



**Universidad
Andrés Bello®**
FACULTAD DE INGENIERÍA
INGENIERIA CIVIL INFORMÁTICA

**ANÁLISIS E IMPLEMENTACIÓN DEL “HETEROGENEOUS MULTI-TYPE FLEET
VEHICLE ROUTING PROBLEM IN FINISHED VEHICLE LOGISTICS”**

Tesis de pregrado para optar al título de Ingeniero Civil Informático

Bastían 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 y Yang Zan(Liu et al., n.d., 2023). El contenido original ha sido utilizado con permiso y según los términos especificados por el autor/propietario original.

Agradecimientos

Felipe Moya:

Quiero agradecer mi familia que nos apoyó en todo, especialmente a mi hermano por no rendirse pese a todas las trabas que nos encontramos tratando de ver este problema.

A mi profesor Gustavo Gatica, que tuvo la paciencia de escucharnos y ver junto a nosotros las distintas alternativas y acercamientos que podríamos tratar de usar. Este trabajo no hubiera funcionado sin su carisma y apoyo.

Y, por último, me gustaría dejar un mensaje a todo el que lea este documento, aunque se vea difícil, nada es imposible. Puede que te pares en seco algunas veces, pero un cambio de perspectiva puede ser todo lo que te haga falta para llegar al resultado que esperas.

De verdad me alegrar terminar todo este trayecto con un trabajo así, me hizo hacer mil cosas y me sentí decaído muchas veces, pero por lo mismo haber hecho esto se sintió demasiado bien.

Solo una cosa, si les piden información para trabajos similares, sean buenos y pásenla, que hagan un trabajo basado en uno tuyo lo encuentro una de las mejores maneras de admiración.

Bastián Moya:

Quiero agradecer a mi familia por todo el apoyo y ganas de trabajar que nos han entregado, y a mi hermano especialmente por escuchar y ayudar cada vez que no entendía algo mientras trabajamos en este problema.

A mi profesor Dr. Gustavo Gatica, por la ayuda, guía y dedicación para que hiciéramos el mejor trabajo posible, como también por la paciencia que tuvo con nosotros cuando no lográbamos una buena explicación, sin su ayuda nada de esto hubiera sido posible.

El finalmente terminar con este trabajo me llena de felicidad, tras tantos momentos de problemas nuevos y repetidos, se siente bien el poder dejarlo atrás con lo mejor que pudimos hacer, aunque también me deje con incertidumbre por lo que vendrá a futuro, pero sé que podre superarlo como hice con este trabajo.

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*" o HVRP–FVL. (Liu et al., 2023). 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 las 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 la funcionalidad propuesta. 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
Resumen	iv
Tabla de contenidos	v
Ilustraciones	vi
Tablas.....	vi
Gráficos.....	vi
Conceptos Clave	vii
1. Introducción	1
2. Marco teórico.....	2
2.1 ¿Que es un VRP?	2
2.2 Evolución Histórica de la formulación de VRP	2
2.2.1 VRP (TSP Generalizado)	3
2.2.2 VRP Homogéneo	3
2.2.3 VRP Heterogéneo.....	3
3. Metodología.....	4
4. Definición de los parámetros y variables.....	6
4.1 Variables especiales:.....	6
4.2 Proceso de asignación:	6
4.3 Pseudocódigo del proceso:	6
4.4 Salida de datos:.....	7
5. Implementación Técnica	8
6. Modelo Matemático	9
6.1 Descripción general:.....	9
6.2 Conjuntos y parámetros:	9
6.3 Variable de decisión:	9
6.4 Función Objetivo:	9
6.5 Restricciones:.....	10
7. Código Fuente	11
7.1 Pseudocódigo de la función principal hvrp_fvl:	12
8. Resultados.....	14
8.1 Casos de estudio.....	14
8.2 Órdenes Generadas	18

8.2.1 Tablas y gráficos Fitness	18
8.2.2 Tabla y gráfico de los tiempos de ejecución.....	24
9. Conclusiones	26
10. Referencias	28

Ilustraciones

Ilustración 1 Modelos originarios del problema VRP	2
Ilustración 2 Diagrama de componentes	8
Ilustración 3 Diagrama de Flujo funcionamiento código	11

Tablas

Tabla 1 The optimized solutions	14
Tabla 2 Resultados uso_casos.ipynb caso 21 clientes.....	14
Tabla 3 Comparativa Resultados Fitness Caso 1	15
Tabla 4 Supplementary Material Solutions	16
Tabla 5 Resultados uso_casos.ipynb caso 100 clientes.....	17
Tabla 6 Resultados 10 Clientes.....	18
Tabla 7 Resultados 20 Clientes.....	19
Tabla 8 Resultados 30 Clientes.....	19
Tabla 9 Resultados 40 Clientes.....	20
Tabla 10 Resultados 50 Clientes.....	20
Tabla 11 Duración y aumento de tiempo por experimento en segundos.....	25

Gráficos

Gráfico 1 Resultados Fitness: 1 vehículo por tipo, 5 máximo por orden	21
Gráfico 2 Resultados Fitness: 2 vehículos por tipo, 10 máximo por orden	21
Gráfico 3 Resultados Fitness: 3 vehículos por tipo, 15 máximo por orden	22
Gráfico 4 Resultados Fitness: 4 vehículos por tipo, 20 máximo por orden	22
Gráfico 5 Resultados Fitness: 5 vehículos por tipo, 25 máximo por orden	23
Gráfico 6 Evolución Fitness vs Máximo por tipo.....	23
Gráfico 7 Duración en horas de cada experimento	26

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.

1. Introducción

El enrutamiento de vehículos (*Vehicle Routing Problem* o VRP) es un problema clásico en la investigación operativa y la logística, centrado en diseñar rutas eficientes para una flota de vehículos que parte desde un depósito y atiende a múltiples clientes. Este problema asume típicamente costos basados únicamente en la distancia recorrida, pero en la práctica, factores como tarifas de peaje, costos laborales, costos de combustible, penalizaciones por tiempo y emisiones de carbono influyen significativamente en el costo total de transporte.

Dentro de este contexto, surge una variante más compleja conocida como el problema de enrutamiento de vehículos heterogéneos para logística de productos terminados (*Heterogeneous Vehicle Routing Problem for Finished Vehicle Logistics*, HVRP-FVL). Este problema considera una flota heterogénea de vehículos con distintas capacidades y costos operativos, así como restricciones adicionales relacionadas con las condiciones de carga y las tarifas variables según las zonas geográficas. Estas características lo convierten en un modelo más representativo de las necesidades actuales en logística.

La gestión eficiente de rutas de transporte es crucial en la logística moderna, especialmente en la industria automotriz y de bienes terminados. La creciente complejidad de los costos asociados y la necesidad de minimizar el impacto ambiental hacen indispensable desarrollar soluciones avanzadas para optimizar el uso de los recursos disponibles. En particular, el HVRP-FVL aborda problemas reales que los enfoques tradicionales del VRP no pueden resolver de manera efectiva, como la asignación de vehículos con características específicas a rutas complejas.

Puntos Clave:

- Existe un único punto de partida y retorno de donde salen los camiones.
- La cantidad por tipo de camión es infinita, lo que permite atender cada grupo de ordenes sin problema.
- Cada tipo de camión y vehículo terminado cuenta con dimensiones y pesos distintos.
- Se conoce la demanda de cada orden.
- Se visita cada cliente únicamente con su orden completa.
- Se calcula dinámicamente el fitness de individuo durante los procesos de mutación y cruzamiento.

La implementación práctica del HVRP-FVL permite abordar problemas logísticos complejos que requieren soluciones eficientes y adaptables. Este trabajo busca no solo validar un modelo teórico, sino también demostrar su aplicabilidad en escenarios reales, proporcionando beneficios tanto para investigadores como para profesionales en el campo de la logística.

2. Marco teórico

En esta sección se hablará sobre que es un VRP(Braekers et al., 2016), su evolución a lo largo de los años, cambios y variaciones recibidos a lo largo de los años, la complejidad y la importancia del problema en la logística.

2.1 ¿Que es un VRP?

Es una generalización del Traveling Salesman Problem(Laporte, 1992), un problema de optimización combinatoria que hace la pregunta de “¿Cuál es el conjunto optimo de rutas para una flota de vehículos para satisfacer las demandas de un conjunto dado de clientes?”.

El objetivo de este tipo de problemas es conseguir el mínimo coste total de ruta, y al ser un tipo de problema NP-Hard, el tamaño de los problemas que puede resolver de manera óptima puede llegar a ser limitadas.

El uso de este tipo de programas puede llegar a ahorrar compañías de 5% a 30% en costos.(Hasle et al., 2007)

2.2 Evolución Histórica de la formulación de VRP

Los problemas tipo VRP han recibido varias variaciones desde que fue planteado 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 a continuación.

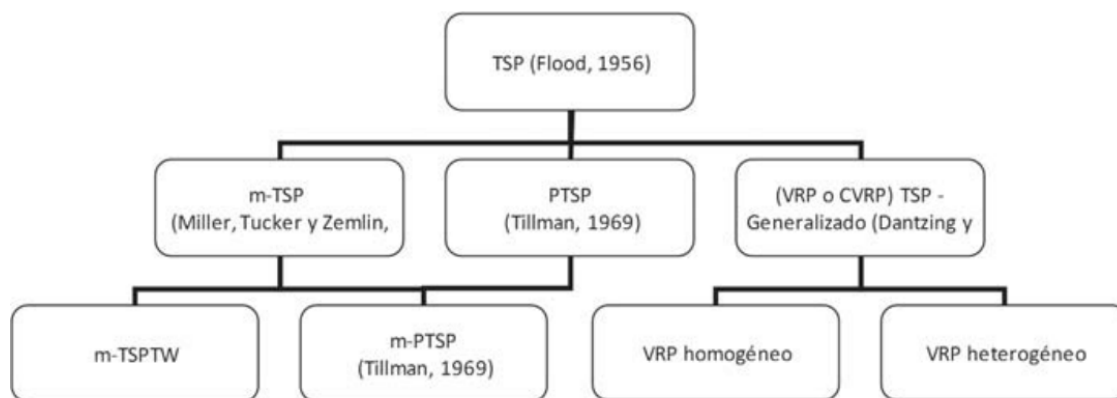


Ilustración 1 Modelos originarios del problema VRP

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.

2.2.1 VRP (TSP Generalizado)

Al ser una generalización, el principal cambio que se recibió de un TSP normal fue que paso de un solo vehículo a ser m cantidad de vehículos saliendo de un punto hacia todos los clientes, añadiendo así una capa extra de complejidad a un TSP.

Un VRP encuentra la ruta optima con menor costo total para los transportes de todos los vehículos de carga disponible, para que solo sean visitados una única vez cada clientes.

De la generalización se desprenden dos categorías, el VRP Homogéneo y el VRP Heterogéneo, el primero siendo manejando de igual manera todos los recursos en sus nodos, como distancia o tiempos, y el segundo teniendo todos los componentes de distinta manera.

2.2.2 VRP Homogéneo

Los modelos del VRP Homogéneo pueden dividirse nuevamente en cuatro subtipos:

- **DVRP:** Distance VRP.
- **VRPTW:** VRP with time windows.
- **VRPB:** VRP with backhauls.
- **SDVRP:** Split delivery VRP.

2.2.3 VRP Heterogéneo

Al igual que el VRP Homogéneo, el VRP Heterogéneo también fue modificado en diferentes subtipos:

- **VRPHF:** VRP with heterogeneous fleet.
- **PVRP:** Periodic VRP.
- **Multi-Trip VRP.**
- **Multi-Depot VRP.**
- **MCVRP:** Multi-commodity VRP.
- **MOVRP:** Multi-objective VRP.
- **SVRP:** Stochastic VRP.

3. Metodología

El desarrollo del presente trabajo se realizó utilizando el lenguaje de programación **Python 3.10.0**(*Python Release Python 3.10.0* | *Python.Org*, n.d.) 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, lo que permite una lectura e interpretación fácil.

Proceso de desarrollo

1. Análisis preliminar:

- Se revisó 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.(Archetti et al., 2014; Rodríguez-Vásquez, 2020; *Vehicle Routing Problem - Wikipedia*, n.d.)
- 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.(Bektaş & Laporte, 2011; Demir et al., 2011)

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. (*OpenStreetMap*, n.d.; *Overpass API Wiki*, n.d.; *OverPass Turbo*, n.d.)

3. Generación de Ordenes y rutas:

- Con los datos utilizados, 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 (*PMX Crossover / Manuel Alabor | Observable*, n.d.)
 - **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.

4. Definición de los parámetros y variables

4.1 Variables especiales:

- N1
- N2

4.2 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 realizará 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.

4.3 Pseudocódigo del proceso:

Leer datos entregados desde txt;
Inicio procedimiento;
Creación de función N_1 ;
 N_1 contiene clientes y orden de visita;
Creación función N_2 ;
 N_2 contiene tipos de vehiculos a ser usados;
Procesamiento de N_1 obteniendo una lista de valores de nodos
junto con 0 correspondientes al uso de vehículos de carga;
De haber desuso se mejorarán las funciones para eliminar los
valores y obtener la lista real que se utilizaron;
Evaluación de variables de vehículos o clientes a traves de un
algoritmo desarrollado en Python para determinar el viaje y
entrega óptimos;
Los criterios de asignación serán dependientes de la carga
máxima de los tipos de vehículo considerando también sus dimensiones;
Fin procedimiento;

4.4 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_Iso:** 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 los 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.

5. Implementación Técnica

Se usaron 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.

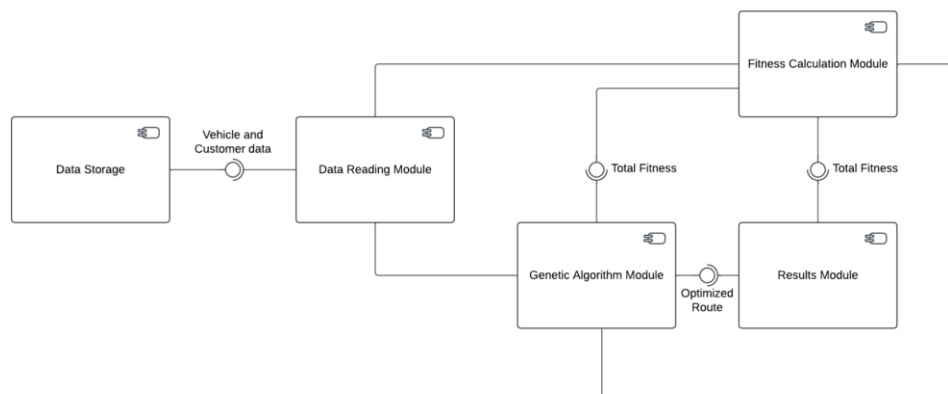


Ilustración 2 Diagrama de componentes

- 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.

6. Modelo Matemático

Se utiliza el modelo matemático provisto en **The finished vehicle routing problem with a heterogeneous transport fleet** (Liu, Yuan, & Zan, 2023).

6.1 Descripción general:

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

6.2 Conjuntos y parámetros:

- V : Conjunto de vehículos.
- N : Conjunto de nodos (clientes), donde 0 es el depósito.
- Q_v : Capacidad máxima del vehículo v .
- w_p : Demanda del cliente i (en toneladas).
- e_i, l_i : Ventana de tiempo para iniciar el servicio en el cliente i (inicio e_i , fin l_i).
- t_{ij} : Tiempo de viaje entre el nodo i y el nodo j .
- c_{ij} : Costo de viaje entre el nodo i y el nodo j .
- d_{ij} : Distancia entre el nodo i y el nodo j .
- D_{ip} : Demanda del cliente i de vehículos tipo p .
- G_k, L_k, W_k, H_k : Limitaciones de carga, largo, ancho y alto del vehículo de tipo k .
- β : Constante específica por vehículo.
- S_p^k : Disponibilidad del vehículo de transporte k para el vehículo terminado del tipo p

$$S_p^k = \begin{cases} 1, & w_p < W_k \text{ y } h_p < H_k \quad \forall p, \forall k \\ 0, & \text{caso contrario} \end{cases}$$

6.3 Variable de decisión:

- x_{iju}^k : Variable binaria que toma valor 1 si el vehículo u del tipo k se encuentra en el arco (i, j) , y 0 en caso contrario
- y_{iu}^k : Variable binaria que toma valor 1 si el vehículo u del tipo k visitará al cliente i , 0 en caso contrario

6.4 Función Objetivo:

La función objetivo es minimizar el coste (rmb) de toda la operación

$$F = f_1 + f_2 + f_3$$

Donde f_1 son los costos de transporte, f_2 las penalizaciones de tiempo y f_3 los costes por emisiones de carbono.

6.5 Restricciones:

- Conservación del movimiento: para cada cliente i , el número de vehículos que entran debe ser igual al salir

$$\sum_{i=0}^N x_{iju}^k = \sum_{i=0}^N x_{jiu}^k, \forall j \in N, \forall k \in K; \forall u \in U_k$$

- Asignación única: Los clientes deben ser visitados por un único vehículo, sin separar los envíos

$$\sum_{u=1}^{U_k} \sum_{k=1}^K y_{iu}^k = 1, \forall i \in N$$

- Restricción de alto y ancho de carga: Limites de altura y anchura del vehículo de transporte

$$y_{iu}^k (S - 1) = 0, \forall i \in N; \forall k \in K; \forall u \in U_k$$

- Restricción de capacidad de carga: El peso total al unir clientes a una ruta no debe superar el peso permitido por el vehículo

$$\sum_{p=1}^P \sum_{i=1}^N D_{ip} w_p y_{iu}^k \leq G_k, \forall k \in K, \forall u \in U_k$$

- Restricción de largo de carga: El largo total al unir clientes a una ruta no debe superar el peso permitido por el vehículo

$$\sum_{p=1}^P \sum_{i=1}^N D_{ip} l_p y_{iu}^k \leq L_k, \forall k \in K, \forall u \in U_k$$

7. Código Fuente

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

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:

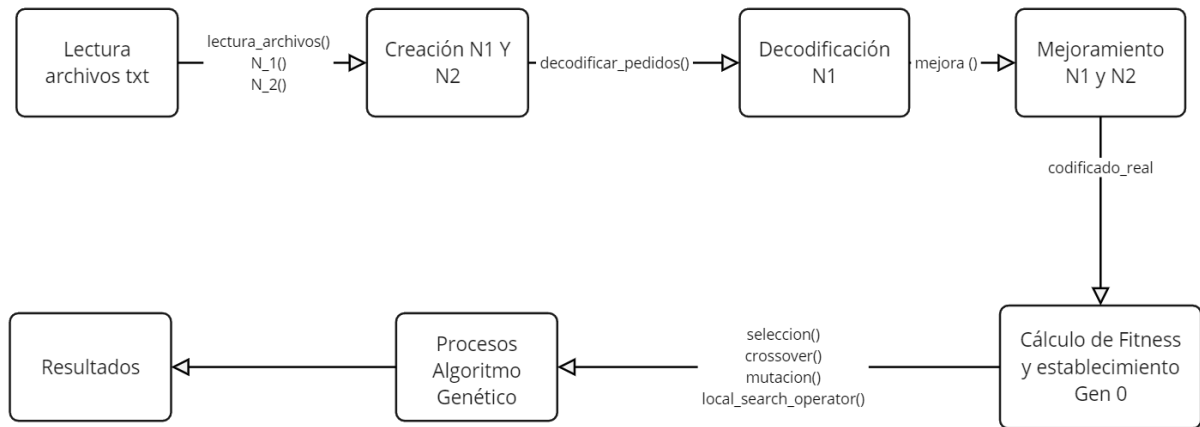


Ilustración 3 Diagrama de Flujo funcionamiento código

7.1 Pseudocódigo de la función principal hvrp_fvl:

Proceso Principal

Paso 1: Leer datos

```
t_v_d, f_v_i, c_o_i_t_w, p_o_m, pop_data <- lectura_archivos(file_1, file_2, file_3, file_4,
file_5, folder_data, n_clientes)
```

Paso 2: Crear N1 y N2 iniciales (Gen 0)

```
poblacion_inicial <- N1(c_o_i_t_w, pop_data)
```

```
vehicle_order_inicial <- N2(t_v_d)
```

```
datos_esenciales <- [poblacion_inicial]
```

Paso 3: Decodificar y mejorar N1 y N2 iniciales

Para cada poblacion En poblacion_inicial Hacer:

```
    poblacion_modificada <- DecodificarPedidos(poblacion, vehicle_order_inicial,
    t_v_d, c_o_i_t_w)
```

```
    poblacion_mejorada, orden_mejorado <- Mejora(poblacion, poblacion_modificada,
    vehicle_order_inicial)
```

```
    Agregar a datos_esenciales: [poblacion_modificada, poblacion_mejorada,
    orden_mejorado]
```

```
FinHacer
```

Paso 4: Calcular Fitness inicial

```
fitness_actual <- CalcularFitness(datos_esenciales, c_o_i_t_w, p_o_m, t_v_d)
```

Paso 5: Algoritmo Genético

```
gen <- 0
```

```
mejores_candidatos_por_generacion <- ActualizarCandidatos(datos_esenciales,
fitness_actual)
```

```

Mientras gen < pop_data[3] Hacer
    Escribir "Generación ", gen + 1
    # Selección
    selected <- Seleccion(datos_esenciales, pop_data[0])
    # Crossover
    fitness_generacion, datos_esenciales <- Crossover(vehicle_order_inicial, selected,
    fitness_actual, datos_esenciales, pop_data, c_o_i_t_w, t_v_d, p_o_m)
    # Mutación
    fitness_generacion, datos_esenciales <- Mutacion(fitness_generacion,
    datos_esenciales, pop_data, t_v_d, c_o_i_t_w, vehicle_order_inicial, p_o_m)
    # Local Search Operator
    datos_esenciales, fitness_generacion <- LocalSearchOperator(fitness_generacion,
    datos_esenciales, pop_data, t_v_d, c_o_i_t_w, vehicle_order_inicial, p_o_m)
    # Actualizar mejores candidatos
    mejores_candidatos_por_generacion <- ActualizarCandidatos(datos_esenciales,
    fitness_actual)
    gen <- gen + 1
FinMientras
# Paso 6: Guardar resultados
Escribir "Guardando resultados en archivo.....txt"
GuardarResultados(mejores_candidatos_por_generacion, tiempos, file_3,
n_clientes)
FinProceso

```

(Núñez & Cotejo, n.d.), (*PSeInt*, n.d.),

8.Resultados

8.1 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 se analizan las rutas resultantes:

Tabla de soluciones optimizadas originaria del paper, teniendo 21 clientes, muestra el tipo de vehículo utilizado junto a la ruta que tránsito.

Ruta n°	Tipo Vehículo	Ruta
1	1	D0→D20→D21→D0
2	3	D0→D19→D18→D15→D0
3	2	D0→D14→D17→D0
4	3	D0→D13→D12→D11→D0
5	2	D0→D8→D10→D0
6	2	D0→D16→D1→D0
7	1	D0→D4→D2→D0
8	3	D0→D3→D6→D9→D0
9	2	D0→D7→D5→D0

Tabla 1 The optimized solutions

Tabla de soluciones optimizadas de nuestro desarrollo, muestra los mismos resultados.

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 2 Resultados uso_casos.ipynb caso 21 clientes

Tabla comparativa de los cálculos de fitness separados junto con el fitness total.

	f_1	f_2	f_3	F
<i>Paper</i>	35078	95	32391	67564
<i>Replica desarrollada</i>	43460	374	590913	634749

Tabla 3 Comparativa Resultados Fitness Caso 1

Tabla de soluciones optimizadas originaria del paper, teniendo 100 clientes, muestra el tipo de vehículo utilizado junto a la ruta que tránsito.

<i>Ruta n°</i>	<i>Tipo Vehículo</i>	<i>Ruta</i>
1	3	D0→D46→D22→D93→D0
2	2	D0→D97→D95→D0
3	3	D0→D1→D42→D63→D40→D0
4	3	D0→D67→D26→D34→D0
5	2	D0→D98→D5→D0
6	2	D0→D32→D9→D0
7	3	D0→D92→D7→D54→D0
8	3	D0→D83→D70→D86→D0
9	2	D0→D35→D81→D0
10	2	D0→D37→D2→D0
11	3	D0→D4→D23→D57→D0
12	3	D0→D27→D52→D21→D0
13	2	D0→D80→D24→D0
14	3	D0→D100→D60→D19→D0
15	3	D0→D50→D94→D18→D0
16	3	D0→D85→D43→D53→D0
17	3	D0→D87→D49→D20→D0
18	3	D0→D55→D76→D71→D0
19	1	D0→D96→D90→D0
20	3	D0→D10→D89→D8→D0
21	3	D0→D99→D72→D0
22	3	D0→D65→D11→D13→D0
23	3	D0→D75→D29→D12→D44→D0
24	3	D0→D82→D73→D91→D0
25	2	D0→D78→D48→D0
26	2	D0→D69→D15→D0
27	2	D0→D17→D84→D88→D0
28	2	D0→D61→D45→D25→D0

29	2	D0→D47→D39→D0
30	2	D0→D64→D58→D0
31	2	D0→D31→D6→D0
32	2	D0→D16→D38→D0
33	2	D0→D62→D36→D0
34	1	D0→D14→D0
35	2	D0→D51→D56→D0
36	3	D0→D28→D33→D68→D0
37	2	D0→D41→D79→D0
38	2	D0→D59→D30→D0
39	2	D0→D66→D3→D0
40	2	D0→D74→D77→D0

Tabla 4 Supplementary Material Solutions

Tabla de soluciones optimizadas de nuestro desarrollo, muestra los mismos resultados.

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 5 Resultados uso_casos.ipynb caso 100 clientes

8.2 Órdenes 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.

8.2.1 Tablas y gráficos Fitness

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 6 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 7 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 8 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 9 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 10 Resultados 50 Clientes

Fitness según el número de clientes

Máximo 1 Vehículo por tipo / Máximo 5 Vehículos por orden

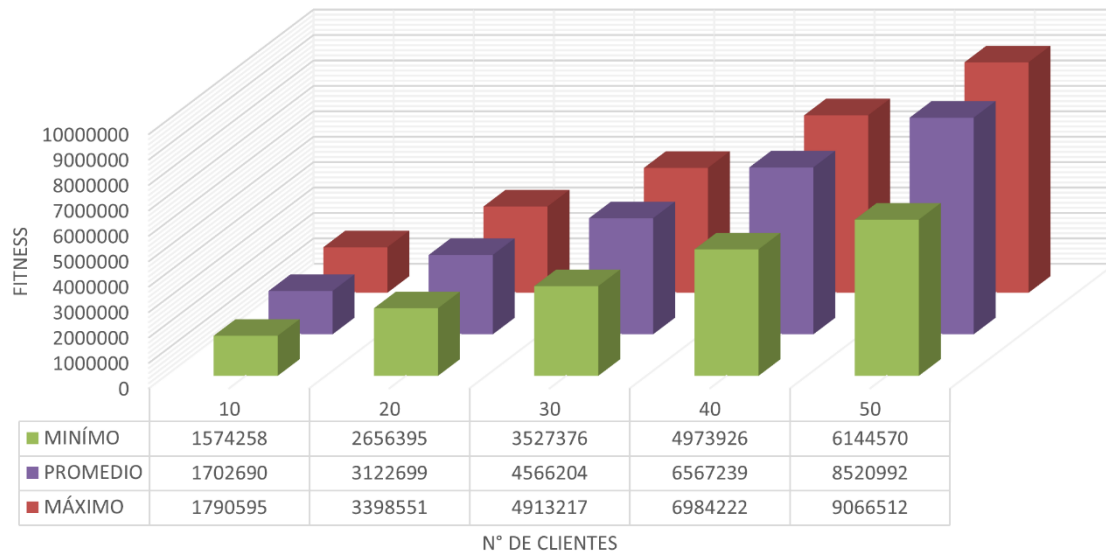


Gráfico 1 Resultados Fitness: 1 vehículo por tipo, 5 máximo por orden

Fitness según el número de clientes

Máximo 2 Vehículos por tipo / Máximo 10 Vehículos por orden

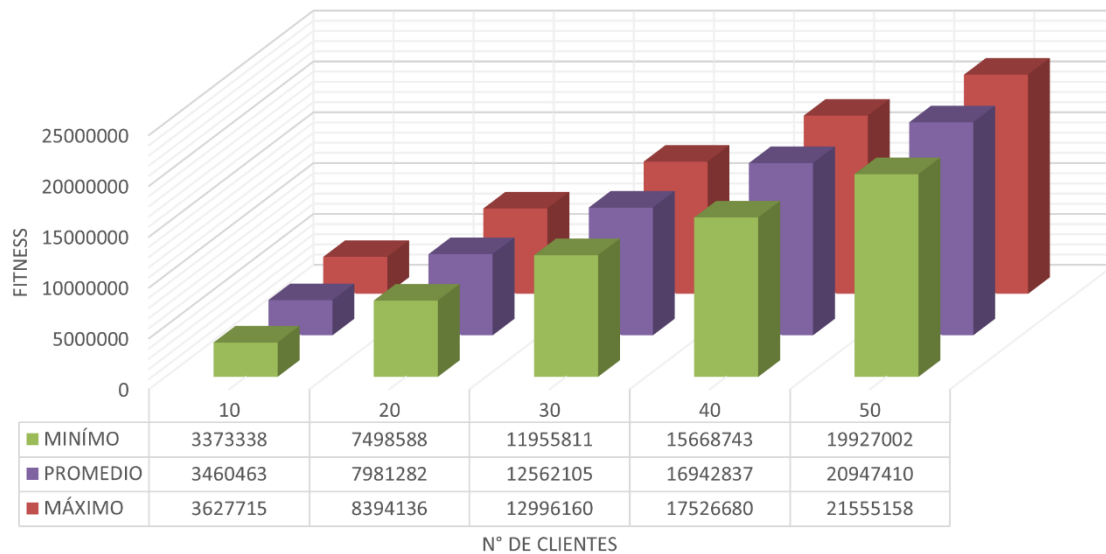


Gráfico 2 Resultados Fitness: 2 vehículos por tipo, 10 máximo por orden

Fitness según el número de clientes

Máximo 3 Vehículos por tipo / Máximo 15 Vehículos por orden

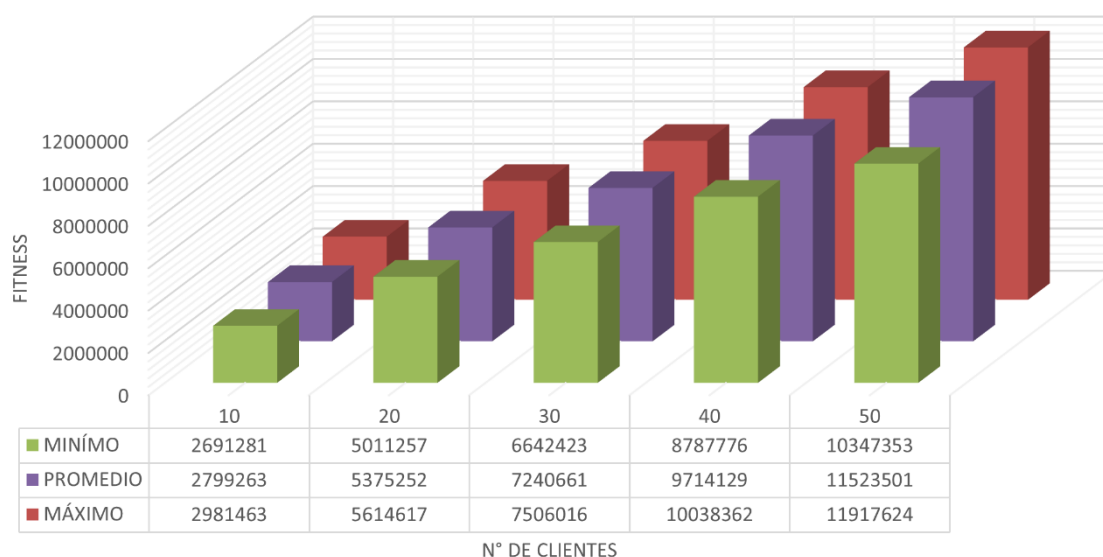


Gráfico 3 Resultados Fitness: 3 vehículos por tipo, 15 máximo por orden

Fitness según el número de clientes

Máximo 4 Vehículos por tipo / Máximo 20 Vehículos por orden

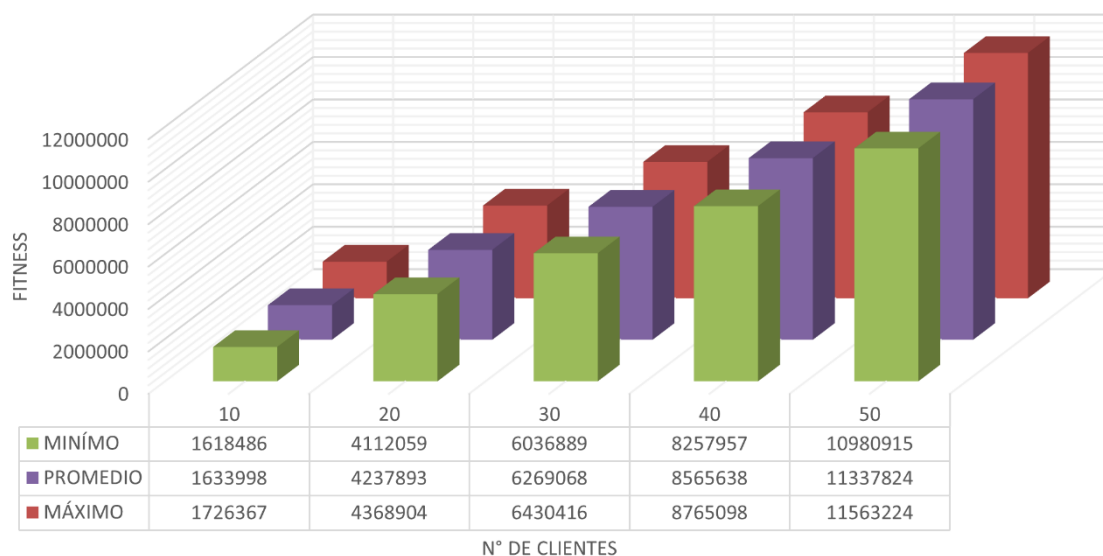


Gráfico 4 Resultados Fitness: 4 vehículos por tipo, 20 máximo por orden

Fitness según el número de clientes

Máximo 5 Vehículos por tipo / Máximo 25 Vehículos por orden

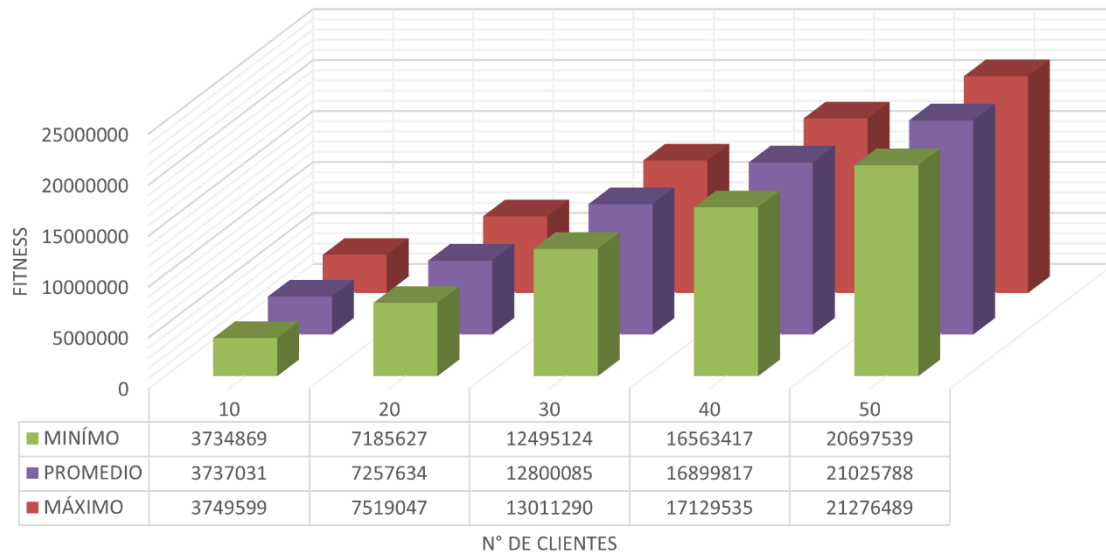


Gráfico 5 Resultados Fitness: 5 vehículos por tipo, 25 máximo por orden

Evolucion Fitness vs Máximo por tipo

5 vehículos por clientes

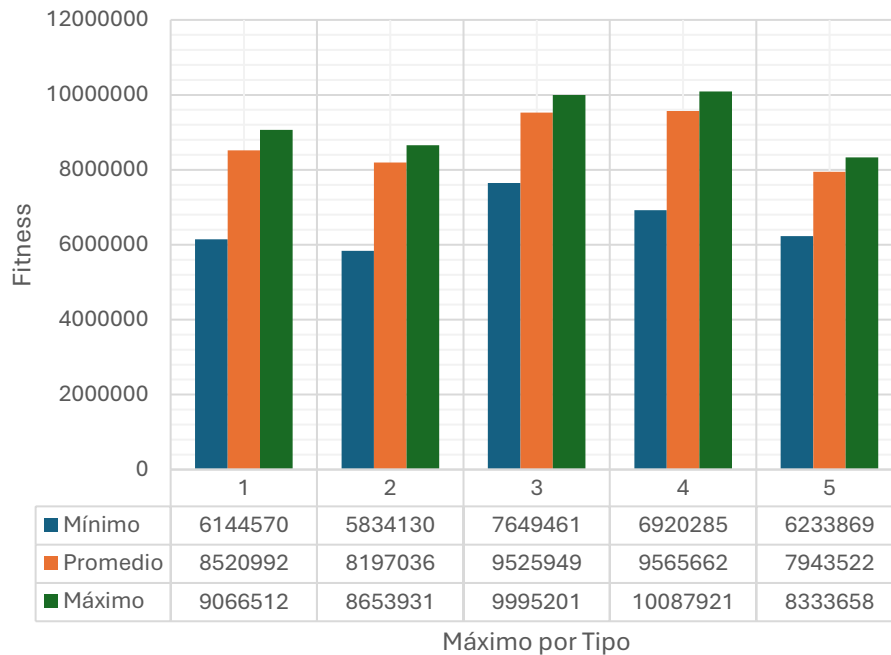


Gráfico 6 Evolución Fitness vs Máximo por tipo

8.2.2 Tabla y gráfico de los tiempos de ejecución

Utilizando 2 computadoras distintas (Acer, n.d.; ASUS, 2023), se utilizó el programa para calcular el uso computacional y su tiempo de ejecución en segundos, dándonos como resultado los siguientes tiempo, ordenados por orden de ejecución de cada caso. Por último, se revisa el tiempo total y se comparan ambos tiempos totales.

N° de Clientes	Máximo por tipo	Máximo por Cliente	Tiempo HVRP-FVL prueba 1	Tiempo HVRP-FVL prueba 2	Porcentaje aumento de tiempo
10	1	5	504,2600968	1096,589978	117,4651505
10	2	5	512,6113949	1069,98632	108,7324493
10	2	10	534,3839018	1087,60136	103,5243495
10	3	5	503,1119533	1053,652601	109,4270656
10	3	10	557,3345597	1178,111215	111,3831261
10	3	15	570,3106961	1219,418222	113,8164742
10	4	5	537,0224183	1083,978823	101,8498271
10	4	10	638,3015432	1169,44656	83,21224072
10	4	15	593,5105164	1361,834662	129,4541756
10	4	20	636,2380197	1346,457758	111,627994
10	5	5	526,4631169	1096,187632	108,2173655
10	5	10	572,9316075	1206,485156	110,5810082
10	5	15	661,9988313	1368,903243	106,7833323
10	5	20	802,8018551	1683,709546	109,7291548
10	5	25	651,130121	1495,290707	129,6454517
20	1	5	996,0138075	2146,286794	115,4876547
20	2	5	1078,693504	2267,060899	110,1672895
20	2	10	1218,78301	2490,112634	104,3114002
20	3	5	1055,127892	2198,383271	108,3523038
20	3	10	1292,70868	2779,090019	114,981926
20	3	15	1368,040882	2737,457226	100,1005425
20	4	5	1078,457647	2307,307332	113,9451037
20	4	10	1367,847096	2871,725614	109,944929
20	4	15	1422,38514	3008,249023	111,4932825
20	4	20	1449,204319	3680,006537	153,9328988
20	5	5	1093,634443	2433,903001	122,5517875
20	5	10	1348,108934	3351,560048	148,6119604
20	5	15	1923,047052	3979,227535	106,9230459
20	5	20	1496,970009	3752,402943	150,6665411
20	5	25	1491,538828	3603,104013	141,5695753
30	1	5	1830,679469	3857,270178	110,7015588
30	2	5	2018,65136	4085,808555	102,4028832
30	2	10	2228,597455	4569,290898	105,0298894
30	3	5	2019,310915	4158,00446	105,912048
30	3	10	2674,950974	5040,663229	88,43946221
30	3	15	3053,537511	5136,339749	68,20948589
30	4	5	2139,252589	4295,053069	100,773536

30	4	10	3001,402707	5767,179578	92,14947615
30	4	15	3513,634161	5771,431268	64,2581727
30	4	20	3629,894938	7232,841739	99,25760558
30	5	5	1998,706403	4143,75046	107,3216182
30	5	10	2765,312026	5956,533413	115,4018554
30	5	15	3084,730931	6743,604326	118,6124001
30	5	20	4265,298401	6873,258236	61,14366664
30	5	25	3892,49312	8615,159964	121,3275579
40	1	5	3278,143418	6056,139053	84,74295604
40	2	5	3505,786018	7122,679243	103,1692524
40	2	10	3690,543758	7655,723416	107,4416107
40	3	5	3252,102358	6924,913423	112,9365149
40	3	10	3773,732839	8167,037522	116,4180103
40	3	15	4040,202857	8365,503292	107,0565164
40	4	5	3167,158711	6678,444734	110,86549
40	4	10	4072,008072	9032,095603	121,8093737
40	4	15	4642,151444	9831,727116	111,7924681
40	4	20	4798,892917	11173,71231	132,8393757
40	5	5	3152,67127	6551,68484	107,8137642
40	5	10	4478,641872	9859,164251	120,1373661
40	5	15	4631,589484	12852,1412	177,488781
40	5	20	4911,068113	10828,09224	120,4834465
40	5	25	6647,369319	12241,73667	84,15911748
50	1	5	4289,665489	8863,292754	106,6196718
50	2	5	4652,014741	9827,152655	111,2450884
50	2	10	5182,379978	11283,76387	117,7332406
50	3	5	4710,481986	10245,52741	117,5048633
50	3	10	5899,426592	12469,51236	111,3682096
50	3	15	6090,601832	12291,0346	101,8032854
50	4	5	4636,020566	10047,62368	116,7294888
50	4	10	6388,394864	13668,48969	113,9581222
50	4	15	8195,897773	16311,26067	99,01737573
50	4	20	7415,063356	16944,79479	128,5185436
50	5	5	4733,422256	9892,037916	108,9827905
50	5	10	6612,791518	14424,0091	118,1228465
50	5	15	9381,482415	22051,08302	135,0490258
50	5	20	7162,221347	18947,70408	164,5506634
50	5	25	9893,294857	21077,62579	113,0496068

Tabla 11 Duración y aumento de tiempo por experimento en segundos

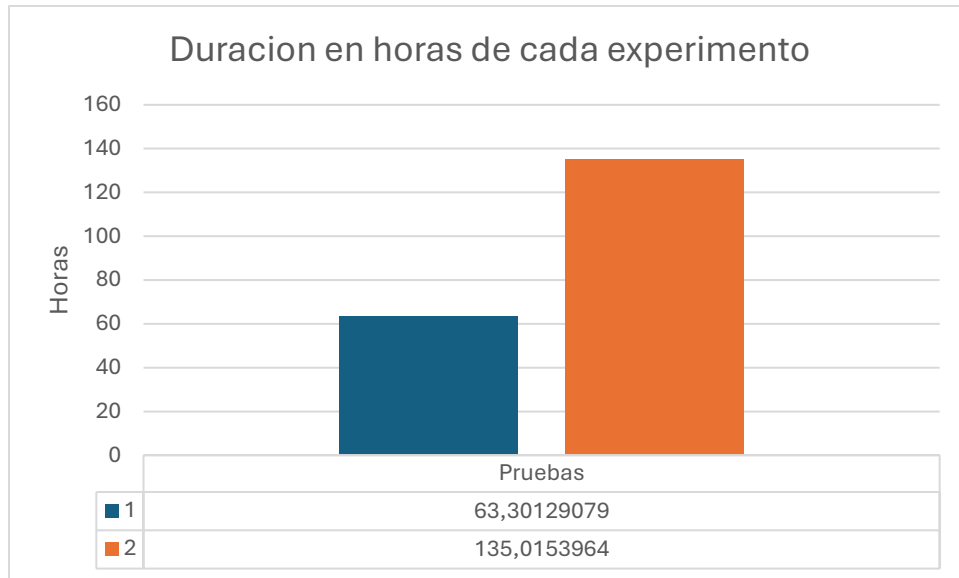


Gráfico 5 Duración en horas de cada experimento

9. Conclusiones

Comparando los resultados obtenidos con los del artículo base, se evidencia que el algoritmo propuesto es capaz de replicar el funcionamiento 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.
- Desarrollar las pruebas con información local.
- Pruebas con hardware de mayor capacidad, con el fin de mejorar el rendimiento y encontrar el balance para la resolución de este tipo de problemas.

10. Referencias

- Acer. (n.d.). *Nitro 5 (AN515-45)* . Retrieved January 5, 2025, from <https://www.acer.com/mx-es/laptops/nitro/nitro-5/pdp/NH.Q7MAL.001>
- Archetti, C., Speranza, M. G. & Vigo, D. (2014). Vehicle Routing: Problems, Methods, and Applications. In *Vehicle Routing* (Second). Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611973594.ch10>
- ASUS. (2023). *ASUS TUF Gaming F15 (2023)* . <https://www.asus.com/cl/laptops/for-gaming/tuf-gaming/asus-tuf-gaming-f15-2023/techspec/>
- Bektaş, T. & Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B: Methodological*, 45(8), 1232–1250. <https://doi.org/10.1016/j.trb.2011.02.004>
- Bibiana Rocha Medina, L., Cristina González La Rota, E. & Arturo Orjuela Castro, J. (n.d.). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución State of the art review of the vehicle routing problem: A historic account with solving methods. In *Ingeniería* (Vol. 16, Issue 2). <https://www.redalyc.org/articulo.oa?id=498850173004>
- Braekers, K., Ramaekers, K. & Van Nieuwenhuyse, 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>
- Demir, E., Bektaş, T. & Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5), 347–357. <https://doi.org/10.1016/J.TRD.2011.01.011>
- Hasle, G., Lie, K. A. & Quak, E. (2007). Geometric modelling, numerical simulation, and optimization: Applied mathematics at SINTEF. *Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF*, 1–558. <https://doi.org/10.1007/978-3-540-68783-2/COVER>
- Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2), 231–247. [https://doi.org/10.1016/0377-2217\(92\)90138-Y](https://doi.org/10.1016/0377-2217(92)90138-Y)

Liu, J., Yuan, B., Hu, Y. & Smith, A. E. (n.d.). *The finished vehicle routing problem with a heterogeneous transport fleet*.
<https://ssrn.com/abstract=4239703>

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

Núñez, J. & Cotejo, A. J. (n.d.). *Pseudocódigo + Diagramas de Flujo + Introducción a los lenguajes: C y Python*. Retrieved January 5, 2025, from https://xn--jesusnuez-r6a.com/wp-content/uploads/pseudoy_cpython.pdf

OpenStreetMap. (n.d.). Retrieved January 5, 2025, from <https://www.openstreetmap.org/>

Overpass API Wiki. (n.d.). Retrieved January 5, 2025, from https://wiki.openstreetmap.org/wiki/Overpass_API

OverPass Turbo. (n.d.). Retrieved January 5, 2025, from <https://overpass-turbo.eu>

PMX Crossover / Manuel Alabor | Observable. (n.d.). Retrieved January 5, 2025, from <https://observablehq.com/@swissmanu/pmx-crossover>

PSelnt. (n.d.). Retrieved January 5, 2025, from <https://pseint.sourceforge.net/slide/pseint.html>

Python Release Python 3.10.0 | Python.org. (n.d.). Retrieved January 5, 2025, from <https://www.python.org/downloads/release/python-3100/>

Rodríguez-Vásquez, W. C. (2020). Modelado de un problema de ruteo de vehículos con múltiples depósitos, ventanas de tiempo y flota heterogénea de un servicio de mensajería. *Información Tecnológica*, 31(1), 207–214.
<https://doi.org/10.4067/S0718-07642020000100207>

Vehicle routing problem - Wikipedia. (n.d.). Retrieved January 5, 2025, from https://en.wikipedia.org/wiki/Vehicle_routing_problem#cite_ref-4