esta primera etapa de la estrategia KDD, se centró en la información suministrada por la Gerencia en relación a su actividad diaria durante los años: 2019, 2020 y 2021. Estas bases de datos se presentan en forma de tablas en Excel con las siguientes columnas: fecha de la facturación, fecha que cliente exige (dado el caso), hora de facturación, número de factura, bodega de la cual sale el pedido, zona, tipo de entrega, nombre del cliente, nombre de la empresa, dirección, hora de llegada, hora de salida, carro utilizado, fecha 1, recorrido, fecha definitiva en la que sale el pedido, días, ajuste del horario en la mañana, ajuste del horario en la tarde, horas, estado, rango de tiempo, mes y valor de la factura.

5.1.2. Preprocesamiento

La información es ingresada manualmente por el responsable de logística de la empresa y no se verifica o cruza de ninguna manera con una base de datos, lo que resulta en la aparición de errores de digitalización. Durante el análisis, se detectaron datos inconsistentes, tales como:

- Nombres de clientes escritos en mayúscula y minúscula para hacer referencia al mismo comprador.
- Nombres con apellidos y sin apellidos para hacer referencia al mismo comprador.
- Nombre del cliente, el establecimiento o del solicitante para hacer referencia al mismo comprador.
- Mismo nombre escrito de manera diferente
- Direcciones de entrega inexistentes
- Diferencias en las direcciones de un mismo cliente de un pedido a otro, verificando que se encuentran vigentes en la misma dirección.

Durante el proceso de limpieza, se abordaron estos errores relacionados con las variables categóricas mediante la unificación de los nombres y direcciones en cada año de manera manual. En el caso de las direcciones, se realizó una búsqueda minuciosa en el software de Google Maps y se verificó que los clientes estuvieran en el mismo lugar a la fecha actual. En la **Tabla 4**, se pueden observar algunos de los errores mencionados previamente para un mismo cliente y la forma cómo se llevó a cabo su pretratamiento.

Tabla 4

Errores más frecuentes en el ingreso de datos.

Bodega (7), depósito (1)	Nombre del cliente	Dirección	Vehículo
7	Helena Gimenez	Cra 16 #15-36	NKR
7	elena isabel gimenez	Cra 16 #15-46	NKR
7	Dcarton	Cra 16 #15-46	NKR
Pretratamiento			
7	Dcarton	Cra 16 #15-46	NKR

Al finalizar esta etapa, se llevó a cabo una retroalimentación con la Gerencia del procedimiento aplicado a las bases de datos, lo que generó interés de su parte en la información debidamente organizada, verificada y unificada. En respuesta, la Gerencia expresó su consideración de que este enfoque equivalía a una *auditoría interna* y de control, la cual utilizaron en la implementación de un nuevo sistema ERP SAP adquirido por la entidad.

5.1.3. Transformación

Durante esta etapa, se llevó a cabo la reducción de dimensiones tanto en filas como en columnas. Dado que las variables contenidas en estas bases de datos eran principalmente categóricas, la transformación de los datos fue limitada. Por lo tanto, la reducción horizontal se efectuó eliminando las filas de tuplas que contenían datos incompletos. En cuanto a la reducción vertical, se procedió eliminando las columnas con variables que no tenían relevancia para la presente investigación, como, por ejemplo: el valor de las facturas, número de factura, las observaciones realizadas, entre otras.

5.1.4. Minería de Datos

Esta etapa se centró en la generación de modelos descriptivos utilizando estadísticas descriptivas simples. El objetivo principal fue la identificación de patrones de interés dentro de la

empresa, que sirvieran de base para la caracterización del problema de ruteo de vehículos más adecuado para el caso. Es importante destacar que la minería de datos se realizó de manera breve y concisa. Para realizar este análisis, se empleó el lenguaje de programación *Python* en la interfaz gráfica de usuario (GUI) *Anaconda Navigator*, analizando los datos de cada año de forma individual y, al mismo tiempo, obtener una visión completa de la situación de la empresa.

5.1.5. Interpretación

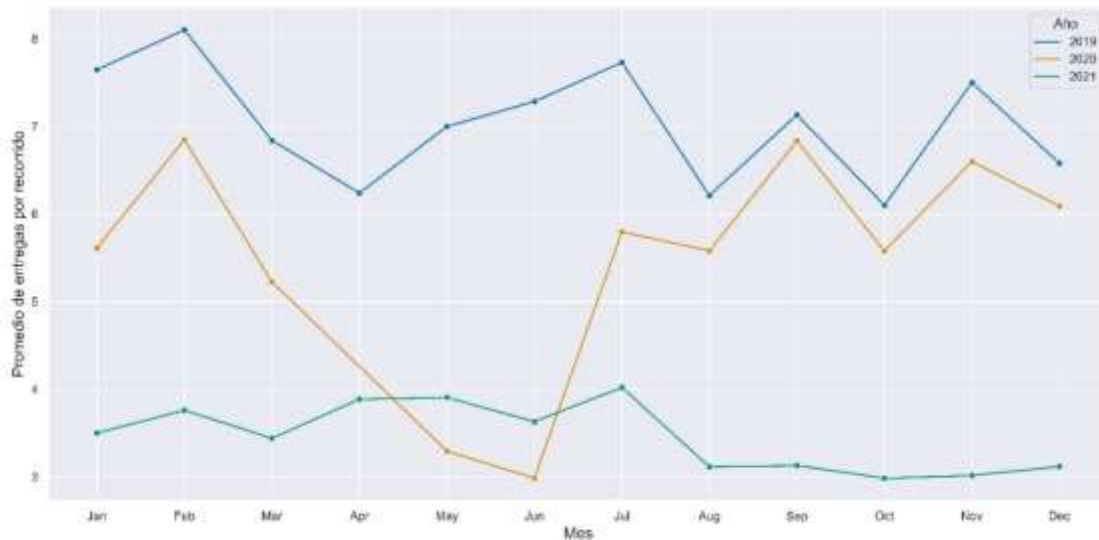
La etapa final se abordó por medio de la creación de visualizaciones e insights para traducir los patrones descubiertos. Estos patrones sirvieron de base para identificar la necesidad de la implementación de un modelo en la empresa. A continuación, se presentan algunos:

Inicialmente, se llevó a cabo un análisis del promedio de entregas mensuales por recorrido, presentado en la **Figura 7**. En esta gráfica, se observó que, durante el año 2019, el promedio de entregas superó al de los años siguientes. Este fenómeno puede deberse a que, en el sector gráfico, las elecciones territoriales representan una oportunidad para incrementar las ventas, y este año no fue la excepción, como se aprecia en los meses de primera y segunda vuelta electoral. En contraste, para el año 2020, se evidenció un decrecimiento en el promedio de entregas a partir del mes de febrero, que se pronunció debido a la pandemia de covid-19, prolongándose hasta el mes de junio. Sin embargo, a partir de ese momento, se observó una recuperación casi inmediata en el promedio de entregas.

Finalmente, para el año 2021, se evidenció un comportamiento diferente a los años anteriores, efecto de crisis de los contenedores que estalló a raíz de los problemas logísticos a nivel mundial ocasionando incremento de costos de materias primas e insumos y fletes, además de la escasez de dichos recursos. Como consecuencia el promedio de entregas se redujo drásticamente, llegando a solo 2 pedidos al mes.

Figura 7

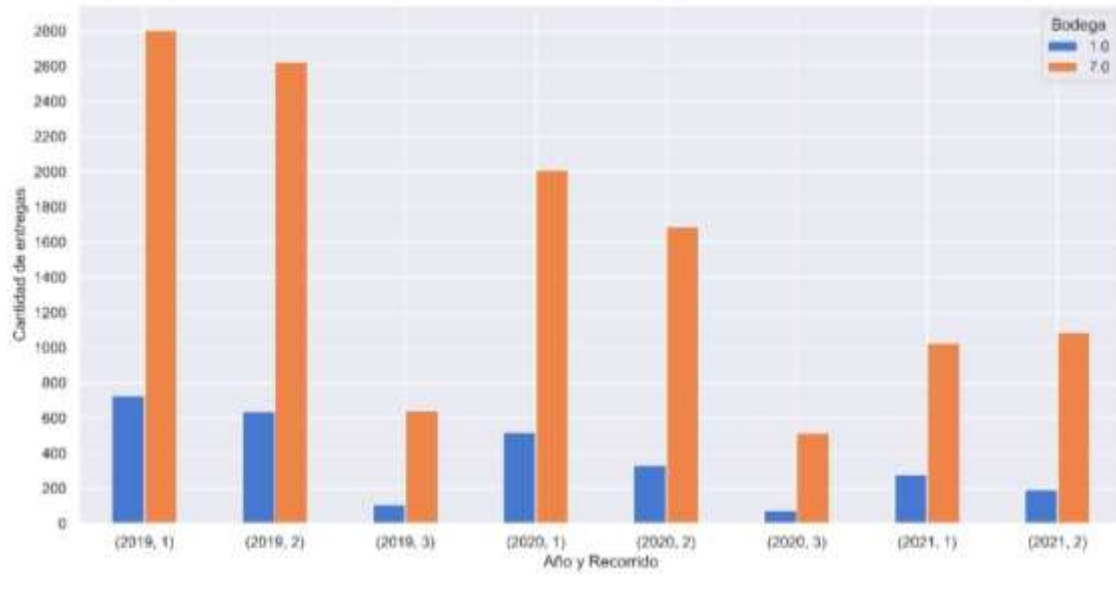
Promedio de entregas por recorrido al mes.



Teniendo en cuenta que la empresa cuenta con una bodega (7) y un depósito (1), se procedió a realizar de manera gráfica (**Figura 8**) el comportamiento de los pedidos que salen de cada uno de ellos en los diferentes recorridos hechos a lo largo del día. A partir de allí, se logra observar que la mayor parte de los pedidos realizados salen de la bodega principal (7) y pueden llegar a ser hasta cuatro veces más que los pedidos que provienen del depósito (1). Asimismo, se pudo observar que, para el primer recorrido, que corresponde al horario diurno, la cantidad de pedidos realizados tanto de la bodega principal como del depósito, son considerablemente mayor que los pedidos que se entregan en la ruta de la tarde. Esto puede significar que, para satisfacer la demanda de la tarde, es necesaria la utilización de múltiples vehículos. Estas observaciones sugieren la necesidad de optimizar el uso de los vehículos, con el objetivo de atender todos los pedidos con la menor cantidad de vehículos posible.

Figura 8

Cantidad de facturas por recorrido y por bodega.

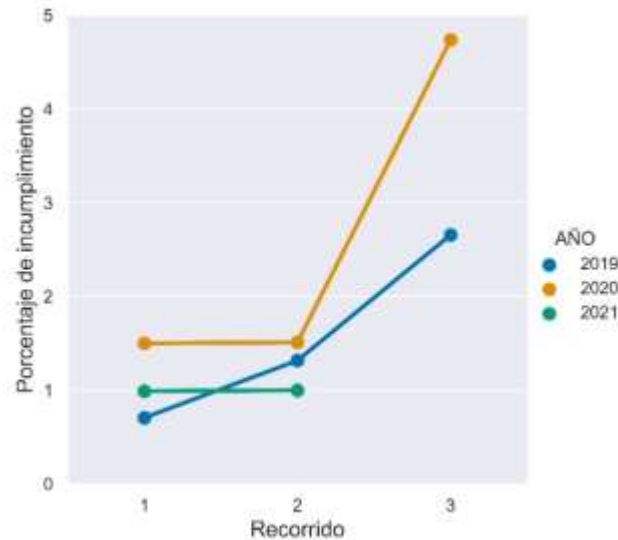


Un patrón relevante que se observó fue en el porcentaje de incumplimiento a través de los años, en relación con la cantidad de recorridos realizados al día. Al analizar la **Figura 9**, se puede observar que cuando se realizan los dos recorridos habituales, el porcentaje de incumplimiento no varía significativamente para los años 2020 y 2021. No obstante, en 2019, se observa una variación mayor en comparación con los otros años, aunque no supera el 1%

En contraste, cuando se lleva a cabo un tercer recorrido, el porcentaje de incumplimiento experimenta un aumento de hasta el 3,22%. Esto sugiere que, en el caso de recorridos adicionales o de “emergencia”, es probable que no estén tomando las precauciones necesarias, lo que resulta en un mayor número de incumplimientos. Este patrón plantea preocupaciones, ya que, durante estos tres años, el promedio se sitúa en un 1,8% lo que podría indicar una posible insatisfacción por parte de los clientes.

Figura 9

Porcentaje de incumplimiento por recorrido.



En resumen, luego del análisis detallado de algunos de los patrones y datos descubiertos, se entendió el comportamiento de las entregas a lo largo del tiempo, la influencia de eventos externos, la relación entre la cantidad de recorridos y el porcentaje de incumplimiento. Estos hallazgos respaldan la imperante necesidad de optimizar la asignación de vehículos y mejorar los procedimientos de entrega para garantizar la satisfacción del cliente y la eficiencia operativa en el futuro. En definitiva, el proceso de descubrimiento de conocimiento en bases de datos se ha revelado como un paso crucial en el desarrollo de la presente investigación, para comprender y poder mejorar estos aspectos del negocio.

5.2. Clientes Clave

El proceso de identificación de clientes clave o estratégicos para la empresa Coimpresores del Oriente, desempeñó un papel crítico en el éxito del presente proyecto. Dado que la empresa mantiene relaciones comerciales con aproximadamente 150 clientes, resultó imperativo realizar un riguroso análisis para determinar cuáles de ellos generaban un mayor volumen de compras en

la empresa. Este análisis se basó principalmente en la cantidad de facturas emitidas para cada cliente a lo largo de cada año de estudio.

Al examinar el número de facturas, se descubrió que había casos en los que diferentes clientes compartían la misma dirección de entrega. Tras consultar con la empresa, se confirmó que algunos clientes optan por recibir sus pedidos en ubicaciones diferentes a su dirección oficial. Ante esta información, se decidió que los nodos en la matriz de distancia no representan a los clientes, sino a las direcciones de entrega.

Para llevar a cabo este proceso de identificación, se utilizó el lenguaje de programación *Python* a través de la interfaz gráfica de usuario (GUI) *Anaconda Navigator*. El proceso inició con el cálculo de las facturas generadas por dirección para cada año. Posteriormente, se emplearon técnicas como, el promedio ponderado y el Principio de Pareto. En el caso del método de promedio ponderado, se aplicó para dar prioridad a las direcciones con entregas más recientes. Esto se realizó asignando valores ponderados de la siguiente manera:

- Para las facturas generadas en el año 2019, se les asignó un valor de ponderación del 15%.
- En el caso del año 2020, se les asignó un valor del 30% a las facturas correspondientes.
- Finalmente, para el año más reciente, 2021, se le asignó un valor de ponderación del 55%.

Este enfoque permitió destacar y priorizar las direcciones que han tenido entregas más recientes en la empresa.

Además, en el proceso se aplicó el Principio de Pareto, una herramienta valiosa en diversos contextos, particularmente cuando se trabaja con grandes volúmenes de datos cuantitativos y cualitativos, permitiendo una focalización efectiva de los esfuerzos en los datos críticos y de mayor valor. En este caso específico, la Ley de Pareto se basó en la premisa de que el 20% del esfuerzo genera el 80% de los resultados, lo que significa que aproximadamente el 20% de los clientes

generan alrededor del 80% de los ingresos en muchas empresas. A partir de lo anterior, se calculó una tabla con las frecuencias relativas y absolutas, a partir del promedio ponderado realizado anteriormente, para cada dirección de cada año con el propósito de identificar las principales direcciones dentro del 80%. Finalmente, se contrastaron los tres años para encontrar los clientes que se mantenían constantes de un año a otro. Como resultado de este análisis, se logró determinar la existencia de 19 clientes clave o fijos que ejercían un impacto significativo en el negocio a lo largo de los años estudiados.

Figura 10

Ubicación de clientes fijos identificados.



Nota: Tomado de Google Earth

5.3. Matriz

En la resolución de cualquier problema de enrutamiento de vehículos, resulta fundamental disponer de una matriz de distancias que abarque todos los puntos de interés, tales como ubicaciones de clientes, almacenes, depósitos, entre otros. Para lograr un cálculo efectivo de esta matriz de distancias en relación con la muestra de clientes clave, se llevaron a cabo una serie de pasos previos, los cuales se describen a continuación:

5.3.1. *Software de Servicio de Mapas*

En la búsqueda de un software de servicio de mapas, se exploraron las características ofrecidas por el software de Google Maps y el software de OpenStreetMap. En la siguiente tabla, se presenta una comparación entre los softwares mencionados:

Tabla 5

Características de Google Maps y OSM.

Característica	Google Maps	OpenStreetMap (OSM)
Origen de los datos	Recopilados por Google y sus socios.	Mapeo colaborativo por voluntarios en todo el mundo
Acceso	Servicio comercial con suscripción por número de consultas	Código abierto y gratuito
API	Ofrece una amplia gama de APIs para integrar mapas	Ofrece una sola API: OpenRouteService (ORS)
Búsqueda	Direcciones	Coordenadas: latitud y longitud

Este análisis, permitió la elección informada del software, priorizando la característica de acceso gratuito que ofrece OpenStreetMap (OSM). Después de la elección del software, se realizó un ejercicio adicional para evaluar la calidad de los datos proporcionados por OSM. En la **Tabla 6**, se encuentran los clientes utilizados para el ejercicio

Tabla 6

Cientes utilizados para el ejercicio.

N.	Nombre	Dirección	Coordenadas (longitud, latitud)
1	Bodega	Parque Industrial de Bucaramanga	(7.110865, -73.161385)
2	Dcarton	Cra 16 #15-46	(7.13164, -73.12949)
3	UIS	Portería Carrera 30	(7.13926, -73.11725)

Se calculó una matriz de distancia para cada uno de los softwares en la **Tabla 7** y la interfaz de búsqueda de ubicaciones en la

Figura 11.

Tabla 7

Diferencia de las distancias entre Google Maps y OSM.

Google Maps				OpenRouteService				Diferencias			
Direcciones	1	2	3	Direcciones	1	2	3	Direcciones	1	2	3
1	0	11	12,3	1	0	11	12	1	0	0	0,3
2	12	0	2	2	12	0	2,3	2	0	0	-0,3
3	14,2	2,6	0	3	14	2	0	3	0,2	0,6	0

Figura 11

Interfaz de Google Maps y OSM.



Nota: Tomado de *Google Maps* y *OSM* respectivamente.

Finalmente, en la **Tabla 7**, también se observa el cálculo de la disparidad entre ambas matrices, lo que condujo a la conclusión de que, a pesar de ser un software de código abierto, OpenStreetMap mostró variaciones que no resultaron significativas. Este ejercicio permitió verificar la precisión y confiabilidad de la información del software.

5.3.2. Geolocalización

Partiendo del hecho de que el software OSM emplea la geolocalización para determinar las distancias entre dos ubicaciones, resultó imperativo trabajar con datos geoespaciales para llevar a cabo el cálculo de la matriz de distancias. Este procedimiento, se realizó manualmente utilizando el software de Google Maps para los clientes clave que habían sido identificados en etapas previas.

5.3.3. *Matriz*

Gracias a los pasos detallados anteriormente, se pudo generar la matriz de distancia para 14 de los 19 clientes fijos identificados previamente. Esta decisión se tomó debido a que, en el proceso, se observó que una matriz de con dimensiones superiores a 14x14 se requería de un tiempo computacional considerablemente alto (Apéndice B).

6. Modelo de Optimización para el Problema Ruteo de Vehículos de Flota Fija Heterogénea con Restricciones de Carga Bidimensional (2L-HFFVRP)

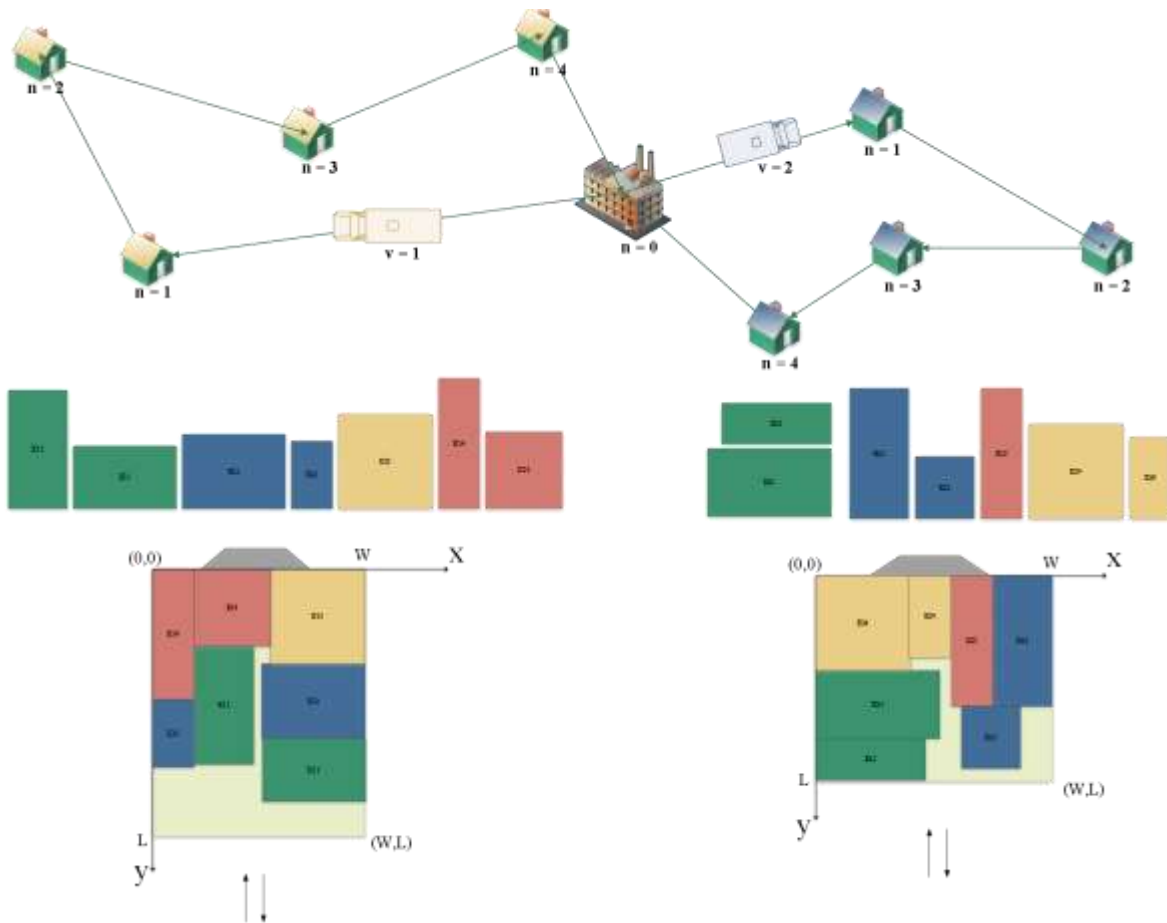
El problema 2L-HFFVRP puede verse como una variante del problema VRP capacitado (CVRP) o un caso específico del HFVRP en el que la capacidad de los vehículos es diferente, con una cantidad limitada de vehículos. Su objetivo es minimizar la distancia total recorrida por todos los vehículos. Integrando así dos problemas de optimización combinatoria: el HFFVRP y el 2L-CVRP. Consta de una restricción de carga bidimensional, teniendo en cuenta la longitud y el ancho de los artículos demandados, los cuales, pueden ser rotados 90° al ser cargados al vehículo. Por otro lado, se considera carga secuencial, que consiste en cargar los artículos de forma contraria al orden de los clientes a visitar. Este problema de carga no orientada-secuencial no ha sido abordado en la literatura de manera conjunta. No obstante, al permitir la rotación de los artículos se eliminan soluciones subóptimas, pero al ser una carga y descarga secuencial dificulta la búsqueda de la solución óptima.

Para esta investigación, se adopta un sistema de coordenadas cartesianas con origen en la esquina delantera izquierda del compartimento de carga del vehículo y se coloca la puerta del vehículo por donde se realizan las operaciones de carga y descarga en el lado entre las coordenadas (0, L) y (W, L), como se muestra en la **Figura 12**.

Se formuló el presente modelo de optimización tomando en cuenta los datos e información proporcionados por la empresa Coimpresores. Se identificaron varios problemas potenciales para abordar, y se llegó finalmente al problema actual. La idealización abstracta de un problema real requiere de múltiples supuestos para la simplificación del modelo (Hillier y Lieberman, 2010). Por lo tanto, se relajaron algunas restricciones en el modelo. En la siguiente figura se ilustra un ejemplo del problema 2L-HFFVRP y el escenario con nueve clientes, uno de estos clientes va a corresponder a la bodega en dónde los vehículos paran a terminar de completar los pedidos de los clientes.

Figura 12

Ejemplo del problema 2L-HFFVRP con carga secuencial.



6.1. Definición Formal del Problema

Formalmente, el 2L-HFFVRP, basado en notaciones similares utilizadas en la literatura, se puede definir como un grafo dirigido $G = (N, A)$, en donde N es el conjunto de vértices o nodos que contiene n clientes en donde $n \in N = \{0, 1, \dots, n\}$, incluyendo el depósito como un cliente, y la planta (n_0), mientras que $A = \{(i, j) \mid i, j \in N, i \neq j\}$ denota el conjunto de arcos posibles. Cada arco $(i, j) \in A$ posee un costo no negativo asociado a la distancia d_{ij} que equivale a la distancia entre el cliente i y el cliente j . En la planta se encuentra una flota fija de v vehículos heterogéneos, es decir una cantidad limitada de vehículos de características diferentes. Cada vehículo v tiene una capacidad de peso máxima P_v y un espacio de carga bidimensional de longitud L_v y anchura W_v . La demanda de cada cliente i ($i = 1, \dots, n$) se define como un conjunto m_i artículos bidimensionales rectangulares denotado por $E = \{1, \dots, \sum_{i=1}^n m_i\}$ y un peso total P_i conocido. Cada artículo $E_{ia} \in E$ ($a = 1, 2, \dots, m_i$) tiene una longitud L_{ia} , y un ancho W_{ia} .

Los artículos no pueden superponerse. Además, los artículos pueden ser rotados 90° durante el proceso de carga. Por otro lado, la carga de cada vehículo debe permitir una fácil descarga, así pues, que al descargar los artículos de un cliente i , no se desplacen los artículos del cliente j ; es por ello que se considera una restricción de secuencia de descarga, es decir que al visitar un cliente i debe ser posible descargar sus artículos realizando únicamente movimientos paralelos a la longitud del vehículo. De forma implícita, se busca reducir el tiempo de descarga y por ende el tiempo que se demora un vehículo de un cliente a otro, dado que la reorganización de la carga en las instalaciones del cliente puede resultar difícil, lenta o incluso imposible debido al peso y el tamaño de los artículos o a las limitaciones de las carretillas elevadoras.

El objetivo de este problema de ruteo es encontrar el conjunto de rutas que atienden a todos los clientes minimizando el costo y cumpliendo con cada una de las restricciones; considerando un esfuerzo computacional razonable, que permita manejar instancias de gran tamaño y obtener resultados útiles para la distribución en el mundo real, partiendo de los siguientes supuestos:

1. Todo vehículo inicia y termina su recorrido en la planta.
2. Cada cliente debe ser visitado una única vez.
3. Todos los artículos de los clientes asignados a la misma ruta deben cargarse en el vehículo correspondiente, no se permiten entregas fraccionadas.
4. Ningún artículo puede superponerse a otro.
5. Los artículos no pueden exceder la capacidad: longitud y ancho de cada vehículo.
6. La demanda de los productos será considerada en m^2 .

6.2. Modelo Matemático

De acuerdo con la descripción formal, se propone un modelo matemático para representar el 2L-HFFVRP. La notación utilizada en el modelo, incluidos índices, parámetros y variables es presentada en la **Tabla 8**.

Tabla 8

Conjunto de índices, parámetros y variables.

Índices	
$i, j, g \in N = \{0, 1, 2, \dots, n\}$	Conjunto de vértices/nodos desde la planta ($n = 0$) hasta los clientes ($n > 0$).
$v \in V = \{1, 2, \dots, v\}$	Conjunto de vehículos.

$e, f \in E = \left\{ 1, \dots, \sum_{i=1}^n m_i \right\}$ Conjunto de artículos demandados por los clientes.

Parámetros

n	Número de clientes
v	Número de vehículos
d_{ij}	Distancia de cada cliente i a cada cliente j .
P_V	Capacidad de peso de cada vehículo v .
W_v	Ancho de la superficie de carga de cada vehículo v .
L_v	Longitud de la superficie de carga de cada vehículo v .
m_i	Número de artículos demandados por cada cliente i .
P_i	Peso total demandado por cada cliente i .
W_e	Ancho de los artículos demandados por el cliente i .
L_e	Longitud de los artículos demandados por el cliente i .

Variables

z_{ij}^v	Variable binaria que indica si el vehículo v viaja del cliente i al cliente j ($z_{ij}^v = 1$) o no ($z_{ij}^v = 0$).
x_e	Variable que indica la coordenada x de la esquina delantera izquierda del artículo e en la carga del vehículo.
y_e	Variable que indica la coordenada y de la esquina delantera izquierda del artículo e en la carga del vehículo.

Ω_e^v	Variable binaria que indica si el artículo e es transportado por el vehículo v ($\Omega_e^v = 1$) o no ($\Omega_e^v = 0$)
r_e	Variable binaria que indica si el artículo e ha sido rotado ($r_e = 1$) o no ($r_e = 0$).
α_{ef}	Variable binaria que indica si el artículo e es colocado completamente detrás del artículo f ($\alpha_{ef} = 1$) o no ($\alpha_{ef} = 0$).
β_{ef}	Variable binaria que indica si el elemento e es colocado completamente enfrente del artículo f ($\beta_{ef} = 1$) o no ($\beta_{ef} = 0$).
γ_{ef}	Variable binaria que indica si el artículo e es colocado completamente en el lado izquierdo del artículo f ($\gamma_{ef} = 1$) o no ($\gamma_{ef} = 0$).
δ_{ef}	Variable binaria que indica si el artículo e es colocado completamente en el lado derecho del artículo f ($\delta_{ef} = 1$) o no ($\delta_{ef} = 0$).

Tal como se indicó previamente, el 2L-HFFVRP se compone de dos problemas de optimización combinatoria: el HFFVRP y el 2L-CVRP. En consecuencia, se procederá inicialmente a presentar la formulación propuesta para el modelo matemático del HFFVRP. Posteriormente, se presentará la formulación propuesta para el problema con restricción de carga bidimensional.

La función objetivo se expresa como:

$$\text{Minimizar } \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} d_{ij} z_{ij}^v, \quad (1)$$

Sujeta a las siguientes restricciones:

$$\sum_{j \in N} \sum_{v \in V} z_{ij}^v = 1, \forall i \in N \setminus \{0\}, \quad (2)$$

$$\sum_{i \in N} \sum_{v \in V} z_{ij}^v = 1, \forall j \in N \setminus \{0\}, \quad (3)$$

$$z_{ij}^v + z_{ji}^v \leq 1, \forall i, j \in N \setminus \{0\}, i \neq j, \forall v \in V, \quad (4)$$

$$\sum_{j \in N \setminus \{0\}} z_{0j}^v = \sum_{i \in N \setminus \{0\}} z_{i0}^v, \forall v \in V, \quad (5)$$

$$\sum_{i \in N} z_{ig}^v - \sum_{j \in N} z_{gj}^v = 0, \forall g \in N \setminus \{0\}, \forall v \in V, \quad (6)$$

$$\sum_{i \in N} \sum_{j \in N} P_i z_{ij}^v \leq P_v, \forall v \in V, \quad (7)$$

$$\sum_{i \in N} \sum_{v \in V} z_{0j}^v \leq V, \quad (8)$$

$$z_{ij}^v \in \{0,1\}, (i,j) \in A, \forall v \in V, \quad (9)$$

Con la función objetivo (1) se busca minimizar la distancia total recorrida por los vehículos al visitar a los clientes. Las restricciones (2) y (3) aseguran que cada cliente sea visitado exactamente una sola vez. La restricción (4) asegura que solo uno de los arcos (i,j) o (j,i) sea recorrido por un vehículo v dado, ayudando a eliminar sub-toures. La restricción (5) asegura que todos los vehículos que parten desde la planta tengan que regresar a la planta. La restricción (6) indica que todo vehículo que visita a un cliente debe salir de él. La restricción (7) asegura que la

capacidad de peso de los vehículos no sea excedida. La restricción (8) asegura que, debido al número fijo de vehículos, el número de rutas no pueda exceder al número de vehículos. La restricción (9) es la restricción del dominio de la variable de decisión.

A continuación, se presenta la formulación propuesta para el problema con restricción de carga de bidimensional no orientada-secuencial:

$$x_e + W_e(1 - r_e) + L_e r_e \leq W_v + W_v(1 - \Omega_e^v), \forall e \in E, \forall v \in V, \quad (10)$$

$$y_e + L_e(1 - r_e) + W_e r_e \leq L_v + L_v(1 - \Omega_e^v), \forall e \in E, \forall v \in V, \quad (11)$$

$$x_e + W_e(1 - r_e) + L_e r_e \leq x_f + W_v(1 - \gamma_{ef}), \forall e, f \in E, e \neq f, \forall v \in V, \quad (12)$$

$$x_f + W_f(1 - r_f) + L_f r_f \leq x_e + W_v(1 - \delta_{ef}), \forall e, f \in E, e \neq f, \forall v \in V, \quad (13)$$

$$y_e + L_e(1 - r_e) + W_e r_e \leq y_f + L_v(1 - \alpha_{ef}), \forall e, f \in E, e \neq f, \forall v \in V, \quad (14)$$

$$y_f + L_f(1 - r_f) + W_f r_f \leq y_e + L_v(1 - \beta_{ef}), \forall e, f \in E, e \neq f, \forall v \in V, \quad (15)$$

$$\alpha_{ef} + \beta_{ef} + \gamma_{ef} + \delta_{ef} \geq S_{ev} + S_{fv} - 1, e, f \in E, p < q, v \in V, \quad (16)$$

$$\Omega_e^v, r_e \in \{0,1\}, \forall e \in E, \forall v \in V, \quad (17)$$

$$\alpha_{ef}, \beta_{ef}, \gamma_{ef}, \delta_{ef} \in \{0,1\}, \forall e, f \in E, \quad (18)$$

$$x_e \in \{0,1, \dots, W_v\}, y_e \in \{0,1, \dots, L_v\}, \forall e \in E, \forall v \in V, \quad (19)$$

Las restricciones (10) y (11) precisan que cada artículo debe ser colocado sobre el espacio de carga. Las restricciones de la (12) a la (15) aseguran que dos artículos cualesquiera que vayan a ser transportados en el mismo vehículo no se superpongan. La restricción (16) garantiza que la

verificación de la superposición sea necesaria sólo si un par de artículos son colocados en un mismo vehículo. Las restricciones (17) a la (19) son las restricciones del dominio de cada variable.

$$y_f + L_f(1 - r_f) + W_f r_f \leq y_e + \left(2 + (\gamma_{ef} + \delta_{ef}) - \left(\sum_{g \in N} z_{gj}^v + \sum_{g \in N} z_{gi}^v \right) \right) L_v,$$

$$i, j \in N \setminus \{0\}, \quad i < j, e \in E, f \in E, \forall v \in V, \quad (20)$$

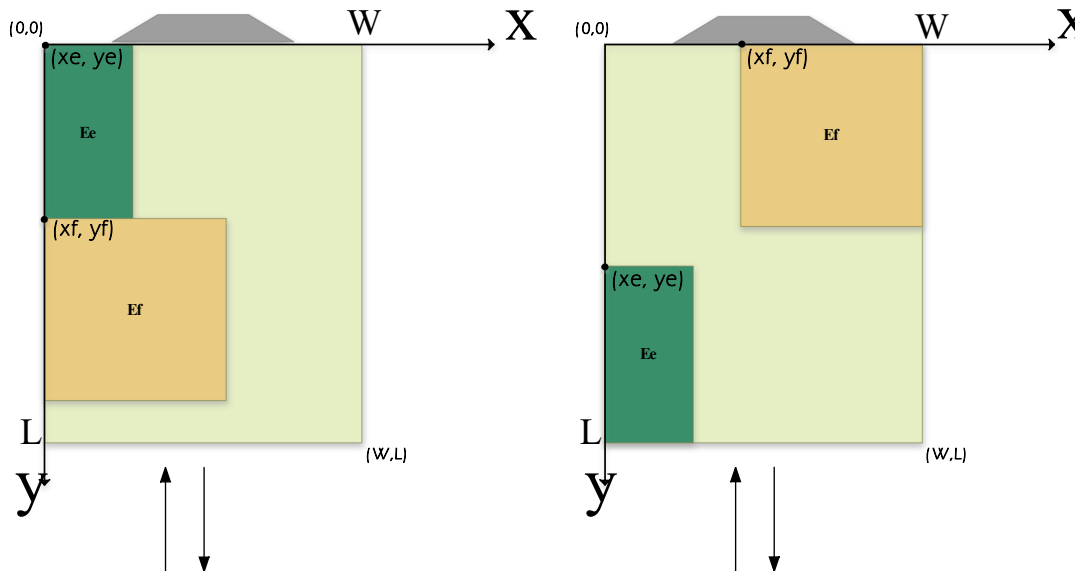
$$y_e + L_e(1 - r_e) + W_e r_e \leq y_f + \left(2 + (\gamma_{ef} + \delta_{ef}) - \left(\sum_{g \in N} z_{gj}^v + \sum_{g \in N} z_{gi}^v \right) \right) L_v,$$

$$i, j \in N \setminus \{0\}, \quad i < j, e \in E, f \in E, \forall v \in V, \quad (21)$$

La restricción (20) previene a los artículos del cliente j de bloquear la descarga de los artículos del cliente i , si el cliente i es visitado antes que al cliente j . Esta restricción verifica si el cliente i es visitado antes que al cliente j . Un ejemplo se ilustra en la **Figura 13** en donde se observa un artículo e de cliente i y un artículo f del cliente j . Si el cliente i no es visitado antes que el cliente j , el artículo e del cliente i y el artículo f del cliente j tienen coordenadas y sin restricciones. Sin embargo, si el cliente i es visitado antes que al cliente j , hay dos posibilidades: Si el artículo e del cliente i está completamente a la derecha o a la izquierda del artículo f del cliente j , tenemos que $\gamma_{ef} + \delta_{ef} = 1$, y no hay intersección de sus proyecciones en el eje x , es decir que un artículo no bloquea al otro. Por otro lado, si el elemento e no está completamente a la izquierda o a la derecha del elemento f , un elemento está detrás o enfrente del otro, y las restricciones imponen la condición que el elemento e debe estar enfrente del elemento f . La restricción (21) tiene una explicación similar, en el caso que el cliente i sea visitado después que el cliente j .

Figura 13

Posiciones relativas de los artículos de dos clientes.



7. Descripción la Metaheurística Algoritmo Genético

Los Algoritmos Genéticos son metaheurísticas adaptativas, que por lo general se utilizan en problemas búsqueda y optimización de parámetros. Estos algoritmos fueron empleados por primera vez por John Holland, en su libro *Adaptation in "Natural and Artificial Systems"* en 1975. Sin embargo, su uso en aplicaciones de la vida real se dio varias décadas después, dado al enorme requerimiento de capacidad computacional para ejecutar todas las operaciones en poco tiempo (Glover & Gary, 2003).

Los Algoritmos Genéticos están basados en la teoría de la evolución enunciada por Charles Darwin en 1859, la cual indica que los individuos con mayores probabilidades de dejar descendencia y cuyos genes pasarán a la siguiente generación, son aquellos que mejor se adaptan al entorno y este fenómeno es conocido como la *supervivencia del más apto* (Reina, D. G., Córdoba, A. T., & del Nozal, 2020). Estas ideas se transfieren a los algoritmos genéticos para

realizar una búsqueda estocástica guiada de forma eficiente. Cada solución factible se presenta dando valores a una serie de parámetros llamados genes, que se descomponen en alelos y se codificados mediante una cadena denominada *cromosoma*; la aptitud de un cromosoma se mide mediante el valor arrojado en la función objetivo, este valor se conoce como valor de bondad, de aptitud o fitness (Gestal et al., 2010). En cada iteración o generación del algoritmo, se producen nuevos individuos mediante la selección, el cruce o la mutación genética dando lugar a nuevas poblaciones; la última generación incluye la mejor o mejores soluciones al problema (Glover & Gary, 2003). En la **Figura 14**, se muestra el diagrama de flujo para el algoritmo genético y se definen cada una de las etapas de éste.

7.1. Generación y Codificación de la Población Inicial

El primer paso para el desarrollo de un algoritmo genético es la generación de la población inicial de cadenas de longitud fija individual. Generalmente la creación de la generación 0 se realiza de forma aleatoria; sin embargo, también existen algoritmos heurísticos constructivos que ayudan con esta tarea. En la medida que una población inicial sea de buena calidad se genera un impacto significativo en la velocidad de convergencia y en la calidad de la solución final del algoritmo (Koza, 2003). Para ello, se requiere una comprensión considerable y detallada del problema, y un buen juicio.

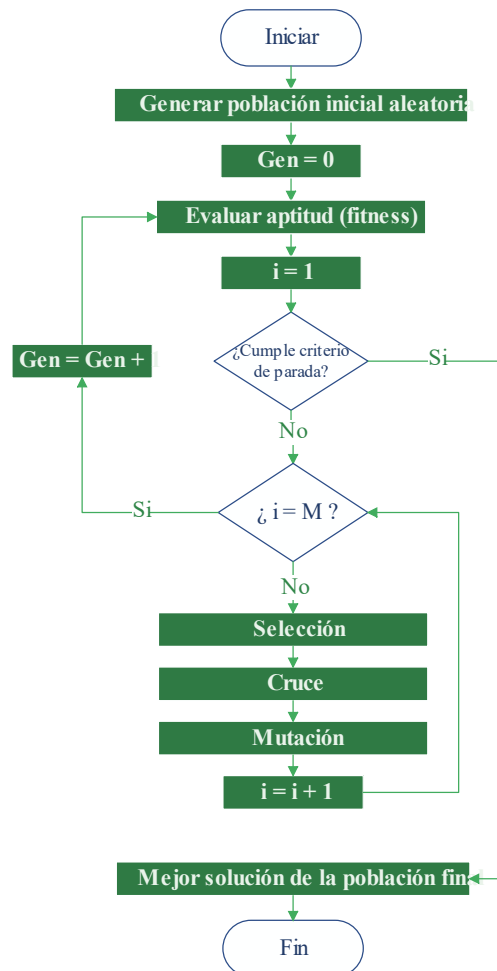
7.2. Evaluación de Aptitud (Fitness)

Como se mencionó anteriormente, la evaluación de aptitud se realiza a cada uno de los individuos a partir de la función objetivo. Se pueden evaluar varias veces en una misma iteración dependiendo de las necesidades del algoritmo. Un individuo se adapta bien al medio si produce un fitness alto, para el caso de maximización; mientras que, para problemas de minimización, el

individuo que mejor se adapta es el que arroja un fitness bajo (Reina, D. G., Córdoba, A. T., & del Nozal, 2020). El índice i se refiere a un individuo de una población de tamaño M (Koza, 2003).

Figura 14

Diagrama de flujo del algoritmo genético.



Nota. Adaptado de Koza, 2003.

7.3. Selección

Es la primera operación que se realiza sobre una población. Para permitir la convergencia hacia soluciones óptimas, los mejores individuos son seleccionados para ser padres en una nueva generación. Un excedente de soluciones descendientes es generado y las mejores son seleccionadas

para lograr un progreso hacia la óptima. Este proceso de selección está basado en los valores de desempeño en la población.

7.3.1. Selección Elitista

Este método clasifica la población de padres y la población de hijos por separado. En él, el mejor individuo de la población de padres y el peor de la población de hijos son identificados y al momento de crear una nueva población, el mejor individuo de la población de padres, reemplaza al peor individuo de la población de hijos. Sin embargo, se puede llegar a perder información genética relevante que puede llegar a existir en otros individuos de la población de padres. Además, al no identificar y preservar esta información este método es incapaz de encontrar múltiples valores óptimos globales (Soremekun et al., 2001).

7.3.2. Selección por Ruleta.

En este método de selección a todos los individuos de la población se les asigna una probabilidad proporcional al valor de su aptitud de manera que los mejores individuos obtendrán una porción más amplia de la ruleta que los peores. Esta probabilidad se determina ponderando la aptitud de cada individuo sobre la aptitud de toda la población. Por lo general, se ordena a la población en base al valor de su aptitud, por lo que los mejores individuos se encuentran al principio de la ruleta. Para seleccionar un individuo se genera un número aleatorio del intervalo [0,1] y será seleccionado como progenitor cuando la suma de probabilidades supere el valor del número aleatorio obtenido (Gestal et al., 2010). A continuación, se ejemplifica el procedimiento:

El orden de los individuos en la ruleta con sus respectivas probabilidades se muestra en la siguiente tabla:

Tabla 9*Ejemplo de selección por ruleta.*

Individuo	Fitness	Probabilidad	Probabilidad Acumulada
5	15	0,3061	0,3061
3	14	0,2857	0,5918
2	9	0,1837	0,7755
1	8	0,1633	0,9388
4	3	0,0612	1

Por ejemplo, si el número aleatorio generado es 0.6, se tomará como progenitor el individuo número 3 dado que 0.6 excede la probabilidad acumulada de 0.5918 del individuo 2, pero no a la de 0.7755 del individuo 3.

Este método resulta ineficiente para poblaciones de gran tamaño además de que el peor individuo puede llegar a ser seleccionado más de una vez.

7.3.3. Selección por Torneo.

Esta metodología consiste en la selección de individuos contrastando los genotipos de dos o más, elegidos al azar. Existen dos versiones de este método de selección, la versión determinística, en donde simplemente se selecciona al más apto para pasar a la siguiente generación, y la probabilística, en la cual, en lugar de elegir siempre al más apto, se genera un número aleatorio dentro del intervalo $[0,1]$ y si este valor es mayor que un parámetro p , determinado para todo el proceso y que generalmente toma valores entre 0.5 y 1, se elige al mejor individuo. En caso contrario, se escoge al menos apto.

Al variar el número de participantes en el torneo, se puede modificar la presión de selección. Con una presión alta, lograda con una gran cantidad de participantes en el torneo, la probabilidad de reproducción de los individuos menos aptos es mínima. Por lo contrario, cuando

la cantidad de participantes en el torneo es reducida, la presión de selección disminuye, incrementando las oportunidades de los peores individuos de ser seleccionados (Gestal et al., 2010).

7.3.4. Muestreo Estocástico Universal

En este método de selección se clasifican los valores de aptitud de los individuos de la población en orden ascendente para después tomar aleatoriamente un individuo como punto de partida inicial y seleccionar a los demás individuos que harán parte de la nueva generación utilizando intervalos espaciados uniformemente. Así, este método da la misma oportunidad a todos los individuos de ser seleccionados para participar en el cruce (Katoch et al., 2021).

7.4. Cruce o Recombinación

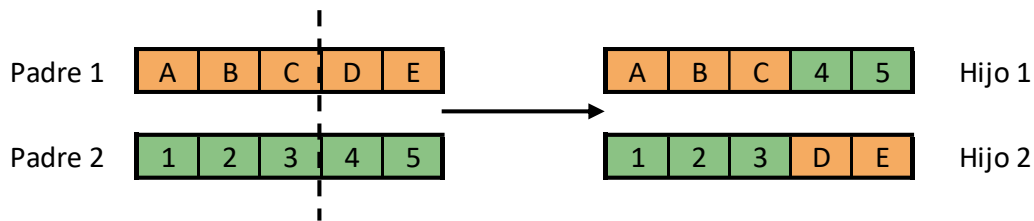
Una vez seleccionados los individuos que serán padres, se genera la operación de cruce de sus cromosomas, siguiendo una distribución de probabilidad uniforme denominada Pr , para crear la descendencia. Esta operación produce dos descendientes diferentes a sus dos padres y diferentes entre sí, y está guiada por los resultados de evaluación de aptitud (Koza, 2003), dado que comparten características de los padres, gran parte de la descendencia tienen una mejor aptitud que cada uno de los padres por separado. A continuación, se detallan las técnicas de cruce más empleados (Gestal et al., 2010):

7.4.1. Cruce de 1 Punto

Como su nombre lo indica, los cromosomas de los padres se cruzan por un único punto determinado aleatoriamente para producir dos segmentos diferenciados en cada padre, los cuales se recombinan para generar los nuevos individuos (ver **Figura 15**). En esta técnica se conserva gran cantidad de información entre padre e hijo.

Figura 15

Ejemplo de cruce de 1 punto.

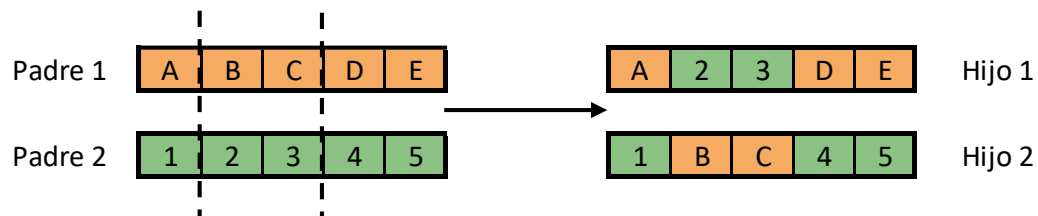


7.4.2. Cruce de 2 Puntos

Asimismo, se pueden establecer dos puntos de corte, garantizando tres segmentos de cada cromosoma padre. Para generar la descendencia, se selecciona el segmento del centro de uno de los padres y los extremos del otro padre, como se muestra a continuación:

Figura 16

Ejemplo de cruce de 2 puntos.

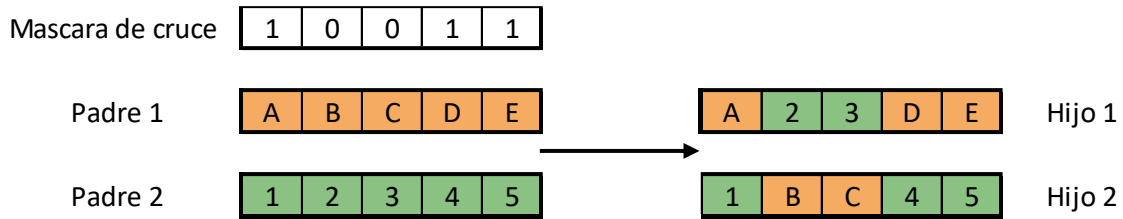


7.4.3. Cruce Uniforme

En esta técnica de cruce, cada alelo tiene las mismas probabilidades de recombinarse. Para ello, se genera aleatoriamente una máscara de cruce entre 0 y 1, de ahí que la probabilidad es de 0,5 para cada padre; el alelo pertenece al primer padre cuando exista 1 en la máscara de cruce y al segundo padre en caso de ser 0 (ver **Figura 17**). En este caso, se pierde información entre padre-hijo dada la búsqueda extensiva que debe realizarse.

Figura 17

Ejemplo de cruce uniforme.



7.5. Mutación

Conjuntamente con la operación de cruce, se realiza la operación de mutación. La mutación es el operador que asegura la diversidad genética de una población con respecto a la anterior (Katoch et al., 2021). Esto se basa en el cambio aleatorio de algún gen de un individuo. Una vez ocurre el cruce entre dos individuos, uno o ambos de los descendientes mutan con cierta probabilidad P_m . Esta probabilidad es muy baja, por lo regular menor al 1%. Esto se debe a que el valor de la aptitud de un individuo se ve reducido una vez es mutado. Las mutaciones mejor conocidas son el reemplazo aleatorio y la mutación codificada. En el reemplazo aleatorio se altera aleatoriamente un gen de un cromosoma (Gestal et al., 2010). Por el contrario, en la mutación codificada, todos los genes de un cromosoma son tomados y colocados de forma aleatoria y se verifica si el valor de la aptitud del individuo mejora (Katoch et al., 2021).

Aunque no es frecuente, existen aplicaciones de algoritmos genéticos donde no todos los individuos tienen la misma longitud en sus cromosomas. Para estos casos, existen otras mutaciones en donde se puede añadir nuevos genes o eliminar genes ya existentes (Gestal et al., 2010).

8. Diseño del Algoritmo Genético para la Solución del 2L-HFFVRP

En esta sección, se aborda la solución al modelo planteado mediante la implementación de un algoritmo genético codificado en el lenguaje de programación *Python*, en su versión 3.9.12, a través de la interfaz gráfica de usuario (GUI) *Anaconda Navigator*. Es importante destacar que el algoritmo genético propuesto en este proyecto no se basó en un autor o referencia en particular, sino que se diseñó específicamente para abordar los requerimientos del problema planteado. Asimismo, se codifica el algoritmo de primera posición (First-fit) para abordar el problema de carga bidimensional.

Por otro lado, el alcance de este trabajo de investigación originalmente comprendía la formulación del modelo de optimización y el diseño del algoritmo genético para la solución. Sin embargo, se desarrolló un prototipo completamente funcional que incluye una interfaz de usuario intuitiva para facilitar la utilización. Este prototipo es completamente original y ha sido desarrollado por los miembros de este proyecto como resultado de su labor investigativa.

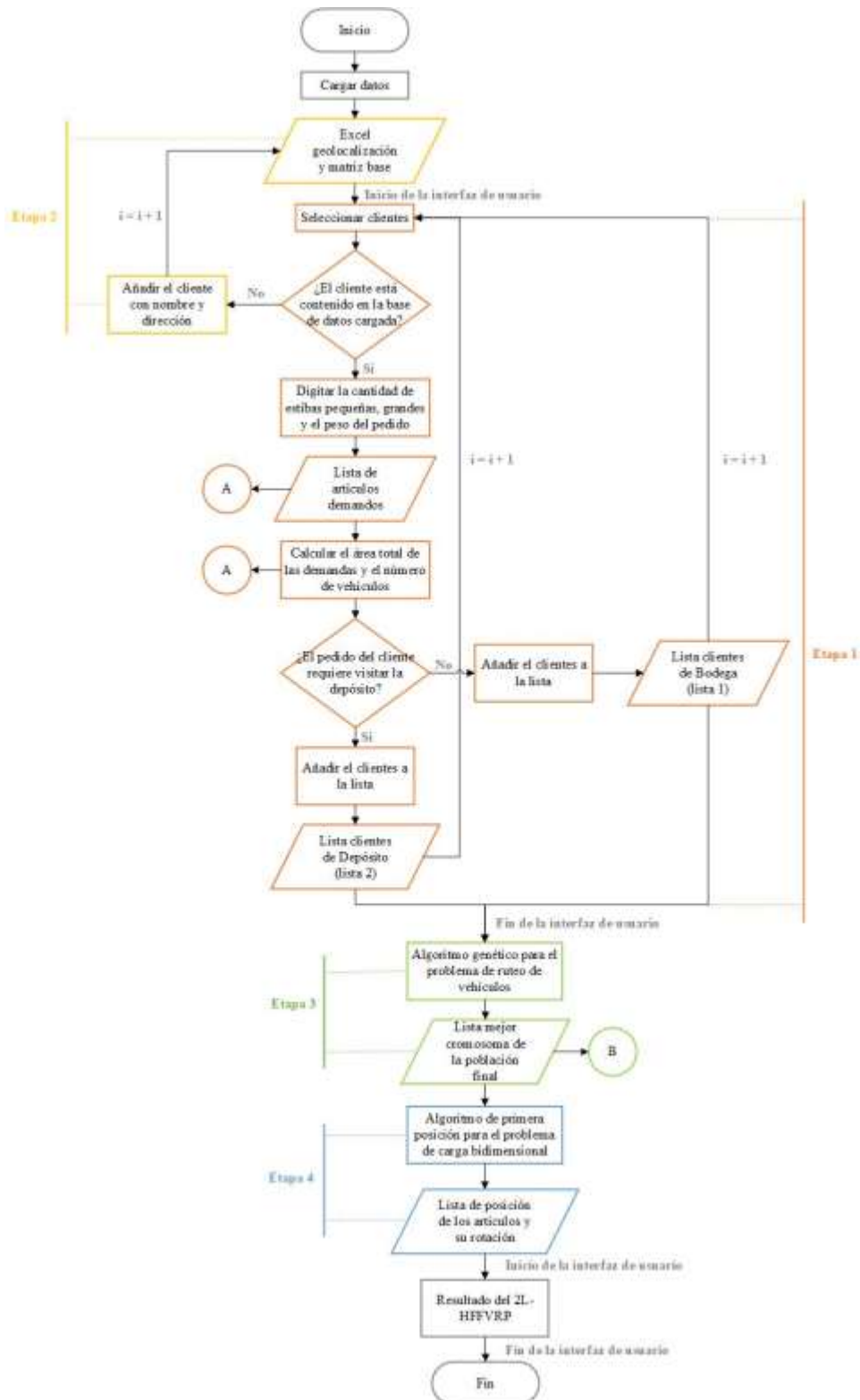
En la **Figura 18**, se presenta el diagrama de flujo del prototipo propuesto para la solución del problema de ruteo de vehículos de flota fija heterogénea con restricciones de carga bidimensional (2L-HFFVRP) y se detalla exhaustivamente cada etapa del proceso.

8.1. Etapa 1. Programa de Ingreso de Datos

En la primera fase, se propone la implementación de un sistema de ingreso de datos diseñado para que el usuario pueda seleccionar entre clientes existentes o nuevos, y así organizar sus pedidos, entre otras funcionalidades.

Figura 18

Diagrama de flujo del prototipo propuesto para la solución del 2L-HFFVRP.

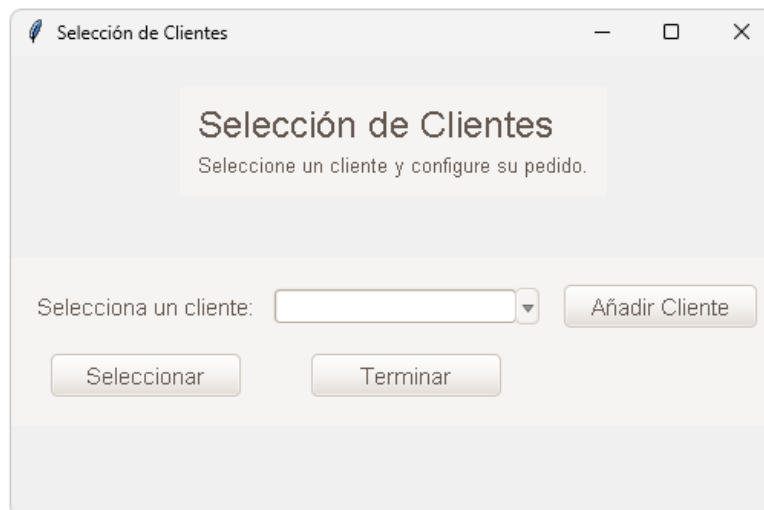


8.1.1. Inicialización y Configuración

Al principio, se crea una clase denominada “Interfaz”, que representa la estructura principal del prototipo de aplicación (**Figura 19**). Al instanciar esta clase, se realiza la configuración inicial de la ventana principal de la aplicación, estableciendo el título de "Selección de Clientes". Además, se definen varios atributos con el propósito de mantener un registro de los clientes seleccionados y calcular tanto el área como el peso total de los pedidos. También se establecen las dimensiones y las características de tres tipos de vehículos y dos tipos de estibas. Estos valores serán utilizados posteriormente para llevar a cabo cálculos relacionados con la organización y distribución de los pedidos.

Figura 19

Ventana principal del prototipo de la aplicación.



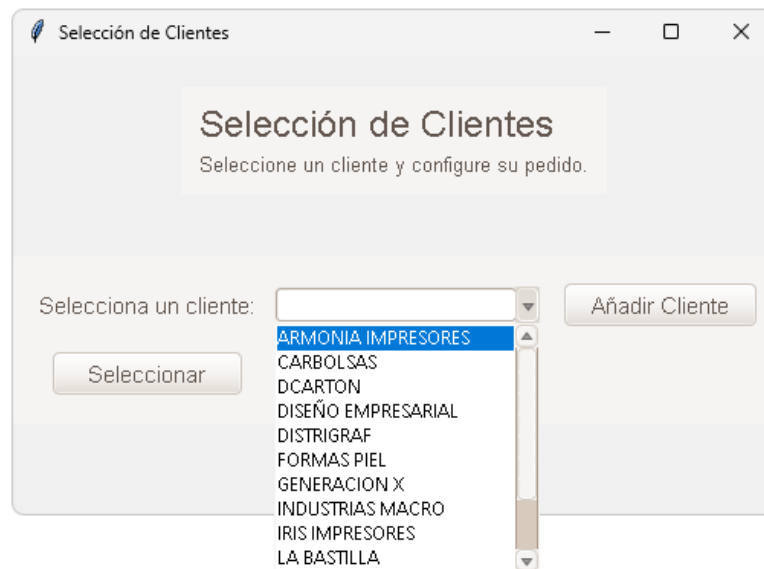
8.1.2. Carga de la Lista de Clientes

La función “*cargar_lista_clientes*” se encarga de leer y cargar una lista de clientes desde un archivo Excel'. Una vez que se ha realizado la lectura, se procede a excluir el nombre de la bodega y el nombre del depósito, como se observa en la **Figura 20**, que no deberían de estar

disponibles en la lista desplegable de selección de clientes. En caso de que surja algún problema durante la carga del archivo, como la ausencia del archivo o archivo vacío, la función muestra un mensaje de error para informar al usuario sobre el problema.

Figura 20

Lista de clientes cargada en el prototipo de la aplicación.



8.1.3. Añadir un Nuevo Cliente

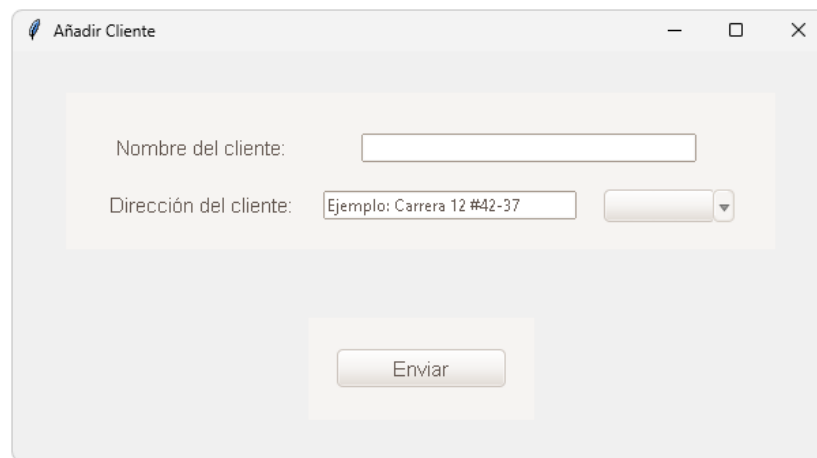
Cuando se selecciona la opción "Añadir Cliente" haciendo clic en el botón correspondiente, se activa la función *emergente_agregar_cliente*, la cual despliega una ventana emergente. En esta ventana, se solicita al usuario que ingrese información específica del cliente, como su nombre y dirección, este proceso se detalla en la **etapa 2**. Para facilitar la entrada de la dirección, se incluye un marcador de posición (*placeholder*), que ofrece un ejemplo de cómo debe ingresar una dirección, además de un menú desplegable para seleccionar la ciudad correspondiente (**Figura 21**).

Una vez que el usuario ha ingresado la información requerida y decide enviarla, se llama a la función *submit_cliente*. Esta función realiza varias comprobaciones. En primer lugar, verifica

que tanto nombre como dirección hayan sido proporcionados. Luego, comprueba si el nombre o la dirección del nuevo cliente ya existen en la base de datos. Si cualquiera de estos datos ya existe, se muestra un mensaje de error para informar al usuario sobre la duplicación. Por el contrario, si el cliente es realmente nuevo, se agrega su información al archivo de Excel y se actualiza la lista desplegable de clientes en la ventana principal para reflejar la adición del nuevo cliente. Finalmente, una vez completado este proceso, la ventana emergente se cierra automáticamente y se muestra un mensaje al usuario, informando que el cliente ha sido añadido con éxito.

Figura 21

Ventana para añadir clientes nuevos al prototipo de la aplicación.



8.1.4. Selección y Configuración de Clientes

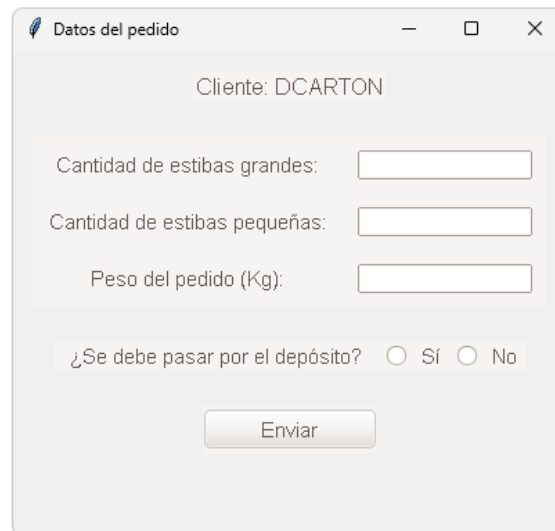
Cuando un usuario elige un cliente del menú desplegable y hace clic en el botón "Seleccionar", se activa la función *mostrar_emergente*. Esta función despliega una ventana emergente que permite al usuario ingresar detalles específicos relacionados con el pedido del cliente seleccionado. Dentro de esta ventana, el usuario puede especificar tanto la cantidad de estibas grandes como pequeñas que el cliente ha solicitado, así como el peso total del pedido (**Figura 22**). Además, el usuario debe indicar si es necesario pasar por el depósito para esta entrega.

Una vez que el usuario ha proporcionado todos los detalles y los envía, entra en acción la función “emergente_submit”. Esta función procesa la información proporcionada, actualiza la lista de clientes seleccionados y realiza varios procesos. En primer lugar, verifica si el cliente ya ha sido seleccionado previamente. Si es así, ofrece al usuario la opción de modificar la orden existente. Luego, se realizan cálculos basados en las dimensiones y el peso de los pedidos para determinar el área total y el peso de todos los pedidos seleccionados hasta el momento, así como el número de vehículos requeridos para transportar todos los pedidos. Si el área o el peso total de los pedidos excede la capacidad de los vehículos, se muestra un mensaje de error y se revierte la adición del último cliente.

Finalmente, después de procesar toda la información, la ventana emergente se cierra y el usuario puede continuar seleccionando otros clientes o finalizado el proceso.

Figura 22

Ventana de ingreso de datos de los pedidos al prototipo de la aplicación.



Datos del pedido

Cliente: DCARTON

Cantidad de estibas grandes:

Cantidad de estibas pequeñas:

Peso del pedido (Kg):

¿Se debe pasar por el depósito? Sí No

Enviar

8.1.5. Finalización de la Selección y Ruteo

Cuando el proceso de selección y configuración de pedidos de los clientes se ha completado, el usuario tiene la opción de finalizar la selección al hacer clic en el botón "Terminar".

Al hacerlo, se activa la función *terminar_seleccion*. Esta función comienza cargando una matriz de distancias desde un archivo Excel que contiene las distancias entre diferentes clientes. A partir de esta matriz, se crea una matriz con las distancias específicas entre los clientes que han sido seleccionados. Una vez se ha creado la matriz de distancia para los clientes seleccionados, la función procede a resolver el problema de ruteo utilizando el algoritmo genético que se detalla más adelante en la etapa 3.

Después de determinar las rutas, la función elimina la hoja del archivo de Excel dónde se generó la matriz para los clientes seleccionados. Esto para mantener los datos y evitar redundancias en futuras ejecuciones. Por último, se presentan los resultados del ruteo al usuario en una nueva ventana. Estos resultados indican cómo deben organizarse y entregarse los pedidos, qué vehículos deben utilizarse y en qué secuencia deben visitarse los clientes.

8.2. Etapa 2. Matriz de Distancia para Clientes Nuevos

Esta etapa se plantea como respuesta a las limitaciones de tiempo computacional encontradas durante el proceso de la generación de la matriz base de 14x14. El objetivo principal es facilitar el ingreso y almacenamiento eficiente de nuevos clientes en el archivo Excel de la matriz base. Este archivo se carga en el código y consta de dos hojas: la primera contiene los geodatos de cada cliente, mientras que la segunda contiene la matriz de distancia.

Como se detalló en la *sección 5.3.2.*, la geolocalización para crear la matriz base se llevó a cabo de forma manual utilizando el software de Google Maps. Esto fue posible debido a que se contaba con información previa de los clientes y sus direcciones. Sin embargo, como no siempre será el caso, se procede a adquirir la prueba gratuita por 90 días de la API de Geocoding proporcionada por Google para automatizar el proceso.

En la etapa anterior, se efectúa el ingreso de datos para direcciones nuevas y se almacenan en una variable. Posteriormente, se emplea una función llamada “geolocalizar(direccion)” que toma esta dirección como entrada y utiliza la API de Google Maps para obtener las coordenadas geográficas (latitud y longitud) de esa dirección. Los resultados se almacenan en la hoja correspondiente del archivo Excel, añadiendo una fila al final de la tabla contenida.

A continuación, se procede a realizar el cálculo de la distancia en metros utilizando la API de OpenRouteService (ORS). Este proceso implica la construcción de una URL con las coordenadas de inicio y destino, junto con la clave de API de ORS. Seguidamente, se realiza una solicitud GET a esa URL y se procesa la respuesta JSON para extraer las distancias requeridas. Estas distancias se calculan mediante una función llamada “*crear_matriz_seleccionados(clientes_seleccionados, matriz_completa_df)*” la cual crea una matriz de distancia seleccionada a partir de la matriz completa de distancias. Esta función toma una lista de clientes seleccionados y genera una submatriz que contiene únicamente las distancias entre esos clientes.

La información de la submatriz de distancias se incorpora al archivo de Excel mencionado previamente. El proceso se realiza mediante la creación de una nueva hoja con el mismo nombre y la eliminación de la hoja anterior, asegurando así la continuidad e integridad del funcionamiento del programa.

8.3. Etapa 3. Algoritmo Genético para el Problema de Ruteo de Vehículos (HFFVRP)

La metaheurística propuesta para dar solución al problema de vehículos de flota fija heterogénea se basó en un algoritmo genético. Este algoritmo emplea una función de aptitud basada en la distancia total recorrida por un vehículo en una ruta determinada, un método de

selección elitista, un operador de cruce de un punto y un operador de mutación por intercambio. Este enfoque se ha destacado en la literatura debido a su efectividad en la resolución del problema de ruteo de vehículos. De hecho, es una de las metaheurísticas más ampliamente utilizadas en este contexto.

8.3.1. *Parámetros de Entrada*

En programación, se define un parámetro de entrada como una variable cuyo valor puede ser modificado según las necesidades del programador o del usuario en cada ejecución. Siguiendo esta premisa, se han establecido los siguientes parámetros de entrada para el algoritmo genético propuesto:

Tabla 10

Parámetros de entrada para la ejecución del algoritmo genético.

Parámetro	Valor
Número de generaciones	1000
Probabilidad de cruce	1
Probabilidad de mutación	0,1
Tamaño de la población	50

8.3.2. *Parámetros de Instancias*

Además de los parámetros de entrada propuestos, el algoritmo genético utiliza parámetros de entrada adicionales incluyendo la matriz de distancia, que se detalla en el Apéndice B, así como el número de vehículos utilizados, que puede variar de uno a tres, y el número de clientes, que puede oscilar entre un solo cliente hasta un máximo de 38, que es cuando se alcanza la capacidad máxima de carga de la flota de vehículos en el mejor de los casos. Estos parámetros serán recibidos por el algoritmo genético desde la etapa 1.

Tabla 11

Parámetros de entrada adicionales para la ejecución del algoritmo genético.

Parámetro	Valor
Número de clientes	1 - 38
Número de vehículos	1 - 3
Matriz de distancia	Apéndice B

8.3.3. Representación de los Cromosomas

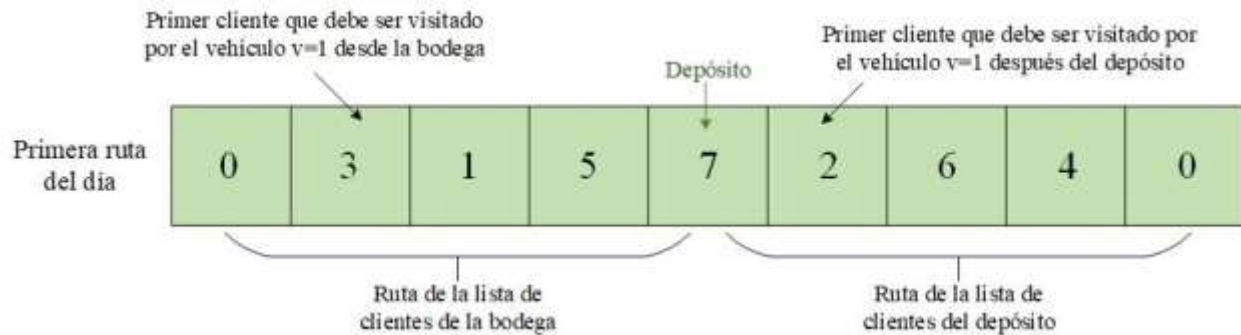
Un cromosoma es definido como un vector que representa una secuencia de clientes que deben ser visitados. El tamaño del cromosoma varía en función del número de clientes asignados, desde 1 hasta n, asegurando que cada cliente aparezca una sola vez en el cromosoma. El valor de “v” responde a la cantidad de vehículos a utilizar en una misma ruta, ya sea diurna o en el horario de la tarde, mientras que los genes que conforman el cromosoma se refieren a las celdas que se completarán de la siguiente manera:

- El vehículo se elige en orden descendente de acuerdo a su capacidad.
- Las cadenas pueden estar compartidas hasta con los 3 vehículos.
- La ruta inicia en el depósito [0].
- Como se tienen dos listas de clientes: lista de clientes de bodega y lista de clientes de depósito, el depósito se considera como el último cliente de la bodega o el primer cliente del depósito. En cuanto a la matriz, el depósito siempre se toma como el último, es decir, si se registran 6 clientes a visitar, el depósito se asignará como el cliente número 7.

A continuación, se muestra la representación de los individuos de manera más visual:

Figura 23

Representación de un cromosoma.



La posición que cada cliente ocupa en el cromosoma indica el orden de visita dentro de la ruta. La ruta se interpreta así: inicia en la bodega (0), visita los clientes contenidos en la lista de clientes de bodega (3-1-5). A continuación, viaja al depósito (7) a cargar los pedidos demandados de allí. Después, se visitan los clientes que se encuentran en la lista de clientes del depósito (2-6-4) y finalmente se regresa a la bodega (0) para concluir la ruta. En esta representación, la ruta no superó la capacidad del vehículo 1 y se utilizó un sólo vehículo.

8.3.4. Generación de la Población Inicial

La población inicial se genera de manera aleatoria. En concreto, para crear cada individuo (o ruta) de la población inicial, el algoritmo toma una muestra aleatoria de todos los clientes sin reemplazo. Dado que el tamaño de la muestra es igual al número total de clientes, esto da como resultado una permutación aleatoria de los clientes. Este proceso se repite un número predefinido de veces, determinado por el tamaño de la población, para crear la población inicial. En consecuencia, la población inicial consta de un conjunto de rutas aleatorias, cada una siendo una permutación aleatoria de los clientes y estas rutas aleatorias se utilizan como punto de partida para el algoritmo genético. Posteriormente, el algoritmo trabaja en la búsqueda y mejora de estas rutas utilizando operadores de selección, cruce y mutación.

8.3.5. Evaluación de Aptitud

Para realizar este proceso, se debe tener presente la función objetivo del modelo matemático planteado, dado que esta es tomada como función fitness del algoritmo. La evaluación de la aptitud o función fitness de un individuo se define como la distancia total que un vehículo recorrería al seguir una ruta específica, desde un punto de inicio hasta un punto final, visitando a todos los clientes en el orden establecido en la ruta. En este contexto, una aptitud mayor se traduce en una menor distancia recorrida por el vehículo, lo que implica la minimización de la distancia total recorrida.

La función aptitud (fitness) emplea una matriz de distancias previamente definida para determinar la longitud total de una ruta específica. Esta ruta inicia desde un punto de partida y, a medida que avanza a lo largo de ella, suma las distancias entre cada par de clientes consecutivos que se encuentran dentro de la ruta. Una vez que se han visitado todos los clientes, se agrega la distancia desde el último cliente hasta un punto final. Al concluir, devuelve la suma total de estas distancias, lo que representa el costo total de la ruta. Cuanto menor sea la distancia, mejor será considerada la ruta.

8.3.6. Creación de Nuevas Generaciones

Después de generar la población inicial, se procede a aplicar los operadores de selección, cruce y mutación con el fin de crear las siguientes generaciones.

8.3.6.1. Selección. La función selección desempeña un papel fundamental dentro del algoritmo genético, ya que su propósito es identificar y seleccionar a las dos rutas más aptas, es decir, aquellas con la menor distancia total dentro de una población de rutas. El método de selección empleado en esta función se basa en un enfoque **elitista**, lo que significa que se seleccionan directamente a los individuos de mayor aptitud en la población sin introducir ningún elemento de azar. Esta elección garantiza que las soluciones más prometedoras identificadas hasta el momento perduren en generaciones futuras del algoritmo.

No obstante, aunque el método de selección elitista asegura la preservación de las soluciones con mayor aptitud para la generación siguiente, presenta sus propios desafíos. Siempre que se elijan y reproduzcan los mismos individuos dominantes, existe el riesgo de que el algoritmo quede atrapado en una solución que, si bien es buena, no necesariamente es la óptima. Para abordar este problema, el algoritmo introduce variabilidad y diversidad mediante los operadores de cruce y mutación, lo que permite explorar nuevas soluciones y evitar que se estanque.

8.3.6.2. Cruce. El operador cruce, al igual que la función de selección, desempeña un papel fundamental en el algoritmo. Su objetivo principal es generar nuevos individuos, en este caso, rutas, a partir de dos individuos preexistentes, conocidos como padres. Para llevar a cabo esta tarea, la función de cruce selecciona aleatoriamente un punto de corte, que determina cómo se combinarán las partes de las rutas de los padres para formar las rutas de los hijos.

El primer hijo se construye tomando la primera parte de la ruta del primer padre (hasta el punto de corte) y luego agregando los clientes del segundo padre que aún no están presentes en la ruta del hijo. El segundo hijo se crea de manera similar, pero invirtiendo los roles de los padres.

Esta técnica garantiza que todos los clientes se visiten exactamente una vez en las rutas de los hijos.

La función cruce se invoca dentro del ciclo principal del algoritmo genético, específicamente después de que se han seleccionado los padres y antes del operador de mutación. Es un paso crucial para introducir diversidad en la población y explorar nuevas soluciones en el espacio de búsqueda del problema.

8.3.6.3. Mutación. El operador mutación desempeña un rol especial en la preservación de la diversidad genética de una generación a la siguiente en el contexto del algoritmo. En este proceso, la función mutación selecciona dos índices aleatoriamente en una ruta y procede a intercambiar las posiciones de los clientes en dichos índices. Este intercambio genera una nueva ruta que puede ser mejor, peor o igual en términos de aptitud con respecto a la ruta original.

Es importante destacar que, en cada generación, después del operador de cruce, cada ruta de la población tiene una probabilidad del 10% de ser sometida a una mutación. Esta probabilidad se encuentra en un punto de equilibrio, ya que, si es demasiado alta, el algoritmo podría mostrar un comportamiento errático y aleatorio, mientras que, si es demasiado baja, podría no proporcionar la diversidad necesaria para explorar de manera adecuada el espacio de soluciones.

8.3.7. Criterio de Parada

El criterio de parada del algoritmo genético se basa en el número de generaciones. El algoritmo se ejecuta durante una cantidad predefinida de generaciones y se detiene una vez que ha alcanzado ese número de iteraciones.

8.4. Fase 4. Algoritmo de Primera Posición para el Problema de Carga Bidimensional

Una vez definida la ruta y el orden de visita de los clientes, es necesario abordar la etapa siguiente: la determinación de la posición de carga de los artículos en los vehículos, con el fin de minimizar el desperdicio de espacio. Esta etapa es esencial para asegurar la máxima utilización del espacio disponible. En este contexto, se propone la implementación de un algoritmo de primera posición o First-fit para abordar el problema de carga bidimensional no orientada-secuencial, en *Figura 24*, se observa el diagrama de flujo. Este algoritmo busca colocar los artículos en la ubicación de menor índice en la que pueda empaquetarse dentro del vehículo.

8.4.1. Inicialización

El algoritmo de primera posición inicia su proceso al recibir la lista del mejor cromosoma de la población final del algoritmo genético, que señala el orden de colocación de los artículos demandados dentro del vehículo. A continuación, se procede considerando la superficie fija del vehículo vacío. En este escenario, es importante tener en cuenta que la empresa Coimpresores dispone de una flota de tres vehículos, cada uno de ellos con dimensiones específicas detalladas en la **Tabla 12**. La cantidad de vehículos y los vehículos a utilizar en cada ruta, está determinado en etapa 1.

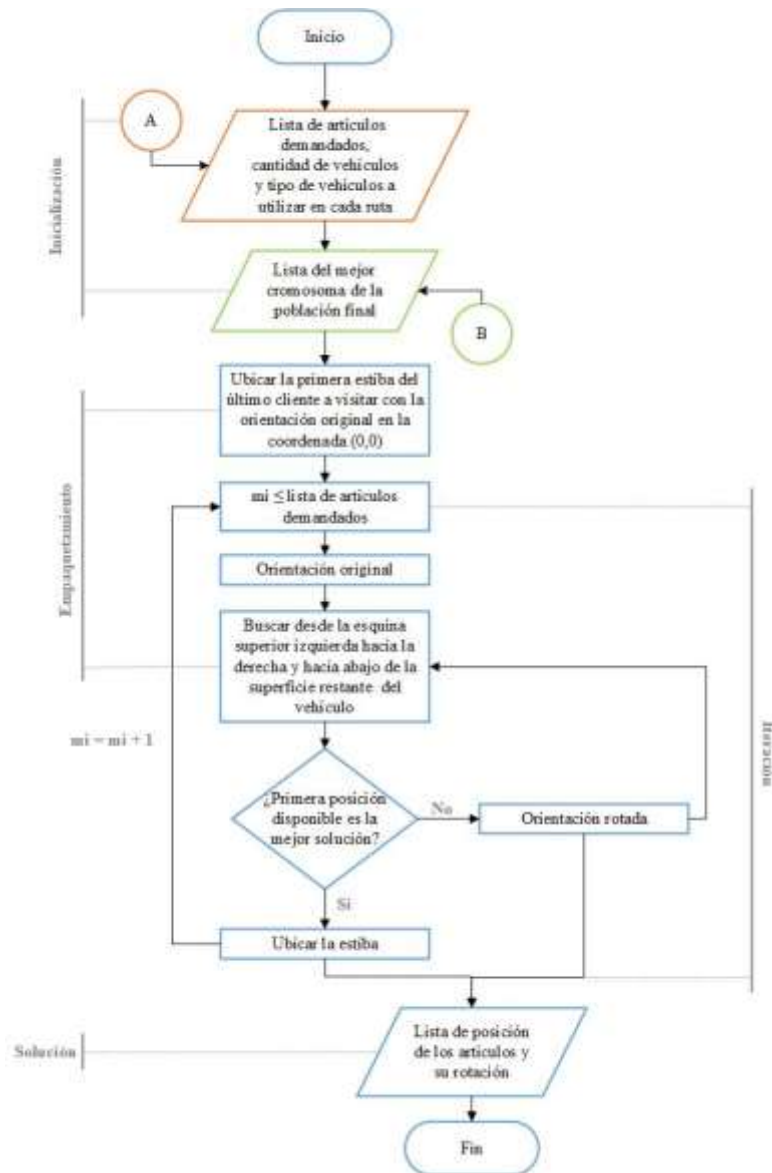
Tabla 12

Dimensiones de la flota de Coimpresores.

Vehículo	Ancho [cm]	Largo [cm]
NKR 1	190	400
NKR 2	190	480
NHR	180	320

Figura 24

Diagrama de flujo del algoritmo de primera posición.



Además, se han categorizado los artículos demandados en dos tipos: “estiba grande” y “estiba pequeña”, con medidas específicas como se detalla en la **Tabla 13**. La cantidad de estibas se guarda a partir de los datos de entrada y los clientes seleccionados en la etapa 1 del programa. Para cada cliente, se generan cargas de estibas grandes y pequeñas, y estas se registran y almacenan en listas denominadas “*Estiba*”.

Tabla 13*Dimensiones de las estibas de Coimpresores*

Tipo de artículo	Ancho [cm]	Largo [cm]
Estiba grande	100	70
Estiba pequeña	90	60

8.4.2. Empaquetamiento

Inicialmente, sólo la esquina delantera o superior izquierda de la superficie de carga del vehículo, correspondiente al punto de coordenadas (0,0), está disponible para cargar un artículo. El proceso de carga se desarrolla de la siguiente manera:

- Se inicia ubicando la demanda de estibas del último cliente a visitar con las estibas grandes en la esquina (0,0) de manera vertical. Esto se hace siguiendo la característica secuencial.
- Luego, se avanza con los artículos restantes, buscamos la primera posición disponible en el espacio de carga restantes del vehículo. Esta búsqueda tiene en cuenta la orientación de las estibas (original y rotada) mediante el método *primera_posicion(self, estiba)*, en la cual determina la posición óptima.
- La búsqueda se lleva a cabo desde la esquina superior izquierda hacia la derecha y hacia abajo de la superficie del vehículo.
- Una vez se encuentra una posición adecuada, se coloca el artículo en esa ubicación y se marcan las celdas ocupadas por el objeto en el espacio de trabajo.

8.4.3. Iteración

El proceso se vuelve iterativo con las dos orientaciones de las estibas (*Estiba* y *Estiba.rotate()*). En la fase inicial, se verifica si la carga en su orientación original puede encajar en la posición (*i,j*) disponible, sin superponerse con otras cargas ya colocadas en el vehículo. Esto se verifica mediante la comprobación de que todas las celdas en la región ocupada por la carga

estén vacías o tiene un valor de cero (0). Este proceso se realiza hasta encontrar una posición válida y la carga en la orientación es óptima.

Este proceso se repite hasta que todos los artículos demandados por los clientes de la ruta se hayan colocado en el vehículo. Cada vez que se inserta un artículo se excluye de la lista de posiciones de carga disponibles.

8.4.4. Solución

Mediante la función *resolver_dosL*, se registran los resultados de empaquetamiento de cada vehículo. Para cada uno de ellos, se genera una lista que contiene las cargas que han sido colocadas en su contenedor respectivo. Al finalizar el proceso, se presenta la información detallada acerca de las cargas que se han ubicado en los contenedores, incluyendo las coordenadas y la orientación de cada artículo.

9. Validación del Algoritmo

En esta sección se aborda la validación del algoritmo propuesto, un proceso destinado a evaluar y comprobar su precisión y eficacia. El objetivo es demostrar que el algoritmo genera resultados sólidos para cualquier conjunto coherente de parámetros de entrada, corroborando que estos sean factibles. La validación se lleva a cabo tanto para el algoritmo genético del HFFVRP como para el algoritmo de primera posición del 2L. A continuación, se detalla cada uno

9.1. Validación del Algoritmo Genético del HFFVRP

Para la validación del algoritmo genético propuesto, se emplearon los datos reales de Coimpresores del Oriente, definiendo 2 escenarios diferentes, cada uno compuesto por 44 instancias. Teniendo en cuenta que la empresa dispone de una bodega que almacena material en pliego y un depósito para el material con cortes específicos. En esta cadena de ruteo, existe un

flujo desde la bodega hacia el depósito, pero no en sentido contrario. Por lo tanto, se considera al depósito como un cliente adicional al que se debe visitar para cargar material en lugar de descargarlo. A partir de esta premisa, se plantean los siguientes escenarios:

Escenario 1: En este escenario, se visita el depósito en la mitad del recorrido, visitando primero a los clientes que no requieren material de este.

Escenario 2: En este caso, se realiza la visita al depósito al inicio y al final del recorrido de una ruta. Es importante resaltar que, si el depósito se visita al final del recorrido, se carga el material requerido de la siguiente ruta para que, al iniciarla, todos los artículos salgan desde la bodega.

La definición de las 44 instancias para la evaluación del algoritmo, se realiza mediante la variación de los días en los que se realizaron rutas. Estas 44 instancias constituyen el 5% de la población total de 841 días. La selección de los días se realiza aleatoriamente y de manera proporcional para cada año. En la **Tabla 14**, se presenta una tabla que muestra la composición de las instancias propuestas.

9.2. Validación del Algoritmo de Primera Posición del Problema 2L

El algoritmo de primera posición propuesto para la solución al problema de carga bidimensional se valida a través de un programa existente diseñado en la web y una validación manual. Esta elección se debe a que, previamente la empresa no contemplaba este problema, lo que resulta en la ausencia de datos históricos para validar este enfoque. Además, cabe señalar que este problema es poco estudiado en literatura, lo que limita la disponibilidad de datos de referencia y/o parámetros para la respectiva validación.

Tabla 14

Instancias para la validación del algoritmo genético del HFFVRP.

Instancia	Día	Cantidad de clientes	Cantidad de vehículos	Instancia	Día	Cantidad de clientes	Cantidad de vehículos
1	03/01/2019	9	2	23	04/06/2020	2	1
2	29/01/2019	5	1	24	25/06/2020	5	1
3	29/01/2019	4	1	25	16/07/2020	10	2
4	22/02/2019	13	2	26	06/08/2020	6	2
5	19/03/2019	5	1	27	31/08/2020	5	1
6	19/03/2019	4	1	28	31/08/2020	10	2
7	13/04/2019	2	1	29	21/09/2020	11	2
8	11/05/2019	4	1	30	23/11/2020	7	1
9	06/06/2019	6	2	31	16/01/2021	4	1
10	04/07/2019	2	1	32	18/02/2021	7	1
11	29/07/2019	10	1	33	18/02/2021	4	1
12	24/08/2019	4	1	34	24/03/2021	6	1
13	18/09/2019	7	2	35	16/04/2021	3	1
14	11/10/2019	10	2	36	14/05/2021	6	1
15	12/10/2019	4	1	37	09/06/2021	5	1
16	08/11/2019	12	2	38	24/07/2021	5	1
17	04/12/2019	6	2	39	18/08/2021	8	1
18	03/01/2020	4	1	40	14/09/2021	6	1
19	23/01/2020	12	2	41	13/10/2021	2	1
20	17/02/2020	11	2	42	05/11/2021	2	1
21	10/03/2020	7	2	43	09/11/2021	2	1
22	11/05/2020	10	2	44	02/12/2021	2	1

Para llevar a cabo la validación, se seleccionan aleatoriamente dos instancias detalladas en la **Tabla 15**. Esto con el propósito de determinar la cantidad máxima de estibas que pueden ser acomodadas en la superficie de cada vehículo y para evaluar su rendimiento bajo distintos enfoques tanto en el orden de carga como en la rotación, manteniendo el mismo número de estibas. En otras palabras, se busca verificar si los resultados permanecen consistentes independientemente de la forma cómo se organice la disposición de las estibas.

Tabla 15

Instancias para la validación del algoritmo de primera posición del 2L.

Instancia	Cliente	Cantidad de estibas grandes	Cantidad de estibas pequeñas	Número de vehículos
1	1	3	4	3
	2	5	1	
	3	5	1	
	4	5	6	
2	1	1	4	1
	2	5	4	

10. Resultados de la Validación

En este capítulo, se presentan los resultados detallados y el análisis correspondiente de las instancias descritas previamente, abarcando tanto para el algoritmo genético, como el algoritmo de primera posición. Los resultados se obtuvieron del prototipo de programa descrito en el capítulo 9, el cual fue desarrollado en el lenguaje de programación *Python* y ejecutado a través de la GUI *Anaconda Navigator*.

10.1. Resultados de la Validación del Algoritmo Genético

La validación del algoritmo genético para la solución del problema HFFVRP se desarrolla a través de dos escenarios, en donde se pretende comparar la distancia real recorrida por la ruta de la empresa con la distancia obtenida por cada uno de estos escenarios en el prototipo de programa propuesto. Para calcular la distancia real recorrida, teniendo en cuenta el orden de visita de los clientes de los datos históricos de cada instancia previamente definida en la sección anterior, se utiliza la matriz de distancia calculada en el mismo prototipo propuesto. En este contexto, se presentan los resultados de cada uno de los escenarios:

10.1.1. Escenario 1: Visita el Depósito en la Mitad del Recorrido

En consecuencia, se exponen los resultados de las 44 instancias para el primer escenario correspondiente a la visita al depósito en la mitad del recorrido. Asimismo, se calcula el error relativo entre la distancia real y la distancia propuesta.

Tabla 16

Resultado de las instancias 1-26 para el escenario 1.

Instancia	Distancia real recorrida	Depósito como cliente en la mitad	Diferencia
1	54.068,20	54.854,70	1,5%
2	26.315,10	24.971,60	-5,1%
3	26.377,10	25.368,00	-3,8%
4	61.125,90	58.484,30	-4,3%
5	35.679,90	35.416,00	-0,7%
6	21.660,40	24.830,60	14,6%
7	56.285,70	56.285,70	0,0%
8	30.361,70	29.580,30	-2,6%
9	48.714,80	45.257,50	-7,1%
10	24.040,40	24.040,40	0,0%
11	72.314,50	55.425,70	-23,4%
12	24.862,40	24.674,50	-0,8%
13	53.775,60	52.199,20	-2,9%
14	65.157,00	61.035,10	-6,3%
15	30.097,60	28.248,30	-6,1%
16	69.122,90	69.305,60	0,3%
17	53.408,50	52.441,00	-1,8%
18	24.173,40	25.393,20	5,0%
19	129.179,30	121.496,70	-5,9%
20	50.837,80	60.381,40	18,8%
21	59.697,70	60.447,39	1,3%
22	73.955,80	66.651,50	-9,9%
23	23.957,90	22.850,30	-4,6%
24	29.022,10	28.711,39	-1,1%
25	82.478,90	66.874,20	-18,9%
26	49.290,20	49.974,70	1,4%

Tabla 17*Resultados de las instancias 27-44 para el escenario 1.*

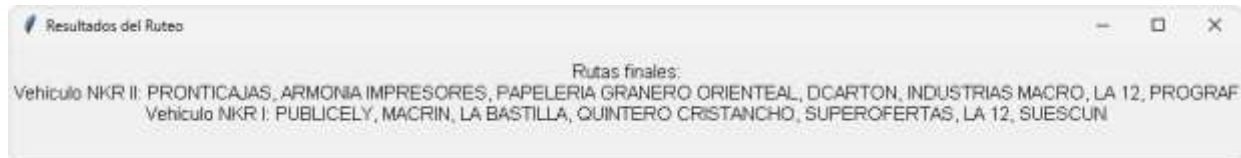
Instancia	Distancia real recorrida	Depósito como cliente en la mitad	Diferencia
27	28.642,80	32.646,10	14,0%
28	59.366,90	56.235,10	-5,3%
29	57.786,70	51.738,60	-10,5%
30	48.807,90	48.807,90	0,0%
31	27.351,00	27.045,50	-1,1%
32	36.977,40	36.977,40	0,0%
33	23.412,50	25.537,30	9,1%
34	28.630,90	26.825,40	-6,3%
35	26.834,50	30.264,40	12,8%
36	35.350,30	30.966,10	-12,4%
37	24.550,00	24.668,10	0,5%
38	32.871,80	30.928,10	-5,9%
39	30.559,00	28.085,60	-8,1%
40	29.710,60	28.083,40	-5,5%
41	23.144,70	24.954,50	7,8%
42	57.683,60	57.683,60	0,0%
43	27.618,70	27.618,70	0,0%
44	23.414,80	23.414,80	0,0%

Durante el proceso de validación por medio de las instancias en este escenario, se observó que el prototipo de programa propuesto calcula distancias más cortas que la distancia real en el 72,7% de los casos. Esta diferencia se encuentra documentada en la **Tabla 16 y 17**, en donde los valores negativos indican una reducción en la distancia propuesta en comparación con la real, con un promedio de reducción de alrededor del 1,7%. Sin embargo, también se registraron instancias en las que ocurría lo contrario. Al investigar las razones subyacentes, se determinó que el prototipo propuesto ofrecía una distancia muy similar a la real en este escenario. En consecuencia, surgió el interrogante de cómo se comportaría la distancia si el depósito fuera visitado al inicio o al final de cada ruta. Por ello, se planteó un escenario 2 que contempla lo anterior. En la **Figura 25**, se observa

la ventana emergente que proporciona el prototipo de programa propuesto para mostrar la ruta a seguir en ese recorrido. En cuanto a la distancia recorrida, se calcula y presenta en la consola.

Figura 25

Resultado del prototipo propuesto para el problema de ruteo de vehículos.



10.1.2. Escenario 2: Visita al Depósito al Inicio o al Final del Recorrido

Como resultado de este escenario, se presentan las distancias tanto del depósito como primer cliente, como el depósito como último cliente del recorrido de la ruta, para las 44 instancias planteadas. Adicionalmente, se presenta el error relativo de cada distancia propuesta.

Tabla 18

Resultado de las instancias 1-7 para el escenario 2.

Instancia	Distancia real recorrida	Depósito como primer cliente	Depósito como último cliente	Diferencia como primer cliente	Diferencia como último cliente
1	54.068,20	51.709,30	49.283,60	-4,36%	-8,85%
2	26.315,10	23.784,60	23.894,80	-9,62%	-9,20%
3	26.377,10	25.368,00	25.574,50	-3,83%	-3,04%
4	61.125,90	52.042,70	52.622,50	-14,86%	-13,91%
5	35.679,90	36.558,40	33.633,60	2,46%	-5,74%
6	21.660,40	22.131,20	21.924,10	2,17%	1,22%
7	56.285,70	48.079,70	54.553,30	-14,58%	-3,08%
9	48.714,80	42.962,20	43.050,30	-11,81%	-11,63%
10	24.040,40	22.886,60	23.234,00	-4,80%	-3,35%

Tabla 19*Resultado de las instancias 9-32 para el escenario 2.*

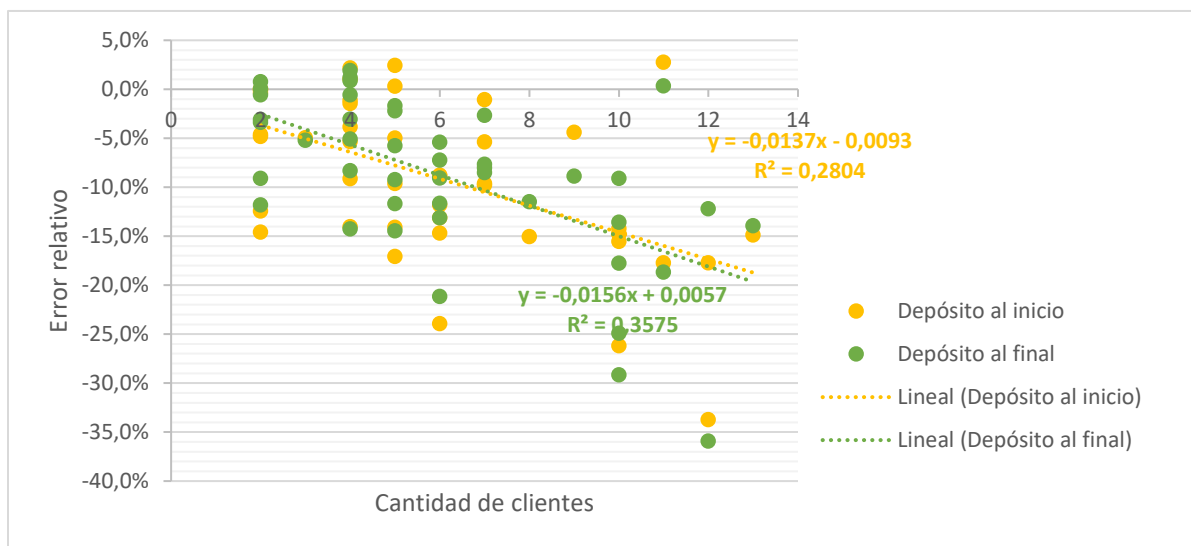
Instancia	Distancia real recorrida	Depósito como primer cliente	Depósito como último cliente	Diferencia como primer cliente	Diferencia como último cliente
11	72.314,50	53.383,89	54.311,30	-26,18%	-24,90%
12	24.862,40	22.595,10	22.802,50	-9,12%	-8,29%
13	53.775,60	48.603,90	49.477,70	-9,62%	-7,99%
14	65.157,00	55.532,00	56.324,20	-14,77%	-13,56%
15	30.097,60	25.876,10	25.809,10	-14,03%	-14,25%
16	69.122,90	56.893,10	60.703,80	-17,69%	-12,18%
17	53.408,50	48.691,60	50.526,10	-8,83%	-5,40%
18	24.173,40	23.826,10	24.041,00	-1,44%	-0,55%
19	129.179,30	85.631,30	82.801,20	-33,71%	-35,90%
20	50.837,80	52.252,10	51.024,30	2,78%	0,37%
21	59.697,70	59.075,29	58.117,60	-1,04%	-2,65%
22	73.955,80	63.435,70	60.833,30	-14,22%	-17,74%
23	23.957,90	22.850,30	21.778,90	-4,62%	-9,10%
24	29.022,10	24.938,00	25.634,00	-14,07%	-11,67%
25	82.478,90	70.313,40	58.454,30	-14,75%	-29,13%
26	49.290,20	44.977,90	45.738,00	-8,75%	-7,21%
27	28.642,80	27.223,70	28.162,00	-4,95%	-1,68%
28	59.366,90	50.144,80	53.974,70	-15,53%	-9,08%
29	57.786,70	47.545,40	47.012,90	-17,72%	-18,64%
30	48.807,90	44.037,30	45.084,10	-9,77%	-7,63%
31	27.351,00	27.045,50	27.589,30	-1,1%	0,9%
32	36.977,40	34.996,70	33.839,70	-5,4%	-8,5%
33	23.412,50	23.656,20	23.863,10	1,0%	1,9%
34	28.630,90	24.431,50	26.044,40	-14,7%	-9,0%
35	26.834,50	25.511,60	25.444,60	-4,9%	-5,2%
36	35.350,30	26.900,30	27.876,10	-23,9%	-21,1%
37	24.550,00	24.628,50	24.012,40	0,3%	-2,2%
38	32.871,80	27.269,80	28.125,60	-17,0%	-14,4%
39	30.559,00	25.966,10	27.052,30	-15,0%	-11,5%
40	29.710,60	25.815,40	25.818,70	-13,1%	-13,1%
41	23.144,70	23.144,70	23.328,90	0,0%	0,8%
42	57.683,60	57.493,70	57.718,00	-0,3%	0,1%
43	27.618,70	24.188,80	24.365,30	-12,4%	-11,8%

44	23.414,80	23.414,80	23.286,00	0,0%	-0,6%
----	-----------	-----------	-----------	------	-------

En este escenario, se observa una clara mejora en las distancias propuestas. Mientras que en el anterior escenario se tenían 12 instancias en las que la distancia real supera a la propuesta, ahora solo se tienen tres para este escenario. Además, las distancias que el algoritmo propone con el depósito en cada uno de los extremos, muestran comportamientos similares, con una reducción promedio de la distancia del 9,3% cuando el depósito se encuentra al inicio de la ruta y del 8,9% cuando se encuentra al final. En la **Figura 26**, se presenta la comparación del error relativo entre una distancia y otra.

Figura 26

Error relativo de las distancias del escenario 2 con respecto a la distancia real



A partir de los datos, se puede observar una tendencia clara: para las rutas con menos de ocho clientes, las instancias donde el depósito se sitúa al comienzo de la ruta arrojan mejores resultados que aquellas en las que el depósito está al final. Esto se debe a que la pendiente en estas rutas es más pronunciada. Sin embargo, en rutas que incluyen más de ocho clientes, que es el punto

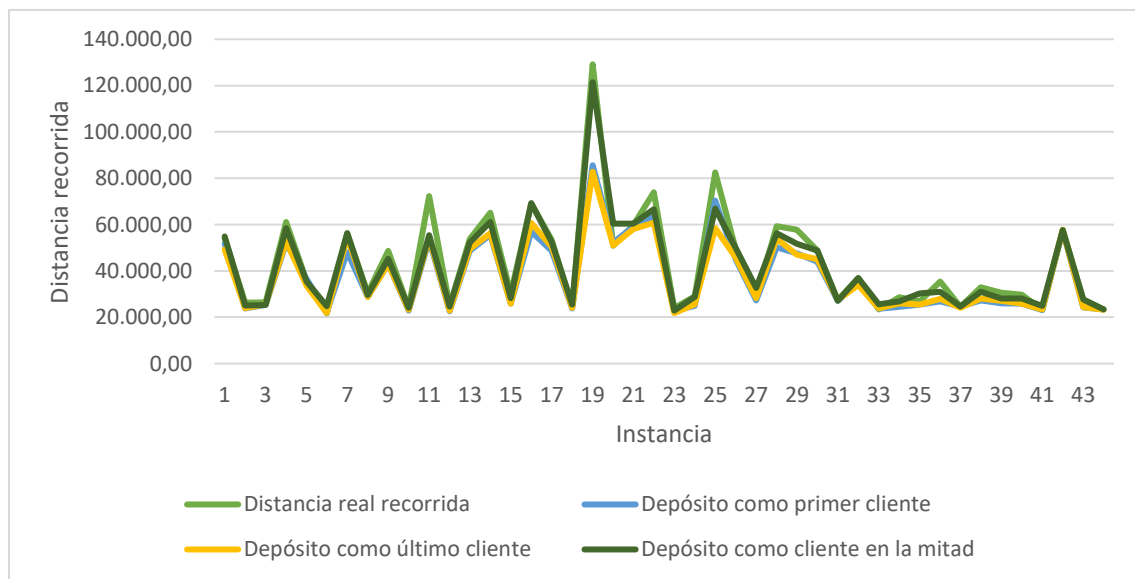
en donde se cruzan las líneas de tendencia, es preferible tener al depósito al final del recorrido de la ruta, ya que muestran un desempeño más favorable.

Considerando lo anterior, y teniendo en cuenta que el promedio de entregas por ruta de la empresa es de 5,2, sería más conveniente para la empresa adaptar una estrategia en la cual el depósito se visite antes de atender a cualquier otro cliente.

A continuación, se muestra un gráfico que compara las distancias recorridas de cada escenario para cada una de las 44 instancias:

Figura 27

Comparación entre las distancias propuesta respecto a la distancia real.



La distancia con el depósito como punto intermedio en la ruta, es la que más se aproxima a la distancia real recorrida. Esto refleja la estrategia de la empresa de atender inicialmente a los clientes que no necesitan productos del depósito, dirigirse luego al mismo para recoger los productos restantes y, finalmente, distribuir los pedidos pendientes.

A continuación, se presenta una tabla que estima los ahorros potenciales de este escenario durante los tres años de estudio, considerando un consumo promedio de los vehículos 55 kilómetros por galón de Diesel y de un precio de \$9065 por galón, que era el precio a nivel nacional a 11 de septiembre de 2023:

Tabla 20

Ahorros potenciales en pesos colombianos de los escenarios propuestos.

Distancia real recorrida: 1.898,67	Depósito como cliente en la mitad	Depósito como primer cliente	Depósito como último cliente
Distancia total	1.837,68	1.676,56	1.677,57
Diferencia respecto a la distancia real	60,99	222,11	221,10
Galones de Diesel no consumidos	1,11	4,03	4,02
Ahorro estimado	\$10.062,15	\$36.607,77	\$36.441,30

Teniendo en cuenta que el promedio de entregas por ruta de la empresa es de 5,2 y que la estrategia donde el depósito es el primer cliente resulta en una menor distancia recorrida y un mayor ahorro, sería conveniente que la empresa adopte esta estrategia, visitando el depósito antes de atender a cualquier otro cliente.

10.2. Resultados de la Validación del Algoritmo de Primera Posición 2L

A continuación, se presentan los resultados de las 2 instancias propuestas y la manera como se abordaron.

10.2.1. Instancia 1

Inicialmente, se realiza la validación de la primera instancia utilizando el software de la herramienta *SeaRates*, la cual se encuentra disponible en el siguiente enlace: <https://www.searates.com/es/reference/stuffing/>. Esta plataforma digital, de acceso abierto, está

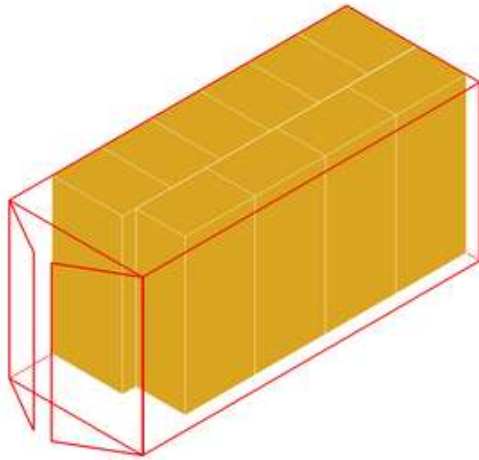
diseñada para simplificar y optimizar el proceso de envío de mercancías, permitiendo a transportistas anunciar la disponibilidad de sus camiones y a propietarios o empresas que busquen transportar mercancías comparar tarifas de transporte (*Searates*, n.d.). Una de sus características destacadas es una calculadora en línea de carga y relleno de contenedores, con la cual se procede a realizar la validación del algoritmo propuesto.

Este proceso de validación, se realiza para los tres tipos de vehículo con los que cuenta la empresa, con el fin de determinar la cantidad máxima de estibas que pueden ser colocadas sobre la superficie de cada vehículo bajo condiciones normales. En la herramienta, se utilizan cubos para representar el espacio que puede ocupar la carga de cada estiba, asegurando que tengan la misma altura que el vehículo, ya que esta dimensión no es relevante para el modelo propuesto. En la **Figura 28**, se presentan los resultados obtenidos en el software de la colocación de las estibas para el vehículo NKR II. Se consideraron inicialmente estibas grandes, seguido de estibas pequeñas y finalmente, se evaluó un conjunto de ambos tipos de estibas.

Figura 28

Cantidad máxima de estibas para el vehículo NKR II.

a)



Transporte Nº1 (NKR II 1 unidades)

Packed: 10 packages: (100%).

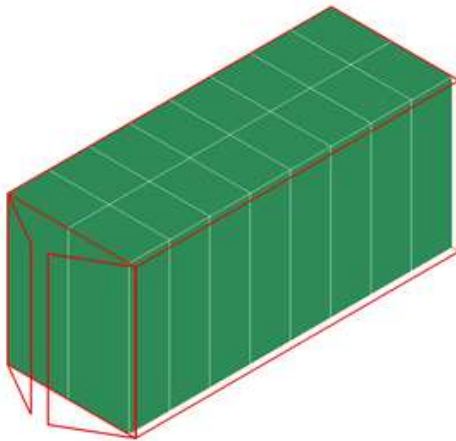
Including:

■ **Estiba grande** - 10 paquetes (100%)

Cargo volume: 15.4 m³
(77% of volume)

Cargo weight: 10 kg.
(0% of max payload)

b)



Transporte Nº1 (NKR II 1 unidades)

Packed: 16 packages: (100%).

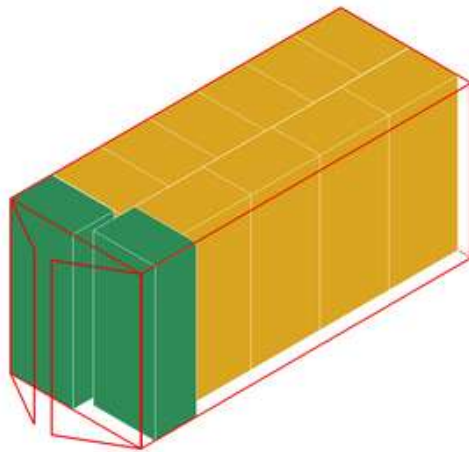
Including:

■ **Estiba pequeña** - 16 paquetes (100%)

Cargo volume: 19.01 m³
(95% of volume)

Cargo weight: 16 kg.
(0% of max payload)

c)



Transporte Nº1 (NKR II 1 unidades)

Packed: 12 packages: (100%),

Including:

Estiba grande - 10 paquetes (100%)

Estiba pequeña - 2 paquetes (100%)

Cargo volume: 17.78 m³

(89% of volume)

Cargo weight: 12 kg.

(0% of max payload)

Nota: Tomado del software *Searates*. a) solamente estibas grandes, b) solamente estibas pequeña y c) los dos tipos de estibas

En la siguiente tabla se detalla la información correspondiente a los tres vehículos disponibles en la empresa, proporcionados por el software mencionado:

Tabla 21

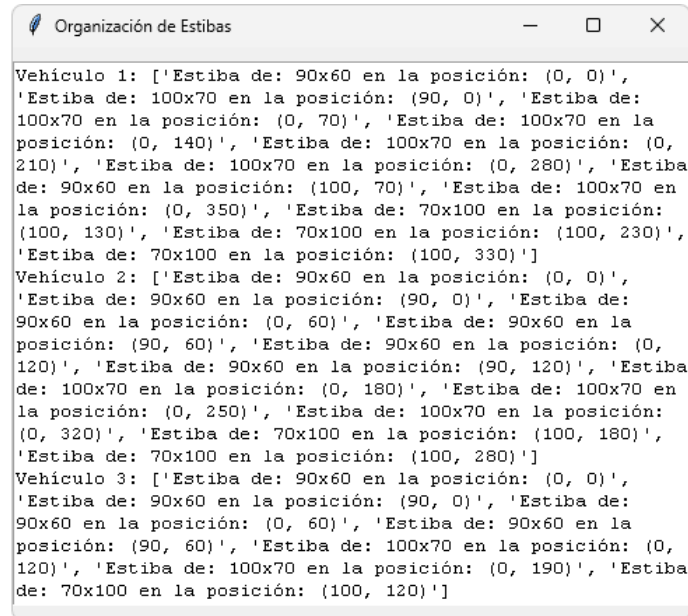
Consolidado de las cantidades máximas de estibas para cada vehículo.

N.	Vehículo	Estibas grandes	Estibas pequeñas	En conjunto	
				Estibas grandes	Estibas pequeñas
1	NKR II	10	16	10	2
2	NKR I	9	12	5	6
3	NHR	7	10	3	4

Una vez calculada la cantidad de estibas máximas que se pueden disponer en cada uno de los vehículos en la herramienta, se procede a realizar el ejercicio en el prototipo del programa propuesto, obteniendo los resultados en la **Figura 29**. Estos resultados, aparecen en una ventana emergente y se componen del número del vehículo, las medidas de la estibas (ancho por largo) y la posición en coordenadas (x,y), en donde x es el ancho y y el largo.

Figura 29

Resultados del prototipo de programa propuesto.



```

Organización de Estibas
-----
Vehículo 1: ['Estiba de: 90x60 en la posición: (0, 0)',
'Estiba de: 100x70 en la posición: (90, 0)', 'Estiba de:
100x70 en la posición: (0, 70)', 'Estiba de: 100x70 en la
posición: (0, 140)', 'Estiba de: 100x70 en la posición: (0,
210)', 'Estiba de: 100x70 en la posición: (0, 280)', 'Estiba
de: 90x60 en la posición: (100, 70)', 'Estiba de: 100x70 en
la posición: (0, 350)', 'Estiba de: 70x100 en la posición:
(100, 130)', 'Estiba de: 70x100 en la posición: (100, 230)',
'Estiba de: 70x100 en la posición: (100, 330)']
Vehículo 2: ['Estiba de: 90x60 en la posición: (0, 0)',
'Estiba de: 90x60 en la posición: (90, 0)', 'Estiba de:
90x60 en la posición: (0, 60)', 'Estiba de: 90x60 en la
posición: (90, 60)', 'Estiba de: 90x60 en la posición: (0,
120)', 'Estiba de: 90x60 en la posición: (90, 120)', 'Estiba
de: 100x70 en la posición: (0, 180)', 'Estiba de: 100x70 en
la posición: (0, 250)', 'Estiba de: 100x70 en la posición:
(0, 320)', 'Estiba de: 70x100 en la posición: (100, 180)',
'Estiba de: 70x100 en la posición: (100, 280)']
Vehículo 3: ['Estiba de: 90x60 en la posición: (0, 0)',
'Estiba de: 90x60 en la posición: (90, 0)', 'Estiba de:
90x60 en la posición: (0, 60)', 'Estiba de: 90x60 en la
posición: (90, 60)', 'Estiba de: 100x70 en la posición: (0,
120)', 'Estiba de: 100x70 en la posición: (0, 190)', 'Estiba
de: 70x100 en la posición: (100, 120)']

```

De acuerdo con los resultados, se puede evidenciar que la cantidad de estibas máxima para cada vehículo de las estibas en conjunto, corresponde a la cantidad de estibas encontrada en el ejercicio realizado en la herramienta SeaRates. Lo anterior indica que el algoritmo de primera posición funciona correctamente, igualando la propuesta de los softwares logísticos comerciales.

10.2.2. Instancia 2

Posteriormente, se lleva a cabo una validación manual, como segunda instancia, mediante el análisis de gráficos a escala. Este enfoque se origina por políticas propias de la empresa, en donde dan prioridad a la carga de las estibas pequeñas debido a su tendencia a ser las más pesadas, lo que implica que transportan mercancía de mayor valor comercial. En este contexto, se busca verificar si los resultados permanecen consistentes independientemente de la forma cómo se organice la disposición de las estibas. Esto implica colocar de primeras las estibas pequeñas y luego las grandes del mismo cliente, sin perder de vista que el problema considera una carga

secuencial, donde es esencial considerar el orden de los clientes en la ruta y cargar las estibas de cada cliente particular de manera ordenada.

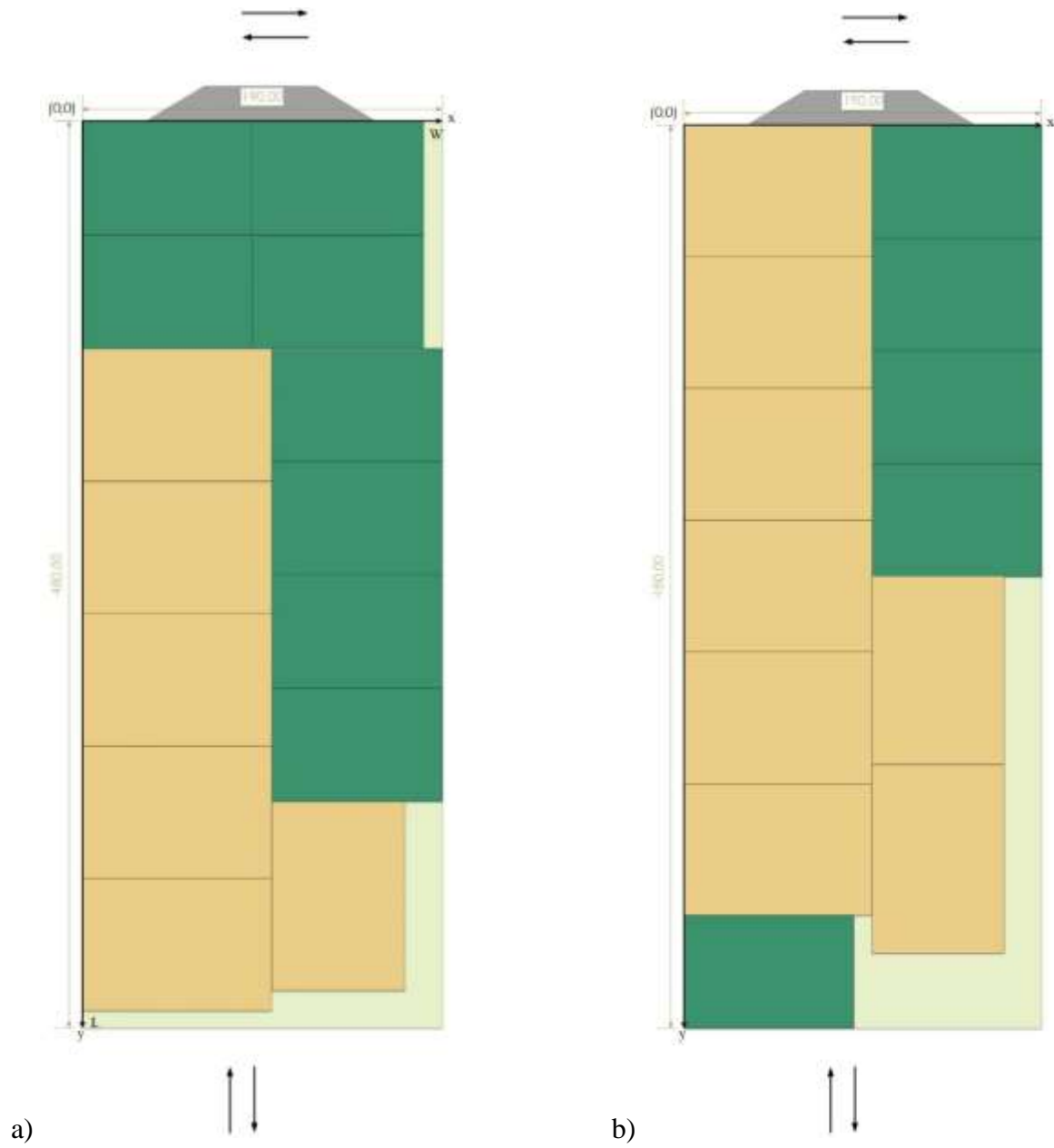
En la siguiente **Figura 30**, se presenta la disposición de estibas priorizando tanto la carga de las estibas pequeñas como la carga de las estibas grandes en el vehículo NKR II de forma manual.

Este ejercicio también se desarrolla mediante el prototipo de programa propuesto y su resultado se presenta en la **Figura 31**. Los resultados concuerdan manualmente y en el programa propuesto. De acuerdo con lo anterior se pudieron identificar las siguientes observaciones:

- Para el vehículo NKR II, al dar prioridad a las estibas pequeñas, la cantidad de estibas grandes disminuyó de 10 a 6, mientras que la cantidad de estibas pequeñas aumentó de 2 a 8 estibas.
- En el caso de los vehículos NKR I y NHR, la cantidad de estibas se mantuvo constante en ambos enfoques.
- Por otro lado, priorizando las estibas grandes en el orden de carga, se observó que la cantidad total de estibas disminuyó en 1 estiba. En otras palabras, se cargó una estiba menos del total al dar prioridad a las estibas grandes.
- Asimismo, se encontró que, a pesar de la prioridad en el orden de carga que se emplee y de que las posiciones de las estibas originales o rotadas cambie, la cantidad de estibas que se pueden cargar sigue siendo constante.
- Además, en el proceso, se comprobó que efectivamente la búsqueda de la primera posición disponible se lleva a cabo desde la esquina superior izquierda hacia la derecha y hacia abajo de la superficie del vehículo.

Figura 30

Priorización en el orden de carga de las estibas.



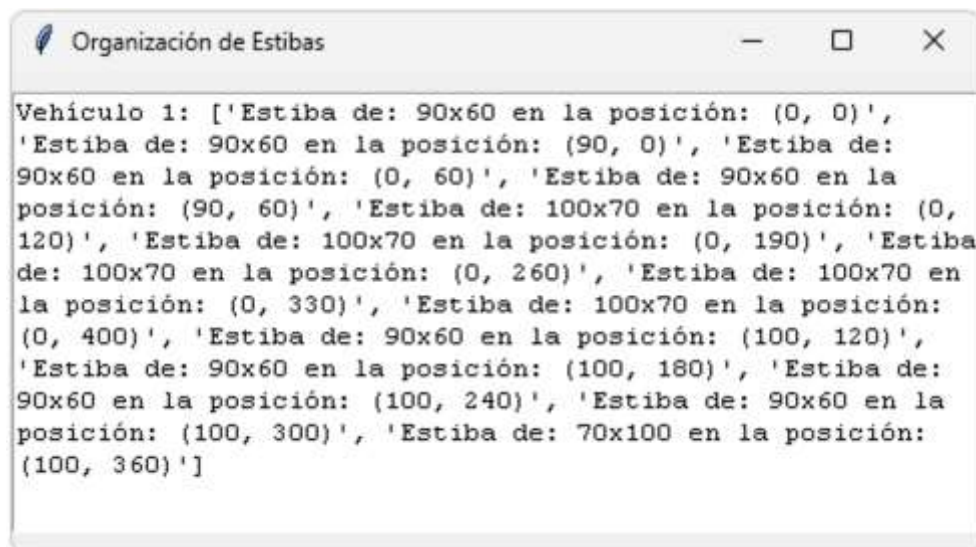
Nota: a) Primero se cargan las estibas pequeñas dentro del vehículo: 6 grandes y 8 pequeñas, b) Primero se cargan las estibas grandes dentro del vehículo: 8 grandes y 5 pequeñas

Esto no sólo confirmó que la política de la empresa de dar prioridad a la carga de las estibas pequeñas por su valor comercial está bien fundamentada, sino que también que la probabilidad de que los resultados permanezcan constantes independientemente de la priorización en el orden de la carga de las estibas es la misma en dos de los tres vehículos. En cuanto a la orientación de las estibas, se evidenció que es independiente de la prioridad en el orden de carga.

Para finalizar la validación, se realiza la comprobación de esta instancia en el software *SeaRates*, obteniendo los mismos resultados manualmente, en el programa propuesto (**Figura 31**) y en el software logístico (**Figura 32**).

Figura 31

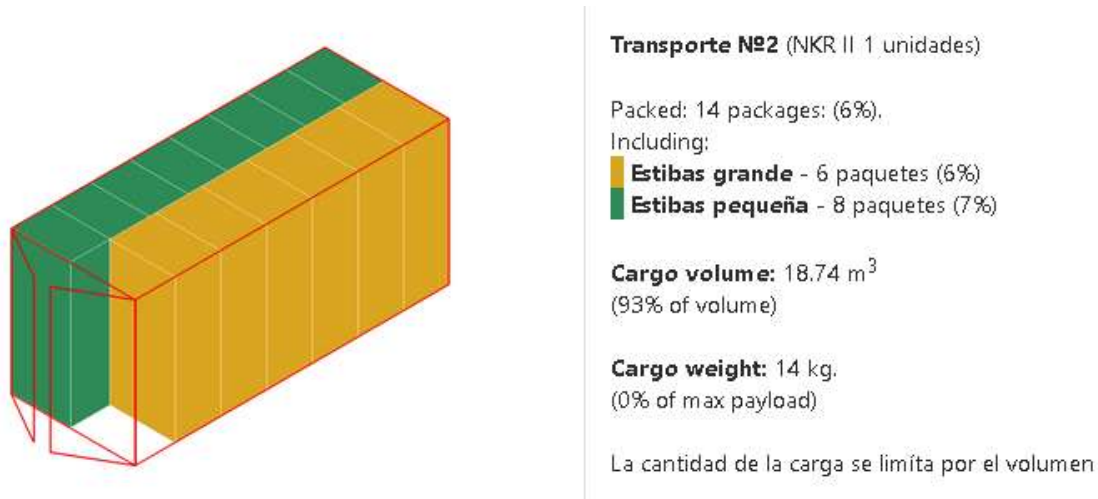
Resultados del prototipo de programa propuesto.



```
Organización de Estibas
Vehículo 1: ['Estiba de: 90x60 en la posición: (0, 0)',
'Estiba de: 90x60 en la posición: (90, 0)', 'Estiba de:
90x60 en la posición: (0, 60)', 'Estiba de: 90x60 en la
posición: (90, 60)', 'Estiba de: 100x70 en la posición: (0,
120)', 'Estiba de: 100x70 en la posición: (0, 190)', 'Estiba
de: 100x70 en la posición: (0, 260)', 'Estiba de: 100x70 en
la posición: (0, 330)', 'Estiba de: 100x70 en la posición:
(0, 400)', 'Estiba de: 90x60 en la posición: (100, 120)',
'Estiba de: 90x60 en la posición: (100, 180)', 'Estiba de:
90x60 en la posición: (100, 240)', 'Estiba de: 90x60 en la
posición: (100, 300)', 'Estiba de: 70x100 en la posición:
(100, 360)']
```

Figura 32

Cantidad máxima de estibas priorizando las estibas pequeñas



Nota: Tomado del software *Searates*.

11. Conclusiones

De acuerdo con el análisis bibliométrico y la revisión de literatura realizada, se pudo evidenciar la creciente importancia de abordar los desafíos del ruteo de vehículos con restricciones de carga bidimensional, específicamente el 2L-CVRP y el 2L-HFVRP. Resaltando también que la mayoría de los estudios previos han pasado por alto la realidad de las empresas, que operan con flota de vehículos limitada debido a restricciones de recursos. Esta brecha entre la teoría y la práctica resalta la necesidad de futuras investigación que aborden este tipo de problemas de ruteo de vehículos. Importancia de desarrollar soluciones más realistas y aplicables considerando las complejidades logísticas del mundo real.

Tanto el diagnóstico realizado al proceso de distribución de la Cooperativa, como la aplicación de las etapas del proceso de descubrimiento de conocimiento en bases de datos (KDD) realizada a los datos históricos, permitieron la comprensión anticipada de la operación del negocio. Esto reveló patrones de interés que sirvieron como base para identificar la necesidad de

implementación de un modelo matemático que optimizara esta parte de la logística y el modelo que más se ajustara a su operación. Además, facilitó la identificación de los clientes clave para el cálculo de la matriz de distancia base. Como resultado de la limpieza de datos, y en respuesta a la Gerencia de la empresa, se generó una auditoría interna, en donde se retroalimentaron las actividades realizadas a las bases de datos, que posteriormente, sirvieron de cimiento para la implementación de un sistema ERP SAP que la empresa adquirió durante el desarrollo del proyecto.

Por otro lado, se derivó el descubrimiento y exploración de la opción gratuita de código abierto OpenRouteService (OSM). Si bien es cierto que cuenta con diferentes limitaciones en cuanto a las dimensiones de la matriz de distancia que se puede calcular, esta herramienta se mostró bastante eficiente e intuitiva, lo que la convierte en una valiosa adición que todos deberían conocer.

La presente investigación presentó un modelo que describe correctamente el problema 2L-HFFVRP, basado principalmente en un algoritmo genético. Sobre la marcha se evidenció la necesidad de utilizar una heurística diferente para abordar el problema de carga bidimensional que fue el algoritmo de primera posición. Es importante destacar que el algoritmo genético propuesto en este proyecto no se basó en un autor o referencia en particular, sino que se diseñó específicamente para abordar los requerimientos del problema planteado. Por otro lado, no se diseñaron solamente los algoritmos anteriormente mencionados, comprendidos en el alcance de este trabajo, sino que se desarrolló un prototipo completamente funcional que incluye una interfaz de usuario intuitiva para facilitar la utilización. Este prototipo es completamente original y ha sido desarrollado por los miembros de este proyecto como resultado de su labor investigativa.

Basándose en los resultados obtenidos en el capítulo 10, se concluye que el desempeño del algoritmo propuesto es comparable al de la estrategia actual implementada por la empresa. Por lo tanto, podría no ofrecer el impacto significativo que se esperaba. No obstante, se propuso una nueva estrategia de distribución en las rutas. Al contrastar con la estrategia previa, se evidenció una notable mejora en las distancias obtenidas. Reducir estas distancias no solo optimiza el tiempo de entrega, sino que también implica un ahorro notable de combustible y una gestión más eficiente de los recursos.

La optimización de recursos en la logística es esencial en la cadena de suministros actual. Modelar estos procesos aporta un valor significativo a cualquier empresa que lo implemente. Es vital reconocer que cada cadena de suministros es distintiva, no sólo por los productos que circulan en ella, sino también por elementos infraestructurales, como maquinaria, mobiliario y el equipo humano encargado de los procesos logísticos. Acometer la tarea de aplicar un modelo de optimización a una situación logística real es un desafío altamente complejo. Para simplificar este proceso, es crucial que el modelo identifique y se adapte a las características intrínsecas del problema específico. Un modelo que refleje fielmente la realidad garantizará resultados más eficientes y efectivos.

12. Recomendaciones

De acuerdo con los resultados obtenidos en este trabajo de investigación, se identificaron diferentes limitaciones que podrían ampliarse y ser objeto de estudio en futuras investigaciones. Estas limitaciones se deben en gran medida a la escasez de investigaciones previas sobre el problema en cuestión. En consecuencia, este trabajo sienta las bases necesarias para abordar el problema 2L-HFFVRP con la restricción de carga secuencial y no orientada, mediante métodos no

exactos. Si se quiere realizar un estudio con resultados significativos, se propone la implementación de instancias más grandes y diversos tipos de artículos.

En el desarrollo de algoritmos genéticos, se propone explorar otros métodos para la generación de la población inicial, teniendo en cuenta que, en esta investigación, la población se generó de manera puramente aleatoria. Además, se recomienda profundizar en los problemas de ruteo de vehículos de flota fija heterogénea. Esto dado a que es un problema más común de lo que puede llegar a parecer.

En el problema de carga, se sugiere la utilización de heurísticas híbridas para su solución. Esta estrategia favorecerá la optimización del espacio de carga de contenedores o vehículos, explorando diferentes enfoques combinados de manera sinérgica para alcanzar soluciones óptimas.

Con respecto al caso de estudio con la empresa Coimpresores del Oriente, se recomienda:

Seguir con el desarrollo y mejora del prototipo de programa propuesto, dado que ha arrojado resultados positivos. Esto no solo se debe a su potencial para general a largo plazo, sino también a su capacidad para fortalecer la consecución de objetivos estratégicos relacionados con logística de la empresa, así como brinda valiosas oportunidades de aprendizaje, mejora continua y gestión del cambio.

Asimismo, se propone a la empresa reevaluar su estrategia de entregas, debido a la similitud encontrada en uno de los escenarios de estudio planteados en esta investigación el cual presentó resultados desfavorables en relación con los demás escenarios planteados. En este contexto se propusieron estrategias que generaban menores distancias recorridas por los vehículos y por ende mayores ahorros en cuestión tiempo, dinero y recurso humano.

Referencias Bibliográficas

- Akpinar, S. (2016). Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications*, 61, 28–38.
<https://doi.org/10.1016/j.eswa.2016.05.023>
- Bischoff, E., & Ratcliff, M. (1995). Issue in the Development of Approaches to Container Loading. *Omega, Int.*, 23(4), 1377–1390.
- Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3), 268–308.
<https://doi.org/10.1145/937503.937505>
- Braekers, K., Ramaekers, K., & Van Nieuwenhuysse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering*, 99, 300–313. <https://doi.org/10.1016/j.cie.2015.12.007>
- Candido, L. C. X., & De Souza, L. V. (2022). Mathematical Model and Simulated Annealing Algorithm for the Two-Dimensional Loading Heterogeneous Fixed Fleet Vehicle Routing Problem. *Mathematical Problems in Engineering*, 2022.
<https://doi.org/10.1155/2022/6012105>
- Clausen, J. (1999). Branch and bound algorithms-principles and examples. *Department of Computer Science, University of ...*, 1–30. <https://doi.org/10.1.1.5.7475>
- Córdoba, E. (2020). Solución a un modelo de optimización para la gestión de inventario y ruteo de vehículos de dos escalones con transbordos laterales (2E-IRPT) a través de una metaheurística híbrida. 1, 105–112.

Cortez, A. (2004). Teoría de la complejidad computacional y teoría de la computabilidad.

Revista de Investigación De, 105(1), 102–105.

http://200.62.146.19/bibvirtualdata/publicaciones/risi/N1_2004/a14.pdfhttp://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/risi/n1_2004/a14.pdf<http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3216>

Côté, J. F., Guastaroba, G., & Speranza, M. G. (2017). The value of integrating loading and routing. *European Journal of Operational Research*, 257(1), 89–105.

<https://doi.org/10.1016/j.ejor.2016.06.072>

De Antonio Suárez, O. (2011). Una aproximación a la heurística y metaheurísticas. *Universidad Antonio Nariño*, 8.

Departamento Nacional de Planeación. (2020). Encuesta Nacional de Logística 2020. *Encuesta Nacional de Logística 2020*, 170.

Descripción general de la API de Geocoding. (n.d.). Retrieved March 15, 2023, from

<https://developers.google.com/maps/documentation/geocoding/overview?hl=es-419>

Dey, B. K., Bhuniya, S., & Sarkar, B. (2021). Involvement of controllable lead time and variable demand for a smart manufacturing system under a supply chain management. *Expert Systems with Applications*, 184(May), 115464. <https://doi.org/10.1016/j.eswa.2021.115464>

Systems with Applications, 184(May), 115464. <https://doi.org/10.1016/j.eswa.2021.115464>

Documentación de Google Maps Platform | Directions API. (2023).

<https://developers.google.com/maps/documentation/directions?hl=es-419>

Documentación de Google Maps Platform | Distance Matrix API. (2023).

<https://developers.google.com/maps/documentation/distance-matrix?hl=es-419>

- Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., & Agustin, A. (2016). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, 236(2), 383–404.
<https://doi.org/10.1007/s10479-014-1551-4>
- Dósa, G., & Epstein, L. (2018). The tight asymptotic approximation ratio of First Fit for bin packing with cardinality constraints. *Journal of Computer and System Sciences*, 96(2018), 33–49. <https://doi.org/10.1016/j.jcss.2018.03.004>
- Duhamel, C., Lacomme, P., Quilliot, A., & Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers and Operations Research*, 38(3), 617–640. <https://doi.org/10.1016/j.cor.2010.08.017>
- Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, 36(3), 655–673. <https://doi.org/10.1016/j.cor.2007.10.021>
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints (Networks (2008) 51 (4-18)). *Networks*, 51(2), 153. <https://doi.org/10.1002/net.20245>
- Gestal, M., Rivero, D., Rabuñal, J. R., Dorado, J., & Pazos, A. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética This page intentionally left blank.*
- Glover, F., & Batista, B. (2006). *Introducción a la Búsqueda Tabu. 1*, 1–36.
- Glover, F., & Gary, A. (2003). Handbook of Metaheuristics. In *International Series in Operations Research & Management Science* (Vol. 21, Issue 3).

<https://doi.org/10.2307/3198208>

Google Maps Platform. (n.d.). Retrieved March 15, 2023, from

<https://developers.google.com/maps?hl=es-419>

Grasas, A., Juan, A. A., Faulin, J., de Armas, J., & Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: A survey and some applications.

Computers and Industrial Engineering, *110*, 216–228.

<https://doi.org/10.1016/j.cie.2017.06.019>

Hillier, F.-S., & Lieberman, G.-J. (2010). Introducción a la Investigación de Operaciones. In

Syria Studies (Vol. 7, Issue 1).

[https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090](https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~reynal/Civil)

<http://www.econ.upf.edu/~reynal/Civil>

wars_12December2010.pdf%0Ahttps://think-

<asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625>

Iori, M., Salazar Gonzales, J. J., & Vigo, D. (2007). An exact approach for the green vehicle routing problem with two-dimensional loading constraints. *Computers and Operations*

Research, *41*(2), 253–264. <https://doi.org/10.1287/trsc.1060.016>

Jokar Arsanjani, J., Zipf, A., Mooney, P., & Helbich, M. (2015). An Introduction to

OpenStreetMap in Geographic Information Science: Experiences, Research, and

Applications. *Lecture Notes in Geoinformation and Cartography*, *0*(9783319142791), 1–15.

https://doi.org/10.1007/978-3-319-14280-7_1

Junqueira, L. (2013). *Modelos e algoritmos de otimização para problemas integrados de roteamento e carregamento de veículos*. 228.

- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. In *Multimedia Tools and Applications* (Vol. 80, Issue 5). Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-020-10139-6>
- Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1), 1–21.
<https://doi.org/10.1016/j.ejor.2015.07.020>
- Koza, J. R. (2003). *Genetic Programming: On the Programming of Computers By Means of Natural Selection Complex Adaptive Systems*.
- Laporte, G., Toth, P., & Vigo, D. (2013). Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, 2(1–2), 1–4.
<https://doi.org/10.1007/s13676-013-0020-6>
- Leung, S. C. H., Zhang, Z., Zhang, D., Hua, X., & Lim, M. K. (2013). A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 225(2), 199–210.
<https://doi.org/10.1016/j.ejor.2012.09.023>
- Linares, P., Sánchez, P., Ramos, A., Sarabia, Á., & Vitoriano, B. (2001). *Modelos matemáticos de optimización*.
- Naanaa, A., & Belghith, S. (2018). An efficient guided local search approach for multiuser detection in UWB systems. *Physical Communication*, 26, 141–148.
<https://doi.org/10.1016/j.phycom.2017.12.006>
- Openrouteservice*. (n.d.). Retrieved March 15, 2023, from <https://openrouteservice.org/>

OpenStreetMap. (n.d.). Retrieved March 15, 2023, from

<https://www.openstreetmap.org/#map=5/4.632/-74.299>

OpenStreetMap Wiki. (n.d.). Retrieved March 15, 2023, from

https://wiki.openstreetmap.org/wiki/ES:Página_principal

ORS Maps. (n.d.). Retrieved March 15, 2023, from

<https://maps.openrouteservice.org/#/place/@8.76709,49.510944,6>

Pabon, A., & Picón, J. (2019). Un algoritmo genético para el problema de distribución de ayuda humanitaria con tiempos de viaje estocásticos y flota heterogénea para la ciudad de Bucaramanga. *Carbohydrate Polymers*.

Poli, R., Kennedy, J., & Blackwell, T. (2006). Particle swarm optimization: An overview. *Swarm Intell*, 1, 33–57. <https://doi.org/10.1007/s11721-007-0002-0>

Reina, D. G., Córdoba, A. T., & del Nozal, Á. R. (2020). *Algoritmos Genéticos con Python: Un enfoque practico para resolver problemas de ingeniería*.

Sang, R. (n.d.). *El proceso de descubrimiento de conocimiento a partir de datos Objetivos , fases y problemática*.

Searates. (n.d.). Retrieved August 15, 2023, from <https://www.searates.com/es/reference/stuffing/>

Soremekun, G., Gürdal, Z., Haftka, R. T., & Watson, L. T. (2001). Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Computers and Structures*, 79(2), 131–143. [https://doi.org/10.1016/S0045-7949\(00\)00125-5](https://doi.org/10.1016/S0045-7949(00)00125-5)

Subramanian, A., Penna, P. H. V., Uchoa, E., & Ochi, L. S. (2012). A hybrid algorithm for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research*,

221(2), 285–295. <https://doi.org/10.1016/j.ejor.2012.03.016>

Taha, H. A. (n.d.). *Operaciones 7*.

Timarán Pereira, S. R., Hernández Arteaga, I., Caicedo Zambrano, S. J., Hidalgo Troya, A., & Alvarado Pérez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. *Descubrimiento de Patrones de Desempeño Académico Con Árboles de Decisión En Las Competencias Genéricas de La Formación Profesional, 2016*, 63–86.

Tolosa A, M. Á. (2017). *Modelo de ruteo de vehículos para la distribución de producto terminado en el área metropolitana de Bucaramanga, caso: Espumas Santander S.A.S.*

Vansteenwegen, P., Souffriau, W., Berghe, G. Vanden, & Oudheusden, D. Van. (2009). A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research*, 196(1), 118–127. <https://doi.org/10.1016/j.ejor.2008.02.037>

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. H. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 265(3), 843–859. <https://doi.org/10.1016/j.ejor.2017.08.035>

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A Guided Tabu Search for the Vehicle Routing Problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3), 729–743. <https://doi.org/10.1016/j.ejor.2007.05.058>

Zhang, D., Dong, R., Si, Y. W., Ye, F., & Cai, Q. (2018). A hybrid swarm algorithm based on ABC and AIS for 2L-HFCVRP. *Applied Soft Computing Journal*, 64, 468–479. <https://doi.org/10.1016/j.asoc.2017.12.012>