



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Matemàtica Aplicada

Optimización multiobjetivo para ruteo de vehículos:  
aplicación de una adaptación BR-Heurística para  
múltiples depósitos.

Trabajo Fin de Máster

Máster Universitario en Investigación Matemática

AUTOR/A: Medina Rodriguez, Veronica

Tutor/a: Conejero Casares, José Alberto

Cotutor/a: Juan Pérez, Ángel Alejandro

CURSO ACADÉMICO: 2023/2024



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

## Trabajo Final de Máster - Curso 2023/ 2024

Título: Optimización multiobjetivo para ruteo de  
vehículos: aplicación de una adaptación  
BR-Heurística para múltiples depósitos.

Autor: **Verónica Medina Rodríguez**

Tutor: ÁNGEL A. JUAN

Cotutor: JOSÉ ALBERTO CONEJERO CASARES



Facultat de  
Ciències Matemàtiques

Máster en Matemáticas

*Dedico este trabajo a mi familia,  
por su amor y apoyo incondicional.*

## Agradecimientos

Agradezco profundamente a mi director de tesis, Dr. Juan Pérez, por su orientación, apoyo y paciencia durante todo el proceso. Mi gratitud también a mi compañero de investigación, Marc Escoto, por sus valiosas aportaciones y por el ambiente de trabajo colaborativo.

A mi familia, por su incondicional apoyo y comprensión a lo largo de mi carrera académica. Sin su ánimo y respaldo, este trabajo no hubiera sido posible.

Finalmente, agradezco a todos aquellos que, de alguna manera, contribuyeron a la realización de esta tesis, en especial a mis compañeros de clase Carlos, Juliana, Daniel, Gaia, Gabriel.

## Resumen

En este trabajo, se presenta un problema de optimización multiobjetivo de enrutamiento de vehículos. El modelo propuesto se enfoca en la modificación de uno existente en la literatura, en el cual se minimiza la distancia total recorrida y se maximiza la visita de Nodos Muy Importantes (VIP) al inicio de cada recorrido. Se presenta un método híbrido para abordar el Problema del Vendedor Viajero Múltiple (mTSP) de ruta cerrada multidepósito mediante la integración de una heurística aleatoria con sesgo con la Búsqueda Local Iterativa (ILS). En el procedimiento propuesto se utiliza el concepto de optimalidad de Pareto para la optimización multiobjetivo. Los experimentos computacionales sobre conjuntos de datos ilustran la eficiencia y eficacia del enfoque propuesto.

## **Abstract**

In this thesis, a multi-objective vehicle routing optimization problem is presented. The proposed model focuses on modifying an existing one in the literature, in which the total distance traveled is minimized and the VIP visit at the beginning of each route is maximized. A hybrid method is presented to address the multidepot closed-route mTSP by integrating a biased random heuristic with the ILS. In the proposed procedure, the concept of Pareto optimality is used for multiobjective optimization. Computational experiments on data sets illustrate the efficiency and effectiveness of the proposed approach.

# Índice

<b>1. Introducción</b>	<b>9</b>
1.1. Objetivos . . . . .	10
1.2. Estructura de la Tesis . . . . .	11
<b>2. Caracterización Teórica y Estado del Arte</b>	<b>12</b>
2.1. Problema de Ruteo de Vehículos . . . . .	12
2.2. Problemas no Convexos y no Suaves . . . . .	15
2.3. Problemas Multiobjetivos . . . . .	16
2.4. Optimalidad de Pareto . . . . .	17
2.5. Metodología de Resolución. Problemas Multiobjetivos . . . . .	18
2.6. Algoritmos Heurísticos . . . . .	19
<b>3. Modelización Matemática del Problema y Metodología de Resolución</b>	<b>23</b>
3.1. Modelo Matemático . . . . .	23
3.2. Metodología de Resolución . . . . .	25
<b>4. Experimentos Computacionales</b>	<b>30</b>
<b>5. Análisis y Resultados</b>	<b>31</b>
<b>6. Conclusiones</b>	<b>46</b>
<b>A. Apéndice</b>	<b>51</b>
A.1. Tablas de Evaluación del Parámetro $\alpha$ . . . . .	51

## Índice de figuras

1.	Ruteo de vehículos. . . . .	10
2.	Función objetivo no convexa y función objetivo no suave. . . . .	16
3.	Frente Óptimo de Pareto. . . . .	17
4.	Clasificación de los procedimientos aleatorios sesgados. . . . .	21
5.	Utilización de distribuciones sesgadas para generar soluciones alternativas. . . . .	22
6.	Diagrama de flujo del método propuesto. . . . .	26
7.	Ilustración de la asignación sesgada de cada nodo a los depósitos. . . . .	27
8.	Soluciones encontradas con 2 depósitos para $\alpha = 0$ . . . . .	31
9.	Soluciones encontradas con 3 depósitos para $\alpha = 0$ . . . . .	32
10.	Soluciones encontradas con 4 depósitos para $\alpha = 0$ . . . . .	32
11.	Soluciones encontradas con 5 depósitos para $\alpha = 0$ . . . . .	33
12.	Soluciones encontradas con 2 depósitos para $\alpha = 1$ . . . . .	33
13.	Soluciones encontradas con 3 depósitos para $\alpha = 1$ . . . . .	34
14.	Soluciones encontradas con 4 depósitos para $\alpha = 1$ . . . . .	34
15.	Soluciones encontradas con 5 depósitos para $\alpha = 1$ . . . . .	35
16.	Impacto de $\alpha$ en la distancia total recorrida, instancia st70. . . . .	37
17.	Impacto de $\alpha$ en la cantidad de nodos VIP visitados, instancia st70. . . . .	37
18.	Impacto de $\alpha$ en la distancia total recorrida, instancia eil101. . . . .	38
19.	Impacto de $\alpha$ en la cantidad de nodos VIP visitados, instancia eil101. . . . .	39
20.	Frente de Pareto para Instancia st70 y 2 depósitos. . . . .	41
21.	Frente de Pareto para Instancia st70 y 3 depósitos. . . . .	41
22.	Frente de Pareto para Instancia st70 y 4 depósitos. . . . .	42
23.	Frente de Pareto para Instancia st70 y 5 depósitos. . . . .	42
24.	Frente de Pareto para Instancia eil101 y 2 depósitos. . . . .	44
25.	Frente de Pareto para Instancia eil101 y 3 depósitos. . . . .	44
26.	Frente de Pareto para Instancia eil101 y 4 depósitos. . . . .	45
27.	Frente de Pareto para Instancia eil101 y 5 depósitos. . . . .	45



## Índice de cuadros

1.	Nomenclatura . . . . .	24
2.	Costos de las soluciones mostradas en las figuras 8, 9, 10, 11. . . . .	36
3.	Costos de las soluciones mostradas en las figuras 12, 13, 14, 15. . . . .	36
4.	Costos de transportación para nodos VIP y depósitos. . . . .	40
5.	Costos de transportación para nodos VIP y depósitos. . . . .	43
6.	Resultados para valores de Alpha en la instancia ch150.tsp . . . . .	51
7.	Resultados para valores de Alpha en la instancia kroB100.tsp . . . . .	51
8.	Resultados para valores de Alpha en la instancia kroC100.tsp . . . . .	52
9.	Resultados para valores de Alpha en la instancia kroD100.tsp . . . . .	52

## Índice de abreviaturas

<b>COP</b>	Problemas de Optimización Convexa .....
<b>CVRP</b>	Problema de Ruteo de Vehículos con Capacidad .....
<b>GATS</b>	Algoritmo Genético basado en Metaheurística con Selección de Torneo .....
<b>ILS</b>	Búsqueda Local Iterativa .....
<b>mTSP</b>	Problema del Vendedor Viajero Múltiple .....
<b>NCOPs</b>	Problemas de Optimización no Convexos .....
<b>NN</b>	Algoritmo del Vecino más Cercano .....
<b>NSPs</b>	Problemas de Optimización no Suaves .....
<b>POP</b>	Principios de Optimalidad Próxima .....
<b>QAP</b>	Problema de Asignación Cuadrática .....
<b>UAV</b>	Vehículos aéreos no tripulados .....
<b>VIP</b>	Nodos Muy Importantes .....
<b>VRP</b>	Problema de Ruteo de Vehículos .....
<b>VRPB</b>	Problema de Ruteo de Vehículos con logística Inversa .....
<b>VRPTW</b>	Problema de Ruteo de Vehículos con Ventana de Tiempo .....

# 1. Introducción

El transporte juega un rol importante en la productividad y competencia de un país, es decir, como se administren los problemas de distribución impactan directamente en su eficacia. Por lo tanto, la creación de recorridos y horarios para los vehículos constituyen un conjunto de problemas comunes que deben ser resueltos de manera óptima.

Las empresas se encuentran inmersas en un entorno de constante cambio, lo que influye en sus relaciones con proveedores, competidores, clientes y personal. Esta dinámica puede afectar la eficiencia de las empresas, causando un aumento en los costos y gastos asociados. Por esta razón, los empresarios buscan constantemente herramientas para mitigar los efectos de la volatilidad del mercado. A nivel global, los costos de transporte representan una parte significativa de los gastos logísticos de las empresas, por lo que es fundamental controlar y planificar su impacto mediante métodos de ruteo eficientes.

Como solución a la problemática se desarrollan muchos algoritmos, como es la generalización del Problema del Vendedor Viajero Múltiple (mTSP) que dado un conjunto de clientes y  $m$  depósitos, tenemos que encontrar recorridos para todos los  $m$  transportes que empezarán y terminarán en el origen y visitarán cada cliente intermedio exactamente una vez, por lo que el coste total y el tiempo deben minimizarse.

Por lo tanto, el problema se divide en dos niveles:

1. Determinar la subdivisión óptima de los clientes en algunos grupos.
2. Encontrar el ciclo de longitud mínima para cada grupo.

El mTSP se resuelve principalmente mediante enfoques heurísticos debido a la complejidad de los modelos. Estos métodos pueden clasificarse en dos grandes grupos, los constructivos y los de búsqueda local. Los métodos heurísticos de construcción son procedimientos iterativos que, en cada paso, añaden un elemento hasta completar una solución. Usualmente son métodos deterministas y están basados en seleccionar en cada iteración, el elemento con mejor evaluación. Los algoritmos heurísticos de construcción son normalmente muy rápidos y devuelven soluciones razonablemente buenas. En cambio los algoritmos de búsqueda por vecindad (también llamados algoritmos de búsqueda local) constituyen un tipo bastante amplio de algoritmos de mejora en los que en cada iteración se obtiene una solución mejorada buscando en el espacio de las soluciones candidatas, permitiendo encontrar soluciones muy cercanas a la óptima pero en menor tiempo (Rosabal 2019).

Muchos investigadores han resuelto el mTSP en diferentes entornos, como preciso, difuso e incierto, con diversos enfoques heurísticos y metaheurísticos. Sin embargo, varios parámetros afectan a la transportación, como las condiciones de la carretera, las condiciones meteorológicas, el tráfico y la distancia, ya que juegan un papel importante a la hora de completar el recorrido y entregar el pedido al cliente a tiempo, así como la introducción de nodos VIP, que emergen como elementos cruciales dentro del mTSP, ya que representan destinos prioritarios que requieren una atención especial en términos de planificación y optimización de rutas.

En situaciones como la distribución de vacunas o medicamentos que requieren entrega urgente a hospitales o centros de salud, ciertos nodos (hospitales prioritarios) deben ser atendidos antes que otros. Incluir estos nodos como VIP asegura que los suministros críticos lleguen a tiempo.

También podemos pensar que durante operaciones de ayuda humanitaria tras un desastre, ciertos nodos (como centros de refugio o zonas afectadas críticas) deben ser priorizados para la entrega de alimentos, agua, y otros suministros vitales. Aquí, los nodos VIP son esenciales para salvar vidas y reducir el sufrimiento humano.

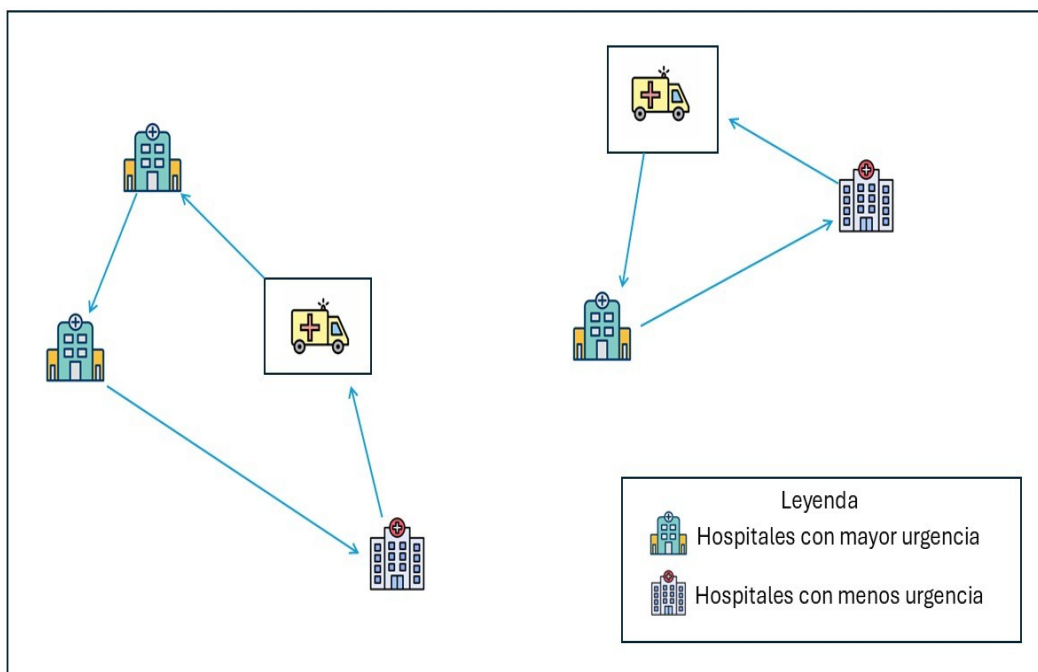


Figura 1: Ruteo de vehículos.

La identificación y correcta gestión de estos nodos VIP pueden mejorar significativamente la eficiencia del sistema de transporte. En el presente trabajo se estudia un método híbrido multiobjetivo para abordar el mTSP de ruta cerrada multidepósito mediante la integración de una heurística aleatoria con sesgo con la búsqueda local iterativa cuyos objetivos son reducir la distancia total recorrida por cada vehículo y a su vez maximizar la cantidad de nodos VIP visitados al inicio de cada ruta.

## 1.1. Objetivos

Los objetivos de la tesis se concretan mediante la siguiente estructura en un objetivo general y objetivos específicos, que se presentarán a continuación.

### Objetivo General

- Modelar matemáticamente y resolver mediante algoritmos heurísticos un problema del tipo mTSP biobjetivo que minimiza la distancia total recorrida por los vehículos y maximiza la cantidad de nodos VIP visitados al inicio del recorrido.

## Objetivos Específicos

- Desarrollar un modelo matemático que aborde el problema planteado.
- Diseñar e implementar una heurística aleatoria con sesgo con la búsqueda local iterativa cuyos objetivos son reducir la distancia total recorrida por cada vehículo y a su vez maximizar la cantidad de nodos VIP visitados al inicio de cada ruta.
- Realizar experimentos numéricos.
- Analizar resultados para varias instancias y comprobar la eficiencia del enfoque propuesto.

### 1.2. Estructura de la Tesis

En el Capítulo 1 de introducción, se comienza exponiendo la motivación de este estudio. También se plantean los objetivos generales y específicos que se llevaran a cabo a lo largo del estudio, así como la estructura de la tesis.

En el Capítulo 2 se abordará información relevante relacionada al tema de estudio. Se tratan los temas de modelos de ruteo desde sus inicios y como han ido evolucionando, los problemas de optimización no convexos y no suaves, sus características y como se resuelven en la actualidad. También introduciremos conceptos importantes como problemas multiobjetivos y la optimalidad de Pareto, así como también mencionaremos metodologías de resolución para este tipo de problemas como los algoritmos heurísticos.

En el Capítulo 3 se planteará el modelo matemático de ruteo a ejecutar y la heurística utilizada para resolver dicho modelo.

En el Capítulo 4 se realizarán experimentos computacionales, teniendo en cuenta diferentes escenarios y valores del parámetro  $\alpha$ . Se examinarán diferentes instancias y números de depósitos para evaluar la eficacia del modelo en diversas situaciones.

En el Capítulo 5 se validará el modelo propuesto a través del análisis de los resultados obtenidos en el capítulo anterior. Además, se realizarán comparaciones con otros modelos existentes para destacar las ventajas y posibles limitaciones del enfoque presentado. Este análisis detallado permitirá comprender mejor la robustez y la aplicabilidad del modelo en escenarios reales, proporcionando una base sólida para futuras investigaciones y desarrollos.

En el Capítulo 6 se presentan las conclusiones, donde se resumen los hallazgos más importantes y se discuten las implicaciones de los resultados obtenidos a lo largo de este estudio. Además se discutirán posibles trabajos futuros.

## 2. Caracterización Teórica y Estado del Arte

En este capítulo, se abordará el estado del arte del Problema de Ruteo de Vehículos (VRP), un tema crucial en el ámbito de la optimización logística.

La primera parte estará dedicada a la caracterización teórica del VRP. Aquí, se presentarán los principales estudios realizados en este campo desde sus inicios. Esta caracterización es fundamental para entender las bases sobre las cuales se desarrollan los modelos y algoritmos de solución.

A continuación, se explorará el estado del arte de los métodos y técnicas más avanzadas utilizadas para abordar este problema, incluyendo tanto enfoques exactos como heurísticos y metaheurísticos. Además, se discutirá la efectividad de estas técnicas en diferentes escenarios prácticos, así como los desafíos y oportunidades que presentan.

### 2.1. Problema de Ruteo de Vehículos

El siguiente análisis de la bibliografía brinda información detallada sobre las investigaciones llevadas a cabo en relación con el desafío del ruteo en tiempos recientes, facilitando una comprensión más profunda del tema. Además, se proporciona una ampliación de datos para situar en contexto el modelo de ruteo a utilizar en este estudio.

El Problema de Ruteo de Vehículos se refiere a una familia de problemas de optimización combinatoria, que consiste en planificar la secuencia en la que se deben visitar distintos puntos de interés con uno o más vehículos, con el objetivo de minimizar la distancia y/o el tiempo recorrido. Este problema tiene diversas aplicaciones en escenarios reales como la entrega de mercancías, el diseño logístico de procesos de fabricación, entre otras.

El desafío del VRP es uno de los más frecuentes en la optimización de procesos logísticos y uno de los más investigados, que plantea la búsqueda de la solución óptima considerando restricciones como: cantidad de vehículos, su capacidad, destinos (clientes) y demanda de estos, entre otros aspectos. Una formulación de este tipo puede contener múltiples variables, funciones objetivos y parámetros diversos.

El primer desafío planteado similar a VRP fue el del vendedor ambulante o TSP presentado por Flood (1956). El enigma se conoce así porque se puede explicar como un vendedor debe ir a cierta cantidad de ciudades en un solo trayecto, comenzando y terminando en la ciudad de partida. El vendedor debe encontrar la mejor ruta para visitar cada ciudad una sola vez y regresar, de modo que la distancia total recorrida sea la menor posible.

A partir de la propuesta de Flood surgen diversas versiones como el TPS extendido con Dantzing y Ramser (1959); proyecto en el que se modela la distribución de gasolina mediante una flota de camiones a distintas gasolineras, desde un punto de partida. Este estudio se convierte en el punto de partida para la evolución de otras formulaciones que van añadiendo más variables y limitaciones.

En 1960, Miller, Tucker y Zemlin presentaron la primera aparición del problema mTSP. Esta variante amplía las posibilidades del clásico TSP al involucrar  $m$  vehículos o agentes viajeros en la búsqueda de rutas eficientes. El principal objetivo es diseñar exactamente  $m$  rutas, una para cada

vehículo, de manera que cada cliente sea visitado una vez por alguno de los vehículos asignados. Cada ruta debe comenzar y terminar en el depósito designado, pudiendo incluir hasta  $p$  clientes en total. En el mTSP, cada cliente tiene una demanda específica y cada vehículo cuenta con una capacidad determinada, lo que convierte al problema del agente viajero en un problema de ruteo (Toth y Vigo 2002).

A partir del trabajo de Tillman, en 1969, se da origen al TSP probabilístico o PTSP. El objetivo de este problema es encontrar el mínimo costo de recorrido esperado a través de un conjunto de nodos con probabilidades asociadas a la presencia o no de consumidores que requieren ser servidos (Bruce Golden y Wasil 2008).

En la década de los 90 se presenta una variación de VRP que se refiere a la idea de VRP con múltiples usos de vehículos, basada en el supuesto que un vehículo puede hacer más un viaje en un periodo de planeación. Esta idea fue introducida por Fleischmann en 1990. En 1997 Brandao y Mercer lo trabajaron como VRP con viajes múltiples resolviéndolo mediante una Búsqueda Tabú (Brandao y Mercer 1997). En 2002 Prins introdujo el VRP con viajes múltiples de flota heterogénea mediante un caso real a larga escala, luego en 2008 aparece el VRP con viajes múltiples periódico y el VRP con viajes múltiples independientes del sitio (Rocha, González y Javier Orjuela 2011).

Las variantes de los modelos de VRP mencionadas anteriormente pretenden asemejar la realidad para poder solucionar los problemas contextualizados. Existen otras modificaciones a este problema muy utilizadas como el Problema de Ruteo de Vehículos con Capacidad (CVRP), en el que se tiene en cuenta la capacidad del vehículo y no se permitirá transportar una carga que exceda su capacidad. En esta variante se cuenta con varios clientes, un depósito único. Luego el desafío consiste en planificar las rutas de los vehículos de forma tal que se reduzca al mínimo la distancia total recorrida, asegurando la entrega programada a todos los clientes. En este tipo de VRP, el inicio y fin del mismo es el depósito central (Narducci 2010).

También pudieramos mencionar el problema de logística inversa, conocido como Problema de Ruteo de Vehículos con logística Inversa (VRPB), en donde el VRP modela un sistema que permite al vehículo recoger productos y hacer entregas en la misma ruta (Goetschalckx 1992); VRP con ventana de tiempo, en la literatura lo podemos encontrar como Problema de Ruteo de Vehículos con Ventana de Tiempo (VRPTW), el cual tiene el mismo planteamiento del VRP pero con la restricción de que existe un periodo de tiempo determinado para cumplir con el abastecimiento de los clientes (Cruz 2008) y la versión de múltiples depósitos, conocida como MDVRP, que implica la presencia de varios puntos de distribución para atender a los clientes. En este escenario, el MDVRP se aborda en dos fases consecutivas: la primera etapa se centra en asignar a los clientes a los diferentes depósitos disponibles, mientras que la segunda fase se enfoca en planificar las rutas óptimas para cada depósito, con el objetivo de minimizar la suma de las distancias recorridas por cada vehículo (Zuluaga 2016).

En investigaciones más recientes Huang et al. (2024) expone una alternativa para la organización de misiones de fumigación en zonas montañosas utilizando varios multicopteros colaborativos. El desafío se plantea como un mTSP y para abordarlo se proponen tres algoritmos distintos: el clásico mTSP, el algoritmo combinado Grouping-TSP, y el algoritmo desacoplado Grouping-TSP. Los resultados de la simulación muestran que el algoritmo clásico mTSP asigna las tareas de manera equitativa, mientras que el algoritmo combinado Grouping-TSP ofrece la solución óptima .

Los estudios realizados por Shao y Xu (2023) se centran en el diseño de rutas para varios drones no tripulados Vehículos aéreos no tripulados (UAV) con el fin de recopilar información después de una catástrofe a través de la inspección cooperativa de vídeo y la carga de información en estaciones base terrestres. Se diseñan rutas tridimensionales para varios UAV con el objetivo de minimizar el tiempo de inspección y maximizar el tiempo de comunicación para la carga de vídeo. Se propone la resolución de un problema de vendedor viajero múltiple para diseñar las trayectorias de los drones y los resultados numéricos muestran que este diseño puede reducir significativamente el tiempo de inspección mediante vídeo, permitiendo una respuesta rápida y una planificación efectiva en caso de catástrofe .

En cambio si analizamos las investigaciones en el campo del VRP multiobjetivo encontramos menos referencias que las temáticas abordadas anteriormente. Algunos de los estudios realizados se describen a continuación.

Murata e Itai (2005) propusieron un algoritmo evolutivo de dos fases para problemas multiobjetivo, que trataba de minimizar al mismo tiempo el número de vehículos y el tiempo máximo de enrutamiento, es decir, la ruta con la máxima duración. Para su resolución, clasificaron los problemas en dos tipos, según las demandas de los clientes fueran normales o grandes. Posteriormente Jozefowicz, Semet y Talbi (2009) propusieron otro algoritmo evolutivo para abordar un problema bi-objetivo del CVRP buscando un equilibrio en distancias de la ruta solución. En dicho problema se minimizaban la distancia total recorrida por los vehículos y la diferencia entre las rutas de mayor y menor distancia recorrida. Gambardella, Taillard y Agazzi (1999) consideraron el problema VRPTW como un problema de optimización bi-objetivo con una función objetivo jerárquica. El primer objetivo trata de minimizar el número de vehículos y el segundo minimizar el tiempo total de viaje. Esto se logró mediante la adaptación de un algoritmo de optimización de colonia de hormigas. En los estudios realizados por Hanshar (2006) se tuvo en cuenta el mismo problema, pero minimizando al mismo tiempo el número de vehículos y la distancia total recorrida. Para su resolución utilizaron respectivamente un algoritmo genético haciendo uso de una técnica de clasificación de Pareto y un algoritmo evolutivo híbrido. En el mismo año, dichos autores modificaron ligeramente el enfoque anterior para abordar la variante con camiones y remolques. En esta variante, se consideran diferentes tipos de vehículos, algunos de ellos con ciertas limitaciones (Lee 2006).

Otro autor que trató el VRPTW como un problema multiobjetivo fue Beham (2007), pero propuso un enfoque de búsqueda tabú para su resolución. En su problema minimizaba el número de rutas, la distancia total del viaje y el tiempo de violación de las ventanas de tiempo. Luego Ghannadpour (2010) presentó un estudio con un enfoque de programación por metas para la formulación del problema multiobjetivo VRPTW, en el que el decisor especifica los objetivos a obtener más optimistas y se minimizan las desviaciones respecto a éstas. Para su resolución, realizaron una adaptación de un algoritmo genético donde se minimizan tanto el tamaño de la flota total requerida como la distancia total del viaje, satisfaciéndose las restricciones de capacidad y las ventanas de tiempo. Martí (2006) aborda el problema de optimización de rutas de autobuses escolares. Este problema consiste en el transporte de un grupo de estudiantes desde sus casas a la escuela. La función objetivo del problema considera la minimización de dos objetivos muy diferenciados: el número total de autobuses, y el tiempo de espera que un estudiante pasa en el bus, que coincide con la ruta de mayor duración. El problema se resuelve teniendo en cuenta ambas funciones objetivo por separado y presenta varios métodos constructivos y un procedimiento de búsqueda tabú para su resolución.

En el ámbito de la logística verde, Chan (2012) propone un enfoque multiobjetivo del problema



de optimización de rutas en el que aparte del objetivo clásico de minimización de costes con restricciones de tiempo, trataron la minimización de las emisiones de CO<sub>2</sub>, incorporándolas como un objetivo secundario del problema así como una restricción adicional.

En cambio en este caso se enfoca en reducir las emisiones de carbono en transportes mediante un algoritmo basado en un algoritmo multiobjetivo quasi oppositional Jaya (RPAL-based MOJO Jaya). Se resuelve el mTSP, considerando múltiples vendedores y restricciones de riesgo y carbono. Se utilizan datos reales de la ciudad de Surat para observar los efectos de la selección de vendedores en diferentes modelos. El rendimiento del algoritmo propuesto se compara y se demuestra su eficacia en la optimización de rutas de transporte (Bajaj y Dhodiya 2024).

El artículo publicado por Linganathan y Singamsetty (2024) se presenta un Algoritmo Genético basado en Metaheurística con Selección de Torneo (GATS) para resolver un MTSP bi-objetivo con la restricción de equilibrio de carga, donde el primer objetivo es minimizar la distancia total de viaje y el segundo objetivo minimiza el tiempo total. El GATS integra estrategias mixtas en la operación de mutación y se comparó con otros enfoques genéticos en diferentes conjuntos de datos derivados de TSPLIB. Los experimentos demostraron que el GATS propuesto obtuvo soluciones Pareto eficientes en el BMTSP.

## 2.2. Problemas no Convexos y no Suaves

Los problemas de optimización podemos clasificarlos en dos grandes grupos, convexos y no convexos. En general, los Problemas de Optimización Convexa (COP) constan de dos componentes: restricciones que delimitan regiones convexas y una función objetivo que se busca minimizar y que, asimismo, es convexa. El problema dual, en el cual la función objetivo es cóncava y se busca maximizar, también clasifica como problema convexo. Es interesante estudiarlos porque tienen muchas aplicaciones, la programación lineal es un ejemplo de ello, ya que las funciones lineales son convexas (Angel Juan et al. 2013).

La ventaja en problemas de optimización convexa consiste en que rango de soluciones está restringido a una región convexa particular. Al tomar la intersección de todas estas regiones, obtenemos el conjunto de soluciones posibles, que también presenta un carácter convexo y por tanto todo óptimo local se convierte en un óptimo global. Diversos algoritmos han sido creados, como el Método del Punto Interior, para hallar la solución óptima. No obstante, la mayoría de ellos no pueden adaptarse fácilmente a casos no convexos.

En los Problemas de Optimización no Convexos (NCOPs), la función objetivo o la región factible pueden no ser convexas, lo que resulta en un espacio de soluciones mucho más complicado en comparación con los problemas de optimización convexas. En este escenario, es posible encontrar múltiples regiones disjuntas y varios puntos óptimos locales dentro de cada una de ellas. Por lo tanto, al aplicar métodos de búsqueda local tradicionales en un NCOP, existe un alto riesgo de converger hacia un óptimo local en lugar del óptimo global. Además, determinar si un NCOP es infactible, si la función objetivo no tiene límite superior, o si una solución encontrada es en realidad el óptimo global puede llevar un tiempo exponencial en función del tamaño de la entrada.

Una función se considera suave si es diferenciable y su derivada es continua (Angel Juan et al. 2013). Luego, una función no suave carece de alguna de estas propiedades. Los Problemas de Optimización no Suaves (NSPs) son similares a los NCOPs en que resultan ser mucho más difíciles de resolver que los típicos problemas suaves y convexas. La función sobre la cual se busca el óptimo

global en este caso no es suave, lo cual lleva a que el espacio de solución pueda contener múltiples regiones separadas y numerosos óptimos locales en cada una de ellas. Esta estructura dificulta la aplicación de herramientas matemáticas tradicionales, como la información de gradientes, tornando la resolución de estos problemas muy complicada e incluso imposible en algunos casos. Las técnicas computacionales empleadas para resolver este tipo de problemas suelen ser extremadamente complicadas y dependen en gran medida de la estructura específica del problema en cuestión. Como consecuencia, el desarrollo de dichas técnicas es generalmente un proceso a largo plazo y su alcance de aplicación resultante es limitado. Sin embargo, la mayoría de las funciones objetivo en situaciones cotidianas son no convexas, no suaves o incluso ambas cosas a la vez. Esto convierte a la optimización combinatoria bajo estas circunstancias complejas pero comunes en un área de estudio crucial.

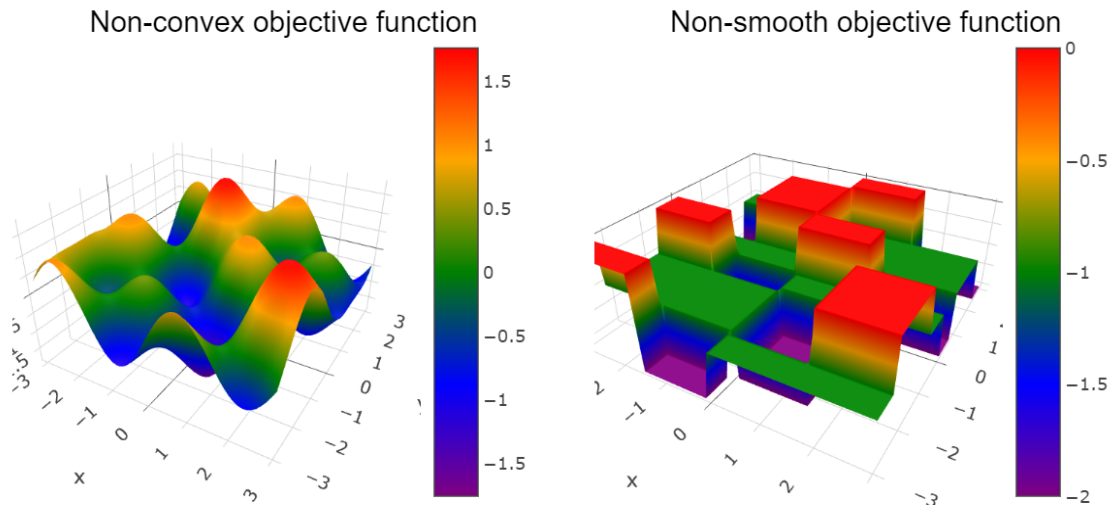


Figura 2: Función objetivo no convexa y función objetivo no suave.

### 2.3. Problemas Multiobjetivos

La programación multiobjetivo es una parte de la programación matemática que se encarga de problemas de decisión con múltiples funciones objetivo que deben ser optimizadas sobre un conjunto factible de soluciones (Marler y Arora 2004). El problema de optimización multiobjetivo general se plantea como:

$$\begin{aligned} \min \quad & F(x) = (F_1(x), F_2(x), \dots, F_k(x)) \\ \text{sujeto a:} \quad & g_j(x) \leq 0 \quad j = 1, \dots, m \\ & g_l(x) = 0 \quad l = 1, \dots, n \end{aligned}$$

Donde  $k$  es el número de funciones objetivo,  $x$  es el vector de variables de decisión y  $m$  es el número de restricciones de desigualdad y  $n$  es el número de restricciones de igualdad. espacio de soluciones factibles.

Marler y Arora (2004), le llama optimización multiobjetivo (MOO) al proceso de optimizar de manera simultánea una colección de funciones objetivo. En estos casos no hay una solución global

única, sino que es necesario determinar un conjunto de puntos que correspondan a las mejores soluciones.

## 2.4. Optimalidad de Pareto

Los vectores que forman parte del conjunto que optimiza el problema multiobjetivo tienen una serie de características en común. La principal de ellas es que no se puede mejorar el valor en un objetivo sin empeorar el valor del otro. La noción de optimalidad se vuelve necesaria para identificar de manera formal que soluciones forman parte del conjunto final, es decir, que soluciones son mejores que otras. El conjunto de Pareto se define de la siguiente forma:

**Definición 1:** Un punto  $x^* \in X$ , es un óptimo de Pareto si y solo si no existe otro punto,  $x \in X$ , tal que  $F(x) \leq F(x^*)$ , y  $F_i(x) < F_i(x^*)$  para al menos una función.

Todos los puntos óptimos de Pareto se encuentran en el límite del espacio de las soluciones factibles (Athan y Papalambro 1996). Véase la siguiente figura, donde se muestra el frente óptimo de Pareto.

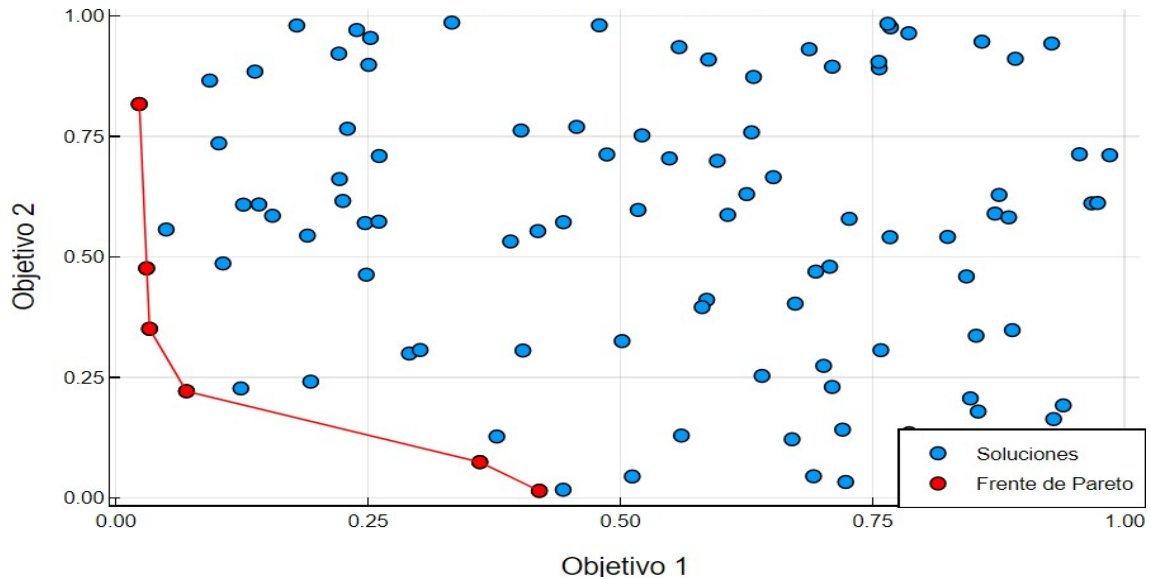


Figura 3: Frente Óptimo de Pareto.

A menudo, los algoritmos proporcionan soluciones que pueden no ser óptimas de Pareto pero que pueden satisfacer otros criterios, lo que las convierte en significativas para aplicaciones prácticas. Por ejemplo, débilmente óptimo de Pareto se define de la siguiente manera (Marler y Arora 2004):

**Definición 2:** Un punto  $x^* \in X$ , es débilmente óptimo de Pareto si no existe otro punto,  $x \in X$ , tal que  $F(x) < F(x^*)$ .

Dado que los vectores no pueden ordenarse completamente, todas las soluciones óptimas de Pareto pueden considerarse igualmente deseables en sentido matemático y necesitamos un decisor para identificar la preferida entre ellas.

El decisor es una persona que puede expresar información de preferencia relacionada con los objetivos. También suele ser necesario un analista que participe en el proceso de solución, es decir una persona o un programa informático responsable de la parte matemática del proceso de solución para seleccionar el método de optimización adecuado.

## 2.5. Metodología de Resolución. Problemas Multiobjetivos

Se han desarrollado numerosos enfoques y estrategias para explorar el espacio de posibles soluciones en busca de la opción óptima. En situaciones donde el espacio de soluciones es fácil de explorar debido a propiedades como la convexidad de las funciones implicadas, es posible resolver el problema de manera eficiente y precisa. Sin embargo, en otros casos, el espacio de soluciones es muy irregular y encontrar la mejor opción puede resultar imposible en un tiempo razonable. En estos casos difíciles, un enfoque exhaustivo que examine todos los puntos del espacio de soluciones sería poco práctico, ya que llevaría un tiempo exponencial.

Veamos varios métodos para hallar la solución de nuestro problema multiobjetivo, que permiten a los usuarios establecer sus preferencias de manera más específica, ya sea en función de sus metas o de la importancia que le asignen a los diferentes objetivos. La mayoría de estas opciones incluyen parámetros, como coeficientes, exponentes o límites de restricción, los cuales pueden ajustarse para reflejar las preferencias del individuo en la toma de decisiones, o modificarse de manera continua en un intento por representar el conjunto óptimo de Pareto completo.

Uno de los métodos más comunes para la optimización multiobjetivo es el método del criterio global en el que todas las funciones objetivo se combinan para formar una sola función. Un criterio global ponderado es un tipo de función de utilidad en la que se utilizan funciones de pesos para modelar las preferencias. Una de las funciones de utilidad más generales se expresa en su forma más simple como una suma exponencial ponderada:

$$U = \min \sum_{i=1}^k w_i [F_i(x)]^p, \quad F_i(x) > 0, \quad \forall i \quad (1)$$

donde  $w$  es un vector de pesos de tal manera que  $\sum_{i=1}^k w_i = 1$  y  $w > 0$  (Aranda y Orjuela 2015). Al igual que con la mayoría de los métodos que involucran pesos de función objetivo, establecer uno o más de los pesos en cero puede resultar ser una débil optimalidad de Pareto. Generalmente, el valor relativo de los pesos, viene dado por la importancia relativa de los objetivos.

Otro enfoque para la optimización multiobjetivo es el método de suma ponderada:

$$U = \min \sum_{i=1}^k w_i F_i(x), \quad F_i(x) > 0, \quad \forall i \quad (2)$$

Esta es una forma particular de (1) con  $p = 1$ . Si todos los pesos son positivos, el mínimo de (2) es óptimo de Pareto (Zadeh 1963); es decir, minimizar (2) es suficiente para la optimalidad de Pareto. Sin embargo, la formulación no proporciona una condición necesaria para la optimalidad de Pareto.

Elegir los pesos de manera errónea puede hacer que la optimización del problema sea ineficiente. En consecuencia, muchos autores han desarrollado enfoques sistemáticos para seleccionar pesos, cuyos

estudios son proporcionados por Eckenrode (1965) y Hobbs (1980). Aquí, describimos brevemente los enfoques generales básicos. Con métodos de clasificación, las diferentes funciones objetivos se ordenan por importancia y la menos importante recibe menor peso. Rao y Roy (1989) proporcionan un método para determinar los pesos basados en la teoría de conjuntos difusos. Para casos en los que la importancia relativa de las funciones objetivo no está clara, Wierzbicki (1986) proporciona un algoritmo que calcula los pesos basados en el punto de aspiración y el punto de utopía.

Diversos estudios han examinado las complicaciones del método de la suma ponderada, en el artículo publicado por Marler y Arora (2004) se hace una recopilación de dichos estudios. En primer lugar, a pesar de los numerosos métodos para determinar los pesos, una selección satisfactoria a priori de pesos no garantiza necesariamente que la solución final sea aceptable; puede ser necesario resolver el problema con nuevos pesos. De hecho, los pesos deben ser funciones de los objetivos originales, no constantes, para que una suma ponderada imite con precisión una función de preferencia.

El segundo problema con el enfoque de la suma ponderada es que es imposible obtener puntos en las porciones no convexas del conjunto óptimo de Pareto en el espacio de criterios. Das y Dennis (1997) y Messac (2000) dan razones teóricas para esta deficiencia. Aunque los conjuntos óptimos de Pareto no convexas son relativamente poco comunes, se señalan algunos ejemplos en la literatura (Koski 1985; Stadler 1995).

La dificultad final con el método de la suma ponderada es que variar los pesos de manera consistente y continua no necesariamente resultará en una distribución uniforme de puntos óptimos de Pareto y una representación precisa y completa del conjunto óptimo de Pareto. Das y Dennis (1997) discuten esta deficiencia en detalle e ilustran las condiciones necesarias para que una serie de iteraciones de suma ponderada produzca una distribución uniforme de puntos en la curva de Pareto (en el espacio de criterios).

Otro método importante en la optimización multiobjetivo es el de  $\epsilon$ -restricciones, donde se selecciona una de las funciones objetivo para ser optimizada y las demás se convierten en restricciones. Luego el problema adquiere la forma:

$$\begin{aligned} \min \quad & f_\ell(\mathbf{x}) \\ \text{sujeto a} \quad & f_j(\mathbf{x}) \leq \epsilon_j \quad \forall \quad j = 1, \dots, k, \quad j \neq \ell, \\ & \mathbf{x} \in S, \end{aligned}$$

donde  $l \in \{1, \dots, k\}$  y  $\epsilon_j$  son límites superiores para los objetivos ( $j \neq l$ ). El método es introducido por Haimes et al., 1971 se demuestra que funciona tanto para problemas convexas como no convexas. En cuanto a la optimalidad, se puede demostrar que la solución del problema planteado es siempre débilmente Pareto (Miettinen 1973).

## 2.6. Algoritmos Heurísticos

Las heurísticas son técnicas que permiten explorar el espacio de búsqueda y encontrar una solución factible en un tiempo razonable. Al combinar una heurística aleatoria sesgada con una búsqueda local adaptada al problema específico, se puede abordar eficazmente desafíos realistas en entornos complejos con funciones objetivo no suaves o no convexas y regiones irregulares.

Estas técnicas han sido empleados para resolver diversos problemas de optimización combinatoria, tales como: problemas de planificación y secuenciación (Funke, Grunert e Irnich 2005; Pinedo 2008),

retos de rutas para vehículos (Laporte 2009), situaciones de ubicación y disposición (Drezner y Hamacher 2002). Además, han demostrado ser eficaces para resolver dilemas reales de optimización combinatoria presentes en distintos sectores industriales, como: Logística, Transporte, Manufactura, Biotecnología, entre otros.

En cuanto al análisis de métodos para resolver VRP primitivos, podemos mencionar los primeros algoritmos heurísticos especialmente diseñados para ello: el algoritmo de ahorro de Clarke y Wright (1964) y el procedimiento de Gillet y Miller (1974). Ambos métodos se emplean en la resolución de problemas reales de pequeña dimensión. La siguiente generación de heurísticas VRP se basó en métodos de doble fase siendo uno de ellos un procedimiento exacto. El algoritmo más representativo de esta familia es el método Fisher y Jaikumar (1981).

Durante la década de los noventa nacen las metaheurísticas con una larga lista de nuevos métodos que hacen uso de números pseudoaleatorios durante los procesos de búsqueda constructiva o local y además los datos de entrada del problema utilizan bits aleatorios para realizar elecciones aleatorias durante la ejecución del algoritmo, por lo que para la misma entrada el algoritmo puede producir diferentes salidas en diferentes ejecuciones. Dentro de estos algoritmos podemos incluir, entre otros, los algoritmos genéticos y evolutivos (Holland 1975; Gendreau y Potvin 2010), GRASP (Festa y Resende 2009), optimización por colonia de hormigas (Dorigo y Stutzle 2009) y búsqueda iterativa (Fisher y Jaikumar 1981).

Por otro lado tenemos las simheurísticas, técnica de optimización que combina métodos de simulación con algoritmos metaheurísticos. Su origen se encuentra en la necesidad de resolver problemas complejos de optimización que incluyen incertidumbre y variabilidad en sus parámetros. La simheurística cierra una brecha importante en la investigación operativa. La mayoría de los estudios incluyen métodos aleatorios sesgados con una Simulación Monte Carlo, utilizando un algoritmo heurístico aleatorizado sesgado como un algoritmo de búsqueda de vecindad (Smurfit 2024).

Se propuso otro algoritmo basado en la combinación de un algoritmo genético con la simulación Monte Carlo que integra algoritmos de búsqueda genética evolutiva con el análisis estocástico de la simulación Monte Carlo para el problema de asignación de tareas y enrutamiento de UAV bajo incertidumbre (Á. Juan et al. 2021). Utilizando el concepto de heurística del ahorro, se desarrolló un algoritmo heurístico basado en el ahorro combinando el concepto con la simulación de Monte Carlo y un modelo de regresión múltiple. Esto es especialmente útil en situaciones en las que es necesario utilizar vehículos aéreos no tripulados, como en las tareas de socorro en caso de catástrofe, y en las que planificación de rutas puede ser crucial. Estos métodos son especialmente eficaces para los COP dinámicos y estocásticos, en los que las entradas pueden cambiar con el tiempo o incluir variables aleatorias.

Los algoritmos heurísticos y la simulación pueden utilizarse conjuntamente para producir soluciones que satisfagan otras propiedades deseables, como la robustez y la flexibilidad. Para resumir estos algoritmos aleatorios sesgados, tenemos la Figura 4, extraída de (Grasas et al. 2017).

A continuación se muestran algunas aplicaciones documentadas de estos algoritmos a problemas no suaves/no convexos.

Potvin (2009) analiza la implementación de algoritmos evolutivos en el VRP. Según su investigación, los algoritmos evolutivos han sido utilizados de diversas formas, principalmente para abordar el VRP Capacitado y el VRP con ventanas de tiempo. Otros trabajos interesantes son los publi-

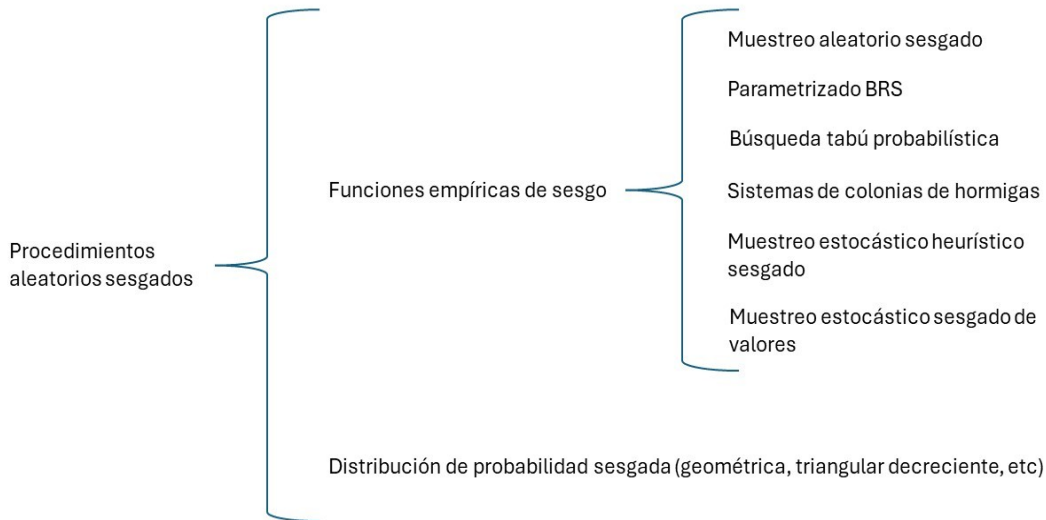


Figura 4: Clasificación de los procedimientos aleatorios sesgados.

cados por Banga (2009) que presentan una optimización de colonia de hormigas extendida para programación no lineal entera mixta no convexa. Toscano y Lyonnet (2010) proponen una nueva heurística para resolver problemas de optimización no convexos, designada como algoritmo heurístico de Kalman. Este método pertenece a la categoría de métodos estocásticos y considera el problema de optimización como un proceso de medición diseñado para dar una estimación del óptimo.

El Problema de Asignación Cuadrática (QAP) es considerado uno de los desafíos más interesantes y difíciles en optimización combinatoria. Fleurent y Glover (1999) proponen un algoritmo constructivo multiarranque mejorado para resolverlo. Este método se basa en el GRASP y emplea principios de búsqueda adaptativa de memoria. Estos principios incluyen una selección sesgada basada en probabilidades adaptativas para determinar qué elementos agregar a la solución parcial durante el proceso constructivo. Además de esto, se incorporan principios como la intensificación, estrategias de listas de candidatos y los Principios de Optimalidad Próxima (POP), que requieren la optimización de varios parámetros. Según los autores, las estrategias propuestas han demostrado obtener resultados excelentes en la mayoría de los casos.

A principios del nuevo siglo, esos métodos se desarrollaron y mezclaron produciendo procedimientos híbridos (Festa y Resende 2009) o nuevas estrategias basadas en ideas similares. Es interesante destacar que el uso de la simulación en el desarrollo de algoritmos de enrutamiento es una metodología adecuada para abordar problemas de enrutamiento con formulaciones complejas (Juan, Faulin y Jorba 2009). Del mismo modo, los algoritmos evolutivos demostraron ser una manera fiable de generar soluciones a VRPs intrincados.

En este estudio, se utilizará un enfoque conocido como Multi-start biased Randomization of classical Heuristics with Adaptive local search (Escoto, Guerrero y Medina 2024), que pertenece a la categoría de métodos no deterministas o estocásticos y se basa en un muestreo aleatorio sesgado. Por lo tanto, cada ejecución del algoritmo puede conducir a diferentes soluciones de calidad,

dependiendo de los puntos aleatorios elegidos.

Esta heurística de aleatorización sesgada propone considerar una lista de candidatos por cada depósito y luego asigna una probabilidad diferente a cada elemento de la lista de ser seleccionado, en nuestro caso hemos considerado la distribución de probabilidad geométrica. Esto es beneficioso pues tendremos diferentes soluciones, dependiendo de la cantidad de listas que generemos aleatoriamente.

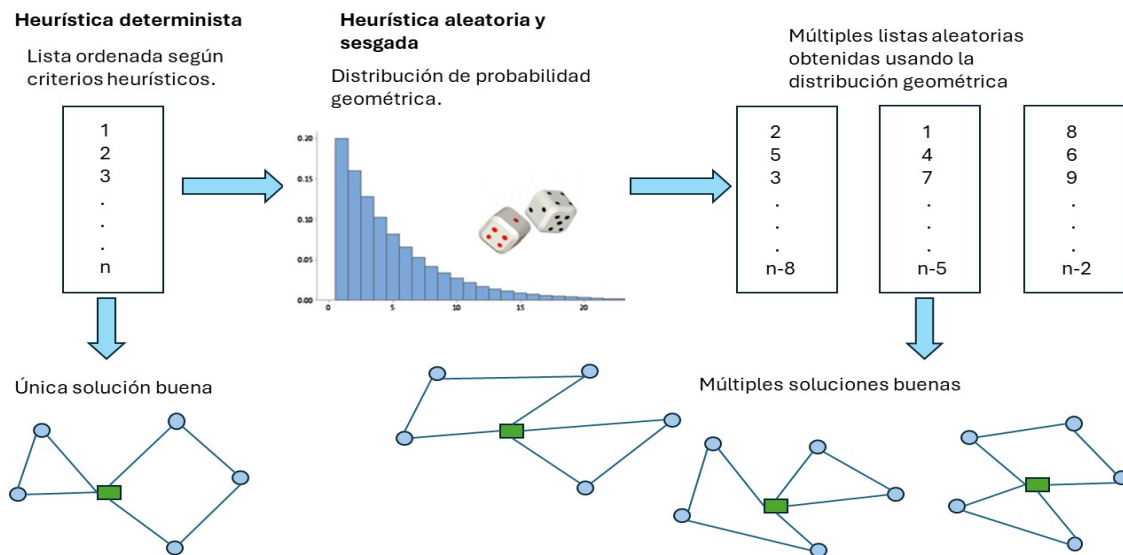


Figura 5: Utilización de distribuciones sesgadas para generar soluciones alternativas.

La heurística aleatoria presentada es integrada con un método de búsqueda local sencilla, llamado 2-opt. Este consiste en eliminar del tour un par de aristas que no sean adyacentes, y reemplazarlas con el único par de aristas con el cual se puede formar nuevamente un tour.

Este procedimiento de búsqueda local al no poder seguir mejorando la solución, se permite tomar otra solución aún cuando el valor no mejore, sino que se degrade, esto permite salir del óptimo local encontrado, pero al mismo tiempo se corre el peligro de caer en un ciclo, de mejorar-empeorar la solución, para evitar esto, se emplea una estrategia que modifica las vecindades a medida que la búsqueda avanza.



### 3. Modelización Matemática del Problema y Metodología de Resolución

Tanto la modelización matemática como la metodología de resolución son componentes fundamentales en la investigación operativa y la optimización, ya que permiten transformar problemas del mundo real en formulaciones matemáticas precisas que pueden ser abordadas de manera sistemática. En el contexto del Problema de Ruteo de Vehículos, estas herramientas son esenciales para desarrollar soluciones efectivas y eficientes.

En esta sección, se presentará un modelo matemático detallado, seguido por una descripción de la metodología de resolución empleada para encontrar soluciones óptimas al problema planteado. Esto incluirá la definición de las variables de decisión, de la función objetivo, los parámetros del problema, y las restricciones que deben cumplirse.

En la segunda parte se explicará la metodología de resolución utilizada que abarca las técnicas y algoritmos utilizados para resolver el modelo formulado.

#### 3.1. Modelo Matemático

Para la formulación de este modelo nos apoyamos en el artículo publicado por Escoto, Guerrero y Medina (2024), donde se estudia un método híbrido para abordar mTSP de ruta cerrada multi-depósito integrando un sistema heurístico de aleatorización con un sistema heurístico de búsqueda local iterativa.

Planteamos el problema como un grafo dirigido completo  $G = (V, A)$ , donde  $A$  es un conjunto de aristas y  $V$  es un conjunto de nodos. La matriz de costes  $C = (c_{ij})$  representa el coste de recorrer la arista  $(i, j)$  en  $A$ . En este problema, el conjunto  $V$  está formado por un conjunto de depósitos  $D$ , un conjunto de clientes  $V'$  y un conjunto  $V^*$  de clientes VIP. Son conjuntos disjuntos, donde denotamos por  $W = V' \cup V^*$  al conjunto total de clientes. Se considera un total de  $m$  vehículos, de los cuales  $m_i = 1$  se encuentran inicialmente en el depósito  $i$ . El objetivo es encontrar recorridos para todos los vehículos, asegurando que cada cliente sea visitado exactamente una vez, y que cada vehículo visite un número de clientes comprendido entre un límite inferior  $K$  y un límite superior  $L$ .

Tuvimos en cuenta varias consideraciones como que todos los nodos están conectados entre sí, y las distancias entre nodos son simétricas. Por tanto, podríamos describir el problema con un grafo dirigido, resultando una matriz de costes simétrica. Además que en cada depósito hay un único vehículo. Por último, como el método presentado garantiza rutas equilibradas en cuanto al número de nodos visitados por cada vendedor, los límites inferior y superior vienen determinados por  $\lfloor \frac{|V'|}{|M|} \rfloor$  y  $\lfloor \frac{|V'|}{|M|} \rfloor + 1$ , respectivamente.

<b>Conjuntos</b>	
$G$	Grafo
$V$	Conjunto de nodos
$D$	Conjunto de depósitos
$V'$	Conjunto de clientes
$V^*$	Conjunto de clientes VIP
$A$	Conjunto de aristas
<b>Variables</b>	
$x_{ij}^p$	Recorrido $(i, j)$ realizado por el vehículo $p$ (0: no, 1: si)
$u_j^p$	Cliente VIP $j$ asignado al vehículo $p$ es visitado antes de cualquier nodo no VIP (0: no, 1: si)
$y_i^p$	Cliente $i$ asignado al vehículo $p$ (0: no, 1: si)
<b>Parameters</b>	
$m$	Cantidad de vehículos
$m_i$	Cantidad de vehículos del depósito $i$
$c_{ij}$	coste de recorrer la arista $(i, j)$
$z_j$	Recompensa por visitar el nodo VIP $j$
$K_p$	límite inferior de cantidad de clientes que puede visitar el vehículo $p$
$L_p$	límite superior de cantidad de clientes que puede visitar el vehículo $p$

Cuadro 1: Nomenclatura

Como habíamos comentado, propondremos un modelo multiobjetivo, el cual minimizará la suma de la distancia recorrida por cada vehículo y a su vez maximizará la cantidad de nodos VIP visitados al inicio de cada ruta.

$$\text{Obj 1: mín} \left( \sum_{p \in D} \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}^p \right) \quad (3)$$

$$\text{Obj 2: máx} \left( \sum_{p \in D} \sum_{j \in V^*} z_j \cdot u_j^p \right)$$

Sujeto a:

$$\sum_{p \in D} y_i^p = 1, \quad i \in W \quad (4)$$

$$\sum_{j \in W} x_{ji}^p - \sum_{j \in W} x_{ij}^p = 0, \quad p \in D \quad i \in W \quad (5)$$

$$\sum_{j \in W} x_{pj}^p = 1, \quad p \in D \quad (6)$$

$$\sum_{j \in W} x_{jp}^p = 1, \quad p \in D \quad (7)$$

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij}^p \leq |S| - 1 \quad p \in D, \quad S \subseteq W, \quad |S| \geq 2 \quad (8)$$

$$K_p \leq \sum_{i \in V'} y_i^p \leq L_p, \quad p \in D, \quad K_p = \lfloor \frac{|V'|}{|m|} \rfloor \quad L_p = \lfloor \frac{|V'|}{|m|} \rfloor + 1 \quad (9)$$

$$O_j^p = M \left( 1 - \sum_{i \in W} x_{ij}^p \right) + \sum_{i \in W \setminus \{i\}} x_{ij}^p (O_j^p + 1) \quad (10)$$

$$O_d^p = 0, \quad d \in D \quad (11)$$

$$\begin{cases} u_j^p \leq \frac{O_i^p - O_j^p}{M} + 1, & \text{si } O_i^p - O_j^p < 0 \\ u_j^p \geq \frac{O_i^p - O_j^p}{M}, & \text{si } O_i^p - O_j^p > 0 \end{cases} \quad (12)$$

$$y_i^p \in \{0, 1\} \quad p \in D, \quad \forall i \in W \quad (13)$$

$$x_{ij}^p \in \{0, 1\} \quad p \in D, \quad \forall i, j \in W \quad (14)$$

$$u_j^p \in \{0, 1\} \quad p \in D, \quad \forall j \in V^* \quad (15)$$

La ecuación (4) garantiza que cada cliente se asigne exactamente a un depósito. En (5) se verifica que cada nodo sea visitado por un vehículo solo una vez y que dicho vehículo salga del nodo, es decir, el grado de entrada del nodo  $i$  es el mismo que su grado de salida. Las restricciones (6 – 7) garantizan que de cada depósito salga solo un vehículo y que entre solo uno también. En (8) se verifica que no se creen subtour en las rutas. La restricción (9) asegura que cada vehículo visite un número de clientes comprendido entre un límite inferior  $K$  y un límite superior  $L$ . En las ecuaciones (10–11) se determina el orden en el que se visitarán los nodos y en la (12) se asegura que la variable binaria  $u_j^p$  sea 1 si el nodo VIP  $j$  es visitado antes de todos los nodos no VIP  $i$ . Por último en las ecuaciones (13 – 14 – 15) se muestran los rangos de valores de las variables utilizadas.

Luego convertimos los dos objetivos individuales en una combinación lineal, con el parámetro  $\alpha \in [0, 1]$  que representa la importancia relativa de cada objetivo para el decisor.

$$\text{mín } Z = \alpha \sum_{p \in D} \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}^p - (1 - \alpha) \sum_{p \in D} \sum_{j \in V^*} z_j \cdot u_j^p \quad (16)$$

### 3.2. Metodología de Resolución

Para abordar el problema multiobjetivo planteado anteriormente creamos un método híbrido mediante la integración de una heurística aleatorización sesgada multiarranque de la heurística clásica con búsqueda local adaptativa. En la Figura 6, tenemos el diagrama de flujo que muestra el método propuesto.

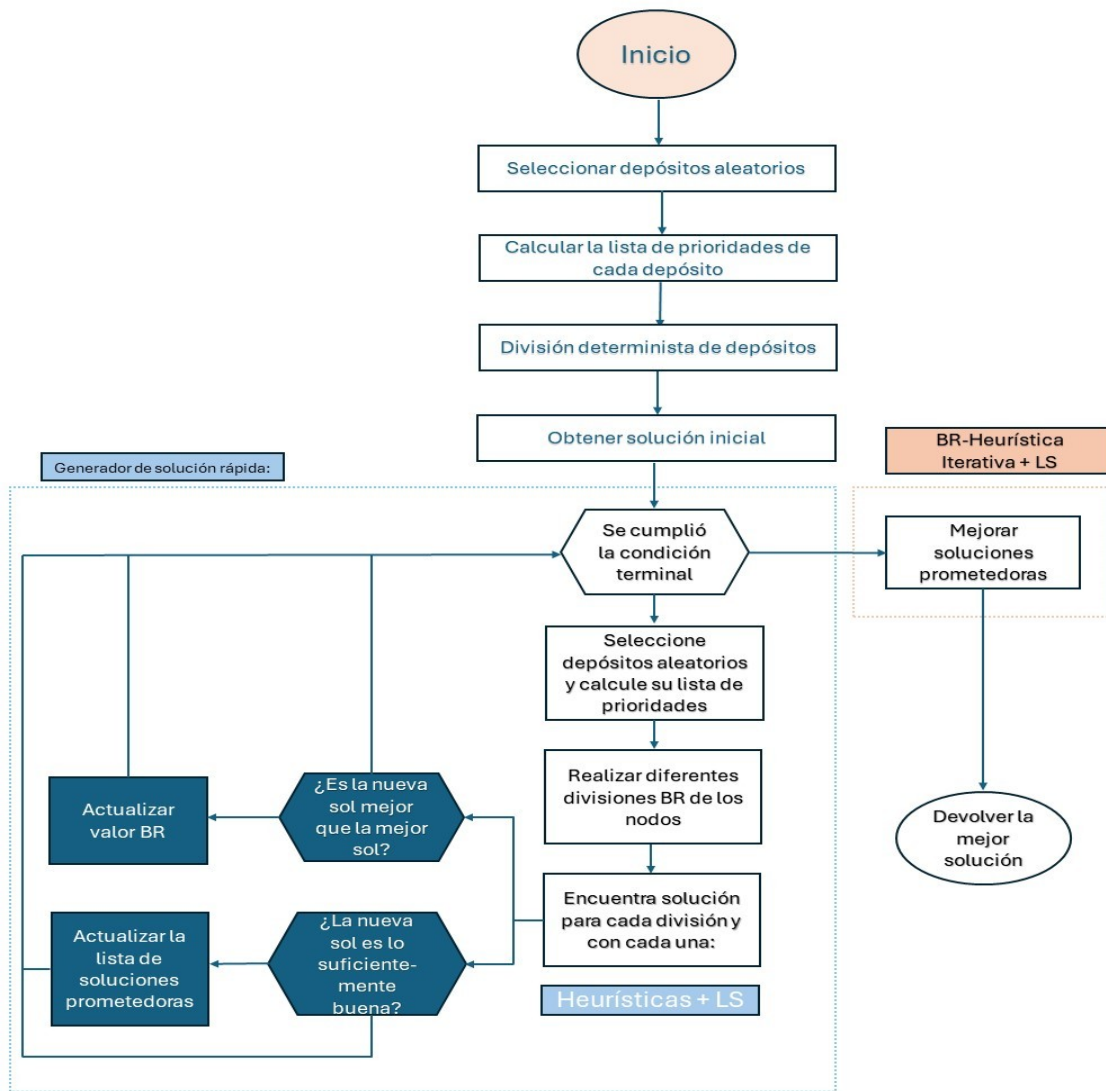


Figura 6: Diagrama de flujo del método propuesto.

Se tiene en cuenta dos fases, primero la asignación de clientes a cada uno de los depósitos y hallar dicha ruta a través de técnicas de ruteo y luego en la segunda fase se mejora la ruta hallada anteriormente:

- Fase 1: En esta fase se busca la asignación de cada cliente a los depósitos existentes, teniendo en cuenta la distancia marginal entre un nodo y un depósito, que se define como la diferencia entre la distancia del nodo al depósito y la distancia del nodo al depósito más cercano entre el resto de depósitos. Matemáticamente, dado un conjunto de depósitos  $\mathcal{D}$  y un conjunto de nodos  $\mathcal{N}$ , la distancia marginal entre un nodo  $N^*$  y un depósito  $D^*$  puede escribirse como:

$$md(N^*, D^*) = d(N^*, D^*) - \min_{D \in \mathcal{D}} \{d(N^*, D)\}$$

donde  $d$  representa una distancia elegida.

Luego para cada depósito, un nodo con una distancia marginal más baja tiene mayor probabilidad de ser seleccionado y una vez calculadas todas las distancias marginales se crea una lista de nodos ordenados por su distancia marginal para cada depósito. Entonces de forma aleatoria y sesgada, utilizando una distribución geométrica, que está determinada por un parámetro  $\beta$  se asigna un nodo a cada depósito (Ver Figura 7). El nivel relativo de avaricia en el comportamiento aleatorio del algoritmo está controlado por el parámetro  $\beta$  ( $0 < \beta < 1$ ) que sirve como valor umbral para construir la solución. El valor de  $\beta$  influye en la probabilidad de que se seleccione un nodo de la EfficiencyList.

Un valor de  $\beta$  más cercano a 0 favorece una selección aleatoria prácticamente uniforme, mientras que un valor más cercano a 1 favorece una distribución según el comportamiento codicioso de la heurística. Sin embargo, debe evitarse un valor de  $\beta = 0$ , ya que esto eliminaría el comportamiento codicioso por completo.

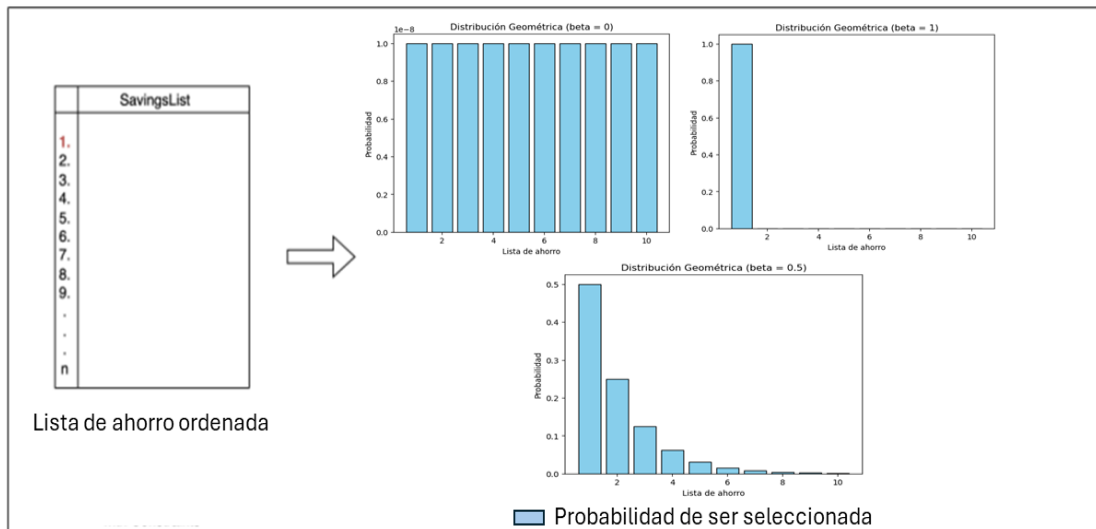


Figura 7: Ilustración de la asignación sesgada de cada nodo a los depósitos.

El algoritmo 1 explica la función de elegir el valor de  $\beta$  y determinar el índice de la Lista Eficiente para la construcción. La función logarítmica de los números aleatorios garantiza que se favorezcan más los valores pequeños que los grandes, mientras que el cálculo de  $\beta$  determina el sesgo implícito a la función logarítmica.

---

**Input:**  $\beta_{min}, \beta_{max}$

**Output:** La lista de ahorro ordenada de forma aleatoria y sesgada.

- 1: Calcular  $\beta \rightarrow \beta_{min} + \log(rand()) * (\beta_{max} - \beta_{min})$ .
  - 2: Calcular la BR-Posición  $\rightarrow Int(trunc(\log(rand())/log(1 - \beta)))$ .
  - 3: Asegurarse de que la posición BR es válida  $\rightarrow index = index \% range$  .
  - 4: **end**
- 

**Algorithm 1:** Algoritmo de selección aleatoria

Después de que un nodo se asigna a un depósito, se elimina de todas las listas y el método continúa con el siguiente depósito. Los depósitos giran para distribuir los nodos de manera uniforme. Éste genera rutas compensadas, ya que todos los depósitos contendrán aproximadamente el mismo número de nodos, lo que garantiza una asignación razonable. Gracias a la aleatorización de la división, este proceso permite el cálculo de 10 soluciones diferentes para la misma combinación de depósitos para evaluar mejor si la selección del depósito puede proporcionar una buena solución. Una vez realizada la división, se emplea el Algoritmo del Vecino más Cercano (NN) y una búsqueda local rápida y superficial, que encuentra una solución para el TSP compuesto por el depósito y los nodos asignados al mismo y una vez seleccionado una ruta para el vendedor que sale de cada depósito, se calcula el costo total como la suma de los costes de cada ruta. Luego este proceso se realiza para cada uno de los 10 soluciones proporcionadas y se selecciona la mejor solución entre las generadas, con el fin de ofrecer más variabilidad a la segunda etapa.

- Fase 2: En la segunda etapa, se emplea un procedimiento de mejora para refinar las soluciones obtenidas en la primera fase. En este método, se aplica una búsqueda local más exhaustiva de la solución, seguida de la aplicación iterativa de un sistema aleatorio de sesgo, versión del algoritmo NN, siguiendo el mismo proceso de destrucción-construcción explicado anteriormente. Esto permite el cálculo de varias soluciones manteniendo las mismas división de nodos como la solución prometedora. Una vez que el procedimiento de mejora se aplica a todos los soluciones prometedoras, se selecciona la mejor solución entre todas las obtenidas (Escoto, Guerrero y Medina 2024).

**Input:** Coordenadas de los nodos, Número de Depósitos, máximo de iteraciones, TIME, TIME2, alpha, nodos vip

**Output:** Un conjunto  $P_{final}$  que aproxima el frente óptimo de Pareto para cada instancia.

**Function main**( $L$ ,  $depot$ , 100,  $TIME1$ ,  $TIME2$ ,  $alpha$ ,  $vip - nodes$ ):

- 1: Seleccionar depósitos y eliminarlos de los nodos iniciales.

**for** *each nodo in Node* **do**

- 2: Se encuentra la distancia mínima a los otros depósitos y se resta la distancia al depósito principal.
- 3: Esta diferencia se ordena de forma descendente.

**end**

**return** una lista de prioridades para cada depósito.

- 4: Se asignan los nodos a diferentes depósitos por la lista de prioridades, utilizando un enfoque de reparto en turnos de manera uniforme.
- 5: Encontrar una solución sencilla con el NN-algoritmo y la búsqueda local.

**Function nn-algoritmo:**

- 6: Seleccionar nodo inicial.

**while** *!isempty(vip-nodesCopy)* **do**

- 7: Se selecciona el nodo VIP más cercano al nodo actual.
- 8: Se actualiza la ruta (Tour), se elimina el nodo VIP de la lista y se actualiza el punto actual.

**end**

- 9: Se visitan los nodos restantes similar al bucle anterior.

**return** Tour, cost.

**Function búsqueda local**( $Tour$ ,  $cost$ ):

**while**  $maxNoImprove > count$  **do**

- 10: Se genera una nueva solución a partir de Tour usando la técnica de "2-opt estocástico".
- 11: Se calcula el costo de la nueva solución.
- 12: Si la nueva solución es mejor, se actualiza la solución y el costo actuales y se reinicia el contador.
- 13: Si la nueva solución no es mejor, se incrementa el contador.

**end**

**while**  $time() - start < TIME$  **do**

- 6: Inicializar nodos y depósitos.
- 7: Asignar depósitos de forma aleatoria y eliminarlos de los nodos iniciales.
- 8: Se encuentra la distancia marginal y se ordena de forma descendente.
- 9: Con una distribución Geométrica se asigna un nodo a cada depósito.

**for**  $j$  *in* 1:10 **do**

- 10: Elimina un porcentaje  $p$  de nodos de cada depósito de manera aleatoria.
- 11: Genera una lista de prioridades para los nodos eliminados basada en su proximidad a los depósitos.
- 12: Utiliza un algoritmo de reparto en turnos (br-round-robin) para reasignar los nodos eliminados a los depósitos.
- 13: Devuelve una nueva asignación de nodos a depósitos (NewDivision).
- 14: Encontrar solución y guardar la mejor encontrada.

**end**

**end**

**return** bestsol

**Algorithm 2:** Heurística aleatoria sesgada.

## 4. Experimentos Computacionales

El método se ha implementado utilizando Julia 1.10, y los experimentos fueron corridos en una computadora personal con Windows 11, con un i7-1165G7 CPU 2.80 GHz y 8 GB de RAM.

Nuestro algoritmo se ha probado en 23 instancias, usadas anteriormente por Escoto, Guerrero y Medina (2024). Cada instancia tiene cantidad de nodos diferentes, y a su vez cada nodo se caracteriza por dos coordenadas. La selección de los nodos VIP es siempre la misma y de los depósitos es aleatoria y se realiza al inicio del algoritmo. Contamos con diferentes cantidades de depósitos, entre 2 y 5, para así obtener más variabilidad en las soluciones.

El número de semillas se fija en 10, y sólo se almacena el mejor resultado (nótese que estas ejecuciones pueden realizarse en paralelo, por lo que el responsable de la toma de decisiones está interesado principalmente en la mejor solución). Este enfoque permite una comparación directa y consistente de las mejores soluciones encontradas. El uso de semillas en todas las instancias garantiza que se genere la misma secuencia de números aleatorios en cada ejecución. La comprobación de múltiples semillas evita la posibilidad de que el algoritmo se atasque en una semilla concreta.

Tenemos en cuenta un parámetro  $\alpha$  que controla que función objetivo tiene mayor prioridad. Hemos probado el algoritmo para valores de  $\alpha$  entre 0 y 1, con una diferencia de 0,1 entre cada valor. La primera experimentación que hicimos consiste en aplicar nuestro algoritmo para diferentes valores de  $\alpha$ . Esto permitió analizar cómo los distintos valores de  $\alpha$  influyen en la calidad y características de las rutas generadas. En cada configuración de  $\alpha$ , se ejecutaron múltiples pruebas para obtener un conjunto diverso de rutas y valores de la función objetivo.

Luego para las segundas experimentaciones fijamos el número de nodos VIP que queremos visitar al inicio del recorrido, aplicamos la heurística sesgada y guardamos la mejor de las soluciones y la media. Este proceso se realizó para varias instancias y diferentes números de depósitos con el fin de evaluar el rendimiento y la robustez de la heurística en diferentes escenarios.



## 5. Análisis y Resultados

Si analizamos los siguientes resultados obtenidos podemos comprobar que un aspecto esencial de este algoritmo es su capacidad para equilibrar las soluciones generadas. Una vez que se ha creado una solución, todas las rutas contienen aproximadamente el mismo número de nodos.

Otro aspecto importante a destacar es que en los problemas multiobjetivos dependiendo del valor asignado al parámetro  $\alpha$ , se puede priorizar una función objetivo sobre otra. En particular,  $\alpha$  actúa como un factor de ponderación que ajusta la importancia relativa de cada objetivo, permitiendo que el enfoque se incline hacia la minimización o maximización de una función específica según las necesidades del problema. Al modificar  $\alpha$ , se puede equilibrar el compromiso entre los diferentes objetivos, obteniendo soluciones que reflejen mejor las preferencias y prioridades del decisor.

Se presenta una solución de prueba visualizada como un mapa que contiene de 2 a 5 depósitos. En las Figuras 8, 9, 10, 11 tomamos un valor de  $\alpha = 0$ , es decir solamente tuvimos en cuenta minimizar la distancia total recorrida, en cambio en las Figuras 12, 13, 14, 15, tomamos  $\alpha = 1$ , dándole así mayor peso a la función objetivo que maximiza la cantidad de nodos vip visitados al inicio del recorrido.

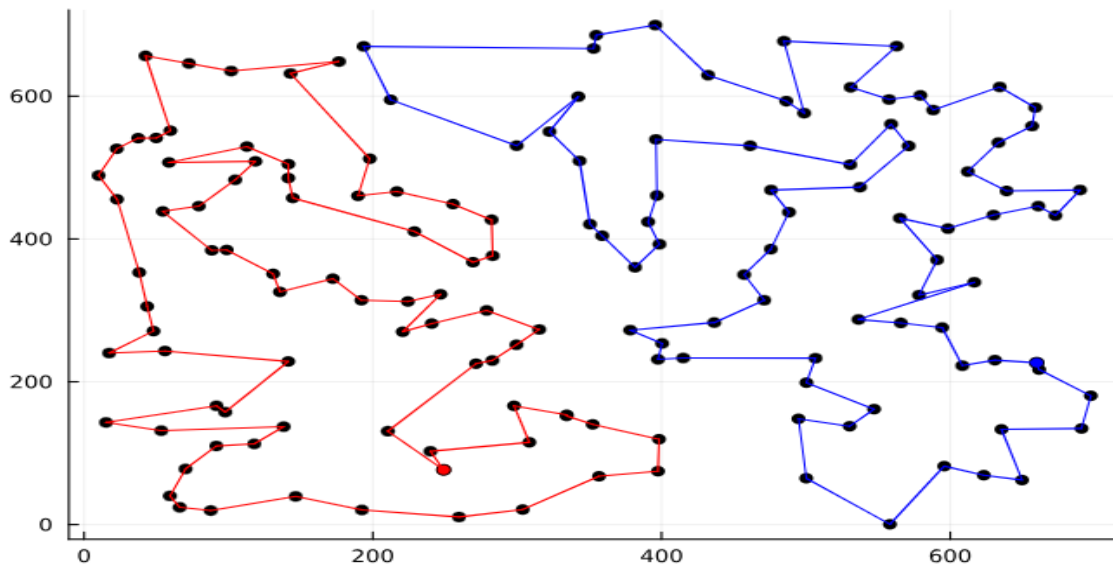


Figura 8: Soluciones encontradas con 2 depósitos para  $\alpha = 0$ .

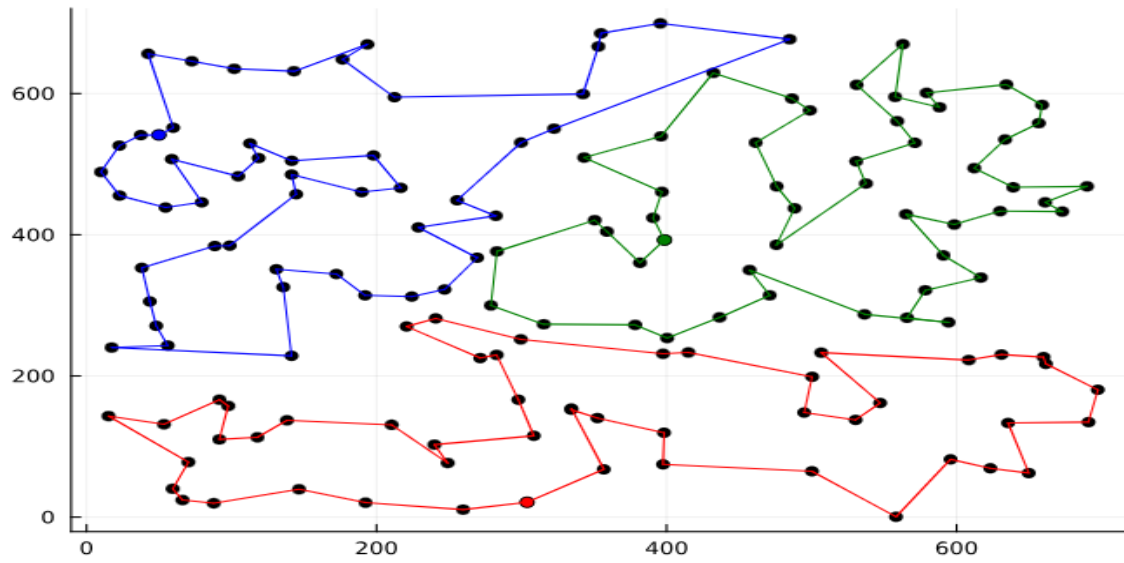


Figura 9: Soluciones encontradas con 3 depósitos para  $\alpha = 0$ .

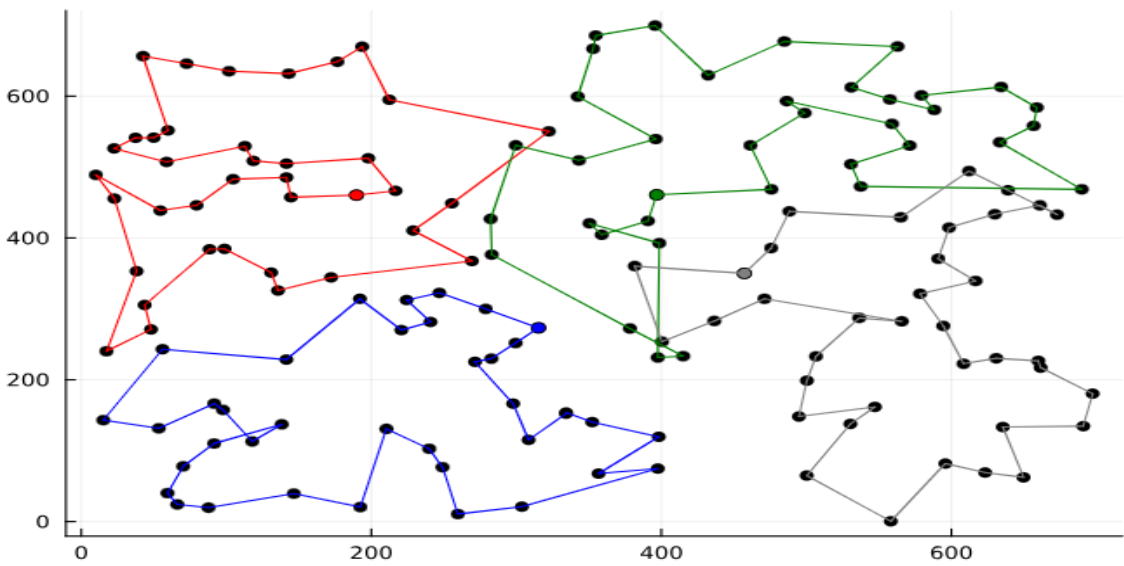


Figura 10: Soluciones encontradas con 4 depósitos para  $\alpha = 0$ .

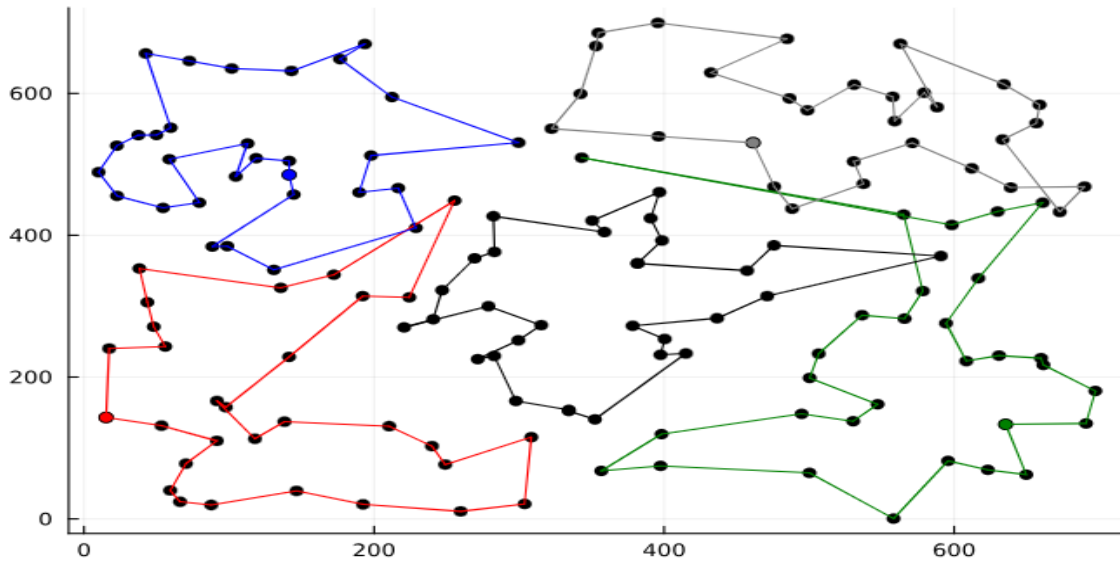


Figura 11: Soluciones encontradas con 5 depósitos para  $\alpha = 0$ .

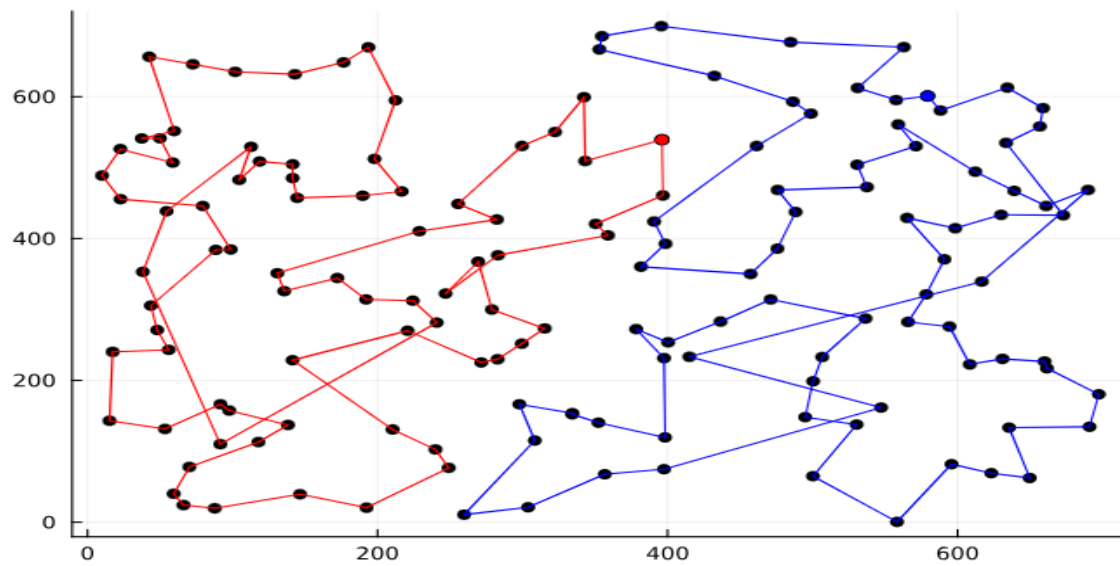


Figura 12: Soluciones encontradas con 2 depósitos para  $\alpha = 1$ .

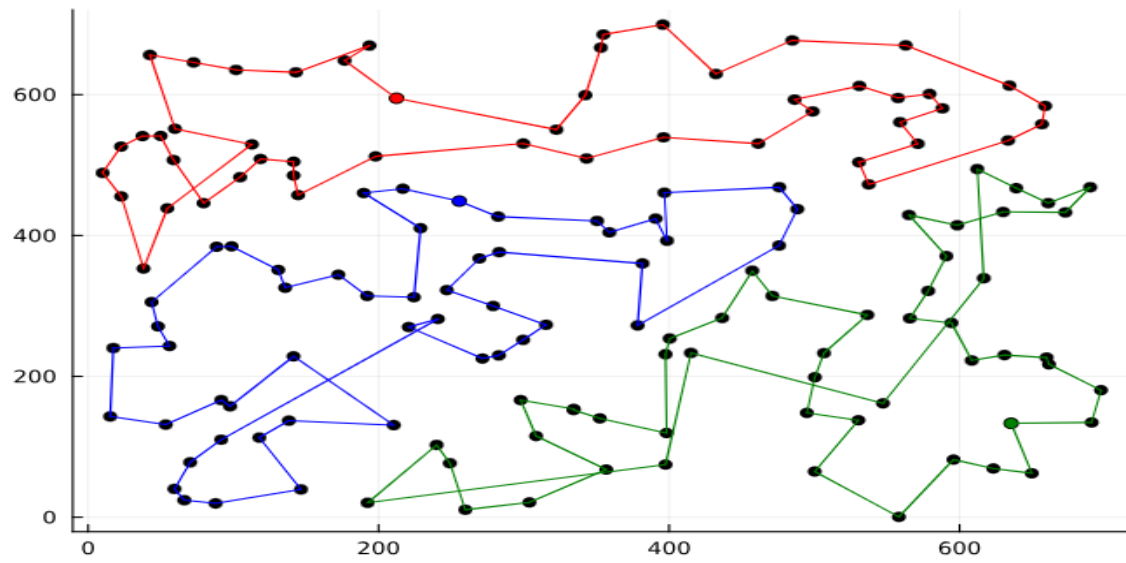


Figura 13: Soluciones encontradas con 3 depósitos para  $\alpha = 1$ .

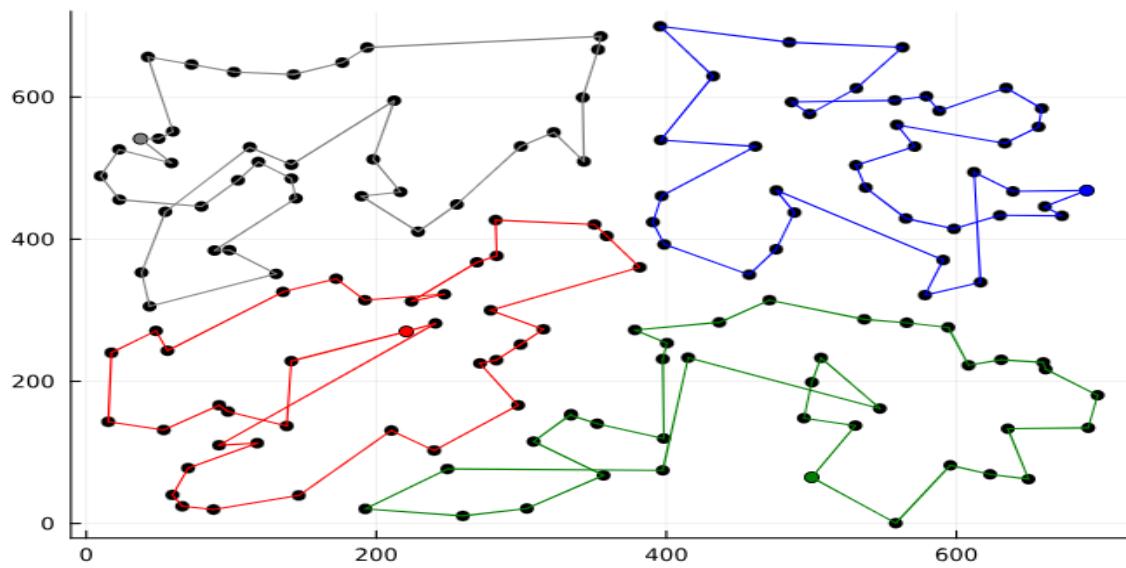


Figura 14: Soluciones encontradas con 4 depósitos para  $\alpha = 1$ .

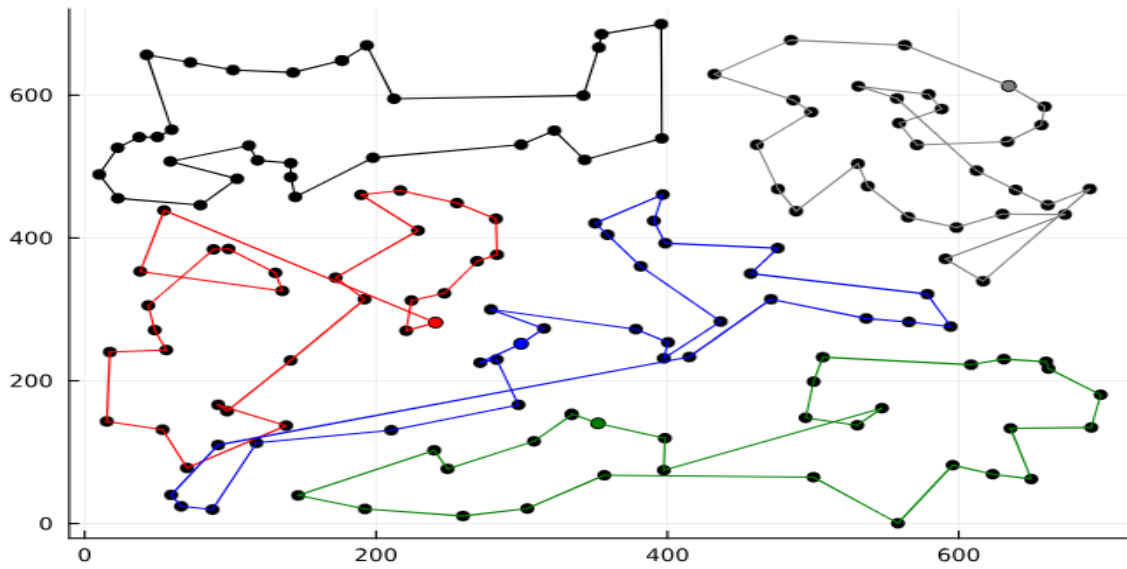


Figura 15: Soluciones encontradas con 5 depósitos para  $\alpha = 1$ .

En las gráficas correspondientes a  $\alpha = 0$ , se puede observar que las rutas están claramente optimizadas. Esto se debe a que, con este valor de  $\alpha$ , el algoritmo prioriza la minimización de la distancia total recorrida, resultando en rutas más directas y eficientes. En estas gráficas, las rutas no se cruzan, lo que indica una planificación efectiva y una menor probabilidad de solapamiento entre las diferentes trayectorias de los vehículos.

Por otro lado, en las gráficas con valores de  $\alpha = 1$ , las rutas tienden a entrelazarse. Esto ocurre porque el parámetro  $\alpha$  ajusta la importancia relativa de los diferentes objetivos en el problema. Cuando  $\alpha$  toma valores distintos de cero, el algoritmo podría estar priorizando otros objetivos, como la maximización de la cantidad de nodos vip visitados, en lugar de la minimización de la distancia total. Como resultado, las rutas se vuelven menos directas y más propensas a cruzarse, lo que refleja una menor eficiencia en la planificación de las trayectorias.

En la Tabla 2 y 3 se compara el costo de cada una de estas soluciones para la instancia ch150. En esta prueba, el problema se resuelve para cada número de depósitos. A medida que aumentan los depósitos, la variedad de las posibles soluciones aumentan exponencialmente y se dedica el mismo tiempo para mejorar la solución de cada sub-ruta una vez realizada la división. Además también se está forzando a que todos los vehículos recorran la misma cantidad de nodos. Por tanto se observa un aumento general en el costo a medida que se incrementa el número de depósitos. También comprobamos que al priorizar el aumento de nodos VIP visitados al inicio del tour, la distancia total recorrida es mayor.

Instance	2 depots	3 depots	4 depots	5 depots
ch150	687.78	696.87	720.34	786.56

Cuadro 2: Costos de las soluciones mostradas en las figuras 8, 9, 10, 11.

Instance	2 depots	3 depots	4 depots	5 depots
ch150	723.44	762.80	789.32	815.37

Cuadro 3: Costos de las soluciones mostradas en las figuras 12, 13, 14, 15.

Es interesante analizar el impacto de la variable  $\alpha$  respecto a las funciones objetivo para cada instancia. A continuación en las Figuras 16 y 17 podemos ver los valores de  $\alpha$  en el intervalo  $[0, 1]$ , con una diferencia de 0,2 para la instancia st70.

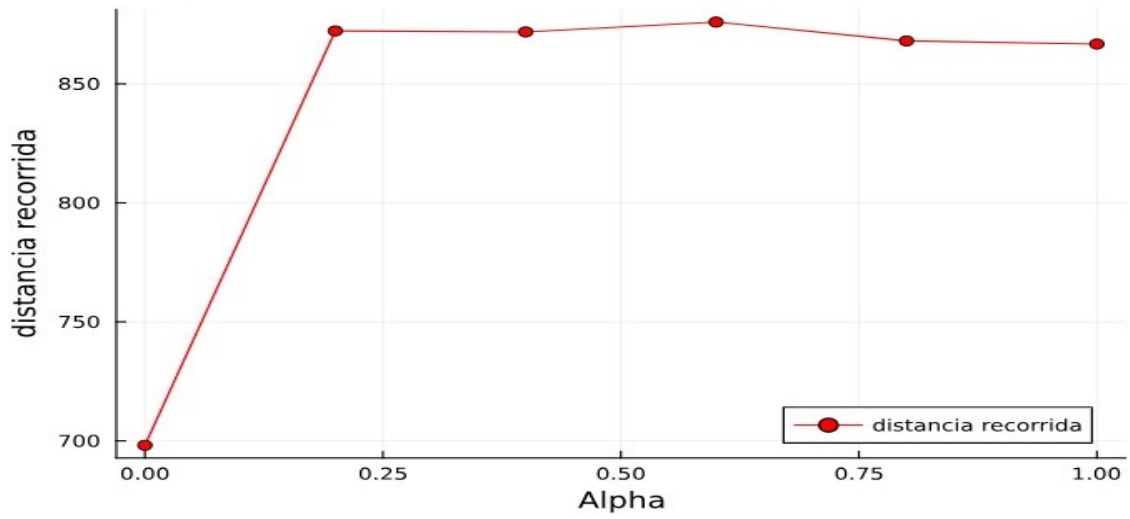


Figura 16: Impacto de  $\alpha$  en la distancia total recorrida, instancia st70.

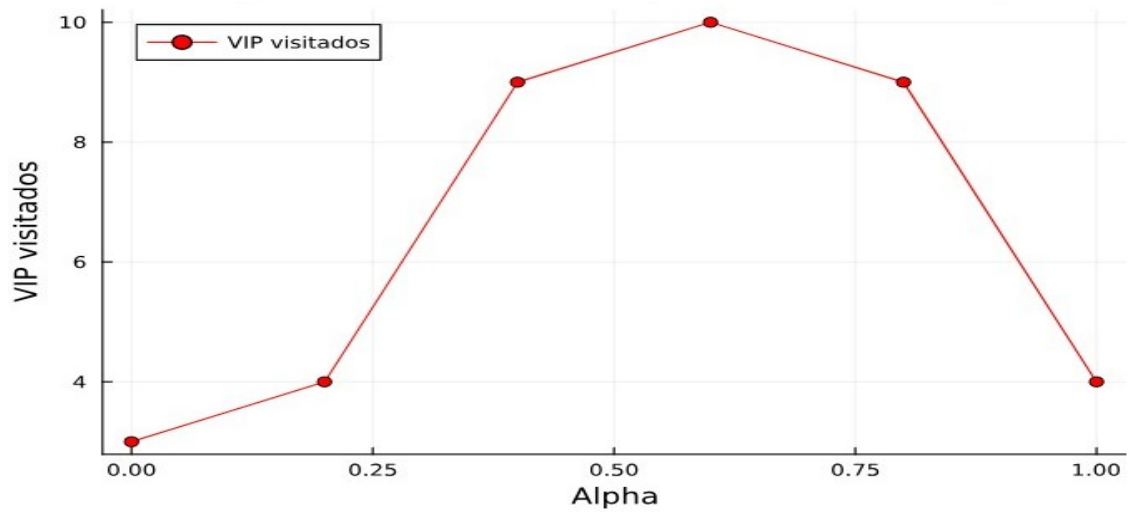


Figura 17: Impacto de  $\alpha$  en la cantidad de nodos VIP visitados, instancia st70.

En la instancia st70 podemos notar que los valores inferiores de  $\alpha$  dan lugar a recorridos de menor distancia, pero a su vez disminuye también la visita de nodos VIP al inicio.

El efecto de  $\alpha$  en la instancia st70 indica que la mayor cantidad de nodos VIP visitados se obtienen cuando  $\alpha$  está entre 0,40 y 0,80, lo que sugiere que este rango puede ser ideal para maximizar esta función objetivo. Obteniendo un valor de nodos VIP visitados entre 8 y 10 y que en ese mismo rango se obtiene la máxima de costes por kilómetros recorridos, superando los 850.

Si analizamos las Figuras 18 y 19 podemos deducir que en la instancia eil101 el parámetro  $\alpha$  se comporta de forma similar entre los valores de 0,20 y 0,60 alcanza la mayor distancia recorrida, con un coste de transportación superior a 800. A su vez la mayor cantidad de nodos VIP al inicio del recorrido, se encuentran en ese mismo intervalo de valores de  $\alpha$ . Obteniendo un valor de 7 nodos VIP visitados.

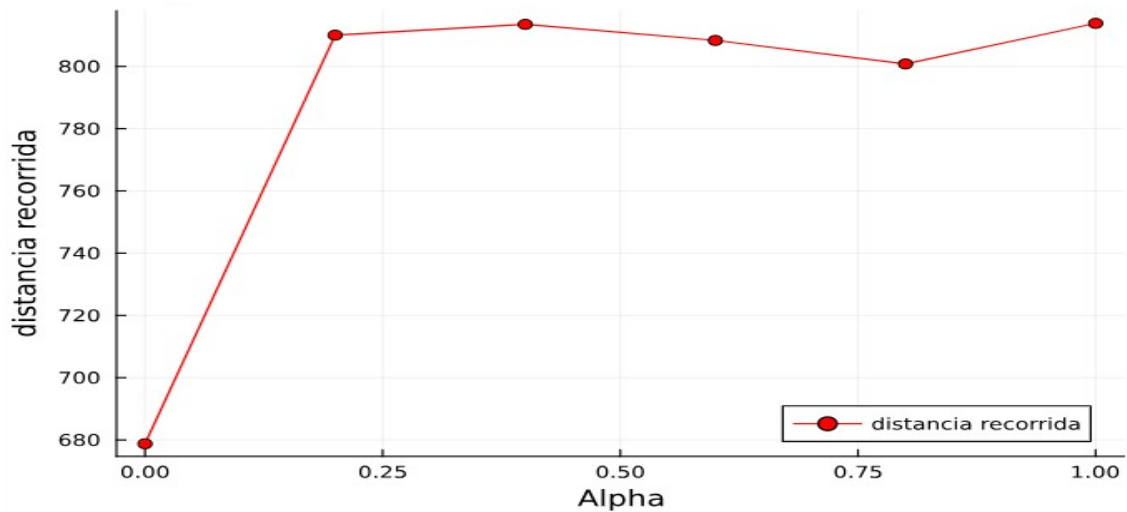


Figura 18: Impacto de  $\alpha$  en la distancia total recorrida, instancia eil101.



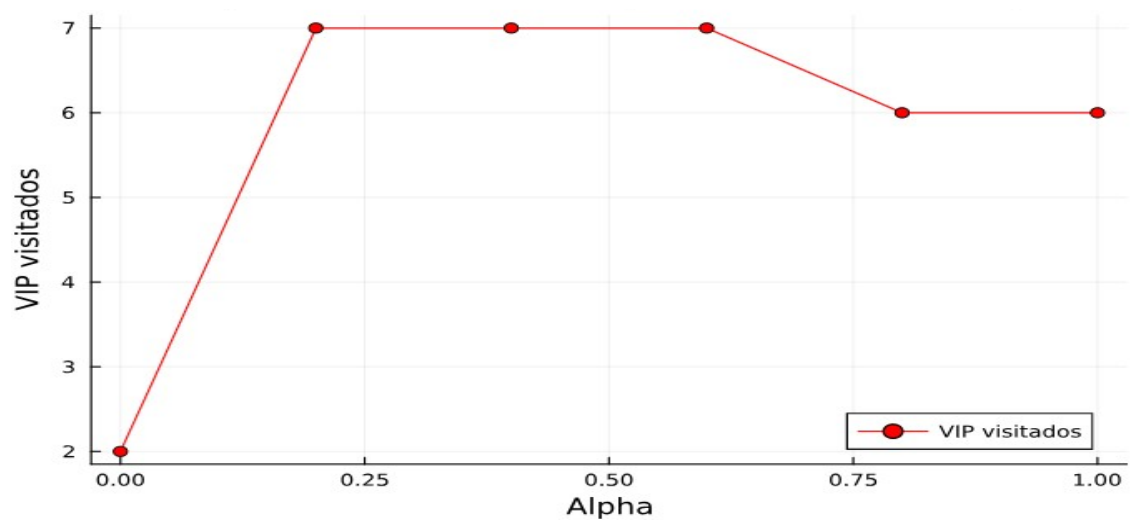


Figura 19: Impacto de  $\alpha$  en la cantidad de nodos VIP visitados, instancia eil101.

Para la siguiente tabla 4, tuvimos en cuenta 10 semillas. Para cada número de nodo VIP visitado al inicio de la ruta, obtuvimos la mejor solución y la media entre todas las soluciones obtenidas.

Si analizamos los datos, notamos que mientras más nodos VIP se visiten al inicio del recorrido es mayor la distancia total recorrida. Al igual que mientras más depósitos se tengan en cuenta, más km se recorren.

Cuadro 4: Costos de transportación para nodos VIP y depósitos.

Instancia st70.tsp								
	2 Depósitos		3 Depósitos		4 Depósito		5 Depósito	
	best sol	media	best sol	media	best sol	media	best sol	media
1 VIP	691.09	691.46	768.34	769.15	743.34	754.00	771.19	773.43
2 VIP	731.77	741.17	712.66	718.48	756.54	765.58	789.11	800.23
3 VIP	746.70	749.15	742.02	749.03	791.31	789.59	864.27	864.27
4 VIP	855.15	862.22	837.89	840.78	849.28	877.39	883.54	884.30
5 VIP	871.14	877.39	844.35	844.35	849.38	850.45	879.67	880.26

En las figuras siguientes veremos el Frente de Pareto obtenido para la instancia st70 y diferentes números de depósitos.

El Frente de Pareto representado por la línea azul muestra cómo al aumentar los kilómetros recorridos se puede maximizar el número de nodos VIP visitados. Estas soluciones son las más eficientes y deben ser consideradas para la toma de decisiones.

En un eje se representa la distancia total recorrida, medida en kilómetros. Los valores oscilan aproximadamente entre 700 y 875 kilómetros. En el otro eje se muestra la cantidad de nodos VIP visitados. Se guardaron las mejores soluciones encontradas fijando el número de nodos VIP que queremos visitar al inicio del recorrido, oscilando este valor entre 1 y 13 nodos VIP.

El incremento más significativo en los nodos VIP visitados ocurre alrededor de los 750 a 850 kilómetros, con un salto notable en los nodos visitados de aproximadamente 4 a 8. También podemos notar que hay soluciones que están por debajo del Frente de Pareto. Esto indica que, para ciertos valores de kilómetros recorridos, hay configuraciones donde no se visitan tantos nodos VIP como en las soluciones del frente óptimo. Estos puntos representan combinaciones subóptimas en las que se recorren distancias similares a las del Frente de Pareto, pero con un menor número de nodos VIP visitados.

El punto más eficiente en términos de kilómetros recorridos es alrededor de 725 km con 1 nodo VIP visitado y el punto más extremo del Frente de Pareto muestra que con unos 875 kilómetros recorridos, se visitan 13 nodos VIP. El resto de las figuras se comportan de forma similar, presentando una relación casi lineal entre las dos funciones objetivo.

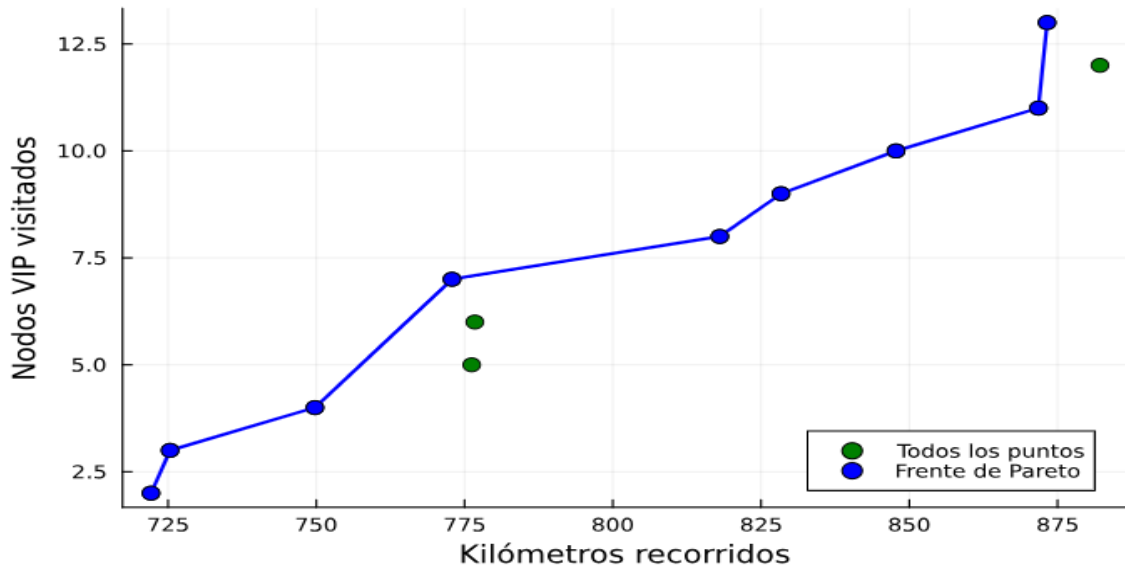


Figura 20: Frente de Pareto para Instancia st70 y 2 depósitos.

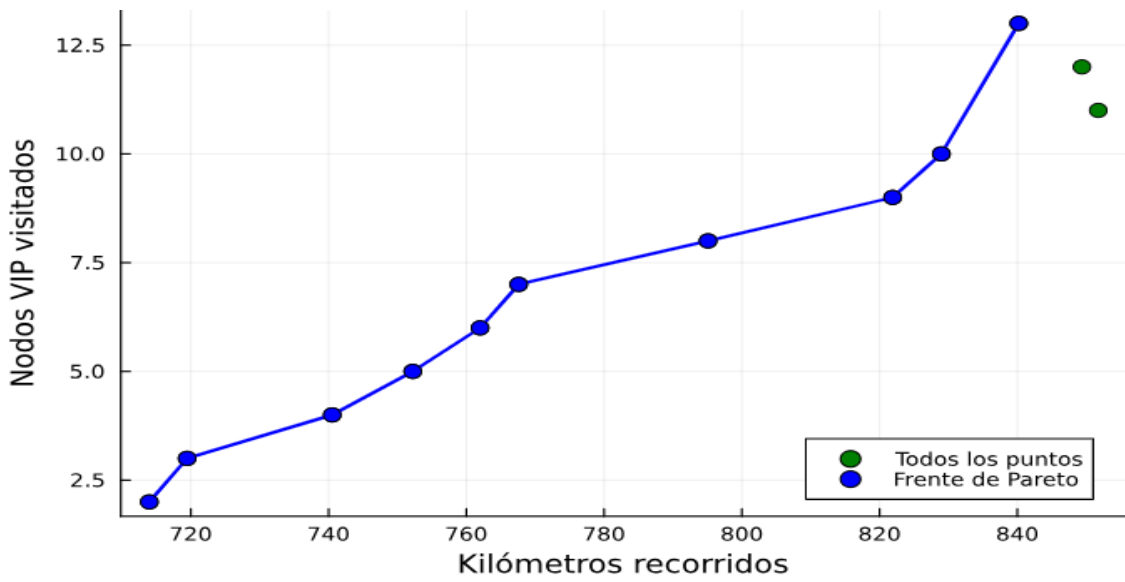


Figura 21: Frente de Pareto para Instancia st70 y 3 depósitos.

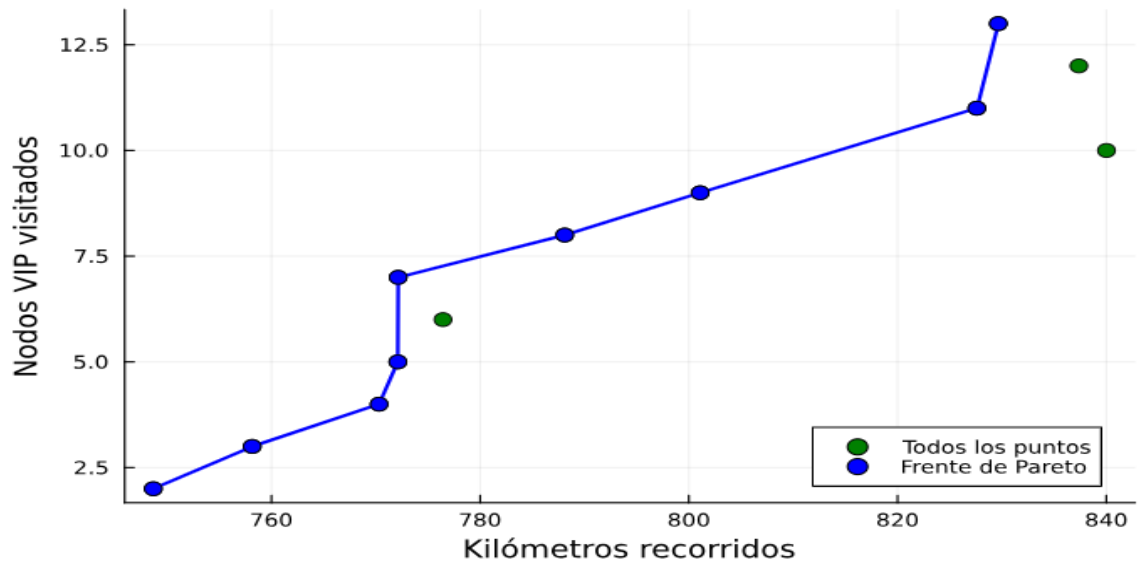


Figura 22: Frente de Pareto para Instancia st70 y 4 depósitos.

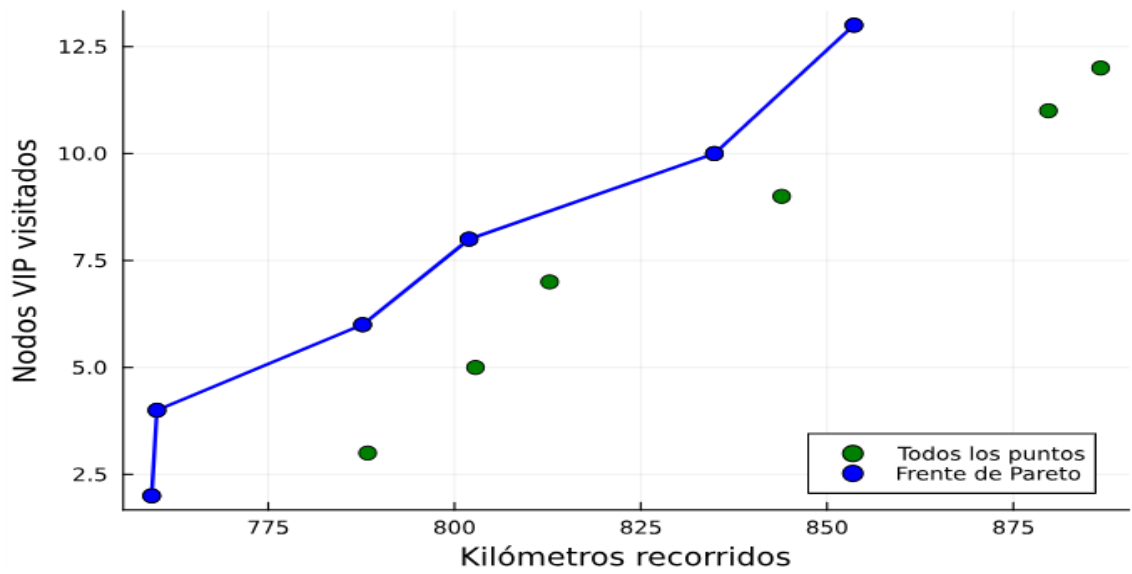


Figura 23: Frente de Pareto para Instancia st70 y 5 depósitos.

Cuando analizamos el comportamiento de las funciones objetivos en la instancia eil101, obtenemos los siguientes resultados.

Cuadro 5: Costos de transportación para nodos VIP y depósitos.

Instancia eil101.tsp								
	2 Depósitos		3 Depósitos		4 Depósito		5 Depósito	
	best sol	media	best sol	media	best sol	media	best sol	media
1 VIP	681.56	684.38	742.70	742.89	691.49	692.67	732.44	734.67
2 VIP	688.78	702.67	743.08	745.80	724.74	724.89	723.85	725.67
3 VIP	723.90	725.78	766.37	766.37	804.33	806.89	779.98	805.78
4 VIP	710.75	711.68	850.67	851.54	793.79	795.78	814.56	814.58
5 VIP	761.94	763,67	851.43	854.58	855.78	860.47	862.65	863.43

Luego en las figuras 24, 25, 26, 27 veremos el Frente de Pareto obtenido. Estas gráficas indican que hay un compromiso (trade-off) entre los kilómetros recorridos y los nodos VIP visitados que para maximizar el número de nodos VIP visitados, se requiere recorrer más kilómetros. En un eje los valores oscilan aproximadamente entre 600 y 820 kilómetros y en el otro eje se muestra la cantidad de nodos VIP visitados, oscilando este valor entre 1 y 13 nodos VIP.

En la Figura 24 la solución más eficiente en términos de kilómetros recorridos visita 1 nodo VIP con un recorrido cercano a los 690 kilómetros. En cambio la solución más eficiente en términos de maximizar los nodos VIP visitados alcanza los 13 nodos VIP, pero requiere recorrer aproximadamente 820 kilómetros. Entre estos extremos, hay una relación casi lineal donde aumentar los nodos VIP visitados implica recorrer más kilómetros. Podemos observar que hay puntos en la solución que no conforman el Frente de Pareto y mientras más se alejen de este, menos eficientes son. Como en la Figura 26 que recorre casi 725 km para visitar solo 3 nodos VIP, en cambio en el frente de Pareto vemos que se visitan 5 nodos VIP recorriendo distancias similares o incluso menores. Las otras figuras se comportan de manera similar.

En algunos segmentos del Frente de Pareto, pequeños incrementos en la distancia recorrida resultan en un aumento significativo en los nodos VIP visitados. Esto sugiere que en esas zonas específicas, la eficiencia del recorrido es alta. Por otro lado, en otros segmentos del Frente de Pareto, se observa que para aumentar un nodo VIP adicional, la distancia recorrida incrementa considerablemente, lo cual indica una menor eficiencia en esas regiones. Si analizamos la Figura 24, podemos ver que en los extremos de la gráfica la eficiencia del recorrido es alta, en cambio en el centro, entre 5 y 8 nodos VIP hay un incremento notable en las distancias recorridas.

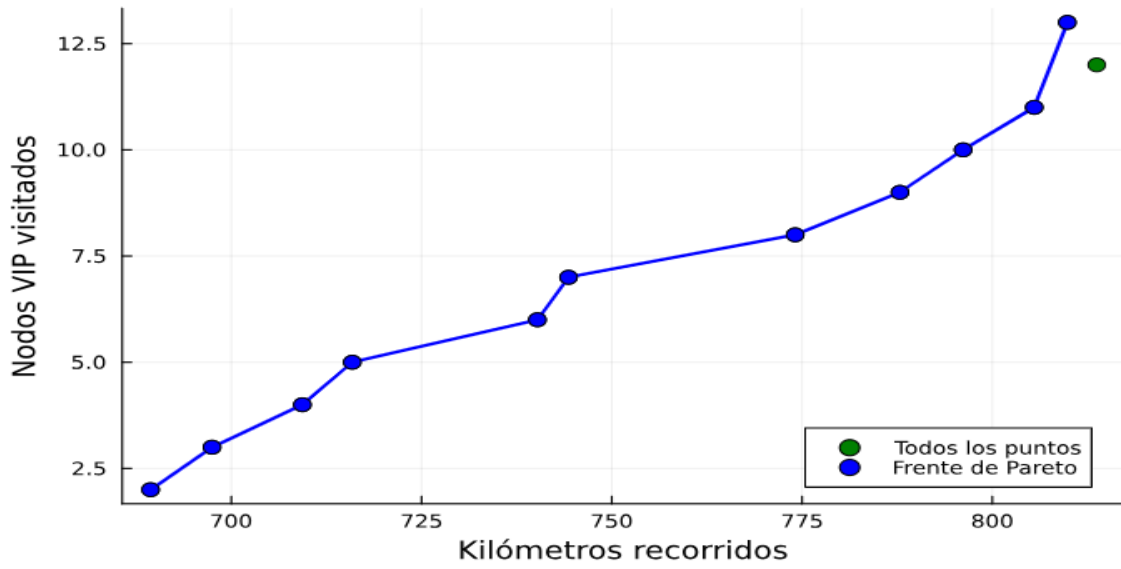


Figura 24: Frente de Pareto para Instancia eil101 y 2 depósitos.

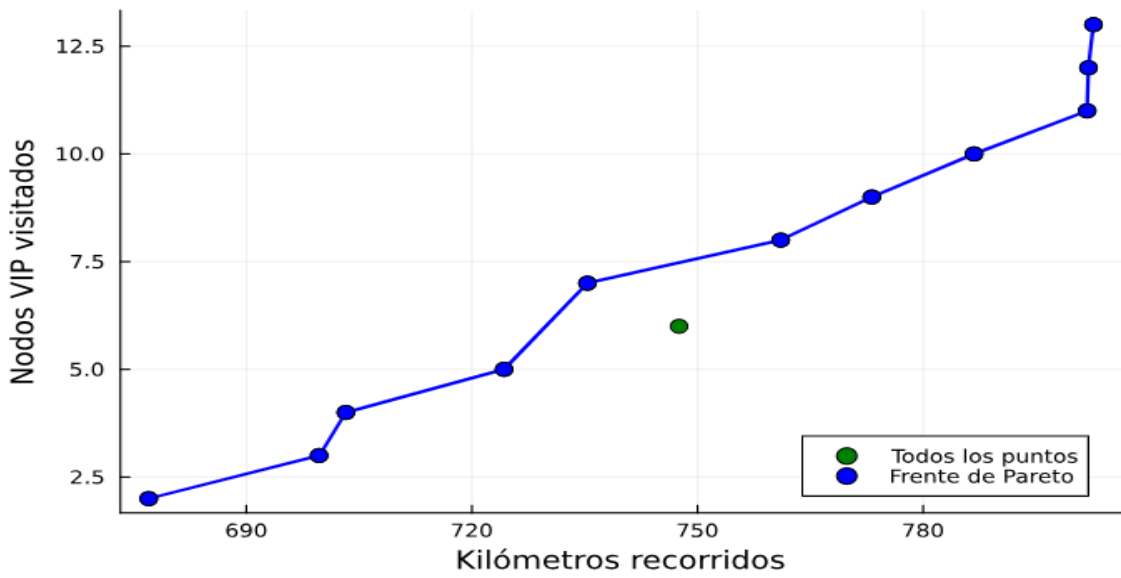


Figura 25: Frente de Pareto para Instancia eil101 y 3 depósitos.

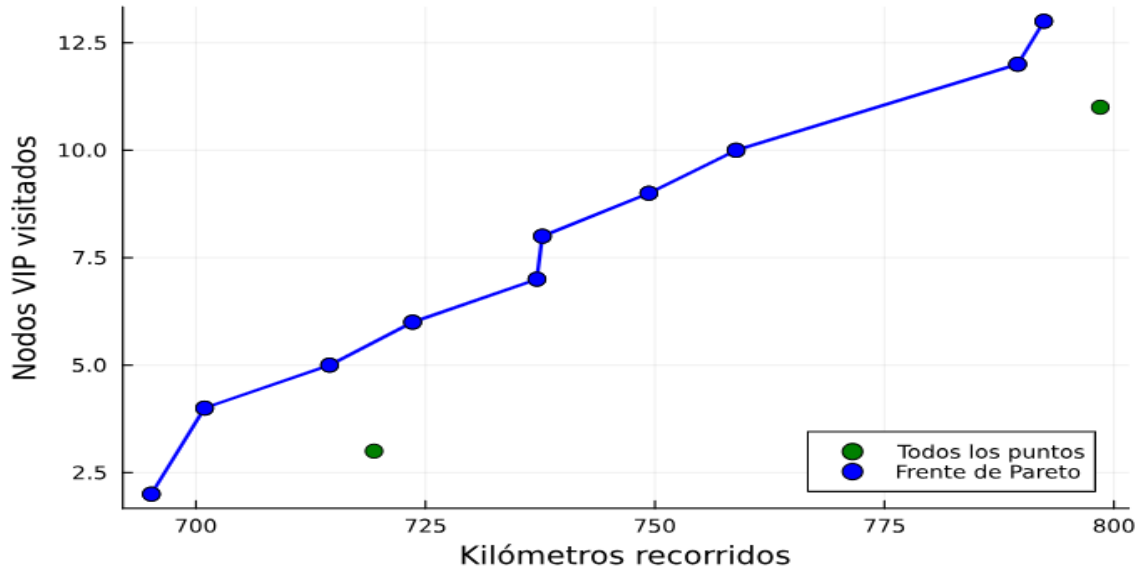


Figura 26: Frente de Pareto para Instancia eil101 y 4 depósitos.

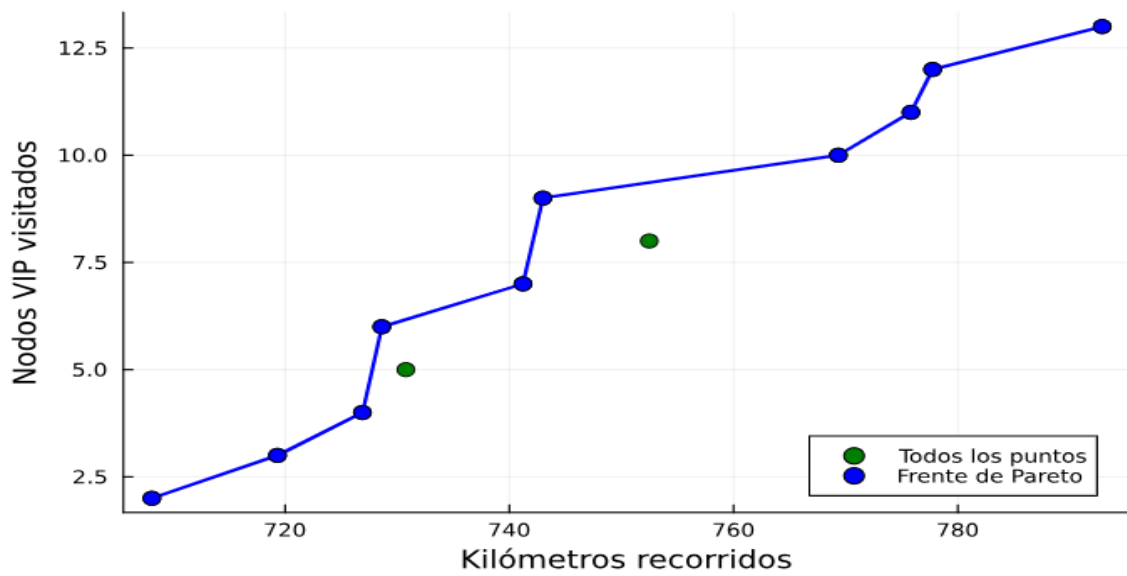


Figura 27: Frente de Pareto para Instancia eil101 y 5 depósitos.

## 6. Conclusiones

En este trabajo se propone un modelo multiobjetivo donde se minimiza la distancia total recorrida y a su vez se maximiza la cantidad de nodos VIP visitados al inicio del recorrido. Con las soluciones obtenidas generamos el frente de Pareto, el cual nos permite visualizar el compromiso entre ambos objetivos. Este frente de Pareto es una herramienta crucial para la toma de decisiones, ya que ofrece un conjunto de soluciones no dominadas, cada una representando un balance diferente entre los objetivos en conflicto. De esta manera, los decisores pueden elegir la solución que mejor se alinee con sus prioridades específicas, considerando tanto la eficiencia del recorrido como la importancia estratégica de los nodos VIP visitados al inicio.

Se introduce un enfoque simple y eficiente para el problema del vendedor viajero con múltiples depósitos y rutas cerradas, orientado a generar rutas equilibradas que permitan a varios agentes visitar todos los nodos de una instancia dada. El método propuesto es una heurística aleatoria y sesgada que se enfoca en la creación de soluciones equilibradas, donde, en la medida de lo posible, todas las rutas contienen un número similar de nodos. No obstante, este enfoque puede resultar en un aumento del costo total de las soluciones a medida que se incrementa el número de agentes involucrados.

En la primera parte de los experimentos se tiene en cuenta un parámetro  $\alpha$  que oscila entre 0 y 1 que nos prioriza una u otra función objetivo dependiendo de su valor y pudimos comprobar las variaciones en las rutas obtenidas y como esto influía en los costos de transportación. Tomamos los valores de  $\alpha$  en los extremos y cuando  $\alpha = 0$  el algoritmo prioriza la minimización de la distancia total recorrida, resultando en rutas más directas y eficientes, en cambio cuando  $\alpha = 1$  el algoritmo prioriza la maximización de la cantidad de nodos VIP visitados, entonces las rutas tienden a entrelazarse, obteniendo para 2 depósitos una diferencia entre las distancias recorridas aproximada de 36 km.

Realizamos un análisis de sensibilidad del modelo, en el que analizamos como influía la variable  $\alpha$  en las funciones objetivos para las instancias st70 y eil101 y obtuvimos que en los rangos de valores entre 0,20 y 0,80 se obtenían la mayor cantidad de nodos VIP visitados, así como la mayor distancia recorrida.

Luego en la segunda parte de la experimentación para diferentes nodos VIP, analizamos como influía en la segunda función objetivo y así obtuvimos el frente de Pareto. Obtuvimos un Frente de Pareto creciente, donde se nota una correlación lineal entre las funciones objetivo.

Mientras se prioricen más nodos VIP al inicio del recorrido, la distancia recorrida obtenida es mayor, por lo que muestra que nuestras funciones objetivos son antagónicas. Es decir, existe un claro conflicto entre ambos objetivos: al dar prioridad a la visita de más nodos VIP al inicio del recorrido, se incurre en un aumento de la distancia total recorrida. Este comportamiento refleja la naturaleza inversa de las funciones objetivo, donde mejorar un objetivo, como maximizar la cantidad de nodos VIP visitados al principio, inevitablemente lleva a un deterioro en el otro, como la minimización de la distancia total. Esto subraya la importancia del análisis del frente de Pareto para encontrar un equilibrio adecuado entre estas metas opuestas.

El enfoque aquí presentado abre múltiples líneas de investigación futuras. Es posible incorporar una variedad de restricciones adicionales para aumentar el realismo del problema, como la capa-



cidad de carga de los vehículos y las demandas de los nodos, convirtiendo el problema en una versión más compleja del problema de rutas de vehículos. Además, resulta crucial considerar las particularidades asociadas con la utilización de vehículos eléctricos, así como las implicaciones que esto conlleva. También se puede tener en cuenta agregar nuevas funciones objetivos. Otros aspectos a explorar incluyen la incorporación de incertidumbres y el dinamismo inherente a los escenarios reales, elementos que podrían influir significativamente en la viabilidad y eficiencia de las soluciones propuestas.

Una interesante dirección para futuros trabajos consiste en estudiar el parámetro  $\beta$  y de qué forma se obtiene el mejor  $\beta$ , que garantice buenas soluciones.

Otra línea de estudio que sería interesante seguir es como se podría extender el problema actual a un entorno estocástico. Esto implicaría considerar la incertidumbre en parámetros claves del modelo, como la demanda de los clientes, los tiempos de servicio, o las condiciones del tráfico, los cuales no son determinísticos en escenarios reales. Incorporar la estocasticidad podría lograrse mediante la formulación de modelos basados en programación estocástica o métodos de simulación, permitiendo evaluar y optimizar la robustez de las soluciones frente a la variabilidad de los datos de entrada. Esta extensión no solo enriquecería el modelo, sino que también haría las soluciones obtenidas más aplicables y resilientes en escenarios reales.

Sería interesante como método de resolución de nuestro problema crear un algoritmo Learnheurístico. Estos métodos son utilizados para resolver problemas de optimización combinatoria con entradas dinámicas, la combinación e hibridación entre metaheurísticas y aprendizaje automático es una herramienta prometedora para encontrar soluciones mejores y más rápidas para problemas combinatoriales.

## Referencias

- Zadeh, L. (1963). "Optimality and non-scalar-valued performance criteria." En: *IEEE Trans. Autom. Control*.
- Clarke, G. y J. W. Wright (1964). "Vehicle scheduling from a central depot to multiple delivery points." En: *Operations Research* 12.4.
- Eckenrode, R. (1965). "Weighting multiple criteria." En: *Manage.Sci.*
- Miettinen, Kaisa (1973). "Introducción a la optimización multiobjetivo: enfoques no interactivos." En: *Springer*.
- Gillet, B.E. y L.R. Miller (1974). "A Heuristic Algorithm for the Vehicle Dispatch Problem." En: *Operations Research* 22, págs. 340-349.
- Holland, J. (1975). "Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor." En.
- Hobbs, B. (1980). "A comparison of weighting methods in power plant siting." En: *Decis. Sci.*
- Fisher, M. y R. Jaikumar (1981). "A Generalized Assignment Heuristic for Vehicle Routing." En: *Networks.*, págs. 109-124.
- Koski, J. (1985). "Defectiveness of weighting method in multicriterion optimization of structures." En: *Commun. Appl. Numer. Methods* 1.
- Wierzbicki, A.P. (1986). "A methodological approach to comparing parametric characterizations of efficient solutions." En: *Berlin: Springer-Verlag*.
- Rao, J. y N. Roy (1989). "Fuzzy set theoretic approach of assigning weights to objectives in multicriteria decision making." En: *Int. J. Syst. Sci.*
- Goetschalckx, C. Jacobs Blecha ; M. (1992). "The Vehicle Routing Problem With Backhauls: Properties and Solution Algorithms." En: *Georgia Tech Research Corporation*.
- Stadler, W. (1995). "Caveats and boons of multicriteria optimization." En: *Microcomput. Civ. Eng.* 10.
- Athan, T. y P. Papalambro (1996). "A note on weighted criteria methods for compromise solutions in multi-objective optimization." En: *Eng. Optim.*
- Brandao, José y Alan Mercer (1997). "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem." En: *European Journal of Operational Research*. 100.1, págs. 180-191.
- Fleurent, C. y F. Glover (1999). "Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory." En: *INFORMS Journal on Computing.*, págs. 198-204.
- Gambardella, L., E. Taillard y G. Agazzi (1999). "MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows." En: *in edited by David Corne, Marco Dorigo, and Fred Glover, New Ideas in Optimization, McGraw-Hill: London*, págs. 63-76.
- Drezner, Z. y H. Hamacher (2002). "Facility Location: Applications and Theory." En: *Springer*.
- Toth, Paolo y Daniele Vigo (2002). "The Vehicle Routing Problem." En: *Society of Industrial and Applied Mathematics (SIAM) monographs on discrete mathematics and applications. Philadelphia, USA*. 16.2, págs. 1-23.
- Marler, R. y J. Arora (2004). "Survey of multi-objective optimization methods for engineering." En: *Struct Multidisc Optim.*
- Funke, B, T Grunert y S Irnich (2005). "Local search for vehicle routing and scheduling problems." En: *Journal of Heuristics* 11, págs. 267-306.
- Murata, T. y R. Itai (2005). "Multi-objective vehicle routing problems using two-fold EMO algorithm to enhance solution similarity on non-dominated set, in: C.A. Coello Coello, A.H. Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization." En: *Third International Conference, Lecture Notes in Computer Science*.
- Hanshar, B. Ombuki ; B. Ross ; F. (2006). "Multi-objective genetic algorithms for vehicle routing problem with time windows." En: *Applied Intelligence.*, págs. 17-30.

- Lee, K. Tan ; Y. Chew ; L. (2006). "A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems." En: *European Journal of Operational Research.*, págs. 855-885.
- Martí, J. Pacheco ; R. (2006). "Tabu search for a multi-objective routing problem." En: *Journal of the Operational Research Society*, págs. 29-37.
- Beham, A. (2007). "Parallel tabu search and the multiobjective vehicle routing problem with time windows." En: *In 21th International Parallel and Distributed Processing Symposium. Computer Society*, págs. 1-8.
- Bruce Golden, S. Raghavan y Edward Wasil (2008). "The vehicle routing problem: latest advances and new challenges." En: *Springer, New York*. 16.2, págs. 3-122.
- Cruz, J. Restrepo ; V. Medina ; T. (2008). "Un problema logístico de programación de vehículos con ventanas de tiempo." En: *Scientia et Technica*.
- Pinedo, M. (2008). "Programación: Teoría, Algoritmos y Sistemas." En: *Springer*.
- Banga, M. Schluter ;J. Egea ; J. (2009). "Extended ant colony optimization for non-convex mixed integer nonlinear programming." En: *Computers and Operations Research*, págs. 2217-2229.
- Dorigo, M. y T. Stutzle (2009). "An Colony Optimization: Overview and Recent Advances." En: *Handbook of Metaheuristics, 2 Edition* 146, págs. 22-24.
- Festa, P. y M. Resende (2009). "An Annotated bibliography of GRASP." En: *Part 1: Algorithms. International Transactions in Operational Research* 16, págs. 1-24.
- Jozefowicz, N, F Semet y E Talbi (2009). "An evolutionary algorithm for the vehicle routing problem with route balancing." En: *European Journal of Operational Research*, págs. 761-769.
- Juan, A, J Faulin y J. Jorba (2009). "Using ori-ented random search to provide a set of alternative solutions to the capacited vehicle routing problem." En: *Operations Research and Cyber-Infrastructure. Springer*, págs. 109-124.
- Laporte, G. (2009). "Fifty Years of Vehicle Routing." En: *Transportation Science*.
- Potvin, J. (2009). "State of the Art Review Evolutionary Algorithms for Vehicle Routing." En: *Journal on Computing* 146, págs. 518-548.
- Gendreau, M y J. Potvin (2010). "Genetic Algorithm." En: *Handbook of Metaheuristics, 2 Edition* 146, págs. 340-349.
- Ghannadpour, K. Ghoseiri ; S. (2010). "Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm." En: *Applied Soft Computing*, págs. 1096-1107.
- Narducci, J. Daza ; J. Montoya;F. (2010). "Solving the capacitated vehicle routing problem using a twophase metaheuristic procedure." En: *Revista EIA*.
- Toscano, R. y P. Lyonnet (2010). "Un nuevo enfoque heurístico para problemas de optimización no convexos." En: *Ciencia de la información.*, págs. 1955-1966.
- Rocha, Linda, Elsa González y Javier Orjuela (2011). "Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución". En: *Ingeniería* 16.2, págs. 35-55.
- Chan, W. Siu ; C. Chan ; H. (2012). "Green cargo routing using genetic algorithms." En: *In Proceedings of the International MultiConference of Engineers and Computer Scientists*.
- Juan, Angel et al. (2013). "MIRHA: Multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problem." En: *DOI 10.1007/s11750-011-0245-1. ISSN: 1134-5764 (print version) ISSN: 1863-8279* 21.4, págs. 109-132.
- Aranda, J y J Orjuela (2015). "Multiobjective optimization in biofuel supply chain management. A review of the literature." En: *Ingeniería*.
- Zuluaga, Eliana Toro Ocampo ; A. Domínguez Castaño ; A. Escobar (2016). "Desempeño de las técnicas de agrupamiento para resolver el problema de ruteo con múltiples depósitos." En: *Scielo*.

- Grasas, Alex et al. (2017). “Biased randomization of heuristics using skewed probability distributions: A survey and some applications.” En: *Computers Industrial Engineering* 110 216–228.
- Rosabal, Yadira (2019). “Heuristic Constructive Greedy for Dynamic Maximal Covering Location Problem.” En: *Revista cuba ciencia informatica vol.13 no.4*.
- Juan, Ángel et al. (2021). “Un algoritmo genético simheurístico para el problema de asignación de tareas y enrutamiento de UAV abierto con tiempos de viaje y servicio estocásticos.” En: *Universidad Autónoma de Barcelona, Instituto de Ingenieros Eléctricos y Electrónicos Inc.*
- Shao, Yihang y Xiaoli Xu (2023). “Three-dimensional Multi-UAV Trajectory Design for Cooperative Video Inspection and Uploading”. En: *IEEE Transactions on Vehicular Technology*.
- Bajaj, Aaishwarya S y Jayesh M Dhodiya (2024). “Sustainable multiple travelling salesman problem solved by reference point aspiration level based multi objective quasi oppositional Jaya algorithm under uncertain environment”. En: *Evolutionary Intelligence*, págs. 1-40.
- Escoto, Marc, Antoni Guerrero y Verónica Medina (2024). “Strategic Route Planning: A BR-Heuristic Adaptation for Multi-depot Logistics.” En: *DSA ISC 2024, DSA ISC 2024 - Optimization Algorithms for DS*. 16.2.
- Huang, Jing et al. (2024). “A Pesticide Spraying Mission Allocation and Path Planning with Multicopters.” En: *IEEE Transactions on Aerospace and Electronic Systems*.
- Linganathan, Shayathri y Purusotham Singamsetty (2024). “Genetic algorithm to the bi-objective multiple travelling salesman problem”. En: *Alexandria Engineering Journal* 90, págs. 98-111.
- Smurfit, Michael (2024). “Optimising a bi-objective team orienteering problem for UAVs with simheuristic.” En: *A Capstone submitted to University College Dublin in part fulfillment of the requirements of the degree of MSc in Business Analytics*.

## A. Apéndice

### A.1. Tablas de Evaluación del Parámetro $\alpha$

Alpha	Nodos VIP visitados	Km recorridos	Alpha	Nodos VIP visitados	Km recorridos
0.00	2	600.78	0.55	6	813.20
0.05	2	573.43	0.60	7	868.48
0.10	4	620.54	0.65	9	910.27
0.15	3	634.98	0.70	8	845.08
0.20	3	625.05	0.75	9	926.33
0.25	4	625.26	0.80	9	910.46
0.30	5	687.94	0.85	9	945.50
0.35	5	758.02	0.90	10	950.47
0.40	5	789.45	0.95	13	968.18
0.45	6	846.07	1	13	950.57
0.50	6	820.16	-	-	-

Cuadro 6: Resultados para valores de Alpha en la instancia ch150.tsp

Alpha	Nodos VIP visitados	Km recorridos	Alpha	Nodos VIP visitados	Km recorridos
0.00	1	957.09	0.55	8	1863.97
0.05	3	1256.58	0.60	9	1842.86
0.10	2	1187.64	0.65	9	1749.63
0.15	3	1476.69	0.70	8	1629.54
0.20	3	1247.80	0.75	8	1635.60
0.25	5	1533.76	0.80	7	1629.39
0.30	5	1547.87	0.85	11	1857.86
0.35	4	1486.49	0.90	12	1829.64
0.40	4	1269.78	0.95	12	1854.76
0.45	6	1674.65	1	12	1868.40
0.50	7	1659.73	-	-	-

Cuadro 7: Resultados para valores de Alpha en la instancia kroB100.tsp

Alpha	Nodos VIP visitados	Km recorridos	Alpha	Nodos VIP visitados	Km recorridos
0.00	0	849.87	0.55	7	1197.96
0.05	1	867.65	0.60	10	1597.54
0.10	1	910.54	0.65	8	1649.76
0.15	2	886.96	0.70	10	1743.68
0.20	3	947.30	0.75	11	1764.43
0.25	3	920.86	0.80	9	1684.86
0.30	5	984.48	0.85	9	1725.29
0.35	4	915.69	0.90	12	1865.83
0.40	4	995.38	0.95	13	1839.42
0.45	8	1233.60	1	13	1828.51
0.50	7	1115.65	-	-	-

Cuadro 8: Resultados para valores de Alpha en la instancia kroC100.tsp

Alpha	Nodos VIP visitados	Km recorridos	Alpha	Nodos VIP visitados	Km recorridos
0.00	0	523.34	0.55	12	989.76
0.05	2	538.54	0.60	10	956.32
0.10	3	570.09	0.65	8	897.46
0.15	3	567.56	0.70	8	976.61
0.20	4	589.35	0.75	8	948.55
0.25	4	658.12	0.80	7	925.09
0.30	6	652.08	0.85	8	945.53
0.35	5	684.58	0.90	10	950.28
0.40	4	649.45	0.95	12	968.19
0.45	6	768.37	1	12	1096.10
0.50	6	735.65	-	-	-

Cuadro 9: Resultados para valores de Alpha en la instancia kroD100.tsp