Icono

Descripción generada automáticamente

**FACULTAD DE INGENIERÍA**

**INGENIERIA CIVIL INFORMÁTICA**

**ANÁLISIS E IMPLEMENTACIÓN DEL VRP WITH HETEROGENEOUS TRANSPORT FLEET**

**Tesis de pregrado para optar al título de Ingeniero Civil Informático**

Bastian Ignacio Moya Carrasco

Felipe Ismael Moya Carrasco

Profesor tutor:

Dr. Gustavo Gatica

Santiago, Chile

2024

Felipe Moya, Bastian Moya.

Facultad de Ingeniería, Universidad Andrés Bello, Antonio Varas 880, Santiago, Chile.

Recibida ; Primera Revisión 28 de Noviembre 2024; Segunda Revisión 5 de Diciembre 2024; Aceptada ;

Se permite el uso, distribución y reproducción en cualquier medio, siempre que se otorgue el crédito adecuado al autor original, se proporcione un enlace a la licencia y se indique si se han realizado cambios.

Esta tesis se basa en el trabajo previo de Jiansheng Liu, Bin Yuan, Yang Zan (<https://doi.org/10.21203/rs.3.rs-2942819/v1>). El contenido original ha sido utilizado con permiso y según los términos especificados por el autor/propietario original.

Agradecimientos

AQUÍ VAN AGRADECIMIENTOS, A QUIEN NO SE.

# Resumen

Este trabajo se centra en replicar y evaluar el modelo presentado en el paper *"The finished vehicle routing problem with a heterogeneous transport fleet"* (HVRP–FVL), que aborda una generalización del problema de ruteo de vehículos. Este modelo considera una flota heterogénea de vehículos e incorpora diversos costos de transporte, como peajes, costos laborales, consumo energético, penalizaciones por tiempo y emisiones de carbono. El objetivo del HVRP–FVL es minimizar el costo total de la flota mediante una metaheurística basada en algoritmos genéticos (GA) con codificación dual, cruzamiento adaptativo, mutación y operadores de escalada.

Además, se revisaron tres versiones de este estudio, para establecer variables no declaradas o mal dimensionadas. A pesar de las inconsistencias encontradas en la formulación matemática y los valores reportados, se trabajó en replicar mecánicamente los casos de estudio provistos en el paper. Esto permitió evaluar la funcionalidad del modelo a nivel práctico, ya que por la falta de información no se puede validar el modelo completo.

El objetivo principal fue diseñar y replicar, mediante el uso de Python, la funcionalidad propuesta, que se utilizó para buscar rutas óptimas que minimizaran el uso de camiones, garantizaran la puntualidad dentro de las ventanas de tiempo y maximice la capacidad de carga de cada vehículo. Si bien no se logró corroborar la veracidad total del modelo teórico, se obtuvieron rutas optimizadas que cumplen con los criterios logísticos establecidos, demostrando la aplicabilidad de este enfoque en escenarios reales de transporte.

# Tabla de contenidos

[Agradecimientos iii](#_Toc185824155)

[1. Resumen iv](#_Toc185824156)

[2. Tabla de contenidos v](#_Toc185824157)

[3. Tabla de contenidos vi](#_Toc185824158)

[3.1 Ilustraciones vi](#_Toc185824159)

[3.2 Tablas vi](#_Toc185824160)

[4. Conceptos Clave vii](#_Toc185824161)

[5. Introducción 1](#_Toc185824162)

[6. Marco teórico 2](#_Toc185824163)

[6.1 Evolución Histórica de la formulación de VRP 2](#_Toc185824164)

[7. Metodología 3](#_Toc185824165)

[8. Funcionalidades del Software 5](#_Toc185824166)

[8.1 Variables especiales: 5](#_Toc185824167)

[8.2 Proceso de asignación: 5](#_Toc185824168)

[8.3 Salida de datos: 6](#_Toc185824169)

[9. Implementación Técnica 7](#_Toc185824170)

[10. Modelo Matemático 8](#_Toc185824171)

[10.1 Descripción general: 8](#_Toc185824172)

[10.2 Conjuntos y parámetros: 8](#_Toc185824173)

[10.3 Variable de decisión: 8](#_Toc185824174)

[10.4 Función Objetivo: 8](#_Toc185824175)

[10.5 Restricciones: 8](#_Toc185824176)

[11. Código Fuente 10](#_Toc185824177)

[12. Resultados 11](#_Toc185824178)

[12.1 Casos de estudio 11](#_Toc185824179)

[12.2 Ordenes Generadas 14](#_Toc185824180)

[13. Conclusiones 20](#_Toc185824181)

[14. Referencias 21](#_Toc185824182)

# Tabla de contenidos

## 3.1 Ilustraciones

[Ilustración 1 Modelos originarios del problema VRP 2](#_Toc185824183)

[Ilustración 2 Diagrama de componentes 7](#_Toc185824184)

[Ilustración 3 Diagrama de Flujo funcionamiento código 10](#_Toc185824185)

[Ilustración 4 Solución optimizada de acuerdo con el paper 11](#_Toc185824186)

[Ilustración 5 Solución optimizada de 100 órdenes 12](#_Toc185824187)

## 3.2 Tablas

[Tabla 1 Resultados uso\_casos.ipynb caso 21 clientes 11](#_Toc185824188)

[Tabla 2 Comparativa Resultados Fitness Caso 1 12](#_Toc185824189)

[Tabla 3 Resultados uso\_casos.ipynb caso 100 clientes 13](#_Toc185824190)

[Tabla 4 Resultados 10 Clientes 14](#_Toc185824191)

[Tabla 5 Resultados 20 Clientes 15](#_Toc185824192)

[Tabla 6 Resultados 30 Clientes 15](#_Toc185824193)

[Tabla 7 Resultados 40 Clientes 16](#_Toc185824194)

[Tabla 8 Resultados 50 Clientes 16](#_Toc185824195)

[Tabla 9 Resultados Fitness: 1 vehículo por tipo, 5 máximo por orden 17](#_Toc185824196)

[Tabla 10 Resultados Fitness: 2 vehículos por tipo, 10 máximo por orden 17](#_Toc185824197)

[Tabla 11 Resultados Fitness: 3 vehículos por tipo, 15 máximo por orden 18](#_Toc185824198)

[Tabla 12 Resultados Fitness: 4 vehículos por tipo, 20 máximo por orden 18](#_Toc185824199)

[Tabla 13 Resultados Fitness: 5 vehículos por tipo, 25 máximo por orden 19](#_Toc185824200)

[Tabla 14 Evolución Fitness vs Máximo por tipo 19](#_Toc185824201)

# Conceptos Clave

\*VRP: Vehicle routing problem (Problema de Ruteo de Vehículos).

\*Heurísticas: Atajos que otorgan decisiones más rápidas que métodos tradicionales de computación.

\*Pseudocódigo: Descripción de un algoritmo o programa escrito en lenguaje natural estructurado.

\*Fitness: Características que miden la calidad del código.

# Introducción

Este informe presenta el análisis, optimización y documentación del código implementado para resolver el **Vehicle Routing Problem (VRP)**, tomando como base el artículo [*"The Finished Vehicle Routing Problem with a Heterogeneous Transport Fleet"*](https://www.researchsquare.com/article/rs-2942819/v1). Este artículo sirvió como referencia principal para la realización de este trabajo y la presente tesis.

El artículo original aborda una variante del VRP enfocada en la planificación de rutas óptimas para la distribución de productos mediante una flota heterogénea de vehículos. Cada vehículo parte de un centro de distribución hacia varios clientes, considerando restricciones como costos, características de los vehículos, camiones y ventanas de tiempo. Aunque el artículo proporciona un esquema en pseudocódigo, este no detalla completamente la implementación de ciertos aspectos del modelo ni especifica cómo se valida el fitness esperado.

Sabiendo eso, se realizó un análisis exhaustivo del artículo y se diseñó un código funcional desde cero, incorporando mejoras y ajustes necesarios para adaptar el modelo a un escenario práctico. Durante este proceso, se logró replicar las rutas optimizadas siguiendo el esquema presentado en el artículo, aunque con algunas diferencias en los valores de Fitness obtenidos.

El objetivo principal es desarrollar un algoritmo para el HVRP-FVL que optimice las rutas de una flota heterogénea de vehículos, tomando en consideración diversas restricciones operativas y logísticas:

* **Capacidad máxima de carga**: Cada vehículo tiene un límite de carga, tanto en términos de peso como de dimensiones, que no debe ser excedido.
* **Ventanas de tiempo**: Cada cliente debe ser atendido dentro de un intervalo de tiempo específico.
* **Cobertura de clientes**: Cada cliente debe ser visitado exactamente una vez con su entrega completa, y cada vehículo debe regresar al centro de distribución después de completar sus entregas.

A través de este trabajo, se busca comprobar la eficacia del modelo propuesto en el artículo y su aplicabilidad en contextos logísticos reales.

# Marco teórico

## 6.1 Evolución Histórica de la formulación de VRP

[1] Los problemas tipo VRP ha recibido varias variaciones desde la primera vez que fue planteado en 1956 por Flood, pero siempre se sigue con la base de salir de un punto hacia los distintos clientes, siendo estos visitados una única vez, para luego volver al inicio.

A lo largo de los años siguientes nacieron las primeras variaciones al VRP, como se muestra en la siguiente figura:

Diagrama

Descripción generada automáticamente

Ilustración 1 Modelos originarios del problema VRP

Cada una de las variaciones presentó modificaciones específicas, como la evaluación de un número determinado de rutas, el análisis del costo mínimo requerido o la incorporación de ligeros ajustes entre ellas.

# Metodología

El desarrollo del presente trabajo se realizó utilizando el lenguaje de programación **Python 3.10.0** en el entorno de desarrollo **Visual Studio Code**.

**Herramientas y entorno de desarrollo**

* **Lenguaje de programación**: Python fue elegido por su flexibilidad y su amplia disponibilidad de bibliotecas para procesamiento de datos, optimización y simulación.
* **Entorno de desarrollo**: Visual Studio Code, una herramienta versátil y ampliamente utilizada en el desarrollo de proyectos de software.
* **Gestión de datos**: Los datos necesarios para probar el problema fueron provistos en el paper, como las características de los vehículos, los pedidos y las distancias entre nodos. Posteriormente se extrajeron datos de ubicaciones reales, las cuales se almacenaron en archivos de texto, permitiendo una lectura eh interpretación fácil.

**Proceso de desarrollo**

1. **Análisis preliminar**:
   * Se revisaron las inconsistencias y elementos faltantes en el artículo original. Este análisis reveló vacíos conceptuales y valores sin declarar, lo que requirió realizar algunos ajustes y reelaboraciones durante el desarrollo del código.
   * Con la información provista por el paper, se obtuvo la información necesaria para probar de manera preliminar el funcionamiento de la generación de N1 y N2, además de los datos a utilizar para los experimentos posteriores.
2. **Extracción de datos:**
   * Primeramente, se extrajeron las coordenadas de 1355 ubicaciones relacionadas al rubro automotriz a través del uso de overpass-turbo.eu, que se guardaron en un archivo kml.
3. **Generación de Ordenes y rutas:**
   * Utilizando estos datos, se generan archivos txt con las coordenadas ordenadas de manera aleatoria, junto con los posibles valores a utilizar en las órdenes.
4. **Diseño del modelo y lectura de datos**:
   * Se implementó un sistema para la lectura de datos desde archivos de texto, permitiendo así estructurar y validar la información básica del problema (capacidad de los vehículos, restricciones de carga, ventanas de tiempo, entre otros).
   * Se crean las variables N1 y N2. Se mejora N1 si es posible, luego se obtiene un N2 correspondiente al N1 mejorado.
5. **Implementación de las funciones de fitness**:
   * Las funciones de fitness fueron diseñadas siguiendo los criterios planteados en el paper de referencia.
6. **Uso de funciones del algoritmo genético propuesto**:
   * El algoritmo genético implementado incluyó las siguientes etapas detalladas en el paper, con unos cambios para evitar la pérdida de candidatos y fomentar la evolución:
     + Cálculo del fitness inicial junto con las probabilidades de cruzamiento y mutación.
     + **Selección**: Se seleccionan las soluciones con los peores valores de Fitness como precandidatos de cruzamiento.
     + **Cruzamiento**: Junto con los candidatos preseleccionados, se usa una ruleta simple para seleccionar a los candidatos de cruzamiento, hasta llegar a la mitad de esta. Luego se procede al acto utilizando un cruzamiento PMX, el cual evitara que se creen candidatos con valores repetidos
     + **Mutación**: Se seleccionan candidatos mediante la ruleta hasta llegar a la mitad de la población nuevamente y se realiza la operación.
     + **Local Search Operator**: Operación que itera sobre toda la población, se realiza sobre cada candidato intercambiando una posición de N1 y luego comparando ambos, manteniendo solamente a los que tengan un Fitness menor.
     + **Actualización de la población**: Se integraron las nuevas soluciones generadas, repitiendo el ciclo hasta converger en una solución óptima.
   * Las probabilidades de cruzamiento y mutación se calcularon dinámicamente en cada iteración para garantizar un equilibrio entre exploración y explotación.

**Resultados esperados**

El proceso descrito permitió comprobar 2 cosas:

* El funcionamiento del código en su función principal (cálculo de fitness, generación de poblaciones, etc.)
* El funcionamiento del código en los casos de estudios provistos, los cuales sirvieron como base para probar el mismo.

# Funcionalidades del Software

## 8.1 Variables especiales:

* N1
* N2

## 8.2 Proceso de asignación:

* Input de data: El software recibe la data a utilizar en los algoritmos desde varios archivos de texto (.txt).
* Se crean las variables N1 Y N2 con los datos extraídos, siendo estas listas aleatorias. N1 contendrá los clientes a visitar y su orden la secuencia en las que deben ser visitados. N2 contendrá los tipos de vehículos a usar, y su orden la secuencia de uso.
* Se procesa N1 para obtener una lista que contiene los valores de los nodos de N1 y una cantidad de 0 correspondiente al uso de los vehículos de carga. En caso de presentar un desuso de vehículos de carga (0 de manera continua), se mejora N1 y N2 para eliminar estos valores y obtener los datos reales utilizados.
* Evaluación y análisis: La evaluación de cada variable de vehículo o cliente se evaluará mediante un algoritmo desarrollado en Python para determinar la manera óptima de viaje y entrega de vehículos.
* Criterios de asignación: Los criterios de entrega serán dependientes de la carga máxima de los tipos de vehículos, considerando también las dimensiones de estos.

## 8.3 Salida de datos:

El programa por cada archivo de datos de clientes ocupado crea archivos txt con los valores de cada generación, que corresponden a los tiempos utilizados por cada sección del algoritmo genético, listas que contienen datos correspondientes a N1 Y N2 y un valor de fitness, estos 3 correspondientes al mejor valor obtenido en esa generación.

* **Generación:** Indica la generación a la que pertenecen los valores siguientes.
* **tiempo\_crossover:** Tiempo que le tomo al programa realizar el Crossover de la generación actual.
* **tiempo\_mutation:** Tiempo que le tomo al programa realizar la mutación de la generación actual.
* **tiempo\_lso:** Tiempo que le tomo al programa realizar el Local Search Operator de la generación actual.
* **N1:** Lista con el orden de visita de los clientes del mejor candidato de la generación
* **N2:** Lista con el orden de uso de los vehículos de carga correspondiente a N1
* **Fitness:** Valor del Fitness del candidato mejor evaluado

Con estos archivos se realiza una última operación, guardando siguientes valores para comparar tiempos de ejecución según los datos utilizados para un futuro análisis de coste computacional:

* **n\_clientes**: Número de clientes utilizados
* **maximo\_por\_tipo**: Número máximo de vehículos del tipo K permitido por orden
* **maximo\_por\_cliente**: Número máximo de vehículos que cualquier cliente del archivo actual puede tener en el pedido
* **tiempo\_HVRP-FVL**: Tiempo que tardo el algoritmo genético en ejecutarse.

# Implementación Técnica

Usando librerías de ampliación de funciones matemáticas, tratamiento de texto y copia de datos. Estas serían math, re y copy respectivamente, además de otras para funciones específicas.

* Entrada de datos: Los datos de los clientes, los camiones, autos y parámetros utilizados se extrajeron del paper y fuentes afines, y se introducen en el programa a través de archivos txt estructurados
* Creación variables N1 y N2: Utilizando los datos extraídos, el programa sigue las instrucciones del paper, creando N1 y N2 los cuáles serán las bases para el GA
* Generación rutas optimas: Habiendo hecho las n° iteraciones el programa le entrega al usuario la solución encontrada, el cual consiste en la ruta y el fitness.

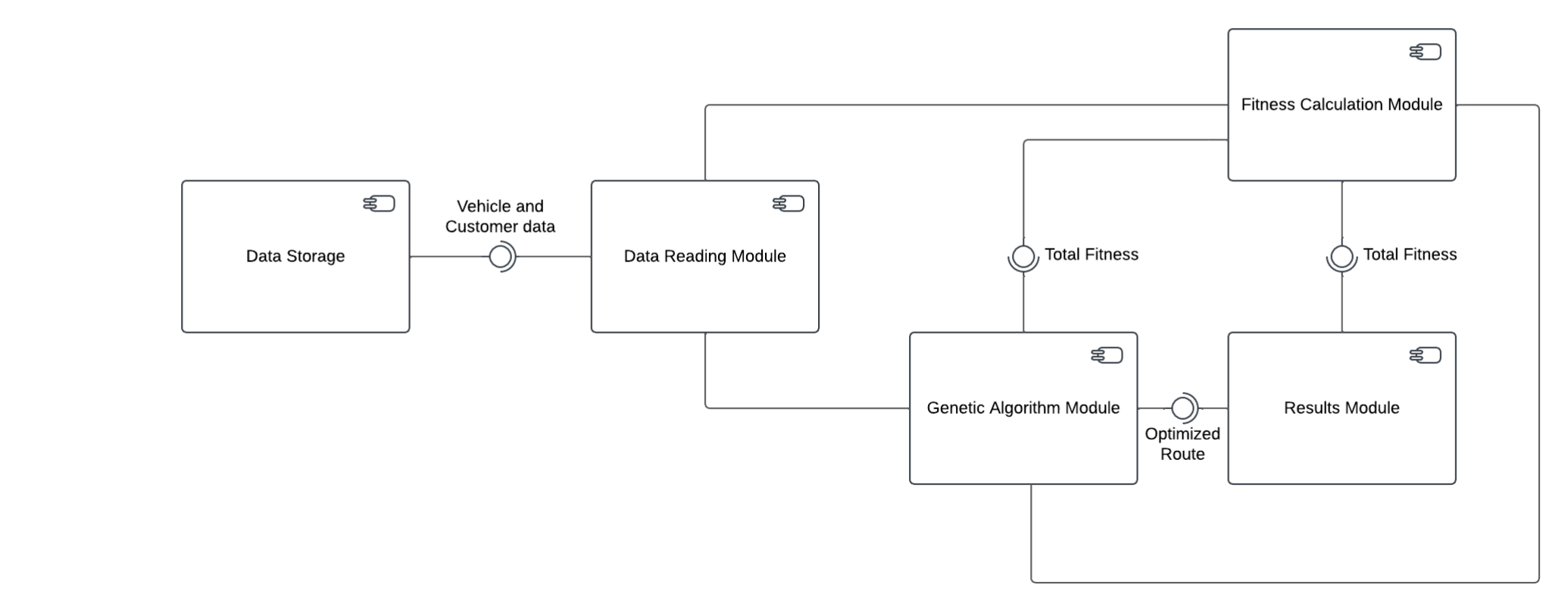


Ilustración Diagrama de componentes

* Data Storage: Este módulo actúa como el repositorio de archivos utilizados por el sistema.
* Data Reading Module: Lee y transforma los archivos almacenados en el módulo de almacenamiento en estructuras de datos listas para su uso.
* Genetic Algorithm Module: Implementa el algoritmo genético para la optimización de rutas.
* Fitness Calculation Module: Evalúa la calidad de cada ruta generada por el algoritmo genético
* Results Module: Maneja el almacenamiento y la presentación de los resultados generados por el sistema.

# Modelo Matemático

Se utiliza el modelo matemático provisto en el paper *“The finished vehicle routing problem with a heterogeneous transport fleet”*:

Liu, J., Yuan, B., & Zan, Y. (2023). *The finished vehicle routing problem with a heterogeneous transport fleet*. Research Square. <https://doi.org/10.21203/rs.3.rs-2942819/v1>

## 10.1 Descripción general:

El problema considera un conjunto de nodos que deben ser visitados por una flota heterogénea de vehículos con distintas dimensiones y cantidad . El objetivo es minimizar el costo total de transporte

## 10.2 Conjuntos y parámetros:

* : Conjunto de vehículos.
* : Conjunto de nodos (clientes), donde es el depósito.
* ​: Capacidad máxima del vehículo.
* : Demanda del cliente (en toneladas).

* ​: Ventana de tiempo para iniciar el servicio en el cliente (inicio ​, fin ​).
* : Tiempo de viaje entre el nodo y el nodo .
* ​: Costo de viaje entre el nodo y el nodo .
* ​: Distancia entre el nodo y el nodo .
* ​: Demanda del cliente de vehículos tipo .
* , , ,: Limitaciones de carga, largo, ancho y alto del vehículo de tipo .
* : Constante especifica por vehículo.

## 10.3 Variable de decisión:

* : Variable binaria que toma valor 1 si el vehículo del tipo se encuentra en el arco , y 0 en caso contrario
* : Variable binaria que toma valor 1 si el vehículo del tipo visitará al cliente , 0 en caso contrario

## 10.4 Función Objetivo:

La función objetivo es minimizar el coste de toda la operación

Donde son los costos de transporte, las penalizaciones de tiempo y los costes por emisiones de carbono.

## 10.5 Restricciones:

* Asignación única: Los clientes deben ser visitados por un único vehículo, sin separar los envíos
* Conservación del movimiento: para cada cliente , el número de vehículos que entran debe ser igual al salir
* Restricción de capacidad: El peso total al unir clientes a una ruta no debe superar el peso permitido por el vehículo
* Restricción de espacio: El largo total al unir clientes a una ruta no debe superar el peso permitido por el vehículo

# Código Fuente

El código desarrollado para este trabajo está disponible en el siguiente repositorio de GitHub:

[https://github.com/moyaxon/tesis\_entrega](https://github.com/Moyaxon/tesis_entrega)

En este repositorio, se pueden encontrar:

* La implementación completa del algoritmo genético.
* Las funciones de fitness utilizadas para evaluar las rutas.
* Los datos de prueba y los resultados obtenidos en los casos de estudio.

A grandes rasgos el funcionamiento se puede resumir en el siguiente diagrama de flujo:

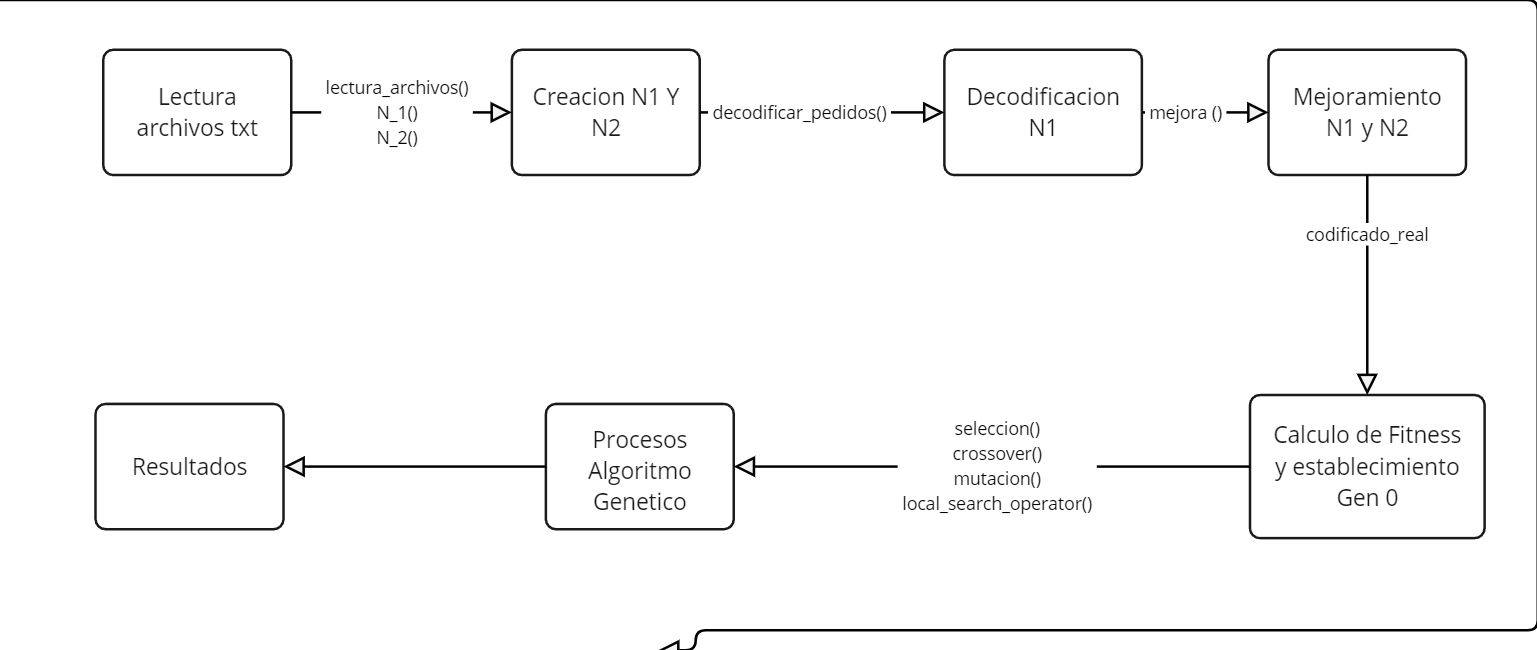


Ilustración Diagrama de Flujo funcionamiento código

# Resultados

## 12.1 Casos de estudio

Dentro del paper utilizado para este trabajo se encuentran 2 casos de prueba con sus rutas obtenidas, y además en el primer caso se entrega el fitness resultante. Por lo tanto, se analizan los resultados de Fitness y las rutas obtenidas del caso 1, y para el caso 2 solo se analizan las rutas resultantes:

Tabla

Descripción generada automáticamente

Ilustración Solución optimizada de acuerdo con el paper

|  |  |  |
| --- | --- | --- |
| **Ruta n°** | **Tipo Vehículo** | **Ruta** |
| 1 | 1 | [20, 21] |
| 2 | 3 | [19, 18, 15] |
| 3 | 2 | [14, 17] |
| 4 | 3 | [13, 12, 11] |
| 5 | 2 | [8, 10] |
| 6 | 2 | [16, 1] |
| 7 | 1 | [4, 2] |
| 8 | 3 | [3, 6, 9] |
| 9 | 2 | [7, 5] |

Tabla Resultados uso\_casos.ipynb caso 21 clientes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Paper | 35078 | 95 | 32391 | 67564 |
| Replica desarrollada | 43460 | 374 | 590913 | 634749 |

Tabla Comparativa Resultados Fitness Caso 1

Tabla

Descripción generada automáticamente

Ilustración Solución optimizada de 100 órdenes

|  |  |  |
| --- | --- | --- |
| **Ruta n°** | **Tipo Vehículo** | **Ruta** |
| 1 | 3 | [46, 22, 93] |
| 2 | 2 | [97, 95] |
| 3 | 3 | [1, 42, 63, 40] |
| 4 | 3 | [67, 26, 34] |
| 5 | 2 | [98, 5] |
| 6 | 2 | [32, 9] |
| 7 | 3 | [92, 7, 54] |
| 8 | 3 | [83, 70, 86] |
| 9 | 2 | [35, 81] |
| 10 | 2 | [37, 2] |
| 11 | 3 | [4, 23, 57] |
| 12 | 3 | [27, 52, 21] |
| 13 | 2 | [80, 24] |
| 14 | 3 | [100, 60, 19] |
| 15 | 3 | [50, 94, 18] |
| 16 | 3 | [85, 43, 53] |
| 17 | 3 | [87, 49, 20] |
| 18 | 3 | [55, 76, 71] |
| 19 | 1 | [96, 90] |
| 20 | 3 | [10, 89, 8] |
| 21 | 3 | [99, 72] |
| 22 | 3 | [65, 11, 13] |
| 23 | 3 | [75, 29, 12, 44] |
| 24 | 3 | [82, 73, 91] |
| 25 | 2 | [78, 48] |
| 26 | 2 | [69, 15] |
| 27 | 2 | [17, 84, 88] |
| 28 | 2 | [61, 45, 25] |
| 29 | 2 | [47, 39] |
| 30 | 2 | [64, 58] |
| 31 | 2 | [31, 6] |
| 32 | 2 | [16, 38] |
| 33 | 2 | [62, 36] |
| 34 | 1 | [14] |
| 35 | 2 | [51, 56] |
| 36 | 3 | [28, 33, 68] |
| 37 | 2 | [41, 79] |
| 38 | 2 | [59, 30] |
| 39 | 2 | [66, 3] |
| 40 | 2 | [74, 77] |

Tabla Resultados uso\_casos.ipynb caso 100 clientes

## 12.2 Ordenes Generadas

Utilizando los datos extraídos y el propio algoritmo, se generaron 75 casos distintos, los cuales variaban entre el número de clientes a utilizar (tamaño de N1 base), el n° máximo que se podía pedir por tipo de vehículo terminado (finished\_vehicle) y el n° máximo de vehículos que requiere el cliente.

Con esos datos se generaron los siguientes resultados:

* Primero se presentan las tablas, separadas por el número de clientes máximo, ordenadas por el tamaño máximo del pedido.
* Luego se presentan gráficos Fitness vs número de clientes, los cuales, según el caso especificado abajo del título, ayudan a comparar el impacto de la cantidad de clientes sobre el valor del fitness, además de como la cantidad de vehículos y la cantidad por tipo afectan al mismo.
* Por último, se presenta la evolución de un caso en particular, se mantiene el n° de clientes en 50 y el máximo por pedido en 5, solo va variando el n° máximo de vehículos por tipo.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N° DE CLIENTES** | **MÁXIMO**  **POR TIPO** | **MAXIMO PEDIDO** | **MÁXIMO** | **POS MAX** | **MINÍMO** | **POS MIN** | **PROMEDIO** |
| 10 | 1 | 5 | 1790595 | 366 | 1574258 | 265 | 1702690 |
| 10 | 2 | 5 | 1325321 | 834 | 1116194 | 14 | 1199228 |
| 10 | 3 | 5 | 1506685 | 285 | 1181822 | 585 | 1324501 |
| 10 | 4 | 5 | 1963758 | 401 | 1740316 | 390 | 1834954 |
| 10 | 5 | 5 | 1384036 | 290 | 1211477 | 655 | 1259900 |
| 10 | 2 | 10 | 3627715 | 892 | 3373338 | 586 | 3460463 |
| 10 | 3 | 10 | 1614419 | 700 | 1440215 | 832 | 1514930 |
| 10 | 4 | 10 | 2391317 | 108 | 2302709 | 203 | 2331271 |
| 10 | 5 | 10 | 2475494 | 74 | 2363874 | 98 | 2402084 |
| 10 | 3 | 15 | 2981463 | 156 | 2691281 | 898 | 2799263 |
| 10 | 4 | 15 | 2926156 | 820 | 2894354 | 682 | 2899268 |
| 10 | 5 | 15 | 3820416 | 853 | 3780466 | 59 | 3787245 |
| 10 | 4 | 20 | 1726367 | 685 | 1618486 | 886 | 1633998 |
| 10 | 5 | 20 | 4782599 | 789 | 4558505 | 3 | 4642575 |
| 10 | 5 | 25 | 3749599 | 900 | 3734869 | 101 | 3737031 |

Tabla Resultados 10 Clientes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N° DE CLIENTES** | **MÁXIMO POR TIPO** | **MAXIMO PEDIDO** | **MÁXIMO** | **POS MAX** | **MINÍMO** | **POS MIN** | **PROMEDIO** |
| 20 | 1 | 5 | 3398551 | 507 | 2656395 | 5 | 3122699 |
| 20 | 2 | 5 | 3432232 | 923 | 2931211 | 636 | 3215273 |
| 20 | 3 | 5 | 3977117 | 718 | 3235101 | 804 | 3719394 |
| 20 | 4 | 5 | 4069705 | 799 | 3457434 | 537 | 3836573 |
| 20 | 5 | 5 | 3193771 | 215 | 2602408 | 0 | 2987488 |
| 20 | 2 | 10 | 8394136 | 366 | 7498588 | 843 | 7981282 |
| 20 | 3 | 10 | 4147520 | 837 | 3677372 | 910 | 3922982 |
| 20 | 4 | 10 | 4923059 | 928 | 4578319 | 793 | 4773304 |
| 20 | 5 | 10 | 4927929 | 90 | 4620946 | 367 | 4772459 |
| 20 | 3 | 15 | 5614617 | 279 | 5011257 | 332 | 5375252 |
| 20 | 4 | 15 | 5802282 | 998 | 5319080 | 672 | 5615458 |
| 20 | 5 | 15 | 6999656 | 251 | 6691934 | 567 | 6893371 |
| 20 | 4 | 20 | 4368904 | 899 | 4112059 | 632 | 4237893 |
| 20 | 5 | 20 | 9747721 | 47 | 9124109 | 911 | 9493251 |
| 20 | 5 | 25 | 7519047 | 456 | 7185627 | 80 | 7257634 |

Tabla Resultados 20 Clientes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N° DE CLIENTES** | **MÁXIMO POR TIPO** | **MAXIMO PEDIDO** | **MÁXIMO** | **POS MAX** | **MINÍMO** | **POS MIN** | **PROMEDIO** |
| 30 | 1 | 5 | 4913217 | 197 | 3527376 | 0 | 4566204 |
| 30 | 2 | 5 | 5111151 | 259 | 4102102 | 4 | 4814569 |
| 30 | 3 | 5 | 5729882 | 514 | 4373662 | 0 | 5371212 |
| 30 | 4 | 5 | 6034727 | 568 | 4628907 | 4 | 5741034 |
| 30 | 5 | 5 | 4904661 | 172 | 3644792 | 4 | 4591994 |
| 30 | 2 | 10 | 12996160 | 365 | 11955811 | 856 | 12562105 |
| 30 | 3 | 10 | 6645333 | 694 | 6046255 | 546 | 6385803 |
| 30 | 4 | 10 | 7511144 | 589 | 7050284 | 0 | 7297975 |
| 30 | 5 | 10 | 7903480 | 141 | 7508567 | 436 | 7689256 |
| 30 | 3 | 15 | 7506016 | 461 | 6642423 | 2 | 7240661 |
| 30 | 4 | 15 | 8338078 | 810 | 7683509 | 790 | 8079272 |
| 30 | 5 | 15 | 10399160 | 745 | 9704706 | 913 | 10154953 |
| 30 | 4 | 20 | 6430416 | 196 | 6036889 | 525 | 6269068 |
| 30 | 5 | 20 | 14708009 | 804 | 13888360 | 847 | 14403044 |
| 30 | 5 | 25 | 13011290 | 449 | 12495124 | 810 | 12800085 |

Tabla Resultados 30 Clientes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N° DE CLIENTES** | **MÁXIMO POR TIPO** | **MAXIMO PEDIDO** | **MÁXIMO** | **POS MAX** | **MINÍMO** | **POS MIN** | **PROMEDIO** |
| 40 | 1 | 5 | 6984222 | 537 | 4973926 | 0 | 6567239 |
| 40 | 2 | 5 | 6469173 | 716 | 4573674 | 0 | 6138550 |
| 40 | 3 | 5 | 7510669 | 484 | 5866541 | 6 | 7196441 |
| 40 | 4 | 5 | 8110609 | 996 | 5562658 | 2 | 7638471 |
| 40 | 5 | 5 | 6493843 | 275 | 4960448 | 0 | 6149441 |
| 40 | 2 | 10 | 17526680 | 878 | 15668743 | 1 | 16942837 |
| 40 | 3 | 10 | 9333921 | 22 | 8506723 | 265 | 9035622 |
| 40 | 4 | 10 | 9531246 | 544 | 8829820 | 269 | 9304046 |
| 40 | 5 | 10 | 10311935 | 801 | 9747694 | 314 | 10097488 |
| 40 | 3 | 15 | 10038362 | 529 | 8787776 | 0 | 9714129 |
| 40 | 4 | 15 | 11701666 | 537 | 10938170 | 672 | 11376914 |
| 40 | 5 | 15 | 13835341 | 322 | 13068415 | 893 | 13574598 |
| 40 | 4 | 20 | 8765098 | 437 | 8257957 | 389 | 8565638 |
| 40 | 5 | 20 | 18938011 | 686 | 18158438 | 519 | 18630480 |
| 40 | 5 | 25 | 17129535 | 404 | 16563417 | 63 | 16899817 |

Tabla Resultados 40 Clientes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N° DE CLIENTES** | **MÁXIMO POR TIPO** | **MAXIMO PEDIDO** | **MÁXIMO** | **POS MAX** | **MINÍMO** | **POS MIN** | **PROMEDIO** |
| 50 | 1 | 5 | 9066512 | 476 | 6144570 | 2 | 8520992 |
| 50 | 2 | 5 | 8653931 | 285 | 5834130 | 0 | 8197036 |
| 50 | 3 | 5 | 9995201 | 373 | 7649461 | 8 | 9525949 |
| 50 | 4 | 5 | 10087921 | 828 | 6920285 | 0 | 9565662 |
| 50 | 5 | 5 | 8333658 | 702 | 6233869 | 5 | 7943522 |
| 50 | 2 | 10 | 21555158 | 244 | 19927002 | 10 | 20947410 |
| 50 | 3 | 10 | 12243889 | 855 | 11252327 | 6 | 11907828 |
| 50 | 4 | 10 | 11722565 | 152 | 10899649 | 888 | 11421702 |
| 50 | 5 | 10 | 13732886 | 301 | 12907263 | 2 | 13525151 |
| 50 | 3 | 15 | 11917624 | 852 | 10347353 | 2 | 11523501 |
| 50 | 4 | 15 | 14238745 | 301 | 13412104 | 478 | 13948576 |
| 50 | 5 | 15 | 17186591 | 540 | 16524785 | 63 | 16920851 |
| 50 | 4 | 20 | 11563224 | 320 | 10980915 | 858 | 11337824 |
| 50 | 5 | 20 | 24735899 | 807 | 23685982 | 192 | 24362351 |
| 50 | 5 | 25 | 21276489 | 584 | 20697539 | 655 | 21025788 |

Tabla Resultados 50 Clientes

Gráfico

Descripción generada automáticamente

Tabla Resultados Fitness: 1 vehículo por tipo, 5 máximo por orden

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Tabla Resultados Fitness: 2 vehículos por tipo, 10 máximo por orden

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Tabla Resultados Fitness: 3 vehículos por tipo, 15 máximo por orden

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Tabla Resultados Fitness: 4 vehículos por tipo, 20 máximo por orden

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Tabla Resultados Fitness: 5 vehículos por tipo, 25 máximo por orden

Gráfico, Gráfico de barras

Descripción generada automáticamente

Tabla Evolución Fitness vs Máximo por tipo

# Conclusiones

Comparando los resultados obtenidos con los del artículo base, se evidencia que el algoritmo propuesto es capaz de replicar el funcionamiento base paper original, Pero con los cambios propuestos por nosotros, se llegó a realizar una nueva serie de pruebas y serie de pasos a realizar, con el fin de resolver el problema de un **HVRP–FVL**

1. **Resultados alcanzados**:
   * Se logró implementar un algoritmo genético desde cero, respetando las funciones de fitness descritas en el paper base. En cambio, se realizaron otras consideraciones eh incluso se llevó a un nuevo paso el cual no estaba documentado en el paper, estas serían, por ejemplo, mantener los valores durante el Crossover, así evitando la perdida población, y más importante aún sería el mejoramiento en si de los candidatos al generar N1 y N2. Paso que ayuda en gran medida al funcionamiento y facilitación del **HVRP–FVL.**
   * El algoritmo entregó soluciones comparables al artículo base en términos de las rutas a usar por cada camión, mostrando que se logró el resultado esperado en los casos de estudio.
2. **Limitaciones**:
   * Aunque el algoritmo genético ofreció buenos resultados, el tiempo de ejecución podría ser mejorado mediante la implementación de optimizaciones adicionales, ya que los datos no clarificados por parte de la fuente pueden cambiar drásticamente el resultado de los ejemplos y afectar en gran medida los valores obtenidos en las pruebas.
3. **Propuestas futuras**:
   * Explorar métodos híbridos que combinen metaheurísticas y enfoques exactos para balancear mejor la calidad de las soluciones y el tiempo de ejecución.

En resumen, este trabajo demostró la viabilidad de aplicar algoritmos genéticos para resolver problemas complejos como el **HVRP–FVL**, y sienta las bases para futuras investigaciones en optimización logística utilizando métodos computacionales avanzados.

# Referencias

* [1] Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. Ingeniería, 16(2),35-55.[fecha de Consulta 9 de Diciembre de 2024]. ISSN: 0121-750X. Recuperado de: https://www.redalyc.org/articulo.oa?id=498850173004
* Liu, Jiansheng Liu and Yuan, Bin and Hu, Yingcong and Smith, Alice E., The Finished Vehicle Routing Problem with a Heterogeneous Transport Fleet. Available at SSRN: <https://ssrn.com/abstract=4239703> or [http://dx.doi.org/10.2139/ssrn.4239703](https://dx.doi.org/10.2139/ssrn.4239703)
* Tolga Bektaş, Gilbert Laporte, The Pollution-Routing Problem, Transportation Research Part B: Methodological, Volume 45, Issue 8, 2011, Pages 1232-1250, ISSN 0191-2615, <https://doi.org/10.1016/j.trb.2011.02.004>. (<https://www.sciencedirect.com/science/article/pii/S019126151100018X>)
* Emrah Demir, Tolga Bektaş, Gilbert Laporte, A comparative analysis of several vehicle emission models for road freight transportation, Transportation Research Part D: Transport and Environment, Volume 16, Issue 5, 2011, Pages 347-357, ISSN 1361-9209, <https://doi.org/10.1016/j.trd.2011.01.011>. (<https://www.sciencedirect.com/science/article/pii/S136192091100023X>)
* OpenStreetMap contributors. (n.d.). OpenStreetMap [Mapa interactivo]. Recuperado de <https://www.openstreetmap.org/#map=4/37.34/92.24>
* OpenStreetMap contributors. (n.d.). Overpass API. En OpenStreetMap Wiki. Recuperado de <https://wiki.openstreetmap.org/wiki/Overpass_API>
* OpenStreetMap contributors. (n.d.). Overpass turbo/Wizard. En OpenStreetMap Wiki. Recuperado de <https://wiki.openstreetmap.org/wiki/Overpass_turbo/Wizard>
* Overpass Turbo. (n.d.). Overpass Turbo [Herramienta interactiva]. Recuperado de <https://overpass-turbo.eu>
* <https://observablehq.com/@swissmanu/pmx-crossover>