



Vigilada Mineducación

Algoritmo evolutivo para resolver el problema de enrutamiento de vehículos tiempo dependiente con ventanas de tiempo en una compañía del sector de alimentos y bebidas en Colombia

Evolutionary Algorithm to solve time-dependent vehicle routing problem with time windows in a company of the food and drinks industry in Colombia

Camilo Ramírez Guilombo

Director

Juan Carlos Rivera Agudelo

Área de Computación y Analítica

jrivera@eafit.edu.co

Maestría en Ciencias de los Datos y Analítica

Escuela de Ciencias Aplicadas e Ingeniería

Medellín

Junio de 2023

Índice

1. Introducción	4
2. Estado del arte y marco teórico	5
2.1. El problema VRP	5
2.2. Tipos de solución	5
2.3. Algoritmos genéticos híbridos	6
3. Metodología	7
3.1. Comprensión del negocio	7
3.2. Comprensión de los datos	8
3.3. Preparación de los datos	10
3.4. Modelado	10
3.4.1. Detalle del algoritmo utilizado	12
3.5. Evaluación	15
3.5.1. Problema 1	15
3.5.2. Problema 2	19
3.5.3. Problema 3	22
3.5.4. Problema 4	24
3.5.5. Problema 5	26
4. Recomendaciones y conclusiones	28

Índice de cuadros

1. Variables utilizadas para la construcción del algoritmo.	8
2. Resultados experimentales.	28

Índice de figuras

1. Distribución de clientes	9
2. Distribución para la ciudad principal	9
3. Solución Problema 1: Algoritmo inicial	16
4. Solución Problema 1: Algoritmo inicial. Detalle en concentración de nodos	17
5. Solución Problema 1: Algoritmo propuesto	18
6. Solución Problema 1: Algoritmo propuesto. Detalle en concentración de nodos	18
7. Evolución del costo de la solución	19
8. Solución Problema 2: Algoritmo inicial	20
9. Solución Problema 2: Algoritmo inicial. Detalle en concentración de nodos	20
10. Solución Problema 2: Algoritmo propuesto	21
11. Solución Problema 2: Algoritmo propuesto. Detalle en concentración de nodos	22
12. Solución Problema 3: Algoritmo inicial	23
13. Solución Problema 3: Algoritmo propuesto	24
14. Solución Problema 4: Algoritmo inicial	25
15. Solución Problema 4: Algoritmo propuesto	26
16. Solución Problema 5: Algoritmo inicial	27
17. Solución Problema 5: Algoritmo propuesto	28

Algoritmo evolutivo para resolver el problema de enrutamiento de vehículos tiempo dependiente con ventanas de tiempo en una compañía del sector de alimentos y bebidas en Colombia

Resumen

La optimización de rutas es un problema común en muchas empresas del sector de transporte y logística, donde la exigencia de los clientes en términos de cumplimiento en cantidades y tiempos de entrega ha tomado gran relevancia. En este proyecto se presenta un caso de estudio en la industria de alimentos y bebidas enfocado en solucionar el problema de enrutamiento de vehículos con ventanas de tiempo y tiempos de viaje dependientes del tiempo de realización conocido como TDVRPTW (sigla de Time Dependent Vehicle Routing Problem with Time Windows). El objetivo principal es encontrar las rutas de distribución más cortas, que permitan minimizar el costo total de distribución. El TDVRPTW es una variante del problema VRP (Vehicle Routing Problem), donde es necesario que los clientes sean atendidos en las ventanas de tiempo en los que ellos están dispuestos a recibir el producto, teniendo en cuenta que el tiempo de distribución es dependiente del tráfico de vehículos que tiene una ciudad en una hora del día específica. Este problema pertenece a los de tipo NP-Hard, por lo que es necesario contar con un modelo eficiente y efectivo para solucionarlo, debido a la complejidad que representa su solución bajo métodos exactos. Para solucionar el caso aplicado se utiliza uno de los métodos heurísticos que ha demostrado ser eficiente para resolver modelos teóricos en la literatura, el cual corresponde a una técnica de Algoritmos Evolutivos. Para determinar las sub rutas óptimas, se aplica el algoritmo Order First Split Second, técnica que garantiza optimalidad dentro de la selección de sub rutas en un cromosoma dado.

El algoritmo es probado para resolver el problema de enrutamiento de vehículos en una ciudad de Colombia, generando planes de distribución para la empresa en cuestión y es comparado con la metodología actual utilizada en la empresa, en la cual se utiliza un software que no permite tener en cuenta la dependencia del tiempo ni las ventanas horarias de los clientes. Los resultados muestran una reducción en la solución del problema del 10 % en comparación con la estrategia actual, fundamentado en la capacidad del algoritmo propuesto de considerar mayores restricciones y la capacidad de tener en cuenta distancias reales medidas en tiempo de desplazamiento del vehículo y no la distancia euclidiana entre nodos.

1. Introducción

Uno de los factores que mayor costo representa para una empresa del sector logístico es el transporte. En Colombia, por su complejidad en las vías y poca diversificación en los medios de transporte, resulta ser de gran interés para las diferentes compañías encontrar estrategias eficientes de distribución que permitan reducir el costo de mover carga. Los costos de distribución suelen separarse en dos grupos: distribución primaria y secundaria. La distribución primaria corresponde a los productos que son transportados de un centro de distribución de la compañía a otro, movimientos que suelen ser de largo desplazamiento como envíos entre ciudades. Por su parte la distribución secundaria suele ser conocida como la operación de última milla donde se entrega el producto al cliente final.

La distribución secundaria, es un problema de interés general en la industria por su alta complejidad. El problema consiste en determinar la mejor forma de distribuir en los vehículos disponibles el producto a entregar, de manera que se pueda encontrar la mejor ruta de entrega de acuerdo a la posición geográfica de los clientes y la demanda de los mismos. La complejidad del problema está dada por la gran cantidad de clientes que generalmente deben ser atendidos, así como la diversa demanda que cada cliente puede representar de acuerdo a su tamaño y distancia [1].

Específicamente en la industria de bebidas en Colombia, la operación de última milla suele atender directamente distintas tiendas en cada una de las ciudades, lo que representa una gran cantidad de clientes a atender. Este problema conocido como el problema de enrutamiento de vehículos, debido a su alta complejidad computacional [2], suele ser simplificado no teniendo en cuenta restricciones de mayor complejidad, como asumir distancias lineales entre los clientes, evitar tener en cuenta sus ventanas de atención y asumir que el tiempo de entrega será el mismo sin importar la hora del día.

Relajar el problema de esta forma, puede llevar a las empresas a incurrir en costos adicionales en su sistema de transporte, en una industria donde el margen sobre los productos es estrecho y la participación en volumen suele ser la principal fuente de competencia entre las empresas que lo conforman; por lo que es de gran interés para las organizaciones mantener sus costos de operación lo más bajos posible para aumentar su EBITDA (Earnings Before Interests, Taxes, Depreciations and Amortizations).

El objetivo principal de este trabajo es optimizar la operación de enrutamiento de vehículos en una empresa del sector bebidas en Colombia, permitiendo aumentar el nivel de servicio entregado a sus clientes y disminuir el costo de operación asociado al proceso de entrega de última milla a través de un algoritmo metaheurístico híbrido. Para esto se desarrolla un algoritmo evolutivo que permite resolver el problema de optimización y brindar una solución eficiente a una de las empresas más grandes del sector de Bebidas en Colombia, la cual a pesar de su dimensión, en su operación actual no cuenta con un sistema que resuelva el problema teniendo en cuenta restricciones de mayor complejidad como son la dependencia del tiempo y las ventanas de operación de los clientes.

2. Estado del arte y marco teórico

2.1. El problema VRP

El problema de enrutamiento de vehículos, consiste en encontrar una asignación de vehículos para realizar entregas en distintos nodos o clientes, en un orden tal que se minimicen los costos de operación. Un problema que suele estar relacionado y representa uno de los puntos de partida para la solución de Vehicle Routing Problem (VRP), es el Traveling Salesman Problem (TSP) teniendo en cuenta que corresponde a una simplificación del mismo, donde solo se tiene un vehículo y éste debe visitar todos los clientes. Es por esto que es común que las técnicas para solucionar un problema sean replicables al otro.

Dantzig y Ramser [3], proponen el primer modelo VRP en el cual plantean una solución para una empresa petrolera, la cual desde su terminal central debe abastecer estaciones de servicio, que corresponden a sus clientes. El problema se plantea como un TSP con restricciones adicionales, en las cuales se añade la condición de tener vehículos homogéneos y que las rutas deben regresar a su nodo origen.

Con el fin de acercar el planteamiento del problema a una mejor representación de las necesidades en aplicaciones en negocios reales, Clarke y Wright [4] proponen, años más tarde, un nuevo problema en el cual se agrega una restricción adicional, en la cual los clientes deben ser servidos en ciertas ventanas de tiempo predefinidas. Este problema es ampliamente investigado y conocido como VRPTW (Vehicle Routing Problem with Time Windows).

Una vez tomada en cuenta la restricción asociada a las ventanas de atención, se han estudiado una serie de problemas que agregan restricciones al problema [5], que pretenden acercar el modelo a situaciones reales. Algunas de las variaciones son flotas heterogéneas (Heterogeneous Fleet Vehicle Routing Problem - HFVRP) donde los vehículos tienen una capacidad y costo diferenciados [6] o dependientes del tiempo (Time Dependent VRP - TDVRP) problema en el cual el tiempo que tarde en recorrer un arco varía de acuerdo a la hora del día en que se está ejecutando la ruta [7].

Los problemas VRP, podrían clasificarse entonces según las condiciones del problema que abordan: su condición estática (único centro o múltiples centros de distribución), restricciones del vehículo (capacidad homogénea o capacidad heterogénea), tipo de operación (solo recoger o solo entregar, recoger y entregar), característica del problema (determinístico o estocástico) [8].

2.2. Tipos de solución

Así como se han abordado distintos tipos de problemas asociados al VRP con diferentes variantes de acuerdo a las restricciones a abordar en el problema, son múltiples las alternativas propuestas para dar una solución. Estos modelos pueden clasificarse en 3 tipos:

- Exactos: estos algoritmos se caracterizan porque dan garantía de encontrar una solución óptima del problema. Sus mayores limitaciones están relacionadas con el tiempo que toma hallar dicha solución, y por ende, su baja escalabilidad a problemas del mundo real, donde la cantidad de nodos y restricciones sea alta [9]. Entre los

métodos más conocidos se encuentra Branch and Bound, con sus variantes Branch and Cut y Branch and Price [10, 11]. Estos métodos han sido probados con buenos rendimientos en problemas de baja cantidad de clientes.

- **Heurísticos:** los métodos heurísticos se caracterizan por limitar la exploración en el espacio de búsqueda, por medio de un conjunto de criterios que los ayuden a encontrar una buena solución en un corto tiempo de respuesta. Dentro de los modelos heurísticos resaltan los métodos constructivos, los cuales son modelos que usan los datos del problema para buscar una solución. Entre los métodos heurísticos que han demostrado encontrar buenas soluciones y han sido utilizados ampliamente a lo largo del tiempo se encuentran el algoritmo del vecino más cercano [12], que consiste en encontrar el punto más cercano de cada nodo y así asociar el punto siguiente en la ruta; el método de los ahorros [4], el cual construye la solución a partir del ahorro que representa ir a un nodo adicional j desde el cliente i , en lugar de volver al nodo inicial 0 y tener que partir de nuevo al nodo j . También han sido utilizados heurísticos de dos fases, donde se incluyen técnicas de clustering y de búsqueda local, los cuales exploran el vecindario de una solución generada por un método constructivo [8].
- **Metaheurísticos:** Los algoritmos metaheurísticos han sido investigados ampliamente en distintos campos y han sido implementados recurrentemente para resolver problemas VRP en la literatura. Estos métodos se caracterizan por explorar de forma profunda un conjunto de soluciones factibles, tratando de seguir aquellas que parecen tener un mejor desempeño. Entre estos destaca Simulated Annealing (SA) [13], basada en mecánica estadística en la que se evalúa estocásticamente si un nuevo vecino aporta en el desempeño de la función objetivo. Mojtahedi et Al [14], proponen el Adaptative Memory Social Engineer Optimizer (AMSEO), como una mejora a SA, el cual utiliza el algoritmo Social Engineering Optimizer (SEO), donde a través del concepto de individuo atacante y defensor, se itera en la búsqueda de soluciones que mejoren el impacto en la función objetivo [15]. Holland [16], propone un algoritmo que simula el proceso de reproducción de las especies, por medio de la representación de cromosomas y genes, llamado Algoritmo Genético (GA), donde a una población inicial se le aplican operadores de cruce y mutación, los cuales combinan los genes de individuos seleccionados, para luego intercambiar algunas de las posiciones de los individuos resultantes. Los algoritmos genéticos han sido utilizados para problemas de mayor complejidad, en los cuales se aumenta el número de restricciones, algunos ejemplos incorporan restricciones asociadas a la dependencia del tiempo, múltiples centros de distribución, entrega y recogida en múltiples destinos [17].

2.3. Algoritmos genéticos híbridos

En los últimos años los modelos híbridos han sido utilizados en mayor medida para resolver el problema VRP con ventanas y dependencia del tiempo. Es común encontrar soluciones basadas en algoritmos genéticos donde el factor híbrido corresponde a la implementación de otros modelos inmersos en el proceso evolutivo que ayuden a encontrar de una forma más rápida una mejor solución del problema [18]. La búsqueda local, es un método común utilizado para acelerar el hallazgo de una mejor solución, en este conjunto de algoritmos resalta el 2-opt, el cual consiste en un proceso iterativo donde se invierte una secuencia de nodos dentro de una ruta [19], este algoritmo inicialmente propuesto para solucionar el problema TSP, ha sido utilizado para resolver problemas más complejos

como es el caso del VRP con múltiples depósitos [20].

La dependencia del tiempo es un factor clave para tener en cuenta altas zonas de tráfico y generar una solución que se adapte a la necesidad real del negocio. Uno de los casos de aplicación fue formulado por Haghani y Jung [21], quienes desarrollaron un algoritmo genético que responde a la variabilidad de la demanda que pueden tener los clientes y el tiempo para recorrer una ruta de un nodo a otro. Fan et al. [22] desarrollan un algoritmo genético híbrido que tiene en cuenta la distancia temporal-espacial para generar un cluster de los clientes que genera la población inicial, además de utilizar la técnica de simulated annealing para balancear la diversificación y explotación.

Una de las dificultades adicionales al implementar algoritmos genéticos para solucionar problemas VRP y sus variantes, está dada por la evaluación de las mejores sub rutas que componen un cromosoma. El algoritmo de ordenar primero y separar en segunda instancia [23], es una propuesta desarrollada para encontrar de forma óptima la mejor repartición de un cromosoma dado, en diferentes rutas. El algoritmo consiste en evaluar el costo de agregar un nodo adicional teniendo en cuenta la restricción de capacidad implícita en el problema. Este tipo de solución suele representar un costo computacional alto en $\mathcal{O}(nB)$ donde n representa el número de clientes y B el número promedio de viajes factibles desde cada nodo del gran tour. Vidal [24], propone una nueva metodología para dar solución al problema computacional al proponer un algoritmo que da solución en tiempo lineal $\mathcal{O}(n)$.

3. Metodología

La solución a desarrollar para responder a la problemática descrita consiste en la implementación de un algoritmo genético para solucionar el problema de enrutamiento de vehículos en una empresa del sector de alimentos y bebidas en Colombia. Para esto es necesario desarrollar una metodología acorde que permita resolver el problema planteado.

Para este trabajo se utilizó la metodología CRISP-DM, ampliamente conocida en proyectos de implementación de soluciones basadas en datos. Se abordaron las distintas etapas del ciclo de vida del proyecto, teniendo en consideración que su desarrollo no estrictamente secuencial, ya que pueden existir iteraciones entre las fases a medida que avanza el proyecto.

3.1. Comprensión del negocio

La empresa en la cual se desarrolló este trabajo tuvo una participación importante para la determinación del problema a plantear, los objetivos y la evaluación de los resultados. Es importante resaltar que la necesidad de solucionar el problema, nace de la incapacidad actual para atender adecuadamente algunos de sus clientes, al no poder cubrir las ventanas de tiempo para la recepción de productos.

Se identificó que una de las principales causas para el incumplimiento de los planes de distribución generados, se debe a la suposición de que el tiempo de recorrido entre nodo y nodo es el mismo sin importar el momento del día, dado que en la práctica el tráfico de la ciudad es cambiante y condiciona los tiempos de operación, además de basar la solución en distancias euclidianas entre clientes, lo cual en la práctica hace que la solución esté basada en caminos que geográficamente no es posible recorrer.

En la revisión con el negocio, se estableció que la demanda será medida en estibas, donde la empresa proporcionará como dato de entrada la cantidad de estibas solicitadas por el cliente en su pedido. Los clientes realizan su pedido en cajas por producto, pero es el negocio quien se encargará de hacer la transformación de esta información para obtener la demanda en la unidad de medida Estibas.

A su vez, se estableció la capacidad en estibas que tiene cada uno de los vehículos que es de 10, por lo que ningún vehículo podrá superar esta capacidad instalada. Finalmente por solicitud del negocio y disponibilidad de la información, se estableció que todas las pruebas serán realizadas en una ciudad del país.

El problema a solucionar consiste en generar el plan de distribución secundaria para un día específico. Cada día representa un problema distinto por el modelo de negocio en el cual se reciben todos los pedidos en el día $n - 1$ y en el día n se procede a realizar las entregas de dichos pedidos. Los pedidos son recibidos en horas de la noche y el cargue de los vehículos de distribución se comienza a realizar una vez se tenga el programa de distribución. Es por esta razón que es necesario contar con un algoritmo eficiente que resuelva el problema en el menor tiempo posible.

3.2. Comprensión de los datos

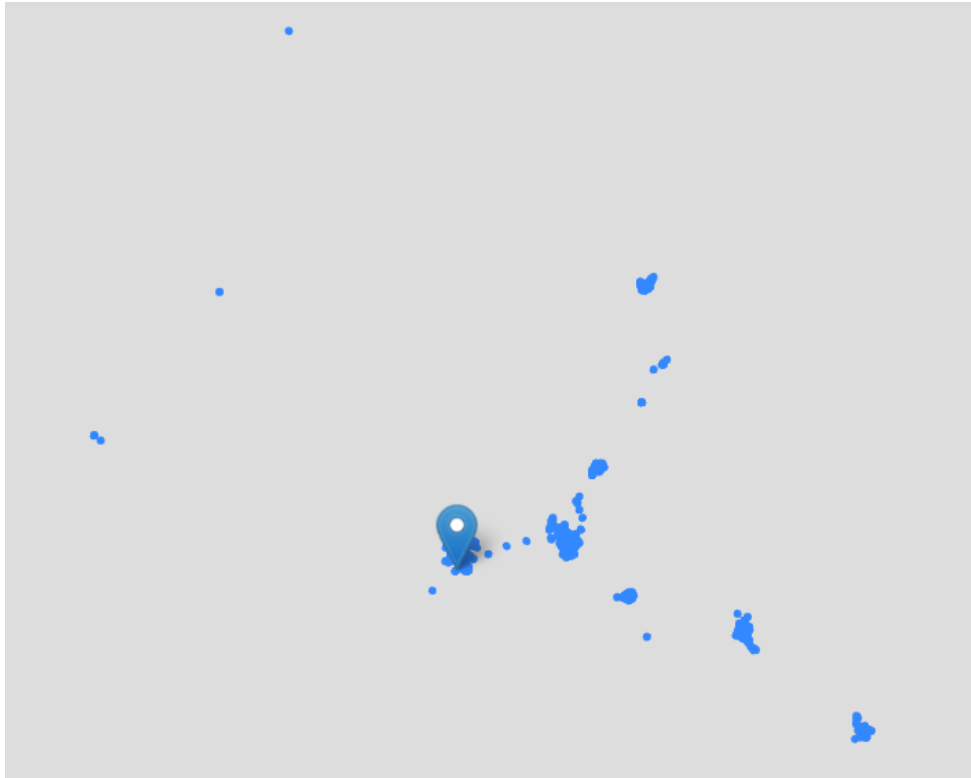
Para este proyecto se utilizan los datos relacionados con los pedidos realizados por cada uno de los clientes de la empresa en mención, su posición geográfica y datos relacionados con las capacidades de operación. La siguiente tabla resume la información utilizada para la solución del problema:

Variable	Descripción
Número_Vehículos	Vehículos disponibles para atender la demanda
Capacidad_Vehículo	Número de estibas que puede transportar un vehículo
Id_Cliente	Identificador único del cliente
Demanda	Número de estibas solicitadas por el cliente
Longitud	Posición georreferenciada del cliente (longitud)
Latitud	Posición georreferenciada del cliente (latitud)
TW_low	Venta de tiempo inferior para la atención del cliente
TW_up	Venta de tiempo superior para la atención del cliente
Time_vector	Ventanas de tiempo utilizadas para la dependencia del tiempo

Cuadro 1: Variables utilizadas para la construcción del algoritmo.

Para mejorar el entendimiento del problema se generó el siguiente gráfico para dimensionar la ubicación de los diferentes clientes a atender y el centro de distribución que debe atender su demanda.

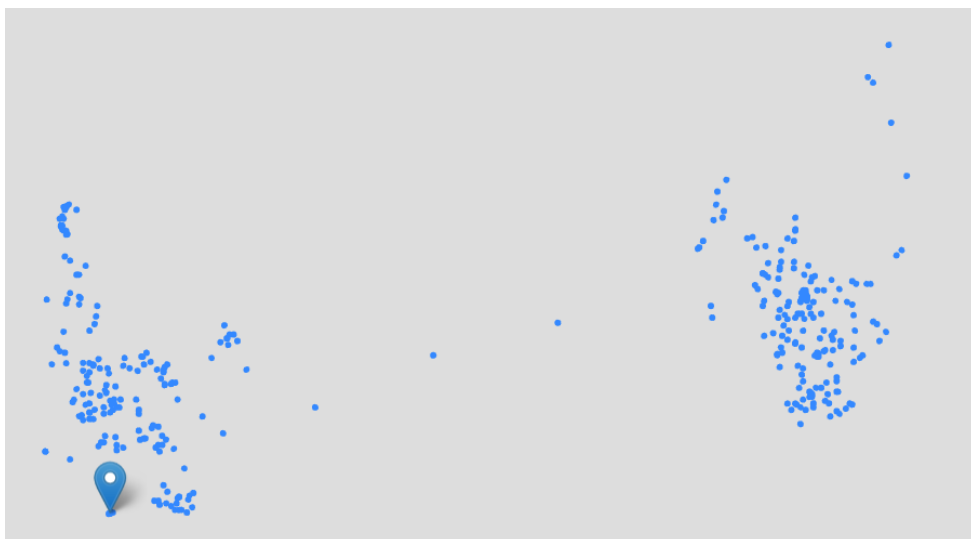
Figura 1: Distribución de clientes



Puede notarse como el centro de distribución debe cubrir clientes que se encuentran en lugares alejados y que tienen baja concentración de clientes a su alrededor.

Para las dos ciudades principales donde se concentran la mayor cantidad de clientes se realiza el siguiente gráfico que permite visualizar con mayor detalle el problema, con el fin de entender la distribución según la ubicación del centro de distribución:

Figura 2: Distribución para la ciudad principal



El centro de distribución resulta estar alejado de la zona en la que están ubicados los clientes. Este tipo de problema es común en algunas ciudades donde el plan de

ordenamiento territorial exige que los grandes almacenes se encuentren en las afueras de la ciudad, lo que representa beneficios en materia de movilidad y gestión ambiental, sin embargo para las empresas representa un aumento en el costo de distribución, el cual aplica para el caso que se está abordando.

3.3. Preparación de los datos

Una vez comprendidos los datos, la siguiente etapa consiste en preparar la información necesaria para que el modelo opere de forma correcta. En esta etapa se realizó la validación de que cada cliente cuente con su posición georreferenciada, su demanda especificada y tenga asociada su restricción de ventana de tiempo correspondiente. En los casos en que el negocio no conocía, o no tenía interés en generar ventanas de tiempo para un cliente particular la ventana de tiempo se dejó abierta, para que el cliente pueda ser atendido en cualquier horario. La hora inicial de operación se definió a las 6 a.m. Para evaluar la capacidad del algoritmo propuesto no se tuvieron en cuenta las restricciones de ventana de tiempo, con el fin de permitir el punto de comparación con el modelo utilizado en la actualidad.

Para obtener la matriz de distancias entre nodos, se utilizó la herramienta de OSRM tomada de <https://map.project-osrm.org/>, quienes permiten obtener la distancia de un punto a a un punto b en auto. El negocio generará una API para consultar en Google Maps las distancias, sin embargo está no logró estar disponible para el desarrollo de este proyecto, por lo que se trabajará con OSRM y se simularán los tiempos entre nodos para las demás ventanas horarias.

3.4. Modelado

El algoritmo a desarrollar busca resolver el siguiente problema matemático [7]:

Constantes:	
n	número de nodos incluyendo el origen
M	número de intervalos de tiempo
K	número de vehículos
c_{ij}^m	tiempo de desplazamiento de i a j en el intervalo m
c_i	tiempo de servicio del nodo i
T_{ij}^m	límite superior del intervalo m para el arco i,j
t	tiempo de inicio del recorrido
w_k	capacidad del vehículo k
d_i	demanda del nodo i
B_1	Un número grande
B_2	Un número grande
B	Capacidad del vehículo más grande
L_i	tiempo más pronto en que se puede atender el nodo i
U_i	tiempo más tarde en que se puede atender el nodo i
Variables de decisión:	
x_{ij}^m	1 si el cualquier vehículo atiende la ruta i,j en el intervalo m
t_j	tiempo de salida de cualquier vehículo del nodo j
w_j	peso acumulado del vehículo a la salida del nodo j

$$\min \sum_{k=1}^K t_{n+k} \quad (1)$$

$$\sum_{i=1}^n \sum_{m=1}^M x_{ij}^m \leq 1, \quad (j = 2, \dots, n + K; i \neq j) \quad (2)$$

$$\sum_{j=2}^{n+K} \sum_{m=1}^M x_{ij}^m = 1, \quad (i = 2, \dots, n; j \neq i) \quad (3)$$

$$\sum_{j=2}^n \sum_{m=1}^M x_{ij}^m \leq K, \quad (i = 2, \dots, n; j \neq i) \quad (4)$$

$$t_1 = t \quad (5)$$

$$t_j - t_i - B_1 x_{ij}^m \geq c_{ij}^m + c_j - B_1 \quad (6)$$

$(i = 1, \dots, n; j = 2, \dots, n + K; i \neq j; m = 1, \dots, M)$

$$t_i + B_2 x_{ij}^m \leq T_{ij}^m + B_2$$

$$t_i - T_{ij}^{m-1} x_{ij}^m \geq 0 \quad (7)$$

$(i = 1, \dots, n; j = 2, \dots, n + K; i \neq j; m = 1, \dots, M)$

$$L_i + c_i \leq t_i \leq U_i + c_i, \quad (i = 1, \dots, n) \quad (8)$$

$$w_j - w_i - B \sum_{m=1}^M x_{ij}^m \geq d_j - B \quad (9)$$

$(i = 1, \dots, n; j = 2, \dots, n + K; i \neq j)$

$$w_1 = 0 \quad (10)$$

$$w_{n+k} \leq b_k, \quad (k = 1, \dots, K) \quad (11)$$

$$x_{ij}^m = 0 \text{ or } 1, \quad (\forall i, j, m) \quad (12)$$

$$t_i \geq 0, \quad (\forall i) \quad (13)$$

$$w_i \geq 0, \quad (\forall i) \quad (14)$$

- **Función Objetivo:** minimizar la distancia recorrida.
- **Restricciones 2 a 4:** garantizar que cada nodo es visitado solo una vez y que máximo K vehículos son utilizados.
- **Restricción 5:** todos los vehículos inician su recorrido en el tiempo t .
- **Restricción 6:** calcula el tiempo de salida en el nodo j .
- **Restricción 7:** restricción de dependencia del tiempo, donde se asegura utilizar el tiempo asociado al intervalo m .
- **Restricción 8:** restricción de ventanas de tiempo.
- **Restricciones 9 a 11:** asegurar que la capacidad del vehículo es respetada y que cada nodo adicional mantiene igual o aumenta la capacidad del vehículo utilizada.

Para este caso la distancia recorrida se da en función del tiempo de recorrido, y esta puede variar de acuerdo a la hora en que se está desarrollando la ruta. A su vez, la distancia (que corresponde al costo de la ruta) tendrá una penalización cada vez que la ruta no cumpla una restricción de ventana de tiempo.

Para la solución del modelo, se implementó un algoritmo evolutivo que sigue los siguientes pasos:

Algorithm 1 Algoritmo Genético para TDVRPTW

$Poblacion \leftarrow$ Generar Población Inicial \triangleright Incluyendo solución del vecino más cercano

$Población \leftarrow$ Obtener sub rutas óptimas \triangleright a través de Split

$Fitness \leftarrow$ Evaluar función objetivo de cada individuo \triangleright Incluyendo restricción TW y TD

for i in Número de iteraciones **do**

$hijos \leftarrow$ aplicar operador de cruce

$hijos \leftarrow$ aplicar operador de mutación

$hijos \leftarrow$ Obtener sub rutas óptimas \triangleright a través de Split

$Población \leftarrow$ Aplicar criterio elitista

end for

$Mejor_ruta \leftarrow$ mínimo fitness

3.4.1. Detalle del algoritmo utilizado

Teniendo en cuenta que el algoritmo utilizado es evolutivo, el primer paso consiste en definir la estructura de un individuo. Un cromosoma estará definido por un conjunto de número enteros entre 1 y n sin repetición, donde n corresponde al número de nodos a visitar [23]. El orden secuencial que contiene el individuo indica el orden en que serán visitados los nodos (cada gen del cromosoma representa el número que identifica el cliente a visitar). A manera de ejemplo, para un problema de 12 nodos, un individuo puede representarse como:

[4 6 1 2 5 9 12 10 3 11 7 8]

Una vez se cuenta con la representación, el primer paso del algoritmo es generar una población inicial. Para el caso se generó una población aleatoria de tamaño m , la cual representa la población inicial y se añadió la solución que se crea a partir de seguir el algoritmo del vecino más cercano [12].

Puede notarse que hasta ahora un individuo representa un gran tour desde el nodo inicial hasta el final, por lo que su representación resuelve el problema TSP. Ahora bien, para el problema en cuestión, es necesario dividir el cromosoma en sub rutas, de manera que un individuo represente una solución para el VRP. Ordenar primero y clusterizar después conocido como split, es un método altamente probado en la academia para resolver este problema, teniendo en cuenta que permite encontrar de forma óptima las sub rutas que componen al gran tour [25]. Se utilizó el algoritmo de Bellman [26] para resolver el problema de las sub rutas, añadiendo la restricción de máximo número de vehículos disponibles, la cual consiste en iterar k veces el algoritmo de Bellman donde k es el número de vehículos disponibles [27]. Finalmente, teniendo en cuenta que se desea resolver un problema que incluye la restricción de dependencias del tiempo, se implementó la evaluación del tiempo entre nodos de acuerdo a la hora en que se encuentra la ruta. Esto permite que el split considere la dependencia del tiempo a la hora de evaluar las sub rutas óptimas.

El algoritmo 2, guardará en la matriz *pred* el mejor nodo precedente para cada cliente teniendo en cuenta la evaluación de k vehículos. Para extraer las sub rutas óptimas se debe aplicar el algoritmo 3.

La complejidad del algoritmo split está dada por $\mathcal{O}(n^2)$ lo que lo mantiene en rangos computacionales que son sencillos de resolver. Para el caso donde se incorpora la restricción de máximo número de vehículos disponibles, la solución del problema se mantiene en un tiempo pseudo polinómico [23].

Una vez obtenidas las sub rutas que sigue el individuo, se evalúa la función fitness de cada uno. Teniendo en cuenta que se desea solucionar un problema TDVRPTW, se incorporan las restricciones de ventanas de tiempo, penalizando en la función de costo aquellas soluciones que violen esta restricción.

Con función de costo calculada para cada individuo, se comienzan los ejercicios poblacionales para comenzar a desarrollar mejores soluciones. En primera instancia se utiliza un operador de cruce para obtener hijos por pares de padres. Los padres son seleccionados mediante el método de la ruleta donde aquellos individuos con mejor función de desempeño tienen una probabilidad más alta de ser elegidos. Una vez seleccionados los padres se utiliza un operador de cruce de para darle mayor profundidad a la exploración de la población inicial, se utiliza un operador de dos puntos, donde se generan dos números aleatorios indicado el punto inicial y final de corte. De esta manera el hijo 1 conservará los valores existentes entre los puntos de corte del padre 1, y los valores externos corresponderán a los clientes restantes por visitar en el padre 2. El hijo 2 por el contrario, conservará los valores interiores del padre 2, mientras los valores restantes serán completados con los genes del padre 1 [28].

Posteriormente se procede con un operador de mutación, cuya utilidad radica en la oportunidad de diversificar el espacio de búsqueda, evitando óptimos locales que

Algorithm 2 Algoritmo Split con máximo número de vehículos y tiempo dependiente

 $w_0 \leftarrow 0$ **for** $t = 1$ to $n - 1$ **do** $w_t \leftarrow \infty$ **end for****for** $k = 0$ to K **do** $p \leftarrow w$ **for** $t = k$ to $n - 1$ **do** $load \leftarrow 0$ $i \leftarrow t + 1$ **while** $i \leq n - 1$ & $load + q_{i-1} \leq Q$ **do** $load \leftarrow load + q_{i-1}$ **if** $i = t + 1$ **then** $cost \leftarrow d_{0,0,i-1}$ $j \leftarrow 0$ **else****while** $cost \geq h_{j+1}$ **do** $j \leftarrow j + 1$ **end while** $cost \leftarrow cost + d_{j,i-2,i-1}$ **while** $cost \geq h_{j+1}$ **do** $j \leftarrow j + 1$ **end while****end if****if** $p_t + cost + d_{j,i-1,0} < w_i$ **then** $w_i \leftarrow p_t + cost + d_{j,i-1,0}$ $pred_{k,i} \leftarrow t$ **end if** $i \leftarrow i + 1$ **end while****end for****end for**

concentren el resultado de la solución e impidan encontrar mejores soluciones. Para este operador se define un parámetro de mutación, el cual indica la probabilidad de que un hijo pueda ser mutado, en caso de tener un resultado positivo, se generan dos números aleatorios que indican las posiciones a ser intercambiadas. En total k posiciones serán mutadas si se cumple el criterio de la probabilidad de ser mutado.

Algorithm 3 Extracción de la solución después del split

 $c \leftarrow \text{máx } k$ $\triangleright \text{ where } \text{pred}_{:,n-1} > 0$ $i \leftarrow n - 1$ $\text{cuts}_0 \leftarrow i$ $j \leftarrow 0$ **while** $i > 0$ **do** add customer $\text{pred}_{c-j,i}$ to the beginning of cuts $i \leftarrow \text{pred}_{c-j,i}$ $j \leftarrow j + 1$ **end while**

Para todos los hijos se aplica el split, obteniendo sus sub rutas óptimas y se calcula su función de desempeño. Para elegir cuales individuos pertenecen a la siguiente generación se utiliza un criterio elitista.

El proceso es iterado hasta que se cumpla con un número de iteraciones definido (como parámetro), o si por p periodos de tiempo consecutivos no se encuentra una mejor solución, el proceso también se detiene.

3.5. Evaluación

Para evaluar la calidad del algoritmo propuesto se realizaron corridas para resolver 5 problemas propuestos por la empresa. Se medirá el tiempo total de recorrido como función objetivo comparando los resultados del algoritmo con el modelo actual que usa la empresa, el cual será llamado Algoritmo inicial, para resolver el problema. Es importante aclarar que la secuencia dada por el Algoritmo inicial para secuenciar las rutas, no permite la evaluación de ventanas de tiempo ni la inclusión de la dependencia de la hora en la cual se está ejecutando la ruta, por lo que para evaluar su calidad se tomará la secuencia y se evaluará según la función de costo calculada en el algoritmo propuesto. Adicionalmente es importante aclarar que dicha secuencia calculada inicialmente por la empresa, busca minimizar las distancias euclidianas entre las coordenadas asociadas, sin considerar los tiempos de desplazamiento reales y las vías por las cuales circulan los vehículos, por lo que al evaluar las rutas con distancias reales según la movilidad de la ciudad, aumenta la asertividad de la decisión a considerar.

Todos los problemas buscan encontrar la mejor forma de secuenciar las entregas de 89 clientes abastecidos desde un único centro de distribución y con un máximo de 5 vehículos disponibles para realizar las entregas. Los vehículos tienen una capacidad máxima de 10 estibas y ningún cliente presenta una demanda superior a esta capacidad.

3.5.1. Problema 1

- Algoritmo inicial Problema 1

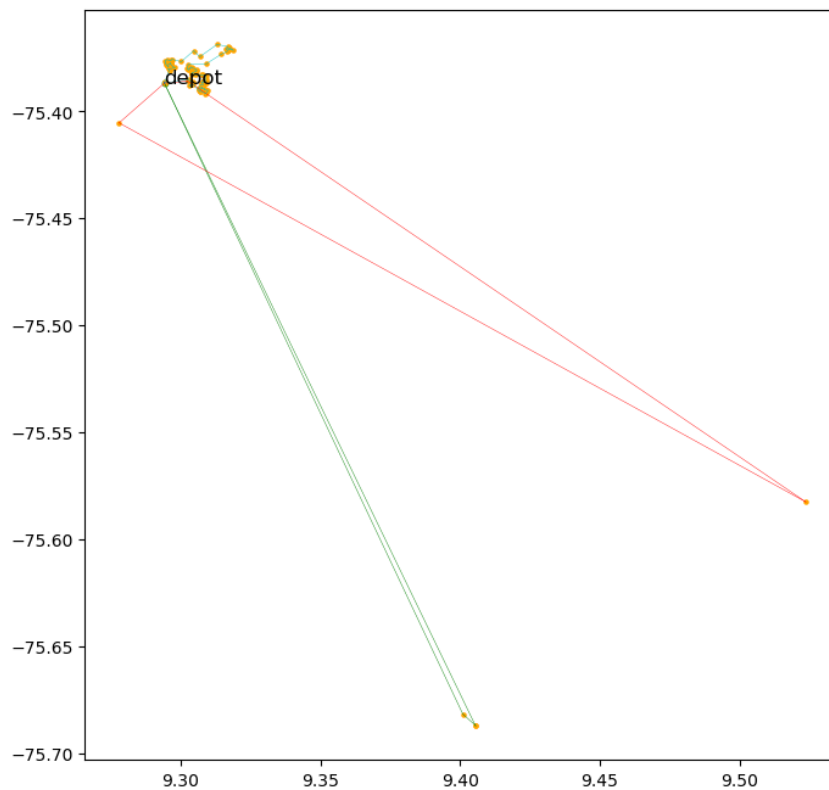
La solución al problema 1 generado por la empresa contiene la siguiente distribución:

- $R1 = [1, 2, 3]$
- $R2 = [4, 5, 6]$
- $R3 = [7, 8, 9]$
- $R4 = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89]$

Todas las soluciones dadas por la empresa son entregadas en el orden de la construcción de la secuencia, por lo que la solución corresponde en el incremental del *id* del cliente desde el primer nodo al 89.

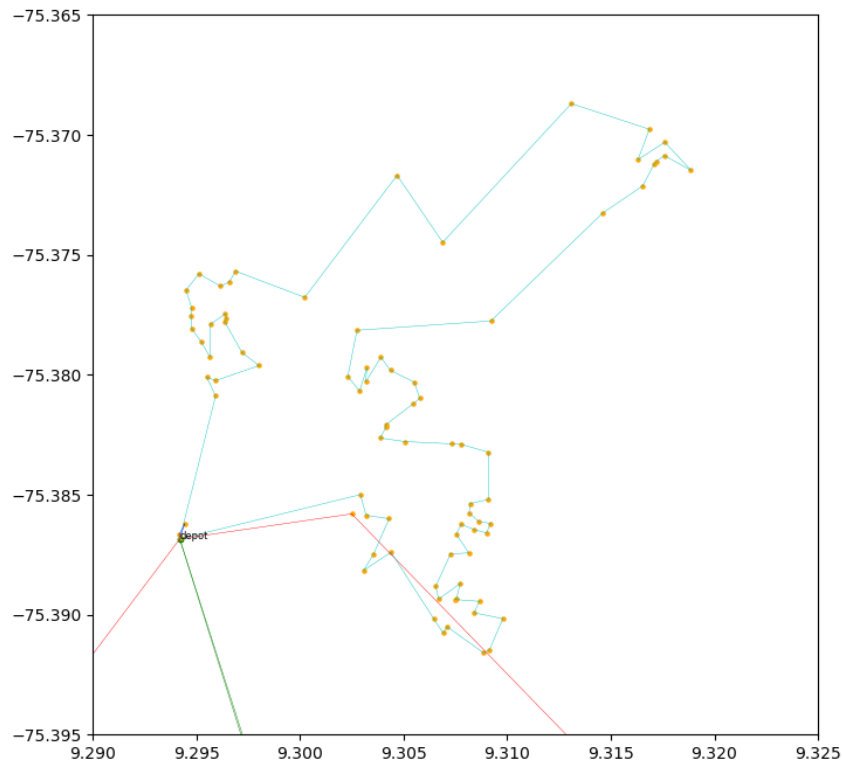
Para facilitar la comprensión de la solución se presenta la representación euclidiana de las rutas propuestas por el Algoritmo inicial de la empresa.

Figura 3: Solución Problema 1: Algoritmo inicial



Teniendo en cuenta la alta dispersión de algunos puntos de entrega, se presenta el siguiente gráfico con el fin de mostrar la secuencia en la mayor concentración de nodos:

Figura 4: Solución Problema 1: Algoritmo inicial. Detalle en concentración de nodos



Puede notarse como la distribución de la ruta 4 (la cuál abarca la mayor cantidad de nodos) presenta una secuencia sin cruces entre líneas, lo cual es deseable en términos de optimalidad, esto se da debido a que la distancia que minimiza el algoritmo evaluado corresponde a la euclidiana entre puntos, sin embargo en la realidad no es posible cubrir las rutas en esa dirección exacta, por lo que de cara a la evaluación del tiempo de la ruta, no necesariamente se garantiza que el no cruce de líneas sea una mejor versión de la solución.

El costo asociado a este programa es de 261.4.

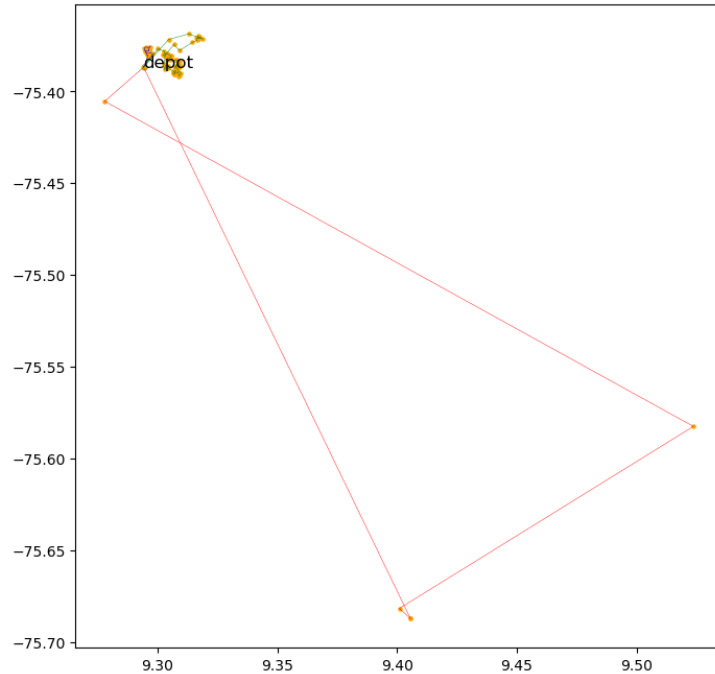
■ Algoritmo propuesto Problema 1

Por su parte el algoritmo propuesto encontró como mejor solución la siguiente secuencia:

- $R1 = [1, 2, 3, 86, 85, 80, 79, 78, 81, 83, 82, 84, 72, 71, 73, 74, 75, 76, 77, 88, 87, 89]$
- $R2 = [70, 56, 53, 52, 10, 9, 14, 13, 11, 12, 15, 28, 27, 26, 25, 24, 18, 17, 16, 22, 23, 19, 20, 21, 30, 33, 36, 37, 32, 31, 29, 38, 35, 34, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 55, 54, 68, 57, 58, 59, 65, 61, 60, 66, 62, 64, 63, 67, 69]$
- $R3 = [7, 8, 4, 6, 5]$

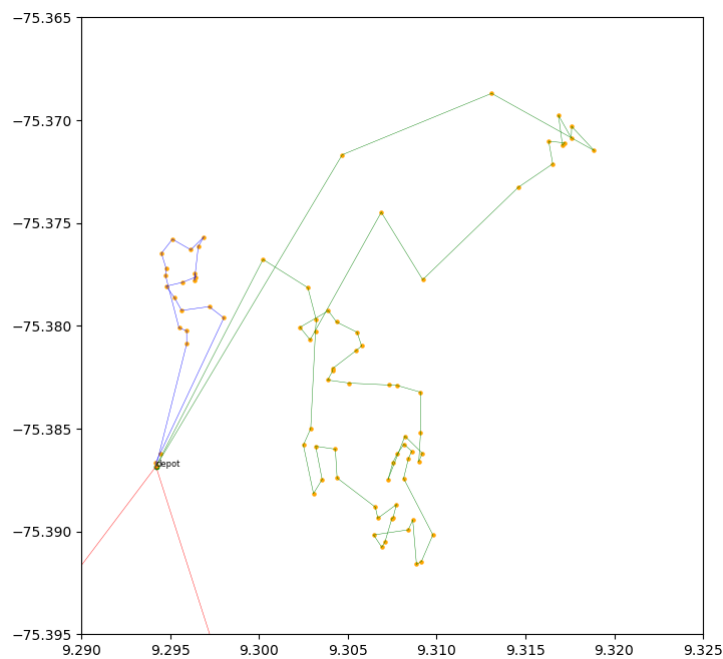
Es importante resaltar que para este problema, el algoritmo propuesto reduce en una unidad la cantidad de vehículos utilizados, pasando de 4 a 3 rutas. El gráfico de la distribución del programa de entregas es el siguiente:

Figura 5: Solución Problema 1: Algoritmo propuesto



Puede evidenciarse como la solución plantea visitar todos los nodos lejanos con el mismo vehículo *R3*, manera por la cual se requiere un vehículo menos para solucionar el problema. Se presenta ahora el detalle en las 2 rutas de mayor cantidad de nodos:

Figura 6: Solución Problema 1: Algoritmo propuesto. Detalle en concentración de nodos



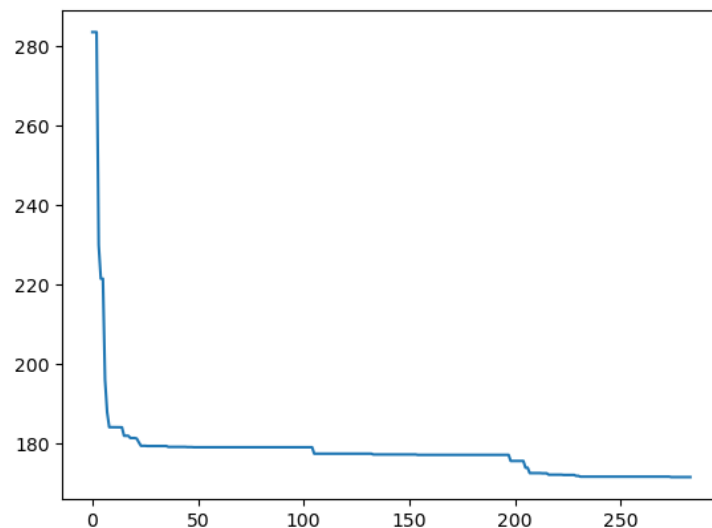
Es de resaltar como la ruta $R1$, toma una mayor cantidad de nodos y atiende un sector de la concentración de clientes, mientras la ruta $R2$ se concentra en los nodos del lado derecho de la imagen, caso que no ocurrió en la solución inicial.

En esta imagen pueden notarse algunos cruces en las líneas que cubren las distancias euclidianas entre nodos, sin embargo como se mencionó anteriormente, los tiempos entre nodos están considerados según la mejor ruta física por la cual se puede dirigir un vehículo de un lugar a otro, por lo que estos cruces son válidos para encontrar mejores soluciones. Se hicieron algunas simulaciones para evaluar si evitar algunos cruces mejoraba la solución, y en ningún caso se encontró positiva dicha evaluación.

El costo asociado a la ruta que resuelve este problema es de 171.4 minutos.

Este problema fue utilizado para calibrar el parámetro asociado al número de iteraciones a realizar, para esto se evaluó el número de iteraciones y como disminuía el costo de la solución en el tiempo, encontrando el siguiente resultado:

Figura 7: Evolución del costo de la solución



Luego de las 200 iteraciones se encuentran mejoras mínimas, sin encontrar una solución mejor luego de la iteración 274.

3.5.2. Problema 2

- Algoritmo inicial Problema 2

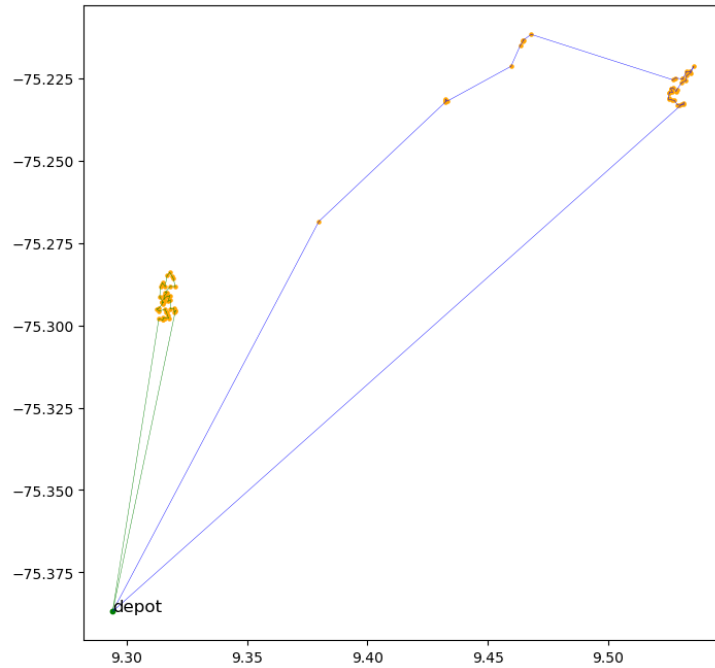
La solución al problema 2 generado por la empresa contiene la siguiente distribución:

- $R1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45]$

- $R2 = [46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89]$

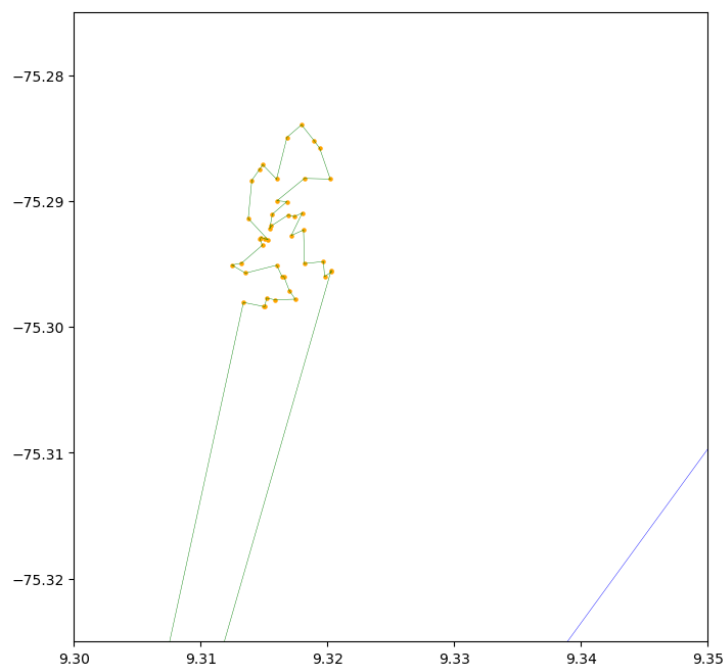
Para facilitar la comprensión de la solución se presenta la representación euclidiana de las rutas propuestas por el Algoritmo inicial de la empresa.

Figura 8: Solución Problema 2: Algoritmo inicial



La secuencia del lado izquierdo de la gráfica que cuenta con una alta cantidad de nodos se representa de la siguiente manera:

Figura 9: Solución Problema 2: Algoritmo inicial. Detalle en concentración de nodos



De nuevo puede notarse como la distribución de la ruta $R1$ presenta una secuencia sin cruces entre líneas, como es lo esperado para el algoritmo utilizado, el cual minimiza distancias euclidianas.

El costo asociado a este programa es de 124.7.

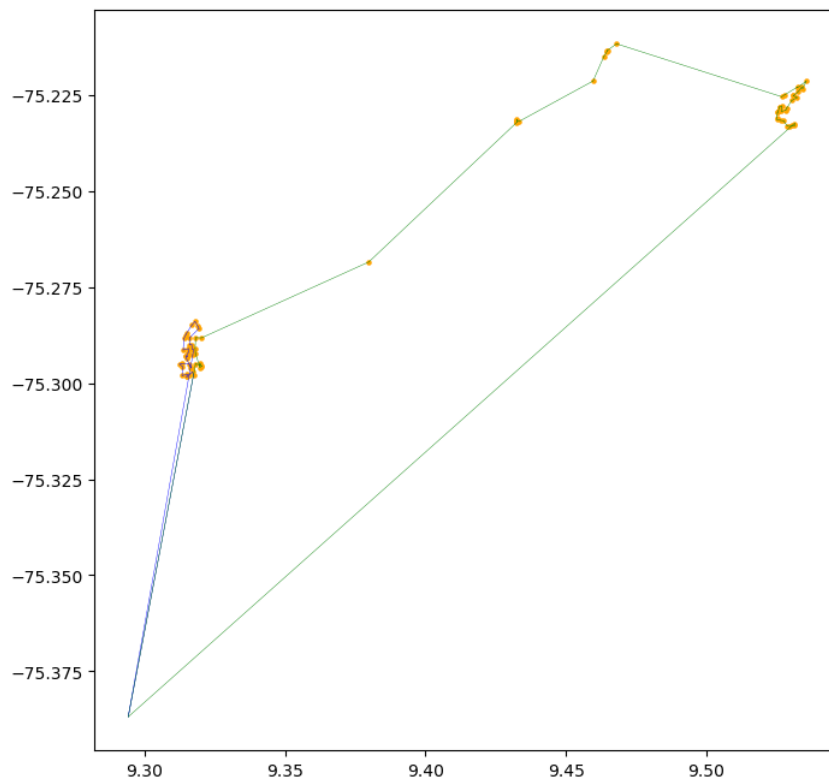
■ Algoritmo propuesto Problema 2

El algoritmo propuesto encontró como mejor solución al Problema 2 la siguiente secuencia:

- $R1 = [51, 52, 50, 48, 47, 49, 46, 56, 57, 58, 55, 53, 54, 59, 60, 61, 62, 63, 78, 79, 77, 64, 65, 66, 67, 69, 70, 71, 72, 68, 75, 76, 80, 81, 83]$
- $R2 = [85, 89, 88, 86, 87, 84, 82, 74, 73, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 26, 27, 23, 22, 25, 24, 21, 28, 29, 20, 19, 18, 17, 13, 16, 15, 14, 12, 10, 11, 9, 8, 7, 6, 5, 4, 3, 2, 1]$

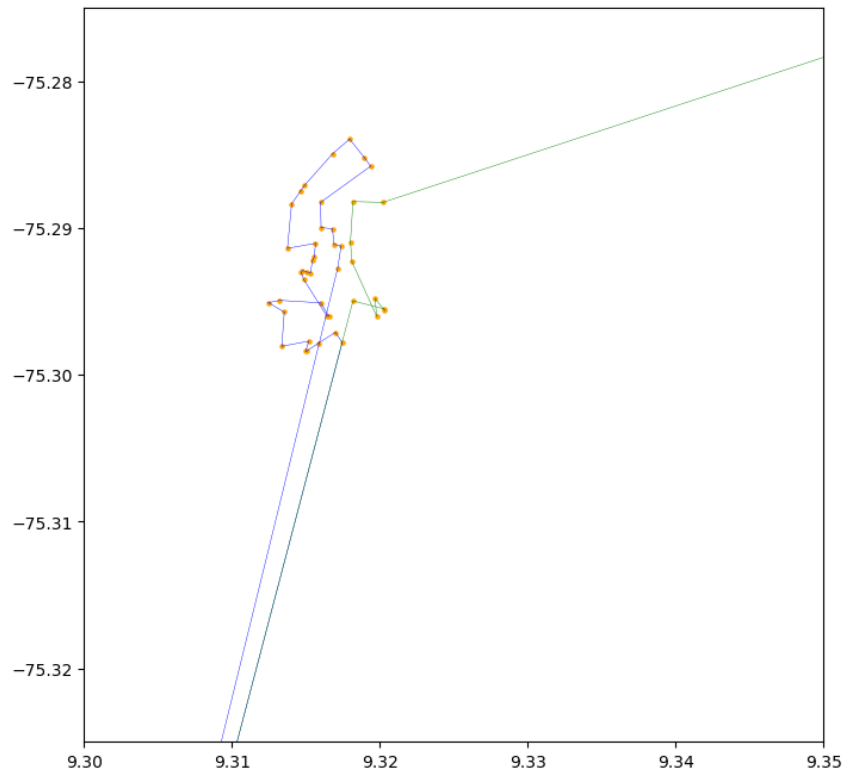
El gráfico de la distribución del programa de entregas es el siguiente:

Figura 10: Solución Problema 2: Algoritmo propuesto



En este caso la ruta $R1$ visita algunos nodos del sector izquierdo de la imagen, como se puede notar en la gráfica siguiente, que presenta esta zona con mayor detalle:

Figura 11: Solución Problema 2: Algoritmo propuesto. Detalle en concentración de nodos



Para este caso, la ruta $R2$ prioriza la entrega de algunos nodos que se encontraban en último lugar en la secuencia propuesta por el Algoritmo inicial permitiendo una reducción en el costo total del programa el cual es de 118.5.

3.5.3. Problema 3

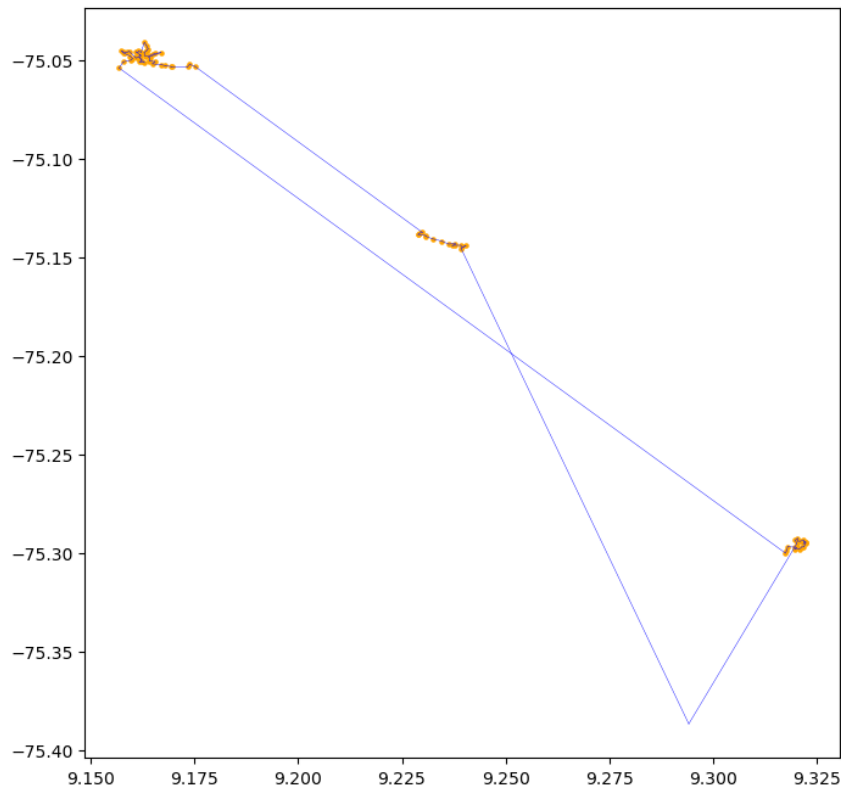
■ Algoritmo inicial Problema 3

La solución al problema 3 generado por la empresa contiene la siguiente distribución:

- $R1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89]$

Puede observarse que toda la demanda puede ser cubierta por una misma ruta y que el algoritmo considera que lo mejor es atender la demanda con un mismo vehículo. Para facilitar la comprensión de la solución se presenta la representación euclidiana de las rutas propuestas por el Algoritmo inicial de la empresa.

Figura 12: Solución Problema 3: Algoritmo inicial



La distribución de la ruta $R1$ presenta una secuencia sin cruces entre líneas en las zonas pobladas, con un cruce en los recorridos entre zonas. Sin embargo parece ser una buena solución.

El costo asociado a este programa es de 141.4.

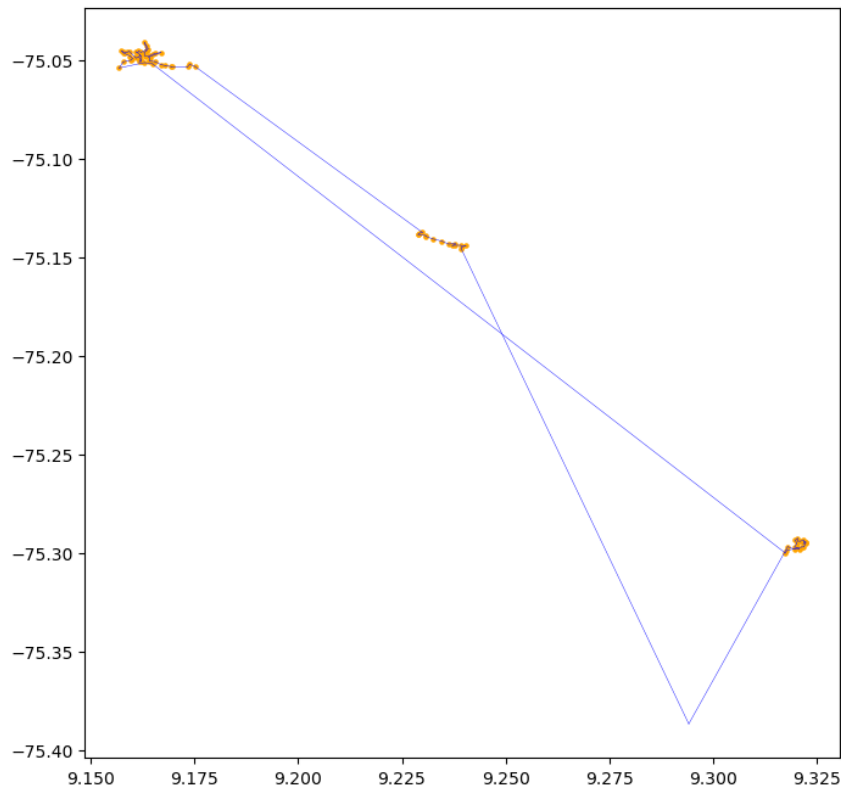
■ Algoritmo propuesto Problema 3

El algoritmo propuesto encontró como mejor solución al Problema 3 la siguiente secuencia:

- $R1 = [19, 16, 15, 17, 18, 1, 2, 3, 4, 5, 6, 7, 8, 10, 9, 11, 12, 13, 14, 20, 21, 66, 37, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 39, 38, 40, 45, 41, 42, 43, 44, 46, 47, 48, 49, 52, 51, 50, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 88, 87, 89]$

El gráfico de la distribución del programa de entregas es el siguiente:

Figura 13: Solución Problema 3: Algoritmo propuesto



El resultado es bastante similar a lo propuesto por el Algoritmo inicial, con algunos pequeños cambios en las poblaciones. El algoritmo propuesto logra de nuevo encontrar una mejor solución con una función de costo de 138.7.

3.5.4. Problema 4

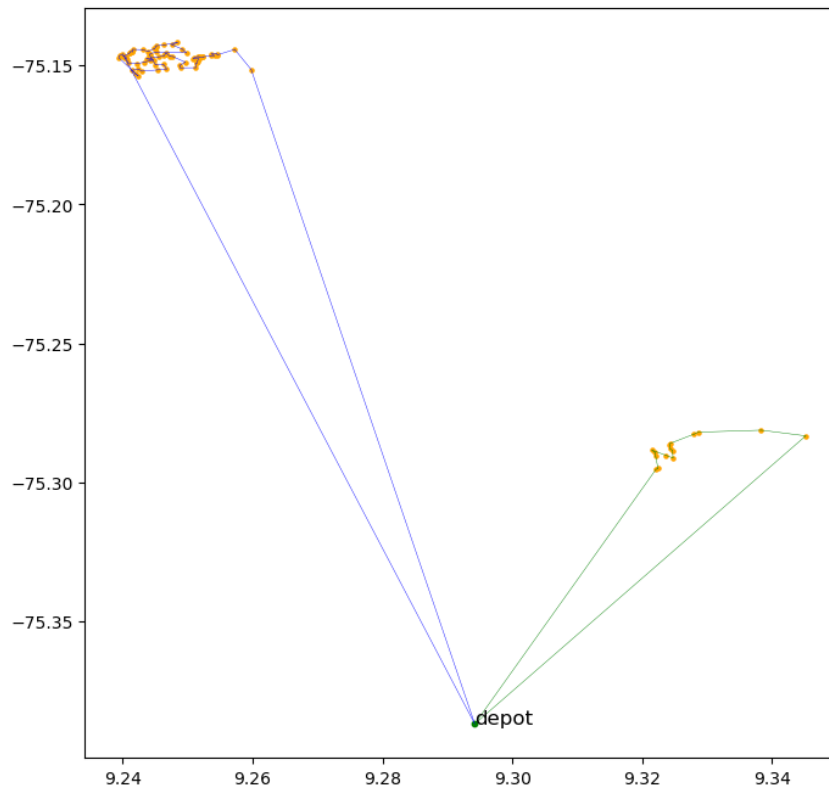
■ Algoritmo inicial Problema 4

La solución al problema 4 generado por la empresa contiene la siguiente distribución:

- $R1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74]$
- $R2 = [75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89]$

Para facilitar la comprensión de la solución se presenta la representación euclidiana de las rutas propuestas por el Algoritmo inicial de la empresa.

Figura 14: Solución Problema 4: Algoritmo inicial



La solución propuesta genera dos rutas dirigidas hacia poblaciones independientes para realizar las entregas. De nuevo de forma euclidiana parece ser una buena solución. Su función de costo asociado es de 133.0.

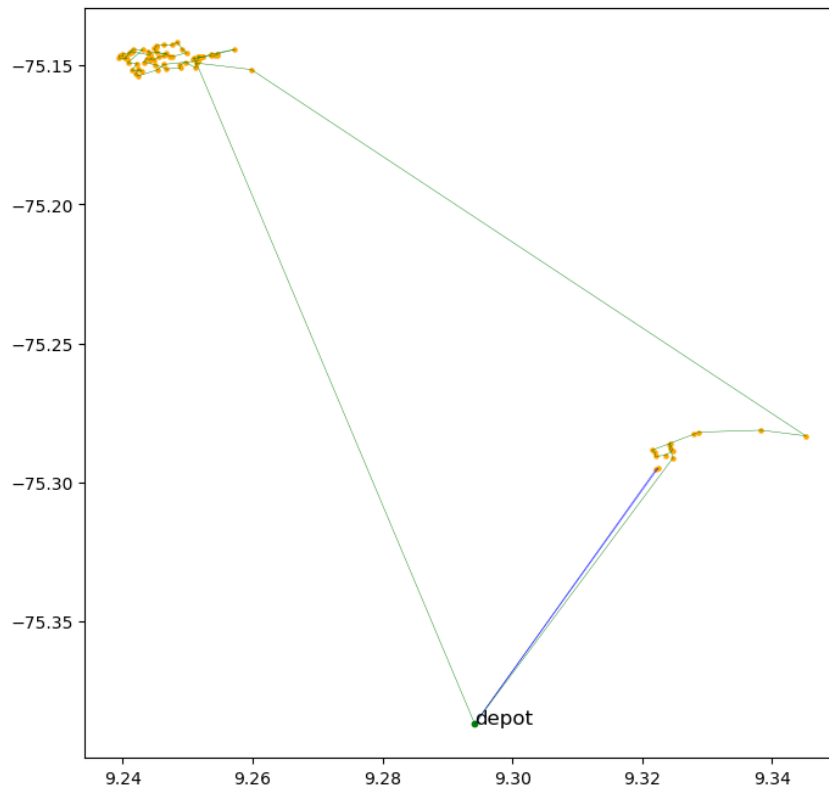
■ Algoritmo propuesto Problema 4

El algoritmo propuesto encontró como mejor solución al Problema 4 la siguiente secuencia:

- $R1 = [76, 75]$
- $R2 = [82, 83, 85, 84, 81, 80, 77, 78, 79, 86, 87, 88, 89, 74, 58, 57, 56, 53, 54, 55, 10, 12, 39, 43, 42, 41, 44, 45, 40, 46, 47, 49, 50, 51, 52, 16, 17, 19, 18, 20, 21, 22, 23, 24, 15, 48, 14, 13, 25, 28, 30, 31, 32, 34, 35, 36, 3, 2, 33, 1, 29, 27, 26, 37, 38, 6, 8, 7, 5, 4, 9, 11, 59, 60, 65, 64, 63, 66, 67, 68, 69, 72, 71, 73, 70, 62, 61]$

El gráfico de la distribución del programa de entregas es el siguiente:

Figura 15: Solución Problema 4: Algoritmo propuesto



En este caso la ruta $R2$ visita algunos nodos del sector derecho de la imagen, aprovechando la capacidad disponible en el vehículo, mientras la ruta $R1$ atiende solo dos clientes pero que concentran gran parte de la demanda a cubrir. Esto permite encontrar una mejor solución que la inicial con un costo de 126.2.

3.5.5. Problema 5

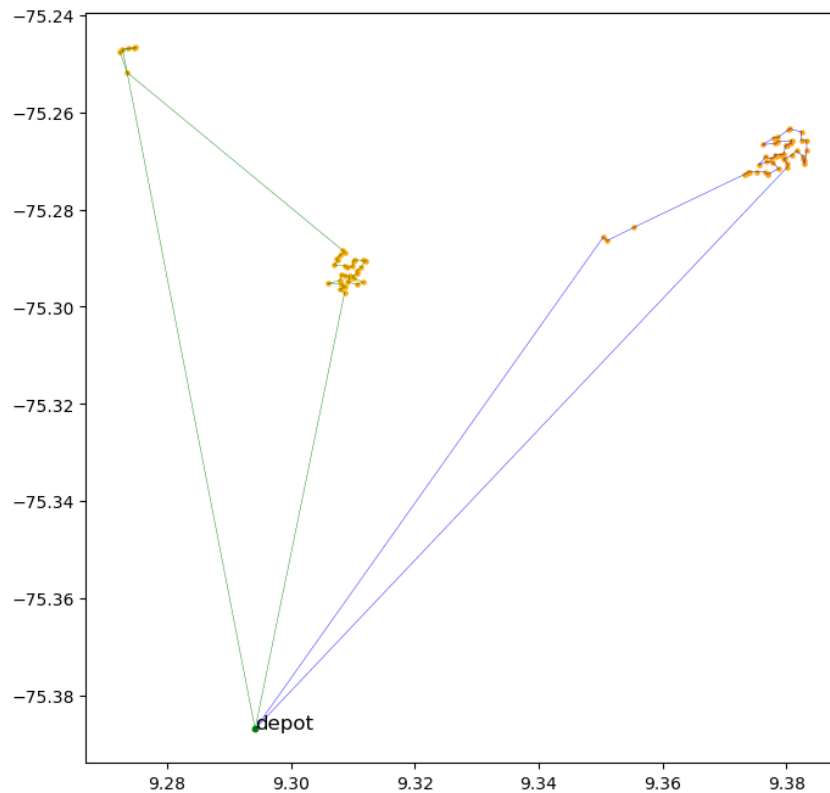
- Algoritmo inicial Problema 5

La solución al problema 5 generado por la empresa contiene la siguiente distribución:

- $R1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]$
- $R2 = [51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89]$

Para facilitar la comprensión de la solución se presenta la representación euclidiana de las rutas propuestas por el Algoritmo inicial de la empresa.

Figura 16: Solución Problema 5: Algoritmo inicial



Para este caso el algoritmo atiende con la ruta $R2$ el sector izquierdo de la imagen y con la ruta $R1$ el sector derecho. Teniendo en cuenta la ubicación geoespacial y la demanda de los clientes del sector parece entregar una buena solución. Su función de costo es de 102.6.

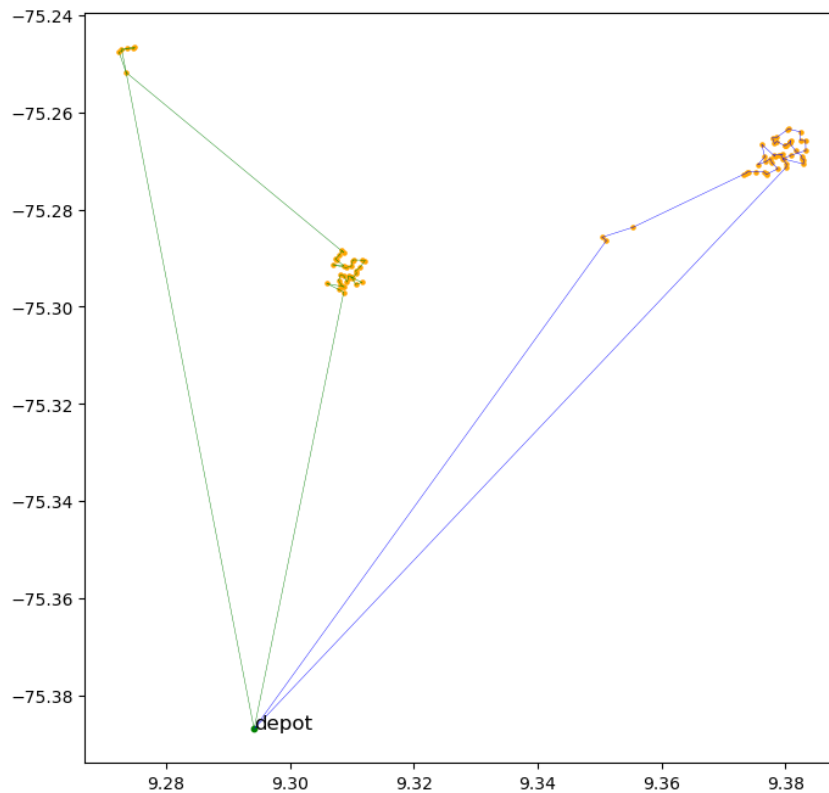
■ Algoritmo propuesto Problema 5

El algoritmo propuesto encontró como mejor solución al Problema 5 la siguiente secuencia:

- $R1 = [2, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 17, 14, 15, 16, 32, 18, 19, 20, 22, 21, 47, 42, 41, 43, 44, 45, 26, 29, 28, 27, 25, 24, 23, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 46, 48, 49, 50]$
- $R2 = [51, 52, 56, 53, 54, 55, 57, 58, 59, 62, 61, 60, 64, 63, 67, 66, 65, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 77, 79, 80, 81, 83, 82, 84, 85, 86, 87, 88, 89]$

El gráfico de la distribución del programa de entregas es el siguiente:

Figura 17: Solución Problema 5: Algoritmo propuesto



Tenemos un resultado muy similar al encontrado por el algoritmo inicial, con algunos cambios en las secuencias internas una vez se atiende la concentración en poblaciones, lo que hace que mejore el costo de la solución establecido en 98.8

4. Recomendaciones y conclusiones

Los resultados obtenidos en los problemas planteados se resumen así:

Problema	Algoritmo Propuesto	Algoritmo Inicial	Diferencia Porcentual
P1	171.4	261.4	-34.4 %
P2	118.5	124.7	-5.0 %
P3	138.7	141.3	-1.8 %
P4	126.2	133.0	-5.1 %
P5	98.8	102.6	-3.7 %

Cuadro 2: Resultados experimentales.

Es de notar como el Algoritmo propuesto en este trabajo mejora los resultados obtenidos por la empresa con su metodología actual, apalancado principalmente en la capacidad de tener en cuenta distancias no lineales entre nodos y la dependencia que existe sobre la hora del día en que se están transitando las vías. Además el algoritmo utilizado despliega la posibilidad de incluir las ventanas de tiempo, algo que va en vía de mejorar los niveles de servicio alcanzados por la organización. En promedio se obtuvo una reducción del 10 % en

el tiempo de transporte en los 5 escenarios planteados. Teniendo en cuenta los problemas 2 a 5 se logró una reducción del 3,9%, excluyendo del cálculo el problema 1, el cual se considera un atípico en la solución propuesta por el algoritmo inicial.

Los puntos claves para obtener buenos resultados pueden resumirse en: garantizar que la población inicial tiene al menos un individuo de buena calidad, operadores de cruce y mutación que permitan mejorar con el paso del tiempo a la población y el uso del algoritmo split que obtiene las sub rutas óptimas de cada individuo. El split es un elemento clave en la solución del problema, ya que le agrega un componente de optimalidad a un proceso estocástico como es el proceso evolutivo. En este trabajo se desarrolla una metodología para incluir la dependencia del tiempo como factor a evaluar en el split y al mismo tiempo considerar la restricción de máximo número de vehículos.

Teniendo en cuenta que no se conocen las soluciones óptimas de los problemas, la comparación con la herramienta actual que dispone la empresa resulta ser un buen indicador para evaluar los resultados del modelo, ya que como se pudo ver gráficamente todas las soluciones planteadas contienen resultados lógicos en la secuenciación de las rutas propuestas y aunque presente limitaciones en las restricciones que puede tener en cuenta es una buena aproximación como solución de cada uno de los problemas evaluados. Cabe resaltar que las soluciones fueron evaluadas teniendo en cuenta la dependencia en el tiempo, caso que no es posible probar en el algoritmo inicial implementado en la empresa.

Para colocar la herramienta en productivo, se recomienda desplegar una API que consuma directamente de una herramienta con acceso a las distancias en tiempo entre nodos, para mantener actualizados los tiempos de desplazamiento entre ellos. Así mismo, es recomendable actualizar los mapas para mejorar la visualización de los resultados del problema. De acuerdo a la dimensión de nodos que se deseen evaluar, ambas herramientas serán de carácter pago.

Referencias

- [1] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):48–58, March 1985.
- [2] Didem Cinar, Gakis Konstantinos, and Panos Pardalos. A 2-phase constructive algorithm for cumulative vehicle routing problems with limited duration. *Expert Systems With Applications*, 56:48–58, December 2016.
- [3] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, October 1959.
- [4] G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, August 1964.
- [5] Shi Yi Tan and Wei-Chang Yeh. The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences*, 11:1–28, March 2021.
- [6] Anand Subramanian, Puca Vaz, Eduardo Uchoa, and Luiz Satoru. A hybrid algorithm for the heterogeneous fleet vehicle routing. *European Journal of Operational Research*, pages 285–295, March 2012.

- [7] Chryssi Malandraki and Mark S Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200, August 1992.
- [8] K Ganesh, Sam Nallathambi, and T.T. Narendran. Variants, solution approaches and applications for vehicle routing problems in supply chain: agile framework and comprehensive review. *Agile Systems and Management*, 2(1):50–75, August 2007.
- [9] A. W. J. Kolen, A. H. G. Rinnooy Kan, and H. W. J. M. Trienekens. Vehicle routing problem with time windows. *Operations Research*, 35(2):266–273, 1987.
- [10] Philippe Augerat, Jose Manuel Belenguer, Enrique Benavent, and Angel Corberán. Computational results with a branch and cut code for the capacitated vehicle routing problem. *Université Joseph Fourier*, 1995.
- [11] Pecin Diego, Claudio Contardo, Guy Desaulniers, and Eduardo Uchoa. New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29(3):489–502, June 2017.
- [12] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, September 1977.
- [13] S Kirkpatrick, CD. Jr. Gellat, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [14] Mohammad Mojtahedi, Amir M. Fathollahi-Fard, Reza Tavakkoli-Moghaddam, and Sidney Newton. Sustainable vehicle routing problem for coordinated solid waste management. *Journal of Industrial Information Integration*, 23:100220, March 2021.
- [15] A.M Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam. The social engineering optimizer (seo). *Engineering Applications of Artificial Intelligence*, 72:267–293, May 2018.
- [16] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [17] Houming Fan, Yueguang Zhang, Tian Pajun, and Hao Fan. Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance. *Computers and Operations Research*, 129:105211, January 2021.
- [18] Abdeljawed Sadok. A genetic local search algorithm for the capacitated vehicle routing problem. *International Journal of Advanced Computer Research*, 10(48):105–115, February 2020.
- [19] G.A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, February 1958.
- [20] Barma Sarathi, Joydeep Dutta, and Anupam Mukherjee. A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem. *Decision Making: Applications in Management and Engineering*, 2(2):112–125, April 2019.

- [21] Ali Haghani and Soojung Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers Operations Research*, 32:2959–2986, April 2005.
- [22] Houming Fan, Yueguang Zhang, Pajun Tian, Yingchun Lv, and Hao Fan. Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance. *Computers Operations Research*, 129:1–14, January 2021.
- [23] Prins Christian. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers Operations Research*, 31(12):1985–2002, October 2004.
- [24] Thibaut Vidal. Technical note: Split algorithm in $o(n)$ for the capacitated vehicle routing problem. *Computers Operations Research*, 69:40–47, May 2016.
- [25] H.Murat Afsar, Christian Prins, and Andrea Cynthia Santos. Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. *International Transactions in Operational research*, 21:153–175, June 2014.
- [26] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT Press, 2009.
- [27] B Jaumard, F Semet, and T Vovor. *A two-phase resource constrained shortest path algorithm for acyclic graphs*. GERAD research report G-96-48, 1996.
- [28] Zaineb Chelly Dagdia and Miroslav Mirchev. *When Evolutionary Computing Meets Astro- and Geoinformatics*. Elsevier, 2020.