

**Modelo de optimización para el problema de gestión de Inventario y Ruteo de dos escalones con ventanas de tiempo (2eIRPTW) aplicado al diseño de la red de distribución de productos perecederos**

**Mateo Rios Pérez**

**Trabajo de Grado para Optar al Título de Ingeniero Industrial**

**Director**

**Javier Eduardo Arias Osorio**

**Magister en Administración**

**Codirector**

**Edgar Eduardo Córdoba Sarmiento**

**Ingeniero Industrial M.Sc.**

**Universidad Industrial de Santander**

**Facultad de Ingenierías Fisicomecánicas**

**Escuela de Estudios Industriales y Empresariales**

**Bucaramanga**

**2025**

### **Agradecimientos**

*A Dios, por guiar mis pasos, renovar mis fuerzas y acompañarme durante todo el camino emprendido.*

*A Alba, mi mamá, por apoyarme y motivarme a no rendirme en ningún punto.*

*A Gilberto, mi papá, por estar siempre presente y por confiar en mis capacidades.*

*A Vanessa, mi hermana, porque aún a la distancia pude sentir su interés y cariño.*

*A Valentina, mi novia, por creer en mí incluso cuando a mí me costaba hacerlo.*

*A los profesores Javier Arias y Edgar Córdoba, mis directores, por su orientación para poder culminar este proyecto satisfactoriamente.*

*A todas las amistades hechas en el camino, porque muchas veces ayudaron a que este foráneo se sintiera como en casa.*

**Tabla de Contenido**

Introducción .....	15
1. Generalidades de la investigación .....	18
1.1 Planteamiento del problema.....	18
1.2 Metodología .....	20
1.2.1. Etapa 1: Revisión de literatura y definición del problema.....	20
1.2.2. Etapa 2: Formulación del modelo matemático .....	21
1.2.3. Etapa 3: Desarrollo del algoritmo de solución .....	21
1.2.4. Etapa 4: Validación y evaluación del algoritmo .....	22
1.2.5. Etapa 5: Síntesis de resultados .....	22
2. Objetivos .....	23
2.1. Objetivo General .....	23
2.2. Objetivos Específicos.....	23
3. Revisión de literatura y Análisis bibliométrico .....	24
3.1. Análisis bibliométrico .....	24
3.2. Revisión de literatura .....	24
4. Marco de Referencia .....	31
4.1. Marco de antecedentes .....	31
4.2. Marco teórico.....	34
4.2.1. Cadena de suministro.....	34
4.2.2. Logística.....	34
4.2.3. Red de distribución. ....	34
4.2.3.1. Decisiones de inventario.....	35

4.2.3.2. Decisiones de transporte. ....	35
4.2.3.2.1 Ventanas de tiempo. ....	35
4.2.4. Transbordos laterales. ....	36
4.2.5. Teoría de optimización. ....	36
4.2.6. Optimización matemática. ....	36
4.2.6.1. Variables de decisión. ....	36
4.2.6.2. Función objetivo. ....	36
4.2.6.3. Restricciones. ....	36
4.2.7. Modelos matemáticos. ....	37
4.2.8. Programación lineal entera mixta (PLEM). ....	37
4.2.9. Optimización combinatoria (COP). ....	37
4.2.10. Principales problemas de optimización combinatoria. ....	38
4.2.10.1. Problema de ruteo de vehículos (VRP) ....	38
4.2.10.2. Problema de inventario y ruteo (IRP) ....	38
4.2.10.3. Problema de inventario y ruteo en dos escalones (2eIRP). ....	39
4.2.11. Métodos de solución para problemas tipo COP. ....	39
4.2.11.1. Métodos exactos. ....	39
4.2.11.1.1. Método Simplex. ....	39
4.2.11.1.2. Método Branch and Bound (B&B). ....	39
4.2.11.2. Métodos aproximados. ....	40
4.2.11.3. Heurísticas. ....	40
4.2.11.4. Metaheurísticas. ....	40
4.2.11.4.1. Recocido simulado (Simulated annealing, SA). ....	41

4.2.11.4.2. Búsqueda tabú (Tabú search, TS). .....	41
4.2.11.4.3. Búsqueda en vecindario variable (Variable Neighborhood search, VNS). .....	41
4.2.11.4.4. Procedimiento de búsqueda voraz adaptativa aleatoria (Greedy Randomized Adaptive search procedure, GRASP). .....	41
4.2.11.4.5. Algoritmos genéticos (Genetic algorithm, GA). .....	41
4.2.11.4.6. Optimización por Enjambre de Partículas (Particle Swarm optimization, PSO). .....	42
4.2.11.4.7. Optimización por colonia de hormigas (Ant Colony optimization, ACO). .....	42
5. Modelo para el 2eIRPTW de productos perecederos .....	42
5.1. Descripción del problema .....	42
5.2. Formulación matemática .....	45
5.3. Representación gráfica de la red de distribución .....	52
5.4. Análisis de sensibilidad .....	56
5.4.1. Periodos .....	57
5.4.2. Distribuidores .....	58
5.4.3. Clientes .....	60
5.4.4. Flota de vehículos K .....	61
5.4.5. Flota de vehículos M .....	62
5.4.6. Ventanas de tiempo .....	63
6. Descripción de la metaheurística .....	64
6.1. Recocido simulado .....	65
6.1.1. Solución factible .....	66
6.1.2. Función objetivo .....	66
6.1.3. Solución inicial .....	66

6.1.4. Vecindario.....	66
6.1.5. Nueva solución candidata .....	67
6.1.6. Temperatura .....	67
6.1.7. Criterio de aceptación .....	67
6.1.8. Enfriamiento .....	68
6.1.9. Relación con cadenas de Markov .....	68
6.1.10. Operador de reparación.....	69
6.1.11. Criterio de terminación o parada .....	69
6.1.12. Solución final.....	70
6.1.13. Representación gráfica del SA .....	70
7. Diseño del algoritmo de Recocido Simulado .....	71
7.1. Etapa 1: Inicialización y generación de solución inicial .....	72
7.1.1. Inicialización .....	72
7.1.2. Generación de la solución inicial .....	72
7.1.2.1. Heurística constructiva para instancias pequeñas .....	73
7.1.2.2. Heurística First Fit Decreasing (FFD) para instancias grandes.....	74
7.1.3. Función de evaluación y modelo de costos .....	77
7.1.3.1. Envejecimiento de inventario y producción .....	77
7.1.3.2. Evaluación de rutas del primer escalón .....	77
7.1.3.3. Evaluación de rutas del segundo escalón.....	78
7.1.3.4. Evaluación de costo de almacenamiento .....	78
7.2. Etapa 2: Estrategia de diversificación .....	79
7.3. Etapa 3: Generación y evaluación de vecinos .....	81

7.3.1. Generación de vecinos .....	81
7.3.2. Evaluación y reparación de vecinos .....	83
7.3.2.1. Sobrecarga de vehículos. ....	83
7.3.2.2. Ajuste de inventarios negativos.....	84
7.3.2.3. Demanda insatisfecha. ....	84
7.3.2.4. Penalización en soluciones inviables. ....	85
7.4. Etapa 4: Criterio de aceptación de soluciones vecinas.....	86
7.5. Etapa 5: Mantenimiento del bucle de Recocido Simulado .....	88
7.5.1. Enfriamiento y control de iteraciones. ....	89
7.5.2. Criterio de parada. ....	91
7.6. Etapa 6: Finalización y selección de la mejor solución .....	92
8. Experimentación y resultados.....	93
8.1. Instancias propuestas.....	93
8.2. Análisis de resultados.....	98
8.2.1. Instancia 1 (1-2-2).....	100
8.2.2. Instancia 2 (1-2-3).....	101
8.2.3. Instancia 3 (1-4-2).....	101
8.2.4. Instancia 4 (1-4-8).....	102
8.2.5. Instancia 5 (1-4-10).....	106
8.2.6. Instancia 6 (1-5-12).....	112
9. Conclusiones.....	118
10. Recomendaciones .....	121
Referencias bibliográficas.....	123

**Lista de Tablas**

Tabla 1. Cumplimiento de objetivos.....	17
Tabla 2. Revisión de tipos de percibibilidad .....	28
Tabla 3. Formulación matemática para problemas de optimización combinatoria .....	38
Tabla 4. Variación de periodos .....	57
Tabla 5. Variación de distribuidores.....	59
Tabla 6. Variación de clientes.....	60
Tabla 7. Variación de vehículos del primer escalón.....	61
Tabla 8. Variación de vehículos del segundo escalón .....	62
Tabla 9. Variación de ventanas de tiempo .....	63
Tabla 10. Función de probabilidad de aceptación .....	87
Tabla 11. Parámetros de estructura – Instancias pequeñas .....	94
Tabla 12. Parámetros de estructura – Instancias grandes.....	94
Tabla 13. Parámetros de demanda – Todas las instancias.....	95
Tabla 14. Parámetros - capacidad de flotas de vehículos.....	95
Tabla 15. Parámetros – capacidad de almacenamiento en nodos .....	95
Tabla 16. Parámetros inventario inicial en nodos – Todas las instancias .....	96
Tabla 17. Parámetros – capacidad de almacenamiento en nodos .....	96
Tabla 18. Parámetros – costos de envío .....	97
Tabla 19. Parámetros – costos de almacenamiento .....	97
Tabla 20. Instancias evaluadas.....	98
Tabla 21. Factores diseño de experimentos .....	99
Tabla 22. Resultados instancia 1.....	100

Tabla 23. Resultados instancia 2.....	101
Tabla 24. Resultados instancia 3.....	102
Tabla 25. Resultados instancia 4.....	103
Tabla 26. Resultados instancia 5.....	107
Tabla 27. Resultados instancia 6.....	112

### Lista de Figuras

Figura 1. Índice de Desempeño Logístico global (2023).....	19
Figura 2. Evolución IRP.....	25
Figura 3. Proceso de modelado matemático.....	37
Figura 4. Diagrama de la instancia 1 para $t = 1$ .....	53
Figura 5. Diagrama de la instancia 1 para $t = 2$ .....	54
Figura 6. Diagrama de la instancia 2 para $t = 1$ .....	55
Figura 7. Diagrama de la instancia 2 para $t = 2$ .....	56
Figura 8. Análisis de sensibilidad – Periodos de tiempo.....	58
Figura 9. Análisis de sensibilidad – Distribuidores.....	59
Figura 10. Análisis de sensibilidad – Clientes.....	60
Figura 11. Análisis de sensibilidad – Flota primer escalón.....	61
Figura 12. Análisis de sensibilidad – Flota segundo escalón.....	62
Figura 13. Diagrama de flujo general Recocido Simulado + Operador de reparación.....	70
Figura 14. Diagrama de flujo algoritmo SA mejorado con repair operator.....	71
Figura 15. Asignación de solución inicial.....	79
Figura 16. Diagrama de flujo etapa 2 SA+RO.....	79
Figura 17. Diagrama de flujo etapa 3 SA+RO.....	81
Figura 18. Repair operator – Sobrecarga de vehículos.....	84
Figura 19. Repair operator – Demanda insatisfecha.....	85
Figura 20. Diagrama de flujo etapa 4 SA+RO.....	86
Figura 21. Nueva solución $<$ solución actual.....	87
Figura 22. Nueva solución $\geq$ solución actual.....	88

Figura 23. Evolución de temperatura + historial de aceptación .....	89
Figura 24. Evolución del historial del valor de la F.O. ....	91
Figura 25. Mejor solución = solución actual.....	92
Figura 26. Diagrama de Pareto estandarizado .....	99
Figura 27. ANOVA para TC – Instancia 4 .....	103
Figura 28. ANOVA para FO – Instancia 4 .....	104
Figura 29. Efectos principales TC – Instancia 4.....	104
Figura 30. Efectos principales FO – Instancia 4.....	105
Figura 31. Interacciones TC – Instancia 4 .....	106
Figura 32. Interacciones FO – Instancia 4 .....	106
Figura 33. ANOVA para TC – Instancia 5 .....	107
Figura 34. ANOVA para FO – Instancia 5 .....	108
Figura 35. Efectos principales TC – Instancia 5.....	109
Figura 36. Efectos principales FO – Instancia 5.....	109
Figura 37. Interacciones TC – Instancia 5 .....	110
Figura 38. Interacciones FO – Instancia 5 .....	111
Figura 39. ANOVA para TC – Instancia 6 .....	112
Figura 40. ANOVA para FO – Instancia 6 .....	113
Figura 41. Efectos principales TC – Instancia 6.....	113
Figura 42. Efectos principales FO – Instancia 6.....	114
Figura 43. Interacciones TC – Instancia 6 .....	115
Figura 44. Interacciones FO – Instancia 6 .....	115

**Lista de Apéndices**

**(Ver en la carpeta comprimida *Anexos TG 2eIRPTW*)**

Apéndice A. PLEM Instancias pequeñas del 2eIRPTW

Apéndice B. SA\_RO Instancias propuestas en MATLAB

Apéndice C. Análisis estadístico Instancias grandes

Apéndice D. Análisis bibliométrico

Apéndice E. Catálogo de operadores de vecindad

Apéndice F. Artículo académico 2eIRPTW

## Resumen

**Título del proyecto:** Modelo de optimización para el problema de gestión de Inventario y Ruteo de dos escalones con ventanas de tiempo (2eIRPTW) aplicado al diseño de la red de distribución de productos perecederos\*

**Autor:** Mateo Rios Pérez\*\*

**Palabras clave:** Problema de inventario y ruteo de vehículos, dos escalones, ventanas de tiempo, productos perecederos, optimización matemática, logística, metaheurística, recocido simulado, operador de reparación, programación lineal entera mixta (PLEM).

### Descripción:

En el presente proyecto de investigación se estudia el problema de inventarios y ruteo de vehículos dentro de una red de distribución en el sector agrícola de dos escalones, en donde se aplican ventanas de tiempo en las entregas de la última milla, además de transportar productos perecederos. El problema en cuestión se programa inicialmente en GAMS haciendo uso de programación lineal entera mixta, en donde el objetivo del modelo es la minimización de los costos totales del sistema, incluyendo los costos asociados al transporte del primer y segundo escalón, costos de almacenamiento y una penalización asociada al estado de frescura del producto. Posterior a esto, se diseña una metaheurística para dar solución al problema, específicamente un algoritmo de recocido simulado, el cual es potenciado con un operador de reparación que permite ajustar las soluciones encontradas para que cumplan con las restricciones planteadas. Seguidamente se proponen 6 instancias en las que se evalúa el desempeño del modelo de optimización, en donde se presentan 3 instancias pequeñas que permiten comparar los métodos de solución, evaluando así la calidad de las soluciones generadas por el algoritmo, y 3 instancias más grandes que asemejan redes de distribución con una envergadura similar a las cadenas de suministro de la industria agrícola. Con el fin de validar el algoritmo propuesto, se realiza un diseño factorial  $2^2$  que permite entender el comportamiento y la influencia que tienen el número de iteraciones, la temperatura inicial del sistema y la interacción entre ellos, sobre la calidad de las soluciones encontradas, y el tiempo computacional requerido para llegar a ellas. Los resultados muestran que el algoritmo es capaz de generar soluciones muy buenas en tiempos computacionales similares a los observados en los métodos exactos, en donde se recomienda una temperatura inicial del sistema elevada, acompañada de suficientes iteraciones para potenciar la exploración del espacio de búsqueda.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Fisicomecánicas - Escuela de Estudios Industriales y Empresariales

Director: Javier Eduardo Arias Osorio - Codirector: Edgar Eduardo Córdoba Sarmiento.

### Abstract

**Title:** Optimization model for the two-echelon inventory and routing problem with time windows (2eIRPTW) applied to the design of a distribution network for perishable products. \*

**Author:** Mateo Rios Pérez\*\*

**Keywords:** Inventory routing problem, two-echelon, time windows, perishable products, mathematical optimization, logistics, metaheuristics, simulated annealing, repair operator, mixed integer linear programming (MILP).

### Description:

This research project studies the inventory and vehicle routing problem within a two-step distribution network in the agricultural sector, where time windows are applied in the last mile deliveries, in addition to transporting perishable products. The problem at hand is initially programmed in GAMS using mixed integer linear programming, where the objective of the model is the minimization of the total system costs, including the costs associated with the transportation of the first and second echelon, holding costs and a penalty associated with the freshness of the product. Subsequently, a metaheuristic is designed to solve the problem, specifically a simulated annealing algorithm, which is enhanced with a repair operator that allows adjusting the solutions found to meet the constraints. Next, 6 instances are proposed in which the performance of the optimization model is evaluated, where 3 small instances are presented that enable comparing the solution methods, thus evaluating the quality of the solutions generated by the algorithm. Then, 3 larger instances that resemble distribution networks with similar size to the supply chains of the agricultural industry are formulated. In order to validate the proposed algorithm, a  $2^2$  factorial design is performed to understand the behavior and influence of the number of iterations, the initial temperature of the system and the interaction between them, on the quality of the solutions found, and the computational time required to reach them. The results show that the algorithm is able to generate very good solutions in computational times similar than those observed in the exact methods, where a high initial system temperature is recommended, accompanied by enough iterations to enhance the exploration of the search space.

---

\* Degree project

\*\* Faculty of Physico-Mechanical Engineering. School of Industrial and Business Studies.

Director: Javier Eduardo Arias Osorio – Co-director: Edgar Eduardo Córdoba Sarmiento.

## Introducción

En esta era dinámica de la digitalización, la gestión de la cadena de suministro atraviesa un proceso transformador caracterizado por la integración de tecnologías avanzadas y la analítica de datos, cuyas fuerzas convergen para favorecer la toma de decisiones en este ámbito. En un horizonte desafiante donde la cadena de suministro enfrenta nuevos retos, miles de organizaciones en todo el mundo están constantemente aumentando la inversión de recursos y esfuerzos dirigidos a esta área de estudio con el fin de encontrar soluciones que permitan mejorar el rendimiento general de las cadenas de suministro y redes de distribución, y de manera simultánea, minimizar los costos logísticos asociados a este tipo de decisiones. El diseño y estructuración de una cadena de suministro robusta y eficiente se encuentra en el núcleo del éxito operativo de las empresas modernas. Este proceso implica diversos tipos de decisiones que se ubican dentro de las cuatro áreas principales de la gestión de cadena de suministro. Entre este tipo de decisiones se encuentra todo lo relacionado a la optimización de la localización, gestión de inventarios, producción y transporte (ruteo de vehículos). Cada una de estas áreas comprende a su vez elementos de carácter estratégico, táctico y/o operativo que generalmente están relacionados con el horizonte de tiempo en el cual se planea implementar la decisión, acompañada de la materialización de sus efectos. Cada decisión requiere diferentes enfoques de planificación, evaluación de riesgos y flexibilidad, adaptándose al horizonte temporal correspondiente.

En la última década, muchas compañías han optado por combinar e integrar sus políticas de transporte y gestión de inventario, en lugar de tratar este tipo de decisiones como dos subproblemas diferentes. El afrontar el problema de esta forma ha demostrado que se pueden generar mejores resultados en términos generales, donde es posible alcanzar mejores niveles de servicio, además de reducciones significativas en los costos totales.

Para este proyecto de investigación, se propone desarrollar un modelo metaheurístico que permita dar solución al problema de optimización de inventario y ruteo de una red de distribución de dos escalones, específicamente para productos perecederos en el sector agrícola. Asimismo, se propone la inclusión de ventanas de tiempo en el modelo, con el fin de priorizar la frescura del producto en toda la cadena de suministro para garantizar la satisfacción del cliente. Con la ayuda de datos sintéticos se planea evaluar el modelo en diferentes instancias, incluyendo la adición de ventanas de tiempo duras, en donde se agrega una penalización de costo a la función objetivo dependiendo del estado en el que se encuentra el producto que fue entregado. La decisión de evaluar el modelo en estas instancias se toma basada en la evidencia encontrada en la literatura, que prueba que estas estrategias suelen ser eficientes a la hora de minimizar costos totales de transporte e inventario.

Seguidamente, la tabla 1 permite evidenciar el cumplimiento de los objetivos planteados para este proyecto de investigación.

**Tabla 1***Cumplimiento de objetivos*

<b>Objetivo específico</b>	<b>Cumplimiento</b>
Realizar una revisión de literatura contextualizando el problema de ruteo de vehículos de dos escalones con gestión de inventario y ventanas de tiempo para productos perecederos, extrayendo insumos para la definición de los métodos de solución a proponer.	Capítulo 3
Formular un modelo matemático que represente de manera precisa el problema de optimización estudiado.	Capítulo 5
Implementar un algoritmo metaheurístico de solución, dentro de los analizados en la revisión de literatura.	Capítulo 6 Capítulo 7 Apéndice E
Validar y evaluar los resultados obtenidos en diversas instancias consideradas, haciendo uso de datos sintéticos.	Capítulo 8
Redactar un artículo académico de carácter publicable a partir de la investigación realizada y los resultados obtenidos.	Apéndice F

## 1. Generalidades de la investigación

### 1.1 Planteamiento del problema

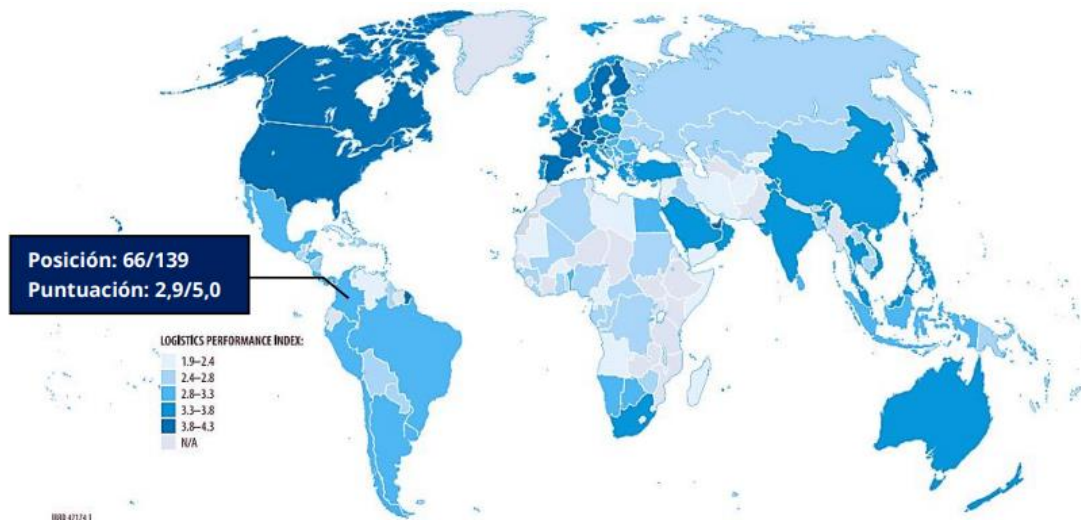
Los costos logísticos juegan un papel sumamente importante no solo dentro de la cadena de suministro, sino para la economía global de todas las empresas. Estos abarcan una amplia gama de actividades que son vitales para el funcionamiento de la cadena de suministro y es por esto que la necesidad de una gestión eficiente y confiable de estas operaciones cobra relevancia. Según la más reciente Encuesta Nacional de Logística realizada por el Departamento Nacional de Planeación (DNP), los elevados costos asociados a transporte, inventarios y almacenamiento han representado un aumento en el promedio del costo logístico sobre las ventas, ubicándose en un 17.9%, lo que representa un incremento del 5% en comparación a la última encuesta realizada por el DNP. Estas cifras representan un notorio retroceso en la forma en la que se gestionan las cadenas de suministro en el país (Analdex, 2024).

De igual manera, Colombia viene enfrentándose a grandes retos en cuanto al desempeño logístico y la facilidad que tiene el país para establecer conexiones confiables en la cadena de suministro con otros países. Esto se puede evidenciar en el Índice de Desempeño Logístico (LPI) publicado por el Banco Mundial, el cual otorga una calificación de 0 a 5 puntos a partir de la evaluación de una serie de indicadores que incluyen eficiencia, calidad y puntualidad asociados a la infraestructura logística del país.

Actualmente, Colombia cuenta con un puntaje de 2.9 sobre 5 puntos posibles, lo que ubica al país en la posición 66 en el ranking de los 139 países a los que se les aplicó esta medición en el año 2023. Esto también representa un retroceso en frente a los resultados de la anterior edición en el 2018, donde el país ocupó el puesto 58 en una lista que contenía un total de 160 economías (Banco Mundial, 2023).

**Figura 1**

*Índice de Desempeño Logístico global (2023)*



*Nota.* Tomado de *Worldbank.org*, por World Bank, 2023, [https://lpi.worldbank.org/sites/default/files/2023-04/LPI\\_2023\\_report\\_with\\_layout.pdf](https://lpi.worldbank.org/sites/default/files/2023-04/LPI_2023_report_with_layout.pdf)

Ahora bien, como lo expone Kumar *et al.*(2020), lidiar con la gestión de cadenas de suministro relacionadas con los alimentos en el sector agrícola siempre ha conllevado desafíos logísticos y operativos, incluidos los crecientes problemas de calidad e inocuidad de los alimentos, la reducción de los alarmantes desperdicios y pérdidas de alimentos y la escasa sostenibilidad económica.

Con base en lo anterior, este proyecto propone un modelo matemático que permita optimizar el problema descrito, el cual es aplicable específicamente al diseño de la cadena de distribución de productos perecederos, en este caso proyectado a dos escalones, incluyendo el uso de restricciones de ventanas de tiempo que afectan directamente la gestión del ruteo en este contexto. Además de esto, el problema será modelado mediante programación lineal entera mixta, en donde los dos escalones estarán conformados por tres niveles. Empezando por el primer nivel

de esta cadena se encuentra el punto donde operan los agricultores y se da la producción de cultivos y materia prima. De aquí se envía el producto a un conjunto de centros de distribución que son los encargados de gestionar un ruteo eficiente que permita hacer las entregas en los periodos de tiempo acordados a los puntos de venta. Para este segundo escalón de la cadena de distribución se planea experimentar evaluando en diferentes instancias con el objetivo de mejorar la variabilidad operativa y de darle al problema un enfoque más práctico y a su vez realista, en donde se puedan incluir ventanas de tiempo más flexibles que acarreen una penalización en la función objetivo. Asimismo, para este caso particular, la introducción de ventanas de tiempo en las entregas agrega una capa adicional de complejidad a este desafío, pero a su vez permite que el problema sea aplicable a la agroindustria en donde es común el transporte de productos que cuentan con características que los hacen susceptibles a caducar o deteriorarse de forma más rápida.

## **1.2 Metodología**

La metodología planteada establece una serie de etapas a seguir con el propósito de completar de forma satisfactoria los objetivos planteados. En este sentido, las fases definidas se muestran a continuación:

### **1.2.1. Etapa 1: Revisión de literatura y definición del problema**

Se plantea una revisión de literatura que permita caracterizar el problema en cuestión e identificar las diferentes metaheurísticas que se han revisado en la literatura, de esta forma se cumple el primer objetivo establecido. Esto implica la realización de las siguientes actividades.

- Definir las palabras claves y construir la ecuación de búsqueda.
- Ejecutar la ecuación de búsqueda en las bases de datos seleccionadas (*Scopus* y *Web of Science*).

- Definir criterios de inclusión y exclusión para validar la calidad de la literatura a estudiar.
- Realizar un análisis bibliométrico de la información.
- Realizar una revisión de carácter narrativo, evidenciando la evolución del problema IRP teniendo en cuenta su aplicación y métodos de solución.
- Extraer insumos clave para fundamentar la selección de la metaheurística a proponer en este trabajo.

### **1.2.2. Etapa 2: Formulación del modelo matemático**

Con el propósito de cumplir el segundo objetivo propuesto, las actividades que se muestran a continuación son planteadas.

- Entender las bases teóricas del problema de gestión de inventario y ruteo, aplicado en una cadena de suministro agrícola de dos escalones, donde se incluyan ventanas de tiempo y transporte de productos perecederos.
- Definir la estructura del modelo, teniendo en cuenta lo encontrado en la revisión de literatura y la formulación de variables y parámetros escogidos por el autor.
- Establecer el objetivo del problema a estudiar.
- Construir el modelo matemático que represente el 2eIRPTW, en donde se definan los parámetros, variables y restricciones con las que se va a trabajar.
- Verificar que los elementos que conforman al modelo estén relacionados entre sí de forma adecuada.

### **1.2.3. Etapa 3: Desarrollo del algoritmo de solución**

Se propone construir una metaheurística capaz de dar solución al modelo con el fin de cumplir el tercer objetivo de esta investigación, realizando las siguientes actividades.

- Identificar y evaluar diferentes alternativas metaheurísticas que sean adecuadas para resolver problemas similares con el fin de seleccionar el enfoque del algoritmo.
- Detallar los elementos necesarios para la construcción del algoritmo.
- Describir el conjunto de pasos lógicos que permitan solucionar el problema en cuestión.
- Programar el algoritmo en un lenguaje de programación adecuado para caracterizar diversas instancias del 2eIRPTW.

#### **1.2.4. Etapa 4: Validación y evaluación del algoritmo**

Se cumple el cuarto objetivo propuesto realizando las siguientes actividades.

- Comprobar que la metaheurística utilizada en el desarrollo del algoritmo corresponde con la programación y los principios teóricos establecidos.
- Definir las diferentes instancias en las que se planea probar el 2eIRPTW aplicado a la distribución de productos perecederos.
- Ejecutar el modelo y hacer ajustes de ser necesario.
- Realizar las pruebas en las instancias previamente definidas y medir indicadores clave de desempeño.

#### **1.2.5. Etapa 5: Síntesis de resultados**

Esta última etapa se enfoca en recopilar y analizar los resultados de la investigación, cumpliendo con el objetivo específico 5 a partir de las siguientes actividades.

- Realizar un análisis de los resultados minucioso con el fin de proponer conclusiones y recomendaciones valiosas para esta área de la investigación.
- Elaborar el libro del proyecto de grado.
- Redactar un artículo de carácter publicable a partir de los resultados obtenidos.

## 2. Objetivos

### 2.1. Objetivo General

Diseñar un modelo de optimización para la gestión de Inventario y Ruteo de dos escalones con ventanas de tiempo (2eIRPTW) aplicado al diseño de la red de distribución de productos perecederos.

### 2.2. Objetivos Específicos

Realizar una revisión de literatura contextualizando el problema de ruteo de vehículos de dos escalones con gestión de inventario y ventanas de tiempo para productos perecederos, extrayendo insumos para la definición de los métodos de solución a proponer.

Formular un modelo matemático que represente de manera precisa el problema de optimización estudiado.

Implementar un algoritmo metaheurístico de solución, dentro de los analizados en la revisión de literatura.

Validar y evaluar los resultados obtenidos en diversas instancias consideradas, haciendo uso de datos sintéticos.

Redactar un artículo académico de carácter publicable a partir de la investigación realizada y los resultados obtenidos.

### 3. Revisión de literatura y Análisis bibliométrico

#### 3.1. Análisis bibliométrico

Se realizó una búsqueda exhaustiva de la literatura existente con el fin de hacer uso de la crítica y los estudios anteriores en el tema de forma analítica para obtener un entendimiento más amplio del trabajo a realizar. Los resultados detallados asociados a las tendencias encontradas en términos de año de publicación, autores, región y palabras clave se recogen en el Apéndice D (Ver carpeta de Anexos).

#### 3.2. Revisión de literatura

Las operaciones relacionadas al funcionamiento de la cadena de suministro de una organización juegan un papel crucial en cuanto a la competitividad de la empresa y en su capacidad para sobresalir en el entorno en el que se mueve. Así mismo, la logística de última milla, incluyendo consideraciones de manejo de inventarios y transporte, se caracteriza por ser un proceso cuya ejecución es sumamente compleja y que generalmente implica costos importantes para la mayoría de las compañías. Para este caso particular, la complejidad del problema es inclusive mayor debido a que los productos que se transportan en la red de distribución en cuestión son productos perecederos, en donde la calidad del producto decrece con el pasar del tiempo. En consecuencia, este modelo propone una red de distribución que incluye restricciones de tiempo, que permitan asegurar la calidad del producto, realizando las entregas en ventanas de tiempo previamente establecidas entre todos los escalones de la cadena de suministro.

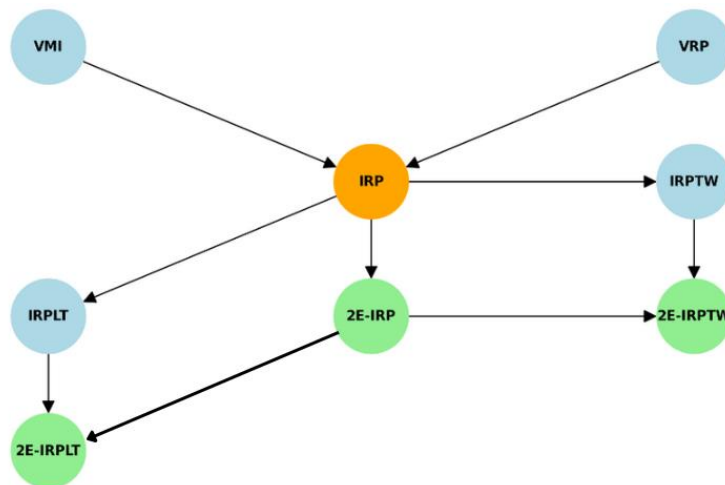
Partiendo de lo anteriormente expuesto, se propone un 2e-IRPTW (*Two-echelon Inventory Routing Problem*) que proviene del modelo IRP original, el cual integra decisiones de inventario y transporte en donde se combina el VRP (*Vehicle Routing Problem*) con el VMI (*Vendor-managed Inventory*).

En la estrategia del VMI, la responsabilidad de la toma de decisiones de las operaciones logísticas recae principalmente sobre el proveedor. El IRP nace de este problema y muchas veces es resuelto teniendo en cuenta los parámetros que caracterizan al VMI. Cuando se resuelve a través del enfoque del VMI, la información relacionada al manejo y niveles de inventario es compartida entre el proveedor y los clientes, permitiendo así que el proveedor monitoree el nivel de inventario de sus clientes con regularidad, con el fin de predecir la tasa de demanda del cliente. Con esta información, el proveedor debe considerar los costos de inventario y de entrega para determinar un plan logístico óptimo en donde el nivel de servicio no se vea afectado (Xiao & Rao, 2016).

A partir del VMI, se han desarrollado una serie de problemas que abarcan la familia del IRP o problemas similares. La figura 2 permite ver la evolución del IRP desde sus inicios en la combinación del VMI con el VRP, hasta algunas de las diferentes variantes existentes, incluyendo el 2eIRPTW.

**Figura 2**

*Evolución IRP*



*Nota.* Adaptado de Córdoba (2020)

Como se evidencia en la figura, el IRP tradicional surge de la combinación del problema VRP con el de ruteo de vehículos, es por esto que durante la construcción de la ecuación de búsqueda se usaron operadores que permitieran encontrar literatura suficiente para evidenciar el desarrollo y evolución del problema de inventario y ruteo. Este modelo busca encontrar una estrategia logística acompañada de una serie de decisiones que permitan definir en qué momento es pertinente servir a cada cliente, con la cantidad de producto que satisfaga sus necesidades y haciendo uso de las rutas de entrega que contribuyan a minimizar los costos totales de las operaciones de esa cadena de suministro en particular.

Una cadena de suministro normalmente se compone de dos partes: una cadena de abastecimiento y una red de distribución. El modelo en cuestión se enfoca en este segundo elemento de la cadena de suministro, en donde la red de suministro a estudiar está diseñada específicamente para distribuir productos perecederos en una cadena de suministro del sector agrícola.

Numerosos estudios sugieren que el IRP surge a partir del VMI y el VRP, por esta razón se decide abordar el problema desde sus inicios, específicamente estudiando variantes del VRP que contienen características similares a las del modelo que se propone en este estudio. Rahmanifar et al.(2023) propone un VRP de dos escalones enfocado en la optimización de un sistema de recolección de desechos añadiendo ventanas de tiempo que permitan disminuir la emisión de CO<sub>2</sub> al medio ambiente, para solucionar el modelo, combina un algoritmo *Social engineering optimizer* (SEO) con recocido simulado (SA). A su vez, Dellaert et al. (2021) también modela un VRP de dos escalones con restricciones de ventanas de tiempo (2e-VRPTW), en este caso, con la particularidad de que se utilizan ventanas de tiempo duras (*Hard time-windows*). El modelo es

solucionado a partir de dos heurísticas y el desarrollo de un método exacto combinando modelamiento matemático con Ramificación y precio (B&P).

Diversos artículos que estudian este mismo problema fueron analizados, por ejemplo, tanto Zhou et al.(2024) como Yu et al.(2023) modelaron un 2eVRPTW con programación lineal entera mixta (PLEM) haciendo uso de parámetros similares con algunas diferencias como la inclusión de instalaciones intermediarias o conductores ocasionales. Los métodos de solución también se asemejan un poco, ya que los investigadores se decantaron por utilizar un algoritmo de búsqueda local adaptativa (ALNS) o variantes de este para resolver el modelo.

Teniendo en cuenta consideraciones medio ambientales, en Akbay et al.(2022) se introducen vehículos eléctricos además de los convencionales en el modelo 2eVRPTW, dándole solución con un algoritmo que combina el método *Clark and Wright* con un VNS. En Liu et al.(2023) también se propone un VRP con vehículos eléctricos, en donde se introducen ventanas de tiempo suaves o flexibles para la entrega de productos perecederos y se da solución enfocándose en el uso de Optimización adaptativa basada en colonia de hormigas (AHACO).

En Zhou et al.(2022) se introduce el uso de entrega y recogida simultáneas y el modelo se resuelve de forma satisfactoria en instancias grandes implementando un algoritmo híbrido que combina las técnicas de búsqueda en vecindarios variables (VNS) y búsqueda tabú (TS). Wang et al.(2024) propone un VRP que busca minimizar costos operacionales y maximizar ganancias asociadas al reciclaje añadiendo la particularidad de que se consideran múltiples depósitos y precios de reciclaje inteligentes (IRPs).

Diversos autores proponen modelos de VRP concretamente en un contexto donde se transportan diferentes productos con naturaleza perecedera. En el caso de Abdelhalim et al.(2015) se propone un modelo con ventanas de tiempo implementado en el contexto de una pastelería, en

donde se utilizó una flota homogénea de vehículos con capacidades previamente conocidas en donde no se permiten entregas fraccionadas (*Split deliveries*) y el cual fue solucionado de forma exacta con el paquete de optimización LINGO. Fue Shaabani (2022) quien resumió los tipos de clasificaciones de perecibilidad encontrados en la literatura de gestión de inventarios para productos de esta naturaleza.

**Tabla 2**

*Revisión de tipos de perecibilidad*

<b>Tipo 1</b>	<b>Tipo 2</b>	<b>Tipo 3</b>
- Vida útil fija <i>(Fixed lifetime)</i>	- Vida útil aleatoria <i>(Random lifetime)</i>	- Productos en descomposición <i>(Decaying products)</i>
	- Tasa de deterioro dependiente de la edad	- Tasa de deterioro dependiente del tiempo o inventario

*Nota.* Adaptado de Shaabani (2022).

Hu et al.(2018) propone un IRP de dos escalones para distribuir productos perecederos con vida útil fija y demanda determinística. Por su parte, Wang et al.(2021) modela la perecibilidad como una función exponencial en donde no asocia la frescura del producto con el tiempo, sino con la distancia que este recorre, es decir, a mayor distancia menor frescura en el producto. Los métodos de solución para estos modelos fueron algoritmos de Búsqueda local y Evolución diferencial (DE) respectivamente.

En Wang et al.(2018) se plantean dos funciones objetivo, en donde la primera busca minimizar los costos logísticos totales mientras que la segunda maximiza la frescura del producto. Se encuentran soluciones efectivas a este modelo mediante una heurística híbrida entre búsqueda

en vecindario variable y un algoritmo genético (ST-VNSGA). Así mismo, en Zulvia et al.(2020) proponen un modelo con cuatro funciones objetivo: tres que minimizan costos operacionales, de deterioro y emisión de CO<sub>2</sub> respectivamente y una que maximiza el nivel de servicio; la solución a este modelo se halló con un algoritmo multiobjetivo de evolución por gradiente (MOGE). Por su parte, Nozar & Behnamian(2021) proponen un modelo con dos funciones objetivo, donde la primera busca minimizar el costo total y la segunda maximizar la probabilidad de que el cliente realice una compra asociando esta variable directamente con el nivel de satisfacción del cliente. Debido a la naturaleza NP-Hard del problema, un algoritmo hiperheurístico basado en la búsqueda Monte Carlo (MCHHA) es propuesto para evaluar el modelo en grandes instancias.

Seguidamente, se procede al análisis de los modelos IRP encontrados en la literatura, en donde se evidencia la variedad de enfoques que puede tomar debido a la versatilidad del problema, ya que es aplicable a diversas áreas o entornos en donde sea necesaria la gestión de inventarios y de ruteo. En Violi et al.(2020) se propone un IRP con incertidumbre en la demanda aplicado específicamente al campo de la gestión una cadena de suministro en el sector agrícola, solucionado a través de un algoritmo *Rolling horizon* (RHMA). Adicionalmente, Xin et al.(2021) enfocan su modelo al campo marítimo, el cual resuelven de forma exacta haciendo uso en un algoritmo de Ramificación y precio (B&P). El transporte de sangre es una tarea desafiante debido a la corta vida útil de este producto, además de la necesidad de refrigeración, por esto en Liu et al.(2021) se propone un modelo 2ePIRP que cuenta con un proveedor de sangre y un grupo de centros de recolección.

De la misma forma, el problema también se ajusta a entornos de logística inversa considerando entregas y acopios simultáneos, como lo propuesto por Iassinovskaia et al.(2017) en su modelo PDIRPTW para la distribución de artículos de transporte retornables (RTIs) o por

Kumar et al.(2022) en su CLGIRP (*Closed loop green Inventory routing*) para productos perecederos, el cual soluciona con un algoritmo de evolución diferencial (DEA).

Numerosas investigaciones han añadido un componente de cuidado medioambiental a sus modelos, como el IRP propuesto en Soysal et al.(2015), que busca minimizar costos operativos y reducir al mismo tiempo la cantidad de desechos e impactos negativos en el medio ambiente, o más recientemente en Alkaabneh et al.(2020), donde se modela el IRP en el contexto del transporte de productos perecederos con el objetivo de no solo maximizar las ganancias del proveedor, sino también de minimizar costos asociados al consumo de gasolina, almacenamiento y emisiones de gases de efecto invernadero. Finalmente, Soysal et al.(2018) también propone un modelo con una perspectiva ambientalista, esta vez enfocado en analizar los beneficios que trae la colaboración horizontal, dándole solución con el *solver* CPLEX 12.6.

Con el objetivo de mejorar el desempeño de la cadena de suministro, su flexibilidad y el impacto en cuanto a los costos que genera, Jia et al.(2014) propone dos tipos de ventana de tiempo en su modelo IRP considerando el deterioro de los productos con el pasar del tiempo. Los autores establecen una ventana de tiempo suave en la etapa del transporte del producto, y una ventana de tiempo dura o estricta en la etapa de la venta del producto. Un algoritmo de dos fases compuesto por Búsqueda tabú (TS) en la primera fase y la utilización tanto de un algoritmo de ahorros como de búsqueda en vecindario (*Saving algorithm and Neighbourhood search*) en la segunda permitieron resolver este modelo de forma satisfactoria. En Mirzaei & Seifi (2015) el modelo está diseñado para asignar el lugar de almacenamiento de bienes perecederos. Una metaheurística eficiente que combina recocido simulado (SA) con búsqueda tabú (TS) es diseñada para solucionar este problema.

Diferentes autores se decantan por métodos de modelamiento y solución un poco más codiciosos haciendo uso de optimización robusta, que se centra en la búsqueda de soluciones que sean efectivas no solo en condiciones ideales, sino también bajo una variedad de escenarios adversos. Tanto en Ji et al.(2020) como en Ji et al.(2021) se modela un IRP de dos escalones para la distribución de productos perecederos mediante programación robusta entera mixta (PREM) y se da solución al mismo a través de tres modelos diferentes (*R-box*, *ellipsoid* y *polyhedron*) de optimización robusta, los cuales son corridos en el algoritmo del *Solver* Gurobi.

Consolidando los resultados de esta revisión, se identificaron diversos enfoques metodológicos para afrontar modelos VRP, IRP y sus variaciones, donde se destaca la preferencia por metaheurísticas sobre los métodos exactos. Además de esto, se evidencia una tendencia creciente en cuanto a las metaheurísticas híbridas donde se combinan diferentes técnicas de solución. Finalmente, la literatura analizada soporta la decisión de modelar el problema a través de programación lineal entera mixta, ya que presenta mayor capacidad para manejar la complejidad de un 2eIRPTW, ofreciendo a su vez flexibilidad en la formulación y escalabilidad a la hora de cambiar el tamaño del problema en las instancias a evaluar.

#### **4. Marco de Referencia**

Se considera un marco de referencia que abarca una delimitación que se hace del problema a estudiar, en relación con los aspectos que servirán de base para el mismo, el cual está conformado por un marco de antecedentes y un marco teórico.

##### **4.1. Marco de antecedentes**

En el repositorio de la Universidad Industrial de Santander (UIS) es posible encontrar algunos proyectos de investigación relacionados con el problema a tratar en este trabajo tanto para tesis de pregrado como de post grado. Algunos de estos proyectos, los cuales sirven como insumo

para la realización del presente trabajo, son expuestos seguidamente. Estos se caracterizan por pertenecer al mismo periodo de tiempo evaluado en las bases de datos (2014-2024), y por aportar valor a esta investigación debido a que estudian modelos IRP o variaciones similares a este, utilizando a su vez diferentes algoritmos de solución para hallar soluciones efectivas.

El primer trabajo por analizar fue realizado por Barragán y Romero (2015). En la realización de este proyecto, los autores consideran técnicas exactas y heurísticas para encontrar soluciones factibles. Para la técnica exacta, hicieron uso del *Solver* que viene integrado en Excel. Además de esto, utilizaron el método Clark and Wright y el método de mejora 2-OPT como heurísticas que permitieran dar solución al modelo planteado. Finalmente, compararon el rendimiento de las metodologías utilizadas evaluando el tiempo de computación requerido en cada uno de los casos, evaluando el modelo en diferentes instancias. Este proyecto sirve como base teórica para la presente investigación, en donde se evidencia la necesidad de identificar metaheurísticas con mayor robustez, que permitan dar solución a un modelo IRP que contenga restricciones de tiempo en las entregas para una cadena de dos escalones.

Más adelante, Niño (2017) estudia en su tesis de pregrado el problema de ruteo de vehículos en una cadena de dos escalones. El modelamiento matemático se realiza a través de programación lineal entera mixta (PLEM). Para la caracterización del problema, el autor define la restricción de envíos directos entre depósitos y clientes, la imposibilidad de dividir la demanda de los clientes entre los vehículos del segundo eslabón, y la posibilidad de repartir la demanda de los satélites entre los vehículos únicamente en el primer escalón. Como método de solución, el autor construye una metaheurística que combina el procedimiento de búsqueda voraz adaptativa y aleatoria (GRASP) con el de Recocido simulado (SA) en dos fases. Los resultados de esta investigación son valiosos para el presente 2eIRPTW. A pesar de tratarse de un problema distinto que no contiene

decisiones de inventarios, el IRP es una extensión del VRP y por ende tienen una base común en la optimización del ruteo.

Córdoba (2020) estudia una variación del IRP tradicional, añadiéndole a este una red de distribución de dos escalones, y habilitando la posibilidad de usar transbordos laterales de ser necesario. Este modelo es programado a través de PLEM, conformado por una planta en el primer nivel y un conjunto de distribuidores y clientes en el segundo y tercer nivel, respectivamente. Se considera una flota homogénea de vehículos para cada escalón, los cuales cuentan con una capacidad máxima previamente definida. El algoritmo de solución propuesto por el autor es una metaheurística híbrida conformada por un algoritmo genético (GA) y el método de búsqueda tabú (TS). Este proyecto es la base para el problema 2eIRPTW a estudiar, debido a su fundamento en el 2eIRP y a el valor obtenido en el análisis de los resultados. Se concluyó que en la mayoría de las instancias analizadas, no es necesaria la utilización de transbordos laterales en la cadena debido a que no representan disminuciones en los costos totales; así mismo, el autor destaca el proceso del diseño de la metaheurística, donde afirma que el algoritmo genético tienen mayor profundidad que la búsqueda tabú, pero la implementación de esta última resulta perfecta para un IRP de dos escalones.

Cataño (2023) propone un modelo matemático basado en un 2eVRPDTW, donde se considera el ruteo de vehículos para un grupo de drones teniendo en cuenta ventanas de tiempo, diseñado específicamente para asistir en operaciones de entrega de suministros en áreas afectadas por desastres naturales. Entre los supuestos del sistema se especifica que a la hora de repartir suministros en el segundo escalón, solo se puede realizar una entrega por cada tipo de vehículo dentro de la ventana de tiempo establecida y en caso de que la entrega de la ayuda sobrepase el límite de tiempo, existirá una penalización en la función objetivo. Para resolver este problema, se

desarrolla un algoritmo evolutivo, específicamente un caso de algoritmo genético, que permite obtener soluciones factibles en tiempos de cómputo aceptables. Los resultados de este proyecto sirven como semilla para la investigación a realizar, donde es posible identificar nuevas metaheurísticas que permiten dar solución a un problema de ruteo para dos escalones.

## **4.2. Marco teórico**

**4.2.1. Cadena de suministro.** La cadena de suministro engloba todos los procesos necesarios que envuelven al producto o servicio desde la obtención de materia prima hasta su resolución final, y que permiten completar de forma satisfactoria las órdenes de los clientes. Entre sus funciones clave se encuentra la gestión de inventarios, almacenes y medios de transporte, además de buscar administrar de manera integral los flujos de productos, información y fondos que se dan a lo largo de la cadena para mejorar el servicio al cliente e incrementar los beneficios (Carreño, 2017).

**4.2.2. Logística.** Este término se puede definir como un proceso que participa determinadamente en la forma como los proveedores, vendedores, clientes y transportistas interaccionan con la empresa. No es solo enviar y recibir, ni transportar y almacenar. La logística es la encargada de gestionar las operaciones de la empresa de forma eficiente, asegurándose de que el papel de cada participante de la cadena sea efectivo (Garay, 2014).

**4.2.3. Red de distribución.** Hace referencia a toda la infraestructura y procesos que se tienen que llevar a cabo para que el producto terminado sea entregado satisfactoriamente a centros de distribución, puntos de venta o clientes finales. Para este trabajo la red a estudiar tiene dos escalones donde los productos se mueven a través de dos niveles antes de llegar al consumidor final. Se estudian dos tipos de decisiones en este caso:

- **4.2.3.1. Decisiones de inventario.** Suelen ser decisiones de tipo táctico y están relacionadas con la determinación de los métodos de registro, formas de clasificación de las existencias y el modelo de reabastecimiento que establece cantidades a pedir o producir y en qué momento hacerlo (Garay, 2014).
- **4.2.3.2. Decisiones de transporte.** Son decisiones operativas e incluyen la selección del modo de transporte adecuado, la programación de envíos, la naturaleza de la mercancía, logística inversa y ruteo de vehículos (Garay, 2014). Además, consideran factores como los costos de transporte, los tiempos de entrega y la capacidad de carga.

**4.2.3.2.1 Ventanas de tiempo.** Estrategia utilizada en el ámbito del ruteo de vehículos, donde se establece un periodo de tiempo específico en el que una entrega debe ser realizada. Pueden ser:

- **Ventanas de tiempo blandas.** También conocidas como *Soft time Windows*, son una variación de la ventana de tiempo tradicional, en donde es permitida la presencia de tiempos de espera y entregas tardías en las instalaciones del cliente, incurriendo en una penalidad en la función objetivo.
- **Ventanas de tiempo duras.** En inglés, *Hard time Windows*. Se caracterizan por la prohibición de tiempos de espera o llegadas tardías. Es decir, si un vehículo llega antes de la ventana de tiempo, deberá esperar hasta que la ventana se abra. Si llega después, la entrega no puede realizarse o se considera como una solución no válida.

La elección del tipo de ventanas de tiempo dependerá del contexto específico en el que se desarrolla el problema y de los objetivos de la cadena de suministro en cuestión (Pérez y Rueda, 2015).

**4.2.4. Transbordos laterales.** Estrategia para gestionar las decisiones relacionadas a inventarios, en donde los miembros de un mismo escalón en una cadena de suministro tienen la posibilidad de agrupar sus existencias permitiéndoles disminuir sus costos y niveles de inventario, manteniendo a su vez, el nivel de servicio requerido (Paterson et al., 2011).

**4.2.5. Teoría de optimización.** Se refiere al estudio de técnicas y métodos numéricos y analíticos que permitan encontrar la mejor solución dentro de un conjunto de posibles opciones.

**4.2.6. Optimización matemática.** Aplicado al campo de la matemática, este término hace referencia a un grupo de técnicas que permiten dar respuesta a diferentes problemas, apoyando de forma eficiente la toma de decisiones. Normalmente un problema a optimizar se puede escribir teniendo en cuenta tres elementos:

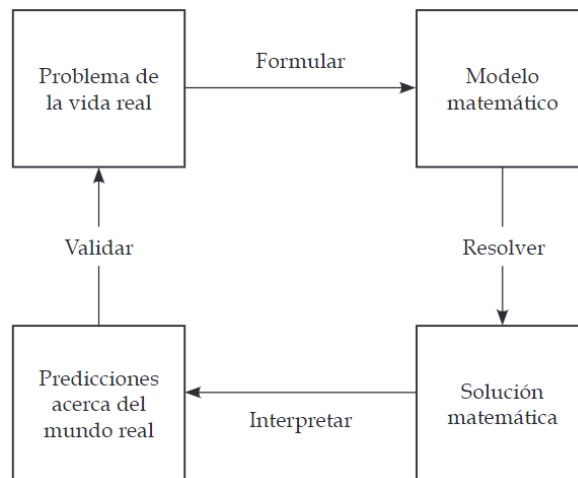
- **4.2.6.1. Variables de decisión.** Se refiere a las diferentes alternativas que pueden considerarse dentro del problema. Estas pueden ser continuas, discretas o en algunos casos, una mezcla de las dos.
- **4.2.6.2. Función objetivo.** También conocida como función de costos en la literatura, es la representación del problema a resolver y hace referencia a la relación matemática entre los diferentes parámetros y variables que componen al sistema y una magnitud que representa el objetivo de este. Esta puede ser lineal o no lineal.
- **4.2.6.3. Restricciones.** Están representadas como igualdades o desigualdades y representan limitaciones impuestas sobre las variables de decisión.

Gracias a este tipo de optimización, es posible modelar situaciones de la vida cotidiana, en ingeniería, finanzas o aplicaciones industriales que permiten dar soluciones óptimas a problemas con todo tipo de aplicaciones y en una amplia variedad de áreas (Yang, 2010).

**4.2.7. Modelos matemáticos.** Estos se usan para describir diferentes sistemas y fenómenos, en donde se representa la naturaleza del problema y la relación existente entre las variables que afectan al mismo. El modelado matemático sigue una serie de pasos que se pueden resumir en el proceso que se muestra en la siguiente figura (Armas y Trujillo, 2017).

**Figura 3**

*Proceso de modelado matemático*



*Nota.* Tomado de *Introducción al modelado matemático con Matlab*, por Armas y Trujillo, 2017

**4.2.8. Programación lineal entera mixta (PLEM).** En este tipo de modelos se requiere que algunas variables tomen valores enteros, mientras que otras pueden tomar valores continuos. En los modelos de PLEM, la función objetivo está sujeta a restricciones lineales donde las variables pueden tomar valores enteros o continuos (Yang, 2010).

**4.2.9. Optimización combinatoria (COP).** Este tipo de problemas se caracterizan por contar con un conjunto de soluciones factibles finito. Además de esto, debido a su alta complejidad entran en la categoría de los problemas NP-Hard, lo que quiere decir que generalmente no existe un

algoritmo conocido que pueda encontrar una solución óptima en tiempo polinómico para instancias grandes.

Según Baghel et al.(2012), un problema COP puede ser definido como se muestra en la tabla 3.

**Tabla 3**

*Formulación matemática para problemas de optimización combinatoria*

Componente	Formulación
Problema de tipo COP	$P = (S, f)$
Conjunto de variables	$X = \{x_1, \dots, x_n\}$
Dominio de las variables	$D_1, \dots, D_n$
Función objetivo	$f = D_1 X \dots X D_n \rightarrow \mathbb{R}^+$
Conjunto de soluciones factibles	$S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\}   v_i \in D_i\}$

*Nota.* Adaptado de Baghel et al.(2012).

**4.2.10. Principales problemas de optimización combinatoria.** Existe una gran variedad de problemas en los que es necesaria la optimización combinatoria, sin embargo, para este proyecto se expondrán a continuación los relacionados con el diseño de redes de distribución y similares.

- **4.2.10.1. Problema de ruteo de vehículos (VRP).** Responde a la necesidad de diseñar un conjunto de rutas óptimas que permitan entregar o recolectar determinado producto haciendo uso de una flota homogénea o heterogénea de vehículos con una capacidad definida (Kritikos & Lappas, 2021).
- **4.2.10.2. Problema de inventario y ruteo (IRP).** Consiste en definir el plan de reabastecimiento óptimo de los clientes de un proveedor a lo largo de un horizonte

de tiempo, así como el programa de enrutamiento en cada período de tiempo de manera que se optimice un objetivo determinado (Alvarez et al., 2020).

- **4.2.10.3. Problema de inventario y ruteo en dos escalones (2eIRP).** Para este problema la logística de última milla contempla un nivel intermedio en donde es necesario considerar el almacenamiento de los productos. Además de esto, se tiene en cuenta la disponibilidad del cliente durante un horizonte temporal determinado, complicando así el proceso de entrega. Por estas razones es tan importante una planificación eficiente que permita optimizar el horario de entregas, el diseño de rutas, el tiempo de almacenamiento y la cantidad de inventario en el almacén (Rohmer et al., 2019).

#### ***4.2.11. Métodos de solución para problemas tipo COP.***

**4.2.11.1. Métodos exactos.** Permiten obtener como solución los óptimos globales del modelo de forma directa, requiriendo de altos recursos computacionales para obtener resultados (Córdoba, 2020). Algunos de los métodos más utilizados se exponen a continuación:

- **4.2.11.1.1. Método Simplex.** Procedimiento iterativo que permite mejorar la solución de la función objetivo en cada paso. Este concluye cuando no es posible continuar mejorando dicho valor, es decir, se ha alcanzado la solución óptima cumpliendo con todas las restricciones planteadas. Una particularidad es que este método únicamente trabaja con restricciones del problema cuyas inecuaciones sean del tipo  $\leq$  y sus coeficientes independientes sean mayores o iguales a 0 (Acosta et al., 2019).
- **4.2.11.1.2. Método Branch and Bound (B&B).** Conocido como método de ramificación y corte, este método contempla una búsqueda del óptimo combinando

la comparación de distintas alternativas función de las variables enteras, con un procedimiento para eliminar combinaciones que no pueden conducir al óptimo y para determinar las condiciones de óptimo (Acosta et al., 2019).

**4.2.11.2. Métodos aproximados.** Debido a los altos niveles de requerimiento computacional en algunos problemas evaluados en instancias grandes, es imposible garantizar un algoritmo de solución que permita llegar a valores óptimos de forma directa. En base a esto, algunas alternativas como las heurísticas y metaheurísticas son propuestas.

**4.2.11.3. Heurísticas.** Esta técnica de solución consiste en programar un algoritmo de solución que suele ser específicamente diseñado para el problema en cuestión, y el cual permite encontrar soluciones factibles cercanas a la óptima en tiempos computacionales razonables. Diferentes técnicas heurísticas han sido aplicadas a problemas como el IRP o similares, entre las más comunes se encuentra el método de ahorros de Clarke-Wright, el algoritmo heurístico del vecino más próximo o diferentes heurísticas de búsqueda local. En términos generales, se pueden clasificar en dos tipos: heurísticas de construcción y de mejora.

**4.2.11.4. Metaheurísticas.** El perfeccionamiento de un algoritmo heurístico es lo que se suele conocer hoy en día como metaheurística. Sin embargo, las tendencias recientes tienden a nombrar todos los algoritmos estocásticos con aleatorización y búsqueda local como metaheurísticos. Los dos componentes claves que distinguen cualquier metaheurística son la selección de las mejores soluciones y la aleatorización (Yang, 2010). Generalmente se dividen en 2 tipos: basados en población o basados en trayectoria. Una de las principales diferencias entre estos dos tipos de técnicas radica en la cantidad de soluciones que se utilizan en el proceso de hallar esos valores óptimos. Los métodos basados en trayectoria reciben este nombre ya que la solución describe una trayectoria desde el punto de partida hasta el final; por otra parte, los

métodos basados en poblaciones tienden a encontrar soluciones de mayor calidad, ya que permiten explorar simultáneamente diversas regiones del espacio de búsqueda. Entre las técnicas basadas en trayectoria más comunes se encuentran:

**4.2.11.4.1. Recocido simulado (*Simulated annealing, SA*).** Promovido por Kirkpatrick en 1983, surge de la necesidad de mejorar la búsqueda local, en donde se utiliza una cadena de Markov que permite evadir el estancamiento en mínimos locales (Yang, 2010).

**4.2.11.4.2. Búsqueda tabú (*Tabú search, TS*).** Algoritmo de memoria adaptativa que aplica una búsqueda local mejorada en la que en cada iteración se selecciona la mejor solución del vecindario como la nueva solución actual (Baghuel, et al. 2012).

**4.2.11.4.3. Búsqueda en vecindario variable (*Variable Neighborhood search, VNS*).** Este método se basa en cambiar dinámicamente de vecindario dentro de una búsqueda local. Conformado por una fase de mejora, una de agitación y una de cambio de vecindario (Baghuel, et al. 2012).

**4.2.11.4.4. Procedimiento de búsqueda voraz adaptativa aleatoria (*Greedy Randomized Adaptive search procedure, GRASP*).** Introducido por Resende en 1995. Este algoritmo de tipo iterativo está conformado por dos fases: una de construcción que permite encontrar una primera solución factible y una de búsqueda local donde se mejora esa solución inicial hasta encontrar un mínimo local (Baghuel, et al. 2012).

Por otra parte, entre los métodos basados en población están:

**4.2.11.4.5. Algoritmos genéticos (*Genetic algorithm, GA*).** Una de las ramas de los algoritmos evolutivos (*Evolutionary algorithm*) está conformada por los algoritmos genéticos. Este modelo iterativo asemeja la teoría de selección natural propuesta por Charles Darwin y tiene en cuenta el proceso de aprendizaje colectivo para una población donde cada miembro representa

un punto en el espacio donde se buscarán soluciones factibles. Para esto se define una función *fitness*, la cual es evaluada en todos los miembros de una población, y esa población va evolucionando hasta llegar a los criterios de parada establecidos (Yang, 2010).

**4.2.11.4.6. Optimización por Enjambre de Partículas (*Particle Swarm optimization, PSO*).** Basado en el comportamiento de enjambres, este método busca el espacio de una función objetivo ajustando la trayectoria de agentes individuales (partículas). El movimiento de estas partículas se basa en dos componentes: uno estocástico y uno determinístico (Yang, 2010).

**4.2.11.4.7. Optimización por colonia de hormigas (*Ant Colony optimization, ACO*).** Inspirado en el comportamiento real de las hormigas, este método permite encontrar el camino más corto entre dos nodos a través de una sustancia química llamada feromona. Esta feromona va dejando un rastro por donde transiten las hormigas, lo que eventualmente genera un camino que sirve como solución parcial, hasta construir una solución final cercana a niveles óptimos que cuenta con niveles de feromonas más elevados (Baghuel, et al. 2012).

## 5. Modelo para el 2eIRPTW de productos perecederos

### 5.1. Descripción del problema

El modelado en este proyecto se desarrolla en base a los estudios realizados por Córdoba (2020), Rohmer *et al.* (2019) y Dellaert *et al.* (2021). El modelo 2eIRPTW a estudiar es presentado a partir de un grafo  $G = (V, A)$ , en el cual  $V$  denota a los vértices o nodos conformados por la planta de producción  $P$ , un conjunto de centros de distribución  $D$  y los clientes  $C$  donde  $e, f \in V$ , mientras que  $A$  representa el conjunto de aristas dirigidas conformadas por pares ordenados de vértices  $(e, f)$  con  $e \neq f$ . De la misma forma, los clientes  $C$  representan los puntos de venta finales, por ende se tiene en cuenta un balance de inventarios en los tres niveles que conforman esta red de distribución.

Para este modelo logístico se considera una flota homogénea para cada escalón (K y M), en donde K representa el conjunto de vehículos disponibles en la planta ( $a \in P$ ) y M el set de vehículos disponibles en los que operan los distribuidores ( $i, j \in D$ ). La capacidad de los vehículos es conocida y está definida por  $Q_K$  y  $Q_M$  y la relación entre ellas está dada por  $Q_K > Q_M$ .

El costo asociado al transporte para cada conjunto de aristas dirigidas perteneciente al conjunto  $A$  está dado por  $C_{ef}$ , y a su vez, cada vértice perteneciente a  $V$  cuenta con un costo asociado a almacenamiento  $Ch_e$  independiente de la frescura del producto. Por otra parte,  $h^g$  representa un valor constante y conocido que introduce un costo adicional que varía dependiendo del estado del producto al aumentar de forma lineal con el tiempo ( $h^{g,t} = h^g * t$ ). El problema considera una planificación para T periodos de tiempo y la programación se realiza de manera dinámica a lo largo del tiempo, en donde en cada periodo ( $t \in T$ ) tanto la planta como los distribuidores pueden realizar envíos. Se considera también una degradación lineal de la calidad de los productos a lo largo del tiempo, para esto se establece un conjunto G que contiene los diferentes estados en los que se puede encontrar el producto, indexados por g ( $g \in G$ ). Se lleva un seguimiento del estado de frescura g del producto desde el momento de su producción. Dado que toda la producción se encuentra en el estado más fresco ( $g = 0$ ), la penalización por el deterioro del producto se aplica exclusivamente en el segundo escalón de la cadena de suministro. Esto se debe a que son los clientes quienes imponen mayores restricciones en la recepción de productos en condiciones no óptimas. El modelo lleva un control riguroso del estado g en el que se encuentra cada lote de producto a través de un mecanismo de envejecimiento que se aplica al inicio de cada periodo t, en donde cada unidad que no fue consumida al final del periodo  $t - 1$  aumenta su estado g en 1 para el inicio del periodo t (el estado de frescura se degrada en un nivel). La demanda  $dem_u^t$  de cada cliente ( $u \in C$ ) en cada periodo de tiempo t es conocida, y debe ser entregada dentro de

los límites de la ventana de tiempo  $(tw_u^l, tw_u^s)$  previamente establecida. Si un vehículo llega al nodo antes del tiempo de llegada  $tw_u^l$ , este deberá esperar a la apertura de la ventana de tiempo para poder hacer la entrega. Además de esto, el inventario inicial en cada uno de los vértices del grafo es igual a cero ( $I_a^0, I_i^0, I_u^0 = 0$ ).

El objetivo del 2eIRPTW en cuestión es minimizar los costos logísticos totales asociados al transporte y almacenamiento de productos perecederos, a lo largo de un horizonte temporal previamente establecido y cumpliendo con los requerimientos de demanda de los integrantes de la red de distribución. Se espera obtener un resultado que permita planificar de forma integral las operaciones logísticas de la cadena de suministro, en donde sea posible determinar las cantidades a enviar desde la planta de producción a los diferentes distribuidores y el momento oportuno para hacerlo. Asimismo, se espera estipular cuándo, cuánto y desde qué distribuidor se realizan envíos a los diferentes clientes, teniendo en cuenta las ventanas de tiempo definidas, y definiendo un conjunto de rutas que permitan minimizar costos totales.

Las restricciones que caracterizan al modelo se enuncian a continuación:

- Cada vehículo puede hacer máximo una entrega por cada periodo  $t$ , iniciando y finalizando la ruta en el mismo vértice.
- El nivel de inventario en cada uno de los vértices tiene que ser menor o igual a su capacidad máxima  $\Omega_e$ .
- La planta de producción del primer nivel cuenta con capacidad infinita  $\Omega_a \approx \infty$ .
- No se permiten inventarios negativos. La demanda de cada periodo debe cubrirse con las existencias que quedaron del periodo anterior junto con los productos recibidos en el periodo actual.

- La producción en planta siempre es por lo menos igual a la cantidad enviada a los distribuidores, y las cantidades producidas en la planta o recibidas por los distribuidores pueden enviarse en el mismo periodo.
- Los vehículos cuentan con una capacidad máxima que no puede ser excedida.
- Cada vehículo (k o m) puede visitar solo a un nodo (CD o cliente) en cada periodo.
- Las entregas en el segundo escalón solo se permiten si ocurren dentro del intervalo previamente definido  $[tw_u^l, tw_u^s]$ , el cual representa los límites inferior y superior de la ventana de tiempo asignada a cada cliente.

## 5.2. Formulación matemática

Los parámetros mencionados anteriormente son acompañados de un grupo de variables de decisión que permiten modelar el problema en lenguaje matemático. Estos son definidos a continuación:

### *Parámetros*

$C_{p,h}$ : Costo de envío del vértice p al h.

$C_{n,q}$ : Costo de envío del vértice n al q.

$Ch_e$ : Costo de almacenamiento unitario en el nodo e.

$h^g$ : Penalización adicional por mantener y enviar producto en estado  $g > 0$  (Se aplica únicamente a  $I^{g,t}$ ).

**BigM**: Constante “Big M” utilizada para activar o desactivar restricciones.

$\tau_{nq}$ : Representa el tiempo de viaje entre los nodos n y q.

### *Variables*

**Z**: Variable dependiente que representa la función objetivo a minimizar.

$I_e^t$ : Representa el nivel de inventario en el vértice e al finalizar el periodo t.

$I^{g,t}$ : Representa la cantidad almacenada de producto en estado  $g$  al finalizar el periodo  $t$ .

$pr_a^t$ : Cantidad producida en la planta agrícola en el periodo  $t$ .

$d_{a,i}^{k,t}$ : Cantidad enviada desde la planta  $a$  al distribuidor  $i$  en el periodo  $t$  usando el vehículo  $k$ .

$b_{i,u}^{m,t}$ : Cantidad enviada desde el distribuidor  $i$  al cliente  $u$  en el periodo  $t$  usando el vehículo  $m$ .

$w_{a,i}^{k,t}$ : Suma de las entregas hechas por el vehículo  $k$  proveniente de la planta  $a$  luego de visitar al distribuidor  $i$  en el periodo  $t$ .

$v_{i,u}^{m,t}$ : Suma de las entregas hechas por el vehículo  $m$  proveniente del distribuidor  $i$  luego de visitar al cliente  $u$  en el periodo  $t$ .

$\theta_{i,u}^{g,m,t}$ : Cantidad de producto en estado  $g$  enviada desde el distribuidor  $i$  al cliente  $u$  en el periodo  $t$  usando el vehículo  $m$ .

$x_{a,p,h}^{k,t} : \begin{cases} 1 & \text{si el vehículo } k \text{ proveniente de la planta } a \text{ viaja del vértice } p \text{ al } h \text{ (} p, h \in A \text{)} \\ 0 & \text{de lo contrario} \end{cases}$

$y_{i,n,q}^{m,t} : \begin{cases} 1 & \text{si el vehículo } m \text{ proveniente del distribuidor } i \text{ viaja del vértice } n \text{ al } q \text{ (} n, q \in A \text{)} \\ 0 & \text{de lo contrario} \end{cases}$

$z_{a,p}^{k,t} : \begin{cases} 1 & \text{si el vehículo } k \text{ proveniente de la planta } a \text{ visita el vértice } p \text{ en el periodo } t \\ 0 & \text{de lo contrario} \end{cases}$

$s_{i,n}^{m,t} : \begin{cases} 1 & \text{si el vehículo } m \text{ proveniente del distribuidor } i \text{ visita el vértice } n \text{ en el periodo } t \\ 0 & \text{de lo contrario} \end{cases}$

$l_n^{m,t}$ : Tiempo de llegada del vehículo  $m$  al vértice  $n$  en el periodo  $t$ .

$$\begin{aligned} \text{Min } Z = & \sum_{a \in P} \sum_{p,h \in A} \sum_{k \in K} \sum_{t \in T} C_{p,h} * x_{a,p,h}^{k,t} + \sum_{i \in D} \sum_{n \in A} \sum_{q \in A} \sum_{m \in M} \sum_{t \in T} C_{n,q} * y_{i,n,q}^{m,t} \\ & + \sum_{i \in V} \sum_{t \in T} Ch_e * I_e^t + \sum_{g \in G} \sum_{t \in T} h^g * I^{g,t} * g \end{aligned} \quad (1)$$

Sujeto a:

$$I^{g,t} = I^{g-1,t-1} - \sum_{m \in M} \sum_{u \in C} \theta_{i,u}^{g-1,m,t-1}, \quad \forall g \in G \setminus \{0\}, \quad t \in T \setminus \{0\} \quad (2)$$

$$I^{0,t} = d_{a,i}^{k,t}, \quad a \in P, \quad k \in K, \quad t \in T \quad (3)$$

$$I_a^t = I_a^{t-1} + pr_a^t - \sum_{k \in K} \sum_{i \in D} d_{a,i}^{k,t}, \quad a \in P, \quad t \in T \quad (4)$$

$$I_a^t, p_a^t \geq 0, \quad a \in P, \quad t \in T \quad (5)$$

$$I_i^t = I_i^{t-1} + \sum_{k \in K} \sum_{a \in P} d_{a,i}^{k,t} - \sum_{m \in M} \sum_{u \in C} b_{i,u}^{m,t}, \quad i \in D, \quad t \in T \quad (6)$$

$$I_i^t \geq 0, \quad i \in D, \quad t \in T \quad (7)$$

$$I_u^t = I_u^{t-1} + \sum_{m \in M} \sum_{i \in D} b_{i,u}^{m,t} - dem_u^t, \quad u \in C, \quad t \in T \quad (8)$$

$$I_u^t \geq 0, \quad u \in C, \quad t \in T \quad (9)$$

$$I_i^t \leq \Omega_i, \quad i \in D, \quad t \in T \quad (10)$$

$$I_u^t \leq \Omega_u, \quad u \in C, \quad t \in T \quad (11)$$

$$\sum_{k \in K} \sum_{a \in P} d_{a,i}^{k,t} \leq \Omega_i - I_i^{t-1}, \quad i \in D, \quad t \in T \quad (12)$$

$$\sum_{m \in M} \sum_{i \in D} b_{i,u}^{m,t} \leq \Omega_u - I_u^{t-1}, \quad u \in C, \quad t \in T \quad (13)$$

$$\sum_{h \in V} (x_{a,h,p}^{k,t} + x_{a,p,h}^{k,t}) = 2z_{a,p}^{k,t}, \quad (14)$$

$$a \in P, \quad p \in V, \quad k \in K, \quad t \in T, \quad p \neq h \quad (14)$$

$$\sum_{h \in V} (x_{a,h,p}^{k,t} - x_{a,p,h}^{k,t}) = 0, \quad (15)$$

$$a \in P, \quad p \in V, \quad k \in K, \quad t \in T, \quad p \neq h \quad (15)$$

$$\sum_{n \in V} (y_{i,n,q}^{m,t} + y_{i,q,n}^{m,t}) = 2s_{i,n}^{m,t} \quad ,$$

$$i \in D, \quad q \in V, \quad m \in M, \quad t \in T, \quad q \neq n \quad (16)$$

$$\sum_{n \in V} (y_{i,n,q}^{m,t} - y_{i,q,n}^{m,t}) = 0 \quad ,$$

$$i \in D, \quad q \in V, \quad m \in M, \quad t \in T, \quad q \neq n \quad (17)$$

$$\sum_{k \in K} z_{a,i}^{k,t} \leq 1 \quad , \quad a \in P, \quad i \in D, \quad t \in T \quad (18)$$

$$\sum_{m \in M} s_{i,n}^{m,t} \leq 1 \quad , \quad i \in D, \quad n \in V, \quad t \in T \quad (19)$$

$$\sum_{i \in D} x_{a,a,i}^{k,t} \geq z_{a,j}^{k,t} \quad , \quad a \in P, \quad j \in D, \quad k \in K, \quad t \in T \quad (20)$$

$$\sum_{n \in V} y_{i,i,n}^{m,t} \geq s_{i,q}^{m,t} \quad , \quad i \in D, \quad q \in V, \quad m \in M, \quad t \in T, \quad i \neq n \quad (21)$$

$$\sum_{i \in D} d_{a,i}^{k,t} \leq Q_K \quad , \quad a \in P, \quad k \in K, \quad t \in T \quad (22)$$

$$\sum_{i \in D} b_{i,n}^{m,t} \leq Q_M \quad , \quad n \in V, \quad m \in M, \quad t \in T, \quad i \neq n \quad (23)$$

$$d_{a,i}^{k,t} \leq \Omega_a z_{a,e}^{k,t} \quad , \quad a \in P, \quad i \in D, \quad k \in K, \quad t \in T \quad (24)$$

$$b_{i,u}^{m,t} \leq \Omega_i s_{i,u}^{m,t} \quad , \quad i \in D, \quad u \in C, \quad m \in M, \quad t \in T \quad (25)$$

$$d_{a,i}^{k,t} \geq z_{a,i}^{k,t} \quad , \quad a \in P, \quad i \in D, \quad k \in K, \quad t \in T \quad (26)$$

$$b_{i,u}^{m,t} \geq s_{i,u}^{m,t} \quad , \quad i \in D, \quad u \in C, \quad m \in M, \quad t \in T \quad (27)$$

$$w_{a,i}^{k,t} - w_{a,j}^{k,t} + Q_K * x_{a,i,j}^{k,t} \leq Q_K - d_{a,j}^{k,t} \quad ,$$

$$a \in P, \quad (i,j) \in D, \quad k \in K, \quad t \in T, \quad i \neq j \quad (28)$$

$$d_{a,i}^{k,t} \leq w_{a,i}^{k,t} \leq Q_K * z_{a,i}^{k,t} \quad , \quad a \in P, \quad i \in D, \quad k \in K, \quad t \in T \quad (29)$$

$$V_{i,q}^{m,t} - V_{i,n}^{m,t} + Q_M * y_{i,n,q}^{m,t} \leq Q_M - b_{i,n}^{m,t} \quad ,$$

$$i \in D, \quad (n,q) \in A, \quad m \in M, \quad t \in T, \quad i \neq q, \quad i \neq n, \quad n \neq q \quad (30)$$

$$b_{i,n}^{m,t} \leq V_{i,n}^{m,t} \leq Q_M * s_{i,n}^{m,t} \quad , \quad i \in D, \quad n \in V, \quad k \in K, \quad t \in T \quad (31)$$

$$\sum_{a \in P} d_{a,i}^{k,t} \geq w_{a,i}^{k,t} \quad , \quad (i,j) \in D, \quad k \in K, \quad t \in T, \quad i \neq j \quad (32)$$

$$\sum_{i \in D} b_{i,n}^{m,t} \geq V_{i,q}^{m,t} \quad , \quad (n,q) \in V, \quad m \in M, \quad t \in T, \quad i \neq n, \quad n \neq q \quad (33)$$

$$l_u^{m,t} \geq tw_u^l - BigM(1 - s_{i,u}^{m,t}) + \sum_{n,q \in A} \tau_{nq} * y_{i,n,q}^{m,t} \quad ,$$

$$\forall u \in C, \quad i \in D, \quad m \in M, \quad t \in T \quad (34)$$

$$l_u^{m,t} \leq tw_u^s - BigM(1 - s_{i,u}^{m,t}) - \sum_{n,q \in A} \tau_{nq} * y_{i,n,q}^{m,t} \quad ,$$

$$\forall u \in C, \quad i \in D, \quad m \in M, \quad t \in T \quad (35)$$

$$I_e^t = \sum_{g \in G} I^{g,t} \quad , \quad \forall e \in V, \quad t \in T \quad (36)$$

$$d_{a,i}^{k,t}, w_{a,i}^{k,t} \in \mathbb{Z}^+ \quad , \quad a \in P, \quad i \in D, \quad k \in K, \quad t \in T, \quad (37)$$

$$b_{i,n}^{m,t}, V_{i,n}^{m,t} \in \mathbb{Z}^+ \quad , \quad i \in D, \quad n \in V, \quad k \in K, \quad t \in T, \quad i \neq n \quad (38)$$

$$x_{a,p,h}^{k,t}, z_{a,p}^{k,t} \in \{0,1\} \quad , \quad a \in P, \quad (p,h) \in V, \quad k \in K, \quad t \in T, \quad p \neq h \quad (39)$$

$$y_{i,n,q}^{m,t}, s_{i,n}^{m,t} \in \{0,1\} \quad , \quad i \in D, \quad (n,q) \in V, \quad m \in M, \quad t \in T, \quad n \neq q \quad (40)$$

La función objetivo (1) minimiza el costo total de envío, incluyendo el costo de transporte y el de almacenamiento. (2) y (3) hacen referencia a las restricciones de inventario relacionado al estado del producto, en donde (2) sugiere una transferencia progresiva de inventario en cuanto a la frescura del producto, mientras que (3) establece que en cada periodo  $t$ , el inventario más fresco se repone directamente con la cantidad recibida desde la planta. Las restricciones (4) – (9) permiten controlar los balances de inventario final para la planta, los distribuidores y los clientes, además de precisar la no negatividad de esas variables. (10) y (11) aseguran que el nivel de inventario final en cada vértice no supere su respectiva capacidad máxima. (12) y (13) restringen la cantidad enviada al distribuidor o cliente, para que esta no exceda la capacidad restante del mismo.

Posterior a esto se definen las restricciones asociadas al ruteo, donde (14) y (16) establecen que cada conjunto de aristas dirigidas entrante requiere uno saliente para cada distribuidor o cliente abastecido, en tanto que (15) y (17) garantizan un equilibrio en el ruteo asegurando que el número de vehículos que entran a un vértice debe ser igual al número de vehículos que salen del mismo. Las restricciones (18) y (19) garantizan que los vehículos  $k$  y  $m$  visiten como máximo una vez a su respectivo vértice en el periodo dado. (20) asegura que si el vehículo  $k$  proveniente de planta visita el vértice  $j$ , debe haber al menos un viaje directo de la planta al distribuidor  $i$ . (21) representa la misma restricción pero para la flota de vehículos  $m$ . (22) y (23) hacen respetar la capacidad máxima de cada flota de vehículos.

Por otra parte, (24) y (25) restringen la cantidad de bienes enviada de acuerdo con la capacidad de almacenamiento del origen si el vehículo ( $m$  o  $k$ ) visitan al vértice de interés. (26) y (27) dan a entender que si el vehículo de la respectiva flota visita al distribuidor o al cliente, al menos una unidad de producto debe ser transportada a estos vértices. Más adelante, (28) evita la existencia de subtours en el primer escalón. Por su parte (29) garantiza que no se entregue más de

lo que se envía y que las tres variables estén en rangos permitidos. (30) y (31) son las mismas restricciones de gestión de balance de carga y eliminación de subtours pero para la flota de vehículos  $m$ . (32) y (33) confirman que la cantidad enviada desde la planta al distribuidor o desde el distribuidor al cliente sea suficiente para cubrir todas las entregas. (34) y (35) son las restricciones asociadas a las ventanas de tiempo duras o estrictas. Por último, (36) garantiza que el inventario total en un nodo sea igual a la suma de los inventarios clasificados por su estado  $g$ , y las restricciones (37) – (40) definen el dominio de las variables.

Es importante resaltar que el modelo formulado en GAMS incorpora una relajación estratégica en las restricciones asociadas a las ventanas de tiempo. Ya que estas se formulan considerando únicamente los tiempos de viaje en el segundo escalón. Esta decisión se toma por dos razones principales, la primera es que el tramo más crítico para el cumplimiento de las ventanas de tiempo es el de la última milla, ya que se considera que los clientes finales son con quienes se tiene ese acuerdo de nivel de servicio, y quienes esperan un producto en las mejores condiciones posibles y en los tiempos que les favorezcan. Por otro lado, incorporar dependencias temporales durante todo el flujo de la red de distribución representa un aumento exponencial en la complejidad del problema. Ya que harían falta restricciones y variables adicionales que vinculen la llegada de un vehículo  $k$  a un CD, con el tiempo de salida de un vehículo  $m$  específicamente desde ese CD. Esto puede convertir el modelo computacionalmente intratable, inclusive para instancias pequeñas. Sin embargo, las ventanas son lo suficientemente amplias para que no afecte la comparación con el algoritmo de MATLAB, y el valor de función objetivo obtenido en GAMS sirve como un punto de referencia teórico de excelente calidad.

Por otra parte, para la validación de este proyecto investigativo se diseñan dos conjuntos de instancias computacionales: un grupo de 3 instancias pequeñas que permiten comparar el

tiempo de ejecución y la calidad de la solución obtenida, y otro grupo 3 de instancias de mayor envergadura para evaluar la escalabilidad del algoritmo en escenarios más realistas. Debido a que el método exacto propuesto no es computacionalmente tratable para las instancias grandes, la validación del modelo matemático comprende solo las instancias pequeñas (1-2-2, 1-2-3 y 1-4-2), caracterizadas por un horizonte de planificación de 2 periodos y un número reducido de nodos en el sistema. Los parámetros específicos como capacidades de almacenamiento, de vehículos, costos de envío y almacenamiento, penalizaciones y ventanas de tiempo se definen detalladamente en la sección 8.1 del presente documento.

### **5.3. Representación gráfica de la red de distribución**

El problema de inventario y ruteo de vehículos en una red de dos escalones con ventanas de tiempo y transporte de productos perecederos (2eIRPTW) formulado, es posteriormente codificado utilizando programación lineal entera mixta (PLEM) en el software GAMS (Ver apéndice A). Para garantizar la implementación del modelo se realiza la respectiva validación de este, verificando su funcionalidad y su coherencia estructural con la formulación matemática presentada anteriormente. Una vez realizado este proceso se concluye que el modelo cuenta con la capacidad de capturar adecuadamente la dinámica propuesta y generar soluciones óptimas en las instancias evaluadas dentro de los parámetros de entrada establecidos.

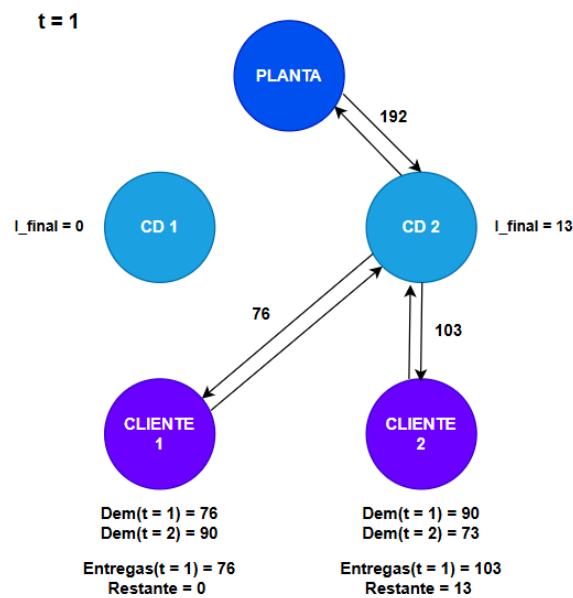
Con el fin de ilustrar la estructura de la red de distribución se presentan a continuación una serie de representaciones gráficas para dos de las instancias evaluadas. En estas se permite apreciar la asignación óptima de rutas propuesta por el software, junto con el flujo de producto a través de la cadena de suministro en los dos periodos de tiempo, el inventario en los nodos, las diferentes decisiones de abastecimiento y el cumplimiento de las ventanas de tiempo. Es importante resaltar que aunque en los diagramas solo se muestran las ventanas de tiempo para el segundo periodo

(donde  $TW(x,y)$  significa que la entrega puede hacerse entre el tiempo  $x$  y  $y$ ), en la codificación del modelo las decisiones de ruteo se ven afectadas por las ventanas de tiempo propuestas en todos los periodos evaluados. Además de esto, se asume que esta red de logística es gestionada de manera centralizada por parte de una misma empresa, lo que implica que todos los vehículos de la flota m pueden ser utilizados desde cualquier distribuidor que los necesite, y atendiendo a su vez a cualquier cliente dentro del grupo de nodos. Desde un punto de vista operativo, este problema de ruteo con múltiples orígenes en el segundo escalón le proporciona al modelo flexibilidad para encontrar soluciones de manera eficiente, sin tener que recurrir a una clusterización de clientes, que los asigne de forma rígida a ciertos distribuidores.

La primera instancia es representada en la siguiente figura, en donde se trata de una red de distribución 1-2-2 con una flota de dos vehículos  $k$  y cuatro vehículos  $m$ , los cuales operan bajo un horizonte de planificación de dos periodos de tiempo.

**Figura 4**

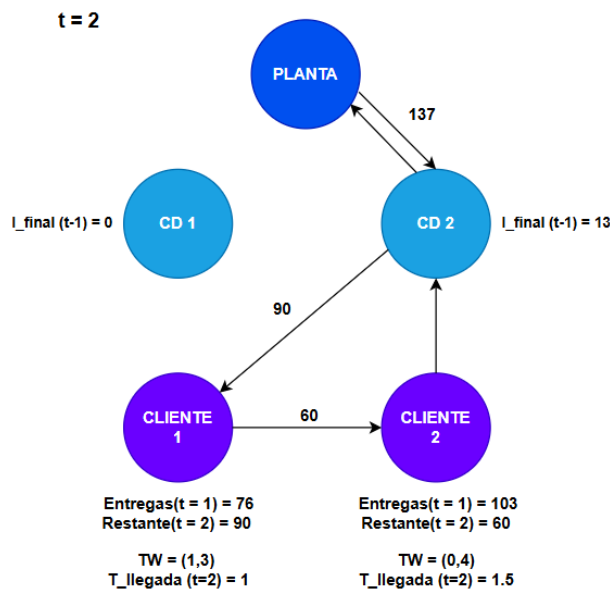
*Diagrama de la instancia 1 para  $t = 1$*



La figura 4 permite apreciar el flujo de abastecimiento en toda la cadena, además de los inventarios finales en los centros de distribución y la interacción entre ellos, en donde la estrategia de optimización busca asignar cantidades exactas de inventario para la entrega final de la demanda restante en el segundo periodo. En este primer periodo se utilizan ambos vehículos pertenecientes a la flota k. Seguidamente, se muestra la figura de la misma instancia pero para el segundo periodo de tiempo.

**Figura 5**

*Diagrama de la instancia 1 para  $t = 2$*



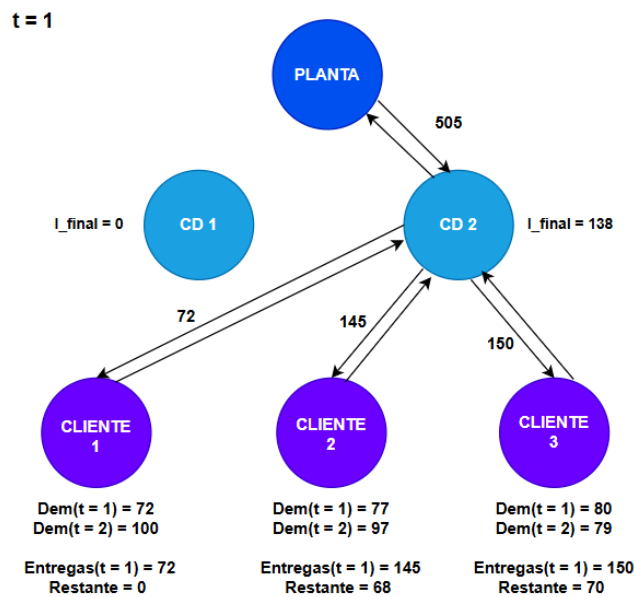
Se observa que para ambos periodos, la estrategia óptima de abastecimiento no incluye envíos al distribuidor 1, minimizando así los costos asociados al ruteo en el primer escalón, además de que en el segundo escalón se incluye el cumplimiento de la ventana de tiempo asociada a cada cliente, lo que muestra que la solución es factible y es una estrategia operativamente viable. Así mismo, se evidencia que se aprovecha la capacidad máxima de los vehículos m (150 para la instancia 1) para, con un mismo vehículo, abastecer a ambos clientes. De la misma forma, es

posible ver el correcto funcionamiento de los balances de inventario, donde la transferencia de productos de un periodo a otro se hace sin ningún inconveniente.

La figura 6 muestra el diagrama del primer periodo de tiempo para la segunda instancia evaluada, en donde aumenta el número de clientes y a su vez, varían ciertos parámetros de entrada como las capacidades tanto de los nodos como de los vehículos.

**Figura 6**

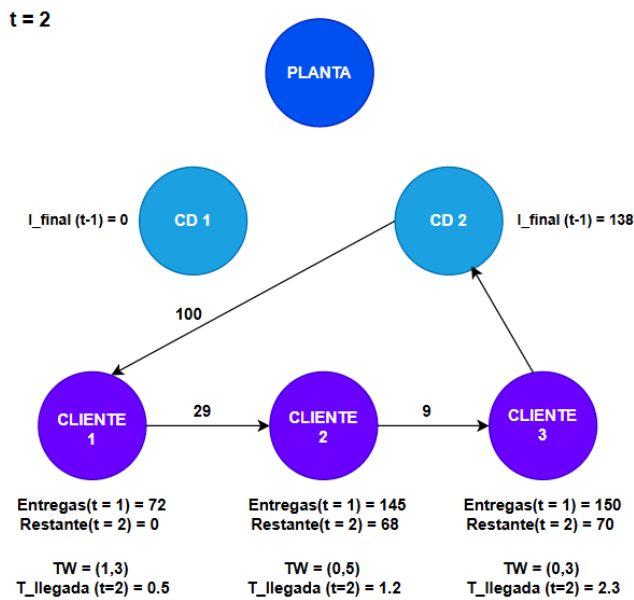
*Diagrama de la instancia 2 para  $t = 1$*



Para esta instancia se modela una red de distribución 1-2-3, con la misma cantidad de vehículos por flota que en la instancia anterior. Debido a que los vehículos de la flota  $m$  cuentan con una capacidad máxima de 170 unidades, los clientes 2 y 3 reciben más unidades de las que necesitan en el primer periodo, con el objetivo de habilitar la posibilidad del ruteo en el segundo periodo, asegurando así la optimización de los costos asociados al transporte. De la misma forma, el inventario final en CD 2 es suficiente para abastecer toda la demanda del segundo periodo, como se muestra a continuación.

**Figura 7**

*Diagrama de la instancia 2 para  $t = 2$*



Como se evidencia en las figuras 6 y 7, el modelo prioriza la operación únicamente desde el distribuidor 2, ya que los costos de envío desde la planta a este distribuidor son menores que los de enviar al distribuidor 1, y envía suficiente producto en el primer periodo para evitar tener que hacer un envío en el segundo periodo. Asimismo, los costos de almacenamiento de este nodo son más atractivos para el modelo que los del otro distribuidor. Finalmente, el modelo demuestra la capacidad de anticipar la demanda futura mediante una distribución estratégica en  $t = 1$ , y la estructura de las ventanas de tiempo valida la factibilidad del modelo.

#### 5.4. Análisis de sensibilidad

En este proceso fundamental en el análisis de modelos de optimización, se busca evaluar el impacto que tiene cambiar algunos parámetros de entrada con relación a las decisiones y resultados del sistema. En el contexto de este trabajo de investigación, realizar un análisis de sensibilidad permite identificar como la variación en los periodos, la cantidad de distribuidores, clientes y vehículos en cada flota afectan la estructura óptima de inventario y ruteo. Así mismo, se

planea evaluar el impacto de las ventanas de tiempo en el modelo, ajustándolas en diferentes escenarios para que sean más flexibles o rígidas. Además, es esencial entender que para los problemas de tipo NP-Hard, la complejidad no reside únicamente en el tamaño del sistema, lo que realmente puede llegar a representar una intratabilidad computacional es el crecimiento exponencial del espacio de soluciones factibles. La adición de un solo nodo al sistema no solo representa un incremento en la cantidad de variables, sino que genera un nuevo conjunto de permutaciones e interacciones logísticas. Por esta razón, el siguiente análisis constituye una aproximación que permite entender como pequeñas variaciones en los parámetros de entrada representan un aumento drástico en la dimensión del espacio de soluciones que debe ser explorado.

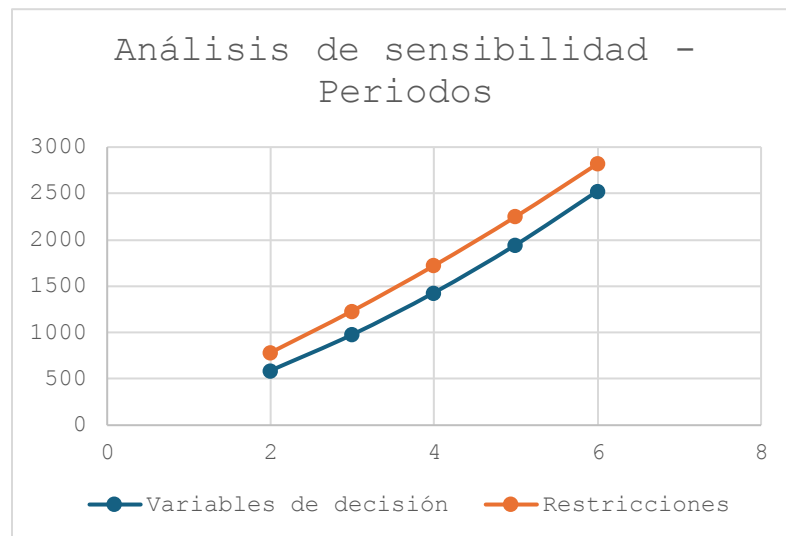
#### 5.4.1. Periodos

Las variaciones en los periodos se implementan mediante incrementos unitarios sucesivos, con el fin de analizar el comportamiento del modelo a lo largo de un horizonte temporal expandido.

**Tabla 4**

*Variación de periodos*

<b>Escenario</b>	<b>T</b>	<b>D</b>	<b>C</b>	<b>K</b>	<b>M</b>	<b>Variables de decisión</b>	<b>Restricciones</b>
1	2	2	2	2	4	590	784
2	3	2	2	2	4	978	1233
3	4	2	2	2	4	1430	1722
4	5	2	2	2	4	1946	2251
5	6	2	2	2	4	2526	2820

**Figura 8***Análisis de sensibilidad – Periodos de tiempo*

Se evidencia que la cantidad de restricciones es mayor a la de variables, lo que es esperado en modelos de optimización con un gran número de condiciones restrictivas, las cuales se encargan de guiar el espacio de soluciones. Es decir, el problema de estudio está altamente condicionado debido a la existencia de numerosas reglas que limitan las soluciones factibles, lo que refleja la necesidad de coordinar múltiples decisiones interdependientes. La tendencia presentada muestra una tasa de crecimiento alta en los primeros periodos y se muestra a continuación:

- De  $t = 2$  a  $t = 3$ , las variables aumentan en 65.8% y las restricciones en 57.2%.
- De  $t = 3$  a  $t = 4$ , el aumento es de 46.2% en variables y 39.6% en las restricciones.
- De  $t = 4$  a  $t = 5$ , el crecimiento es 36.1% en variables y 30.7% en restricciones.
- De  $t = 5$  a  $t = 6$ , el crecimiento disminuye a 29.8% en variables y 25.3% en restricciones.

#### 5.4.2. Distribuidores

La intención inicial consideraba formular una estrategia de análisis basada en incrementos de intervalos de dos distribuidores por escenario, pero al aumentar el número de distribuidores a

6, el modelo excede el límite de 5000 filas o columnas impuesto por la licencia de uso comunitario de GAMS, impidiendo su resolución en las condiciones propuestas. Por consiguiente, los escenarios evaluados se muestran a continuación.

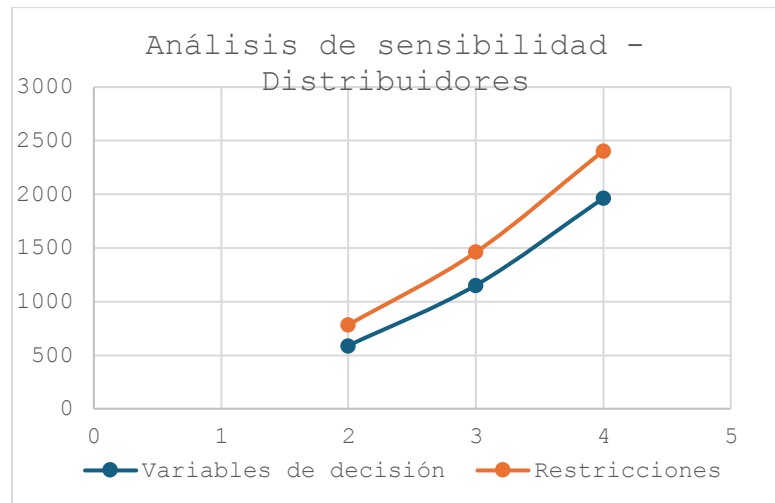
**Tabla 5**

*Variación de distribuidores*

Escenario	T	D	C	K	M	Variables de decisión	Restricciones
1	2	2	2	2	4	590	784
2	2	3	2	2	4	1153	1461
3	2	4	2	2	4	1964	2406

**Figura 9**

*Análisis de sensibilidad – Distribuidores*



Como se evidencia en la tabla 5, cada distribuidor adicional influye de forma significativa en el total de variables y restricciones del sistema, ya que introduce nuevas rutas, decisiones de asignación y restricciones operativas dentro del modelo. Se evidencia que al aumentar el número

de distribuidores en un 100%, la cantidad de variables de decisión aumenta en un 232.9%, mientras que las restricciones totales aumentan en un 206.9%. De la misma manera se concluye que la proporción de restricciones con respecto a las variables tiende a disminuir levemente a medida que el número de distribuidores aumenta.

**5.4.3. Clientes**

El análisis de sensibilidad asociado a este parámetro considera una variación progresiva de dos unidades en la cantidad de clientes y se muestra a continuación.

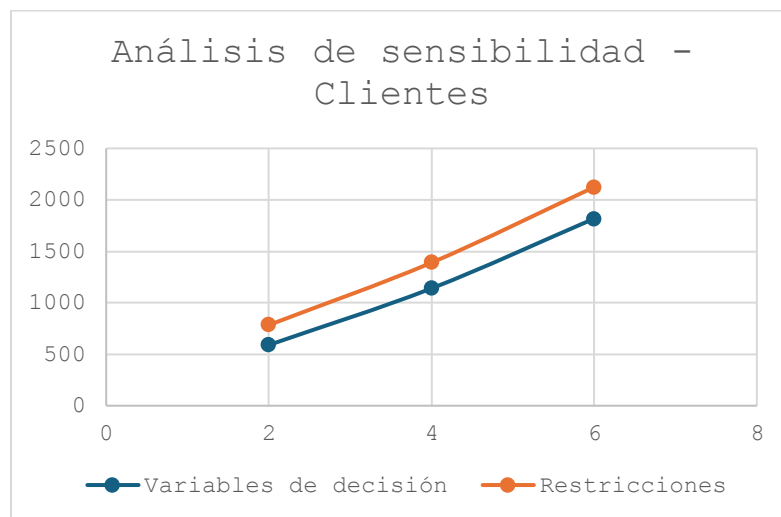
**Tabla 6**

*Variación de clientes*

Escenario	T	D	C	K	M	Variables de decisión	Restricciones
1	2	2	2	2	4	590	784
2	2	2	4	2	4	1140	1390
3	2	2	6	2	4	1818	2124

**Figura 10**

*Análisis de sensibilidad – Clientes*



Se observa que el aumento de este parámetro genera un incremento significativo tanto en la cantidad de variables como en las restricciones del modelo, aunque el crecimiento en las restricciones es ligeramente menor aumentando en un 170.9%, en comparación al 208.1% de las variables de decisión.

**5.4.4. Flota de vehículos K**

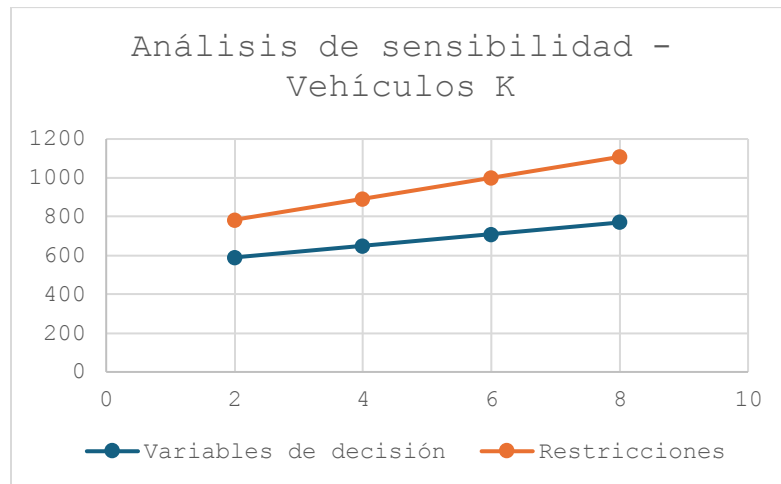
**Tabla 7**

*Variación de vehículos del primer escalón*

Escenario	T	D	C	K	M	Variables de decisión	Restricciones
1	2	2	2	2	4	590	784
2	2	2	2	4	4	650	892

**Figura 11**

*Análisis de sensibilidad – Flota primer escalón*



Debido a que el incremento de los vehículos de la flota del primer escalón no necesariamente genera nuevas rutas, ya que estas siguen restringidas por la demanda de los clientes y la capacidad en cada nodo, se puede concluir que la cantidad de vehículos del primer escalón no

es un factor crítico en la complejidad del problema, sino un recurso que se va ajustando dentro de los límites y requerimientos operacionales.

**5.4.5. Flota de vehículos M**

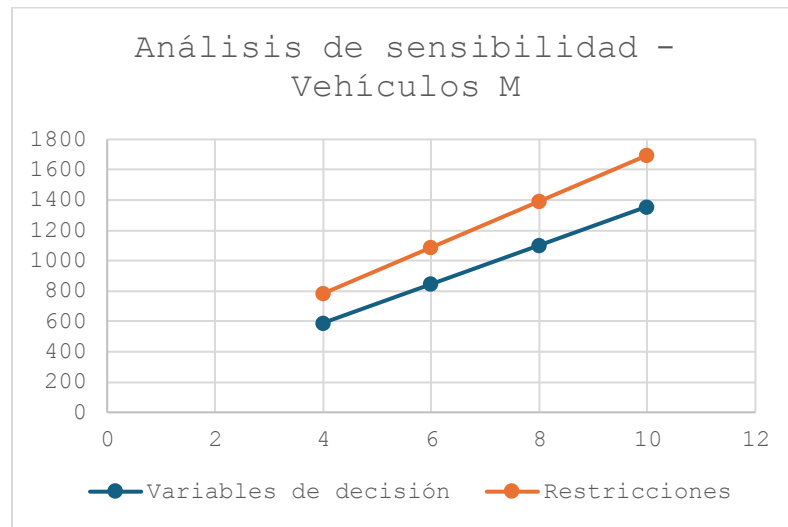
**Tabla 8**

*Variación de vehículos del segundo escalón*

Escenario	T	D	C	K	M	Variables de decisión	Restricciones
1	2	2	2	2	4	590	784
2	2	2	2	2	6	846	1088

**Figura 12**

*Análisis de sensibilidad – Flota segundo escalón*



Para la flota de vehículos correspondiente al segundo escalón se evidencia un comportamiento similar al de la flota del primer escalón, pero se observa que los cambios en estos parámetros son más significativos en comparación. Esto se debe principalmente a que los vehículos de la flota m tienen la posibilidad de empezar la ruta en diferentes orígenes

(distribuidores), y deben cumplir restricciones de capacidad y ventanas de tiempo para la entrega en la última milla.

#### 5.4.6. Ventanas de tiempo

Para este análisis, se toman los parámetros correspondientes al escenario 1 evaluado anteriormente, con la particularidad de que se evalúa el comportamiento del modelo haciendo variaciones en las ventanas de tiempo. Cabe aclarar que para efectos de presentación, en la tabla 9 se asume una única ventana de tiempo común para ambos clientes simplificando así la comparación entre escenarios. Se valoran tres instancias diferentes: sin restricciones asociadas a ventanas de tiempo, ventanas de tiempo amplias y ventanas de tiempo estrechas.

**Tabla 9**

*Variación de ventanas de tiempo*

<b>Escenario</b>	<b>TW (L.I.)</b>	<b>TW (L.S.)</b>	<b>Costo total</b>	<b>Variables</b>	<b>Restricciones</b>
1	$-\infty$	$+\infty$	4461,5	558	720
2	0	5	4461,5	590	784
3	0	1.5	5052,5	590	784

Se observa una menor cantidad menor de variables de decisión y restricciones para el escenario 1, lo cual es lógico ya que no se tienen en cuenta las restricciones temporales. Para el escenario con ventanas de tiempo amplias, se mantiene el costo de la corrida sin ventanas de tiempo, lo que sugiere que las entregas siguen siendo factibles sin mayores ajustes en la planeación. Por último, con las ventanas de tiempo estrechas, el modelo logra ajustarse operativamente a los requerimientos, pero con un incremento importante en el costo total.

En términos generales, el presente análisis permite identificar cuales parámetros tienen más incidencia en el comportamiento del modelo y proporciona información clave para entender el proceso de optimización del sistema de distribución estudiado. Entre las observaciones más relevantes se encuentra el hecho de que el aumento de distribuidores y clientes aumenta de forma significativa la complejidad del modelo, por lo que la expansión de la presente red de distribución impacta directamente la escala computacional del problema.

## **6. Descripción de la metaheurística**

En el marco de antecedentes formulado para la presente investigación, se observa la utilización de métodos de solución aproximados además de las soluciones exactas. Con base en esto se codifica para este modelo un algoritmo de solución para el 2eIRPTW estudiado. A partir de lo analizado en la revisión de literatura, se identifica al recocido simulado (SA) como una de las técnicas de solución mayormente usadas para este tipo de problemas. Diferentes autores utilizan esta metaheurística con algunas variaciones, por ejemplo, en Mirzaei & Seifi (2015) se formula un algoritmo metaheurístico híbrido que combina el recocido simulado (SA) con la búsqueda tabú (TS) para dar solución a un 2eIRP para transporte de productos perecederos. Por otra parte, en el 2eVRPTW formulado por Rahmanifar *et al.*(2023) se propone un algoritmo de recocido simulado (SA) y otro algoritmo de optimización basado en ingeniería social (SEO). Finalmente, para el estudio en cuestión se propone un algoritmo de recocido simulado (SA) mejorado con un operador de reparación, el cual demuestra ser una herramienta eficaz para resolver problemas de optimización combinatoria, en donde su capacidad de explorar soluciones de manera amplia para escapar de óptimos locales, acompañada de un operador de reparación y un mecanismo de diversificación (utilizado únicamente en las instancias de mayor envergadura) permiten encontrar soluciones de alta calidad en tiempos razonables.

### 6.1. Recocido simulado

El recocido simulado es un algoritmo de búsqueda local bastante utilizado en el campo de la optimización matemática. Este método probabilístico fue propuesto en 1983 por Kirkpatrick, Gellet y Vecchi, quienes trataban de encontrar el mínimo global para una función de costos caracterizada por poseer múltiples mínimos locales (Bertsimas & Tsitsiklis, 1993). El nombre de recocido simulado viene de la analogía al proceso que ocurre en la vida real en el campo de la metalurgia, donde se produce un recocido físico en los metales o sólidos, en el cual un sólido cristalino se calienta, para posteriormente enfriarlo de forma controlada hasta que alcance un estado donde la energía en la red sea mínima, obteniendo un sólido sin defectos cristalinos. Si el proceso de enfriamiento es lo suficientemente lento, el sólido que se obtiene cuenta con una integridad estructural muy superior gastando mínimas cantidades de energía (Henderson et al., 2006). A diferencia de un método de búsqueda local simple (como un algoritmo de descenso), donde el algoritmo explora un vecindario más reducido tendiendo a estancarse siempre en óptimos locales (posiblemente lejanos al óptimo global), el recocido simulado utiliza una estrategia en donde en algunos casos es posible elegir una solución que aumente el costo, evitando estancarse en óptimos locales y permitiendo una exploración más amplia del campo de soluciones. La decisión de aceptar o no un movimiento ascendente depende directamente de la temperatura en la que se encuentre el sistema, y viene determinada por una secuencia de probabilidad controlada. A partir de lo expuesto en Eglese (1990), se procede a definir detalladamente cada uno de los elementos que conforman la metaheurística de recocido simulado, con el fin de garantizar la coherencia teórica del modelo y fundamentar su implementación en criterios metodológicos validados en la literatura.

### ***6.1.1. Solución factible***

Se define a  $S$  como el conjunto de soluciones factibles que satisfacen todas las restricciones matemáticas impuestas por el modelo, aunque no necesariamente sea la mejor dentro del campo global de soluciones.

### ***6.1.2. Función objetivo***

La función objetivo (F.O.) cuantifica el criterio de optimización en el problema. Para este algoritmo, es la magnitud que se busca minimizar dentro del conjunto de soluciones factibles y el número de iteraciones definido.

### ***6.1.3. Solución inicial***

Se define una serie de parámetros de entrada para encontrar una solución inicial, la cual es definida como el estado  $i$ . Esta solución es el punto de partida, y a partir de ella el algoritmo empieza a iterar una vez definidas las demás características del recocido simulado. Si el problema cuenta con restricciones, se debe escoger dentro del espacio de soluciones una solución que satisfaga esas restricciones o de lo contrario permitir soluciones que no cumplan con todas las restricciones impuestas a expensas de una penalización convenientemente definida.

### ***6.1.4. Vecindario***

Defínase al vecindario como el conjunto de soluciones cercanas, que pueden generarse a partir de la solución actual. Muchos autores han demostrado que la eficiencia de un algoritmo SA está directamente relacionada con la estructura de vecindad utilizada, llegando a la conclusión de que una estructura de vecindad que impone una topología suave, en la que los óptimos locales son pocos profundos, es preferible en términos de eficiencia y recurso computacional que una estructura de vecindad irregular, en donde la cantidad de mínimos locales con una profundidad representativa es mucho mayor.

### **6.1.5. Nueva solución candidata**

Se genera un estado  $j$ , vecino de  $i$ , a partir de una serie de perturbaciones efectuadas sobre la solución actual. Así mismo, se codifican mecanismos que aseguren la factibilidad de la solución modificada.

### **6.1.6. Temperatura**

La temperatura dentro del bucle de recocido simulado se define como  $T$ , y hace referencia al parámetro que controla la exploración del algoritmo, la cual afecta la probabilidad de aceptar soluciones peores. El funcionamiento de este parámetro implica que un aumento pequeño en el valor de la función objetivo tiene más posibilidades de ser aceptado que un aumento más grande. De igual forma cuando  $T$  es alto, la mayoría de los movimientos serán aceptados, pero conforme  $T$  va disminuyendo su magnitud acercándose a cero, la mayoría de los movimientos que representen un aumento en el valor de la F.O. serán rechazados. El algoritmo procede a intentar un cierto número de movimientos de vecindad a cada temperatura, donde el parámetro  $T$  se va reduciendo de forma gradual. Generalmente, el SA se inicia con un valor  $T$  relativamente alto, para evitar un estancamiento temprano en óptimos locales.

### **6.1.7. Criterio de aceptación**

En el SA, el criterio de aceptación de una nueva solución está dado por una secuencia de números aleatorios, los cuales tienen una probabilidad controlada. La diferencia entre la solución actual ( $i$ ) y la vecina ( $j$ ) se define como  $\delta$ , y la probabilidad de aceptar un movimiento que causa un incremento  $\delta$  en la función objetivo generalmente se establece como  $e^{(-\frac{\delta}{T})}$  visualizado con mayor claridad de la siguiente manera:

$$P = e^{\frac{\{F.O.(j) - F.O.(i)\}}{T}}$$

### 6.1.8. Enfriamiento

Es necesario definir una función  $T(t)$ , que controle la transición entre los diferentes valores de temperatura a evaluar. Los principales métodos encontrados en la literatura son:

- **Enfriamiento geométrico.** Este método hace referencia a la utilización de una función de temperatura proporcional, el cual es el más común debido a su simplicidad y eficiencia computacional. Un ejemplo puede ser:

$$T(t + 1) = \alpha T(t)$$

En donde  $\alpha$  actúa como una constante que representa el factor de enfriamiento, generalmente definido como un valor cercano a 1.

- **Enfriamiento logarítmico.** La disminución de la temperatura ocurre de forma más lenta en comparación con el método geométrico, lo que en problemas particulares puede permitir una mayor exploración en el campo de soluciones. Está dado por:

$$T(t) = \frac{T_0}{1 + \beta \log(1 + t)}$$

Donde  $t$  representa la iteración actual,  $T_0$  la temperatura inicial y  $\beta$  controla la tasa de enfriamiento.

- **Enfriamiento lineal.** Este método es particularmente útil en problemas donde se requiere un descenso gradual, ya que permite reducir la temperatura a una velocidad constante. Un ejemplo de este método sería:

$$T(t) = T_0 - t\Delta T$$

Donde  $\Delta T$  representa la tasa de disminución de la temperatura de forma lineal.

### 6.1.9. Relación con cadenas de Markov

El algoritmo de recocido simulado puede ser modelado usando la teoría de las cadenas de Markov. Manteniendo el parámetro  $T$  como una constante se podría modelar el problema como

una cadena de Markov homogénea, en donde la matriz de transición  $P_{ij}$  que representa la probabilidad de pasar del estado  $i$  al  $j$  es independiente del número de iteración. En otro orden de ideas, el SA también puede ser descrito como una serie de cadenas de Markov homogéneas de longitud finita, con una reducción progresiva de  $T$ . De esta forma se podría considerar como una única cadena de Markov no homogénea de tiempo discreto, en la cual la probabilidad de cambiar de un estado a otro es dependiente del número de iteración.

#### ***6.1.10. Operador de reparación***

Es una herramienta opcional diseñada para ajustar soluciones no factibles generadas al ejecutar la metaheurística. En términos generales, este operador se encarga de detectar si una solución candidata viola alguna de las restricciones, aplicar modificaciones estratégicas que conviertan esta solución en una que cumpla con los requerimientos restrictivos del modelo, todo esto minimizando el impacto en la calidad de la solución y siendo computacionalmente eficiente.

Los operadores de reparación pueden clasificarse en:

- Operadores de reparación basados en eliminación.
- Operadores basados en redistribución.
- Operadores de expansión (Adicionan elementos faltantes).

#### ***6.1.11. Criterio de terminación o parada***

Determina cuando debe finalizar la ejecución del algoritmo. La estructuración del criterio de parada es clave para balancear la exploración y explotación del espacio de soluciones. Algunas de las condiciones de terminación más comunes en un SA engloban:

- Número máximo de iteraciones.
- Temperatura mínima alcanzada.
- Estancamiento en la mejora de la solución.

**6.1.12. Solución final**

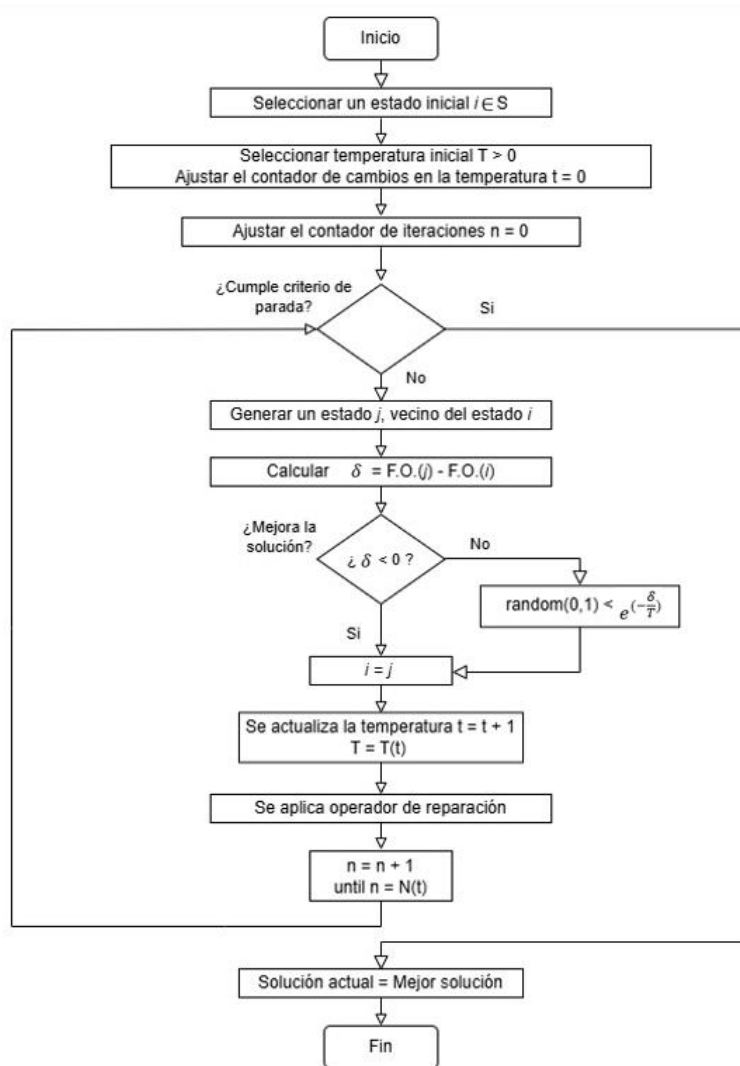
Es el punto final de la corrida del SA, definida como la mejor solución encontrada una vez se cumple el criterio de parada definido.

**6.1.13. Representación gráfica del SA**

Un diagrama de flujo general de un algoritmo de recocido simulado incluyendo un operador de reparación se muestra a continuación.

**Figura 13**

*Diagrama de flujo general Recocido Simulado + Operador de reparación*



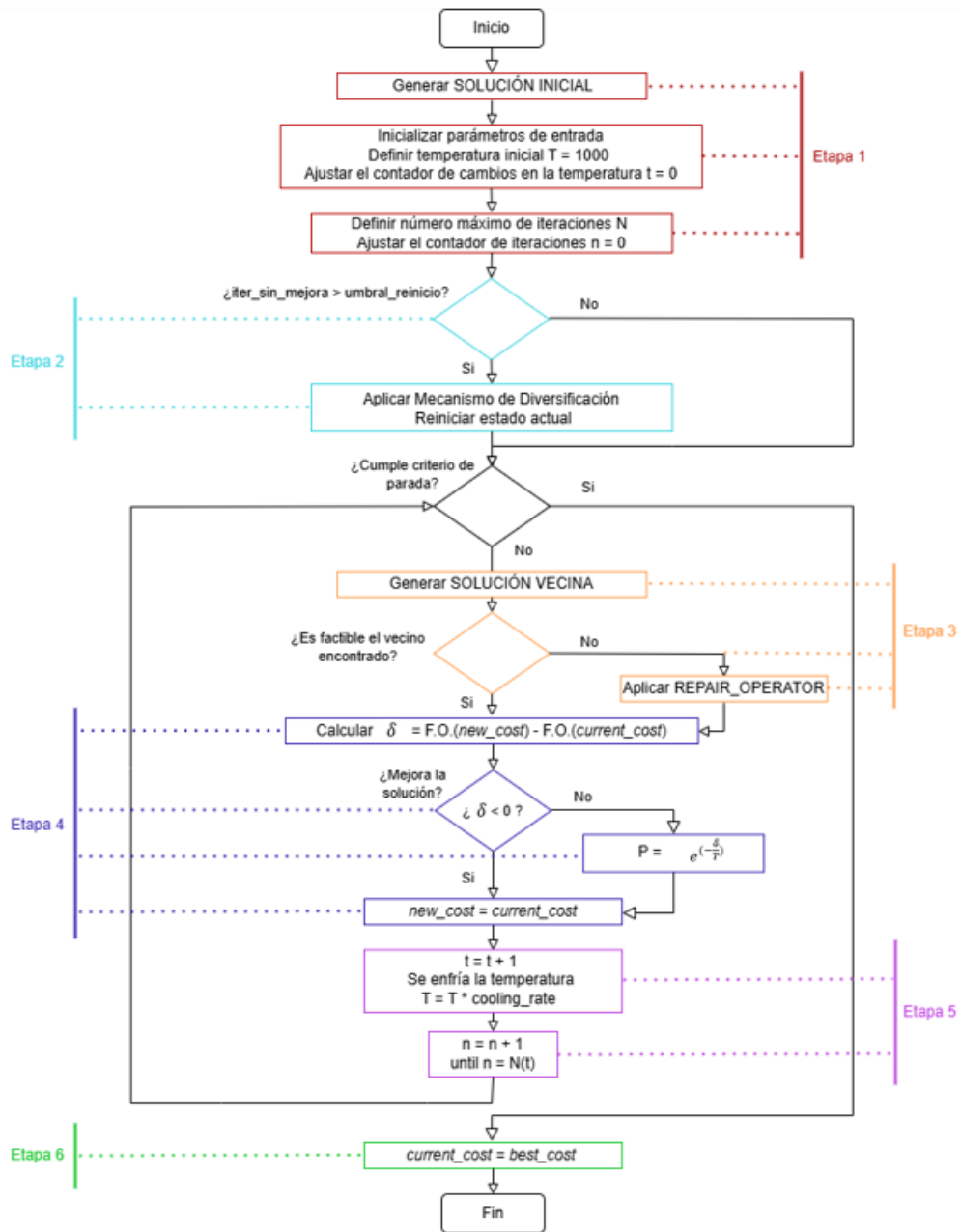
Nota: Adaptado de Eglese, 1990

### 7. Diseño del algoritmo de Recocido Simulado

A continuación se muestra el diagrama de flujo representativo del SA+RO construido específicamente para el presente trabajo.

**Figura 14**

*Diagrama de flujo algoritmo SA mejorado con repair operator*



## **7.1. Etapa 1: Inicialización y generación de solución inicial**

En esta primera etapa de la construcción del algoritmo de solución se establecen los datos de entrada con los que el sistema ejecutará el SA. Es el punto de partida del algoritmo y la correcta definición de estos parámetros es clave para el funcionamiento y eficiencia del SA en cuestión.

### ***7.1.1. Inicialización***

Se define la estructura del problema, en donde se dispone la cantidad de nodos con su respectiva enumeración y se asignan los valores de entrada de demanda, capacidades de las flotas y los nodos, restricciones asociadas al tiempo, costos de envío y almacenamiento y otros parámetros operativos. De la misma forma, se fijan los parámetros que conforman la estructura del bucle de SA, incluyendo la temperatura inicial, la cual debe ser lo suficientemente alta para permitir una mayor exploración del espacio de soluciones; la tasa de enfriamiento, que se trabaja de forma geométrica en el presente trabajo con un valor definido de 0,99 para las instancias pequeñas y 0,9999 para las últimas tres instancias evaluadas; y el número máximo de iteraciones, que actúa como el criterio de parada principal dentro de este algoritmo.

### ***7.1.2. Generación de la solución inicial***

En esta fase se define una subfunción “*createInitialSolution*”, capaz de generar la solución inicial del problema, a partir de la cual el algoritmo realiza las debidas comparaciones con las soluciones vecinas generadas posteriormente. La heurística construida dentro de esta función evoluciona notoriamente desde las instancias pequeñas a las grandes, debido al incremento en el requerimiento computacional y por ende, en la complejidad del problema. Por lo anterior, en el presente documento se evidencian las diferencias en el algoritmo construido dependiendo del tamaño de la instancia en el que fue aplicado.

### 7.1.2.1. Heurística constructiva para instancias pequeñas

La versión del algoritmo ejecutada en las instancias pequeñas utiliza una heurística compleja y estratégica, efectiva en instancias con un número reducido de nodos. A continuación, se describe su funcionamiento.

- **Cálculo de necesidades y producción:** Inicialmente, el algoritmo calcula la necesidad neta de inventario para cada CD por cada periodo. Este cálculo se basa en una proyección de la demanda de los clientes que se estima que ese CD va a servir. Esta estimación es temporal y simplificada y se usa únicamente dentro de esta subfunción. A partir de ese cálculo, se ajusta la producción en planta en cada periodo con el objetivo de cubrir esas necesidades proyectadas, acompañada de un factor de holgura que facilita la creación de soluciones factibles.
- **Creación de rutas del primer escalón:** Para esta decisión el algoritmo utiliza una estrategia dirigida por costo. En primer lugar, se ordenan los diferentes CD según el costo de transporte desde la planta, a partir de lo establecido en la matriz de costos  $c_1$ . Una vez hecho esto se toma uno de los vehículos  $k$  y se crea una ruta de forma codiciosa, en donde el algoritmo busca el CD que tiene el costo de transporte más bajo desde la planta y lo añade a la ruta actual y carga el producto. Esto se hace porque se considera la mejor opción inmediata. A partir de ese punto, se intenta añadir una nueva parada en otro CD dentro de esa ruta siempre que la capacidad del vehículo no sea excedida. Una vez el vehículo está en su límite de capacidad, esa ruta se cierra volviendo a la planta y se toma un nuevo vehículo para repetir el proceso hasta que toda la necesidad proyectada en ese periodo sea satisfecha.

- **Creación de rutas del segundo escalón:** Para este escalón, se toma un enfoque de servicio abierto, en donde el primer paso consiste en la creación de un mapa que refleja la demanda pendiente de los clientes para cada periodo. Una vez creado este mapa, el algoritmo empieza a iterar sobre cada CD de forma secuencial, uno después del otro. Esta estrategia intenta enviar todo el producto posible desde el primer CD en el mapa, antes de considerar despachar un segundo camión desde algún otro CD. A partir de aquí, se vuelve a tomar un enfoque codicioso para el llenado de las rutas, en donde el algoritmo decide añadir un cliente al vehículo actual siempre y cuando el CD de origen cuente con el suficiente inventario y la demanda de ese cliente no hace que se supere la capacidad del vehículo. Finalmente la ruta regresa al CD de origen y se actualizan las variables globales dentro del mapa construido.

#### 7.1.2.2. Heurística First Fit Decreasing (FFD) para instancias grandes

Debido a problemas de escalabilidad al aumentar el tamaño del problema, la heurística propuesta evoluciona para ser integral y consciente de las restricciones del problema, mejorando a su vez la lógica de enrutamiento simplista propuesta para las instancias pequeñas.

- **Asignación capacitada de almacenamiento:** Se desarrolla un mecanismo que busca evitar la sobrecarga de la capacidad de almacenamiento en los CD. Para esto se construye nuevamente un mapa que ordena de menor a mayor el costo de transporte para llegar a cada cliente, con el fin de asignarlo al CD más barato si la capacidad de este lo permite. Si no, prueba con el segundo CD más económico y así sucesivamente hasta encontrar un CD disponible para realizar el envío.

- **Creación de rutas del primer escalón:** Para esta heurística, se plantea una estrategia previamente definida a partir del mapa construido anteriormente. Primero, el algoritmo calcula la demanda agregada por cada CD sumando las demandas de los clientes que le fueron asignados en el paso anterior, y a partir de ese valor, se genera una ruta desde la planta que envíe a cada CD específicamente la cantidad que le fue asignada a repartir, asegurando que el producto se distribuya donde realmente se necesita.
- **Creación de rutas del segundo escalón:** Con el objetivo de crear rutas que puedan abastecer las demandas agregadas de los clientes, se utiliza ahora si la estrategia *First Fit Decreasing (FFD)*. Esto es posible porque ya el inventario está ubicado estratégicamente en cada uno de los CD por el ruteo realizado en el primer escalón. Primeramente, la fase *Decreasing* brinda inteligencia a la heurística ordenando de mayor a menor la demanda agregada del conjunto de clientes. En este punto, el algoritmo prioriza a los clientes con mayor demanda, asegurando que las necesidades agregadas más grandes se atiendan primero y ocupen eficientemente el espacio disponible dentro de los vehículos, dejando así espacios más pequeños para los clientes con demandas agregadas más manejables. Después, viene la fase *First Fit* en donde, con la lista de clientes ya creada, se ejecuta una lógica de primer ajuste para cada cliente. Al comenzar la corrida, el conjunto de rutas para el primer CD seleccionado está inicialmente vacío. A partir de aquí, el algoritmo inicia un bucle anidado que itera sobre las rutas que se van creando, comenzando siempre por la primera ( $k = 1$ ). Para cada ruta que se va creando, se calcula la carga actual dentro del vehículo y se verifica si añadir la demanda del cliente actual sobre el que

se itera excedería la capacidad del vehículo  $k$ . Si esa condición se cumple, el algoritmo inmediatamente asigna el cliente a esa ruta y actualiza la secuencia y las cantidades entregadas de momento. En este punto, se termina el bucle anidado y el algoritmo no sigue buscando un mejor lugar para ese cliente, sino que se compromete con la primera opción factible que encuentra. Si el bucle se completa sin que el algoritmo encuentre un lugar en el vehículo para el cliente, se crea una nueva ruta siempre y cuando haya vehículos disponibles en la flota, esta ruta se añade al final del conjunto de listas, para que los clientes con menor demanda agregada intenten ajustarse a ella si la capacidad del vehículo lo permite. Este proceso iterativo continua hasta que todos los clientes estén asignados a una ruta que permita que sus necesidades del periodo sean abastecidas.

Es fundamental clarificar que el objetivo de esta solución inicial obtenida es generar una solución estructuralmente factible, pero se omiten algunas de las verificaciones clave del problema, las cuales delegan a otra subfunción que evalúa las restricciones más complejas y costosas en términos computacionales. En “*createInitialSolution*” se hace una planificación más rápida, y es “*evaluateSolution*” la encargada de someter esa solución a un escrutinio más riguroso para verificar el cumplimiento de las ventanas de tiempo, rastrear el estado de cada lote de producto y calcular costos de transporte, almacenamiento y frescura. La intención es que el algoritmo siempre sea capaz de generar una solución inicial factible, ya que esto ayuda a explorar el espacio de soluciones de forma más eficiente, por eso se construyen estas heurísticas que son lo suficientemente robustas para garantizar que esto suceda, sin embargo, si en algún caso excepcional no se encuentra una solución inicial factible, el algoritmo pasa a generar una solución vecina y de este punto parte la exploración del espacio de soluciones.

### **7.1.3. Función de evaluación y modelo de costos**

La función “*evaluateSolution*” es el centro de esta fase, y actúa como un componente transversal dentro del algoritmo SA+RO propuesto. Esta función actúa como un auditor exhaustivo durante todo el flujo del algoritmo, recibiendo cada una de las soluciones candidatas que se generan durante la corrida con el fin de verificar su factibilidad, y en caso de que cumpla con todas las restricciones operativas del modelo, también se encarga de calcular su costo total como la suma de los costos de transporte, almacenamiento y penalización por frescura. Cuando recibe una estructura de solución que incumple alguna de las restricciones, esta función la declara inviable y automáticamente le asigna un costo de penalización elevado, asegurando que sea rechazada por el SA. A continuación se detalla su funcionamiento para cada periodo  $t$  evaluado.

#### **7.1.3.1. Envejecimiento de inventario y producción**

Como primera operación, el algoritmo se encarga de simular el paso del tiempo sobre el inventario que no se consumió en el periodo. Para esto, se toman los inventarios finales en cada nodo del periodo  $t - 1$  y el estado de cada lote de producto se incrementa en uno reflejando su pérdida de frescura. La cantidad producida en el periodo  $t$  se añade al inventario final en planta en  $t - 1$ , y ese producto nuevo siempre entra con el estado de mayor nivel de frescura ( $g = 0$ ).

#### **7.1.3.2. Evaluación de rutas del primer escalón**

En este punto se auditan las rutas que van desde la planta a los CD. Se realizan varias validaciones críticas, entre las cuales se incluye una revisión del inventario en planta, asegurándose de que la cantidad de producto que se planea enviar en los vehículos  $k$  no supere la cantidad disponible en este nodo. Así mismo, se verifica que la carga total de cada vehículo no supere su capacidad máxima, y que el inventario que se va a entregar a los CD no supere su capacidad de almacenamiento. Es importante resaltar que el algoritmo diseñado sigue la metodología FIFO

(*First in, First out*) a la hora de cargar los vehículos. Se priorizan los lotes de producto más antiguos disponibles con la intención de minimizar la penalización por frescura. Finalmente, el algoritmo rastrea efectivamente las diferentes edades que puede tener un mismo envío y transfiere esa composición de frescura al CD escogido como destino. Una vez hecho esto se calcula el costo total de transporte en el primer escalón sumando los costos de cada arco recorrido.

#### **7.1.3.3. Evaluación de rutas del segundo escalón**

Debido al aumento en el número de clientes y a la inclusión de ventanas de tiempo para las entregas, esta es la fase más compleja dentro de la subfunción de evaluación. Primero, antes de crear cualquier ruta, se verifica que el CD de origen tenga suficiente inventario para cubrir la carga proyectada. Después, el algoritmo calcula el tiempo de llegada acumulado para cada parada de la ruta sumando los tiempos de la matriz de tiempos de viaje previamente definida. Para todos los clientes, se verifica que la entrega se realice dentro de los límites establecidos, y si no es posible, se declara esa solución como inviable. De la misma manera, el algoritmo verifica que la carga de cada ruta no supere la capacidad del vehículo  $m$ . Seguidamente, así como en el primer escalón, el algoritmo sigue una política FIFO buscando minimizar pérdidas y aumento en los costos, y una vez hecho esto procede a calcular el costo de penalización para todas las unidades que fueron entregadas con un estado de frescura  $g > 0$ . Finalmente, se acumula el costo total de transporte usando la matriz de costos propuesta.

#### **7.1.3.4. Evaluación de costo de almacenamiento**

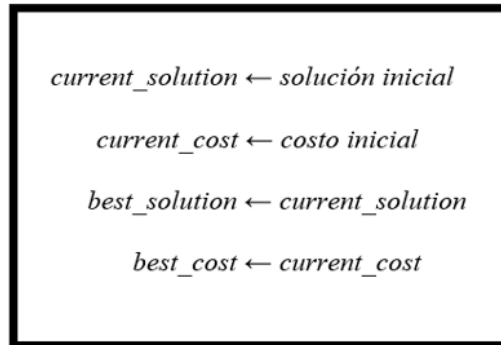
Al final de la simulación del periodo  $t$ , se calcula el costo total de almacenamiento multiplicando el inventario final en cada uno de los nodos por su respectivo costo de almacenamiento, luego se suma este costo al costo total de la solución actual. Finalmente, el algoritmo hace un último chequeo para asegurar que la demanda de todos los clientes sea cero

después de haber recibido producto desde los CD. De encontrar que alguna demanda queda insatisfecha, la solución actual se considera como inviable y se penaliza automáticamente.

Finalmente, se asigna la solución inicial como se muestra en la figura 15.

**Figura 15**

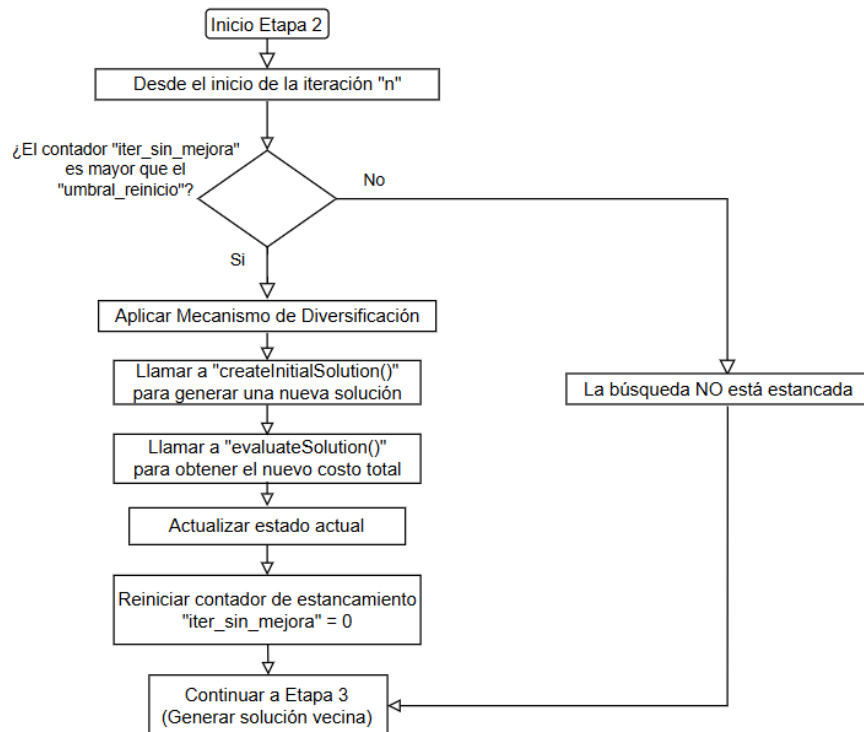
*Asignación de solución inicial*



**7.2. Etapa 2: Estrategia de diversificación**

**Figura 16**

*Diagrama de flujo etapa 2 SA+RO*



La figura 16 permite entender el funcionamiento de este control de estancamiento impuesto únicamente en las 3 instancias de mayor envergadura. Esta decisión se toma considerando que, en problemas con espacios de búsqueda muy amplios es común que se le dificulte al algoritmo salir de óptimos locales y esta situación se estaba presentando específicamente en las instancias más complejas. El objetivo principal de esta etapa no es el de necesariamente mejorar la solución actual de forma gradual, sino forzar al algoritmo a escapar del estancamiento cuando la búsqueda se ha vuelto improductiva. El funcionamiento detallado de esta estrategia se expone seguidamente.

Primero, la variable “*iter\_sin\_mejora*” se define para hacer un monitoreo del estancamiento, y es la encargada de rastrear la cantidad de iteraciones consecutivas en las que no se ha encontrado una mejor solución global. Cada vez que “*current\_soution*” no mejora el “*best\_cost*” el contador incrementa en una unidad, y por otra parte, cada vez que el algoritmo mejora el “*best\_cost*” actual, el contador se reinicia permitiendo que la exploración se siga realizando por ese mismo espacio de soluciones.

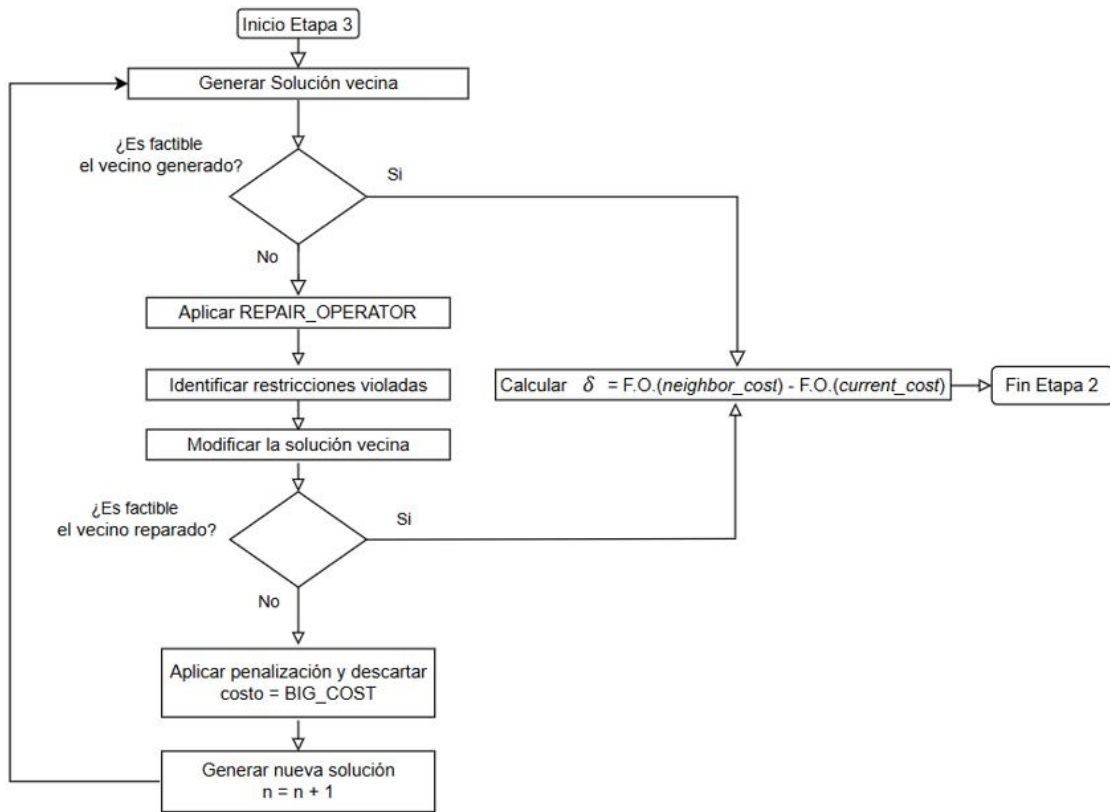
En segundo lugar, se predefine un parámetro que tiene la función de delimitar el número de iteraciones sin mejoras en el costo total. Expresado como un parámetro de paciencia, se define “*umbral\_reinicio*” con un valor de 5000 para determinar si la búsqueda local está siendo efectiva o no, y en caso de que una búsqueda se extienda por más de 5000 iteraciones sin encontrar una mejor solución, se activa el mecanismo propuesto para diversificar la búsqueda.

Finalmente, el mecanismo ejecuta un salto drástico en el espacio de soluciones y para esto se llama nuevamente a la heurística de la solución inicial con el fin de obtener una solución que no tenga relación con el óptimo local en el que existe un atasco. En las corridas realizadas, este mecanismo proporciona un balance entre la intensificación y la diversificación, y es fundamental para obtener resultados de alta calidad.

**7.3. Etapa 3: Generación y evaluación de vecinos**

**Figura 17**

*Diagrama de flujo etapa 3 SA+RO*



En la figura 17 se evidencia de forma precisa el flujo del algoritmo propuesto dentro de esta etapa. El núcleo de esta fase es la generación de las soluciones vecinas que permiten explorar el campo de soluciones de una forma mucho más eficiente y robusta. A continuación se presenta de forma detallada cada sección del flujo presentado.

**7.3.1. Generación de vecinos**

Una vez que el algoritmo tiene una solución inicial de partida, el recocido simulado entra en acción y comienza un bucle iterativo con el objetivo de promover algunos movimientos en la solución actual que permitan mejorar la solución del problema. Para este proceso de búsqueda local se plantean 11 diferentes operadores que permiten generar perturbaciones controladas con el

objetivo de encontrar mejoras incrementales. Dentro de los operadores de vecindad propuestos, algunos son más simples y controlados, mientras que otros son más disruptivos y robustos con el fin de navegar eficazmente el espacio de soluciones. Se realiza una descripción detallada de cada uno de los operadores con el objetivo de entender su estructura y funcionamiento, acompañado de un ejemplo ilustrativo que permite entender el funcionamiento de estos operadores de forma gráfica (Ver Apéndice E).

Estos operadores no se eligen de forma secuencial o determinista, sino que en cada iteración, la selección del operador que se ejecuta depende de una probabilidad ponderada, introduciendo un elemento estocástico a la metaheurística. Para esto, se define previamente el vector “*op\_weights*”, el cual contiene 11 valores numéricos que representan la probabilidad relativa de cada uno de los operadores propuestos. La asignación de esos pesos no es arbitraria ni equiprobable, sino que busca equilibrar la intensificación y la diversificación dentro del espacio de soluciones. Para esto, se realiza un proceso de calibración de parámetros en donde la asignación de pesos se define a través de una experimentación preliminar, en el que se monitorea y registra la contribución de los diferentes operadores propuestos. Con base en esto, a los operadores que generan un mayor impacto en la calidad de la solución se les asigna una probabilidad más alta, así mismo, a los operadores que demostraron durante las corridas de prueba una alta efectividad para el refinamiento de soluciones se les asignan probabilidades más controladas. Por otro lado, a los operadores más disruptivos y computacionalmente intensivos se les asigna una probabilidad media-baja dependiendo del tamaño de la instancia. Finalmente, se escoge una distribución de probabilidades que proporciona tiempos de cómputo razonables y soluciones de alta calidad, evitando a su vez la convergencia en óptimos locales prematuros.

Dentro del algoritmo construido, la trazabilidad de la relevancia de cada operador se establece en esta experimentación previa que se menciona, durante la fase de calibración de parámetros. Esto debido a que, al incluir un componente de aleatoriedad en el proceso de selección del operador que se va a utilizar en cada iteración, se evalúa el comportamiento del algoritmo a través de la contribución agregada de todos los operadores a lo largo de múltiples corridas. Por ende, los 11 operadores construidos contribuyen de forma sinérgica al equilibrio entre la intensificación y la exploración.

### ***7.3.2. Evaluación y reparación de vecinos***

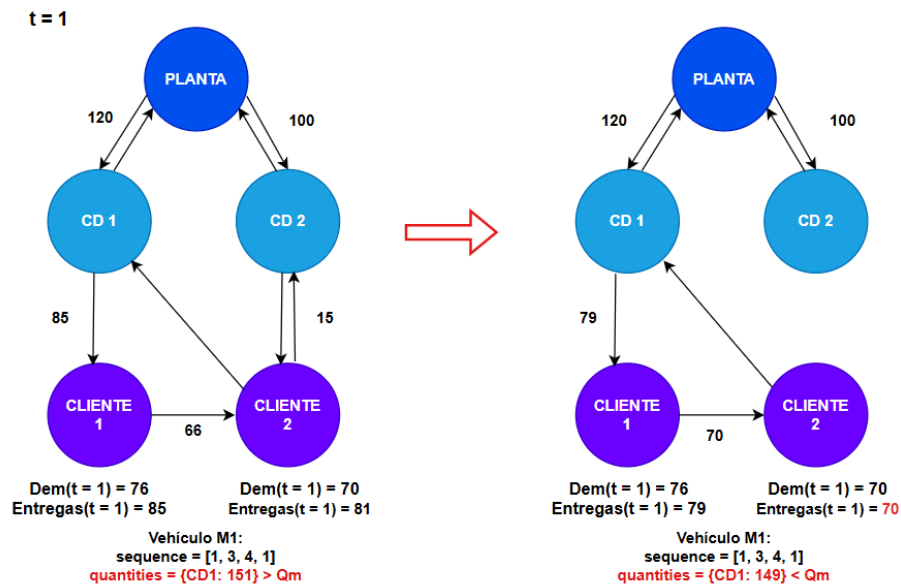
Una vez la solución vecina es generada esta pasa de nuevo a ser evaluada, y puede suceder que se viole alguna de las restricciones impuestas, como por ejemplo exceder la capacidad de los vehículos ( $Q_k$  y  $Q_m$ ), tener inventarios finales negativos en distribuidores o clientes, demanda insatisfecha a alguno de los clientes o que se incumpla la ventana de tiempo establecida para las entregas del segundo escalón. Inicialmente, en este punto se les asigna a las soluciones no factibles un costo muy alto denominado BIG\_COST. Un SA tradicional descartaría esta solución directamente, posiblemente perdiendo algunas soluciones prometedoras, no obstante, en el SA+RO formulado el operador de reparación entra en acción en este punto para modificar esas soluciones, intentando ajustar su factibilidad a través de una serie de cambios. Los diferentes mecanismos de reparación que se pueden aplicar se exponen a continuación.

#### **7.3.2.1. Sobrecarga de vehículos.**

Después de evaluada la solución actual, si se encuentra que la capacidad de algún vehículo fue excedida, la reparación identifica la ruta con sobrepeso y procede a eliminar clientes de esa ruta tratando de que la carga total esté dentro de los límites de los vehículos  $k$  o  $m$ .

**Figura 18**

*Repair operator – Sobrecarga de vehículos*



**7.3.2.2. Ajuste de inventarios negativos.**

Se analiza también el inventario final en cada uno de los nodos, para asegurarse de que estos no tomen valores negativos. En caso de que se encuentre algún inventario negativo, el operador de reparación entra en acción para aumentar y redistribuir la producción a través de los distribuidores del grafo. En este punto también se validan otras restricciones, por ejemplo, después de redistribuir la producción se hace un recuento de inventarios, y si algún CD no tiene suficiente inventario para cubrir una de las rutas que hacía originalmente, se elimina esa entrega evitando inconsistencias en la solución.

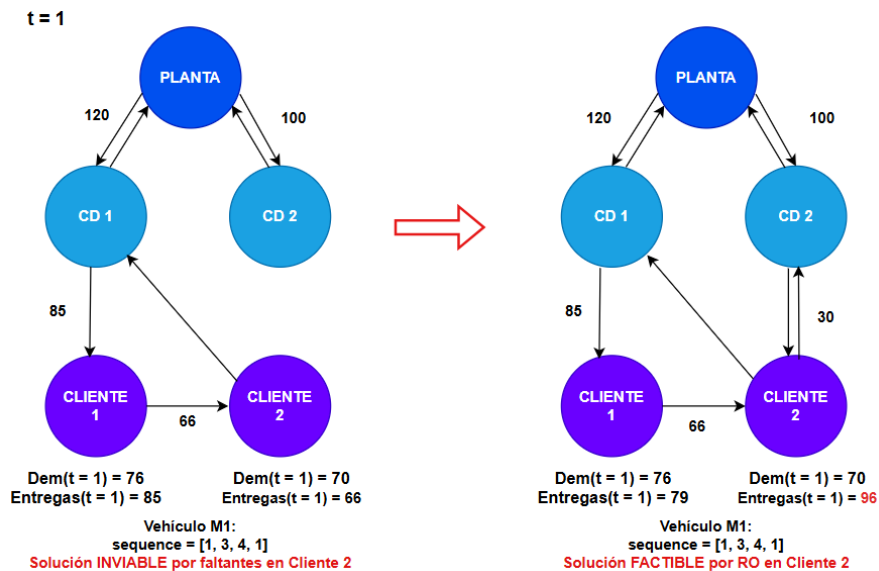
**7.3.2.3. Demanda insatisfecha.**

El tercer mecanismo de reparación que se habilita en el algoritmo es la modificación de la cantidad de rutas en el segundo escalón, específicamente cuando la demanda de alguno de los clientes no se satisface durante el periodo t. Debido a que en este modelo no se aceptan faltantes, este operador se encarga de asegurarse de que los clientes reciban toda la demanda requerida en

cada periodo  $t$ , a través de la creación de nuevas rutas directas que lleven el producto restante para satisfacer sus necesidades. Estas rutas se crean siempre y cuando el CD de origen cuente con suficiente inventario, y se verifique que efectivamente hay vehículos disponibles dentro de la flota  $M$ .

**Figura 19**

*Repair operator – Demanda insatisfecha*



En este ejemplo, el *repair operator* se encarga de crear una nueva ruta  $m$  después de hacer las verificaciones necesarias, con el objetivo de satisfacer la demanda del Cliente 2.

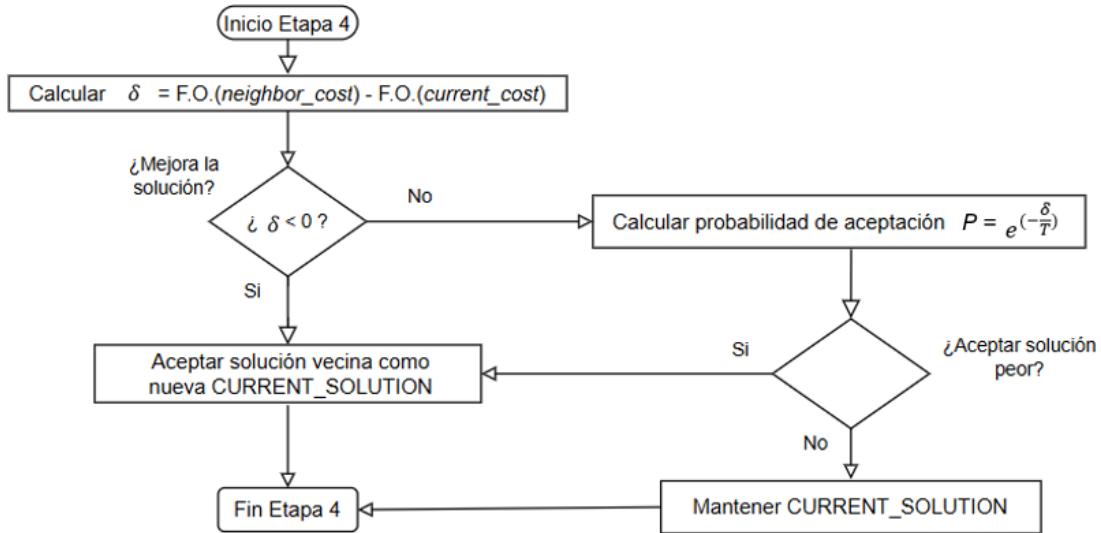
**7.3.2.4. Penalización en soluciones inviables.**

Una vez aplicados los mecanismos mencionados anteriormente sobre la solución actual, se evalúa la solución para verificar su factibilidad en el contexto de los parámetros y condiciones restrictivas impuestas por el modelo. En caso de que el operador no sea capaz de reparar una solución, este procede a asignarle un valor muy grande de penalización denominado **BIG\_COST**, con el fin de descartar la solución y proceder a generar un nuevo vecino.

**7.4. Etapa 4: Criterio de aceptación de soluciones vecinas**

**Figura 20**

*Diagrama de flujo etapa 4 SA+RO*



En esta etapa, el recocido simulado decide si se acepta o no la solución vecina generada en la etapa 2, basado en la comparación de costos y la aplicación de una función probabilística que permite aceptar soluciones peores que la actual, teniendo en cuenta la temperatura en la iteración. El primer paso es la evaluación de costos totales en la solución vecina, con el objetivo de identificar el valor de  $\delta$ . Si la solución nueva permite minimizar los costos totales en comparación a la solución actual, esta se acepta automáticamente, un ejemplo de esta situación se presenta a continuación.

**Figura 21**

*Nueva solución < solución actual*

<i>current_cost = 6500</i>	<i>new_cost &lt; current_cost</i>
<i>new_cost = 6350</i>	<i>solution = new_solution</i>
<i>Temperatura = 750</i>	<i>current_cost = new_cost</i>

*Como 6350 < 6500 la solución se acepta automáticamente*

En contraste, si la nueva solución no es mejor que la actual, con el fin de escapar de mínimos locales y mejorar la exploración del algoritmo se evalúa una función de probabilidad de aceptación  $P = e^{(current\_cost - new\_cost)/T}$ , en donde la probabilidad de que se acepte una solución peor es mucho mayor cuando la diferencia entre los costos es pequeña, o al inicio de la exploración en el espacio de soluciones, donde la temperatura del sistema es más alta.

**Tabla 10**

*Función de probabilidad de aceptación*

Escenario	Valor $\delta$	T°	Probabilidad de aceptación $P$	Decisión esperada
1	Pequeña diferencia	Alta	$P \approx 1$	Se acepta casi siempre
2	Pequeña diferencia	Baja	$P \approx 0.6$	Se acepta a veces
3	Gran diferencia	Alta	$P \approx 0.3$	Se acepta pocas veces
4	Gran diferencia	Baja	$P \approx 0$	Casi nunca se acepta

Un ejemplo numérico para ilustrar el funcionamiento de esta función se muestra en la figura 22.

### Figura 22

*Nueva solución  $\geq$  solución actual*

<i>current_cost = 5200</i>	<b>if</b> <i>rand</i> < <i>probability</i>
<i>new_cost = 5500</i>	<i>solution = new_solution</i>
<i>Temperatura = 800</i>	<i>current_cost = new_cost</i>
$P = e^{(5200-5500)/800} = e^{-0.375} \approx 0.687$	

En donde *rand* representa un número en el intervalo [0,1] generado de forma aleatoria. A partir de esto, se presentan dos posibilidades:

- Si *rand* = 0.45, como  $0.45 < 0.687$  la solución peor se acepta.
- Si *rand* = 0.8, como  $0.8 > 0.687$  la solución se rechaza y se mantiene la actual.

### 7.5. Etapa 5: Mantenimiento del bucle de Recocido Simulado

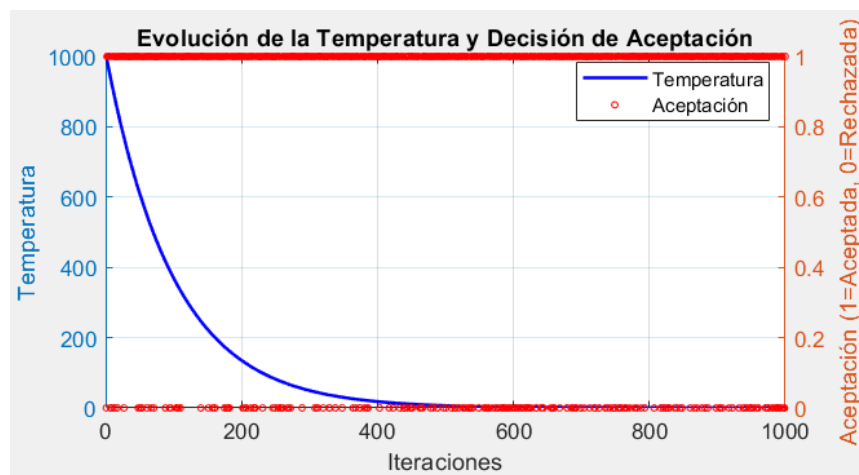
Para esta etapa se realiza un reajuste de los parámetros asociados directamente al recocido simulado para mantener su estructura y garantizar un correcto funcionamiento. Debido a que los procesos de la etapa 5 y 6 son cortos y secuenciales, se han priorizado los diagramas de flujo para las etapas anteriores, las cuales representan una mayor complejidad. Se considera que una representación textual detallada es suficiente para explicar el funcionamiento de estas etapas de forma clara y concisa.

### 7.5.1. Enfriamiento y control de iteraciones.

La primera fase de esta etapa se encarga de mantener un control del parámetro asociado a la temperatura en la que se encuentra el sistema. La temperatura dentro del SA+RO actual permite controlar el proceso de exploración, y sin la actualización de este parámetro, el algoritmo pierde su capacidad de exploración global dentro del espacio de soluciones, y se comportaría como una búsqueda local, limitando su eficiencia. La ecuación de enfriamiento está dada por  $T = T * cooling\_rate$  y en este caso se utiliza un enfriamiento de tipo geométrico usando un factor de enfriamiento denominado *cooling\_rate*, el cual propicia un enfriamiento gradual que permite pasar de una etapa de exploración amplia, a una fase de explotación donde se intenta refinar soluciones cercanas al óptimo local en el que se encuentre el algoritmo. A continuación, la figura 23 permite evidenciar el comportamiento de la temperatura en el sistema, y su relación con el historial de aceptación para el ejemplo que se ha venido trabajando.

**Figura 23**

*Evolución de temperatura + historial de aceptación*



Nota: Tomado de MATLAB R2024b

Para esta corrida se asume una temperatura inicial del 1000, con una tasa de enfriamiento de 0.99 y en donde el número total de iteraciones se fija en 1000. Estos parámetros se definen en

estos valores ya que permiten una exploración lo suficientemente grande del espacio de soluciones, y al mismo tiempo permiten visualizar gráficamente el comportamiento de los elementos de interés. Esta gráfica facilita entender el comportamiento de la temperatura, y cómo esta afecta la probabilidad de que una solución vecina se acepte o no. La curva azul representa la temperatura, y se evidencia su enfriamiento gradual conforme avanza el número de iteraciones, hasta estabilizarse en valores menores a 50 a partir de la iteración 400 aproximadamente. Al inicio, el SA está en una fase de exploración agresiva, donde se puede ver que la decisión de aceptar soluciones peores es completamente dominante. La decisión de aceptación de soluciones está definida por los puntos rojos en la gráfica, los cuales representan si se acepta (1) o se rechaza (0) la solución evaluada. Para la decisión de aceptación, se ve como en muchas iteraciones los puntos en 1 son tantos que quedan superpuestos, al punto de formar casi una línea continua en la coordenada vertical  $y = 1$ . Esto sucede porque se tienen en cuenta todas las soluciones posibles, no solo las peores que la actual, por esto se evidencia que se acepta la solución en casi todas las iteraciones. Lo verdaderamente interesante es evaluar la coordenada vertical  $y = 0$ , que representa el rechazo de soluciones peores que la actual. Se refleja en la figura que cuando la temperatura del sistema es elevada, la probabilidad de aceptación de soluciones peores es baja, por lo que la cantidad de puntos rojos en las primeras 400 iteraciones es baja, lo que quiere decir que no hay muchas iteraciones en las que se decida rechazar la solución evaluada. De la misma forma, se puede evidenciar que conforme aumenta el número de iteraciones y la temperatura baja, el algoritmo se vuelve más estricto y rechaza muchas más soluciones.

### 7.5.2. Criterio de parada.

**Figura 24**

*Evolución del historial del valor de la F.O.*



Nota: Tomado de MATLAB R2024b

Para el SA+RO diseñado, se define un parámetro para definir la parada del algoritmo y que no itere infinitamente, asociado al número máximo de iteraciones. La figura 23 permite evidenciar como la temperatura del sistema va disminuyendo de forma gradual hasta llegar a valores cercanos cero. Asimismo, la figura 24 permite ver como el algoritmo va iterando hasta llegar a las 1000 iteraciones definidas inicialmente para este ejemplo.

La gráfica facilita observar el comportamiento y la evolución de las soluciones evaluadas en el sistema conforme aumenta el número de iteraciones. Tal como en la figura 23, en la figura 24 es posible evidenciar el funcionamiento correcto del algoritmo, en donde la línea roja representa la solución actual que se evalúa. Se demuestra que la primera mitad de la exploración tiene características más agresivas, y las soluciones que se aceptan varían mucho una de la otra. Contrariamente, alrededor de la iteración 400 se evidencia un comportamiento mucho más conservador del algoritmo, validando así lo expuesto anteriormente y verificando a su vez que el SA+RO formulado efectivamente se comporta como un algoritmo de recocido simulado.

### 7.6. Etapa 6: Finalización y selección de la mejor solución

En esta fase, el SA+RO asigna la mejor solución encontrada a lo largo de todas las iteraciones y esta se guarda como la solución final definitiva. Para esto se usan en MATLAB las variables *best\_cost* y *best\_solution*.

#### Figura 25

*Mejor solución = solución actual*

```
if current_cost < best_cost :  
  
    best_solution ← current_solution  
  
    best_cost ← current_cost  
  
Enfriamiento de T° :  
  
Temperature ← Temperature * cooling_rate
```

La figura 25 permite ver como el SA+RO va almacenando la mejor solución encontrada hasta el momento en el sistema, este mejor costo se registra dentro de *best\_cost*, el cual es representado en la figura 24 por la línea azul. Se evidencia como esta línea azul va disminuyendo cada vez que se encuentra una solución mejor a la actual, y cada vez que esto sucede, la variable *best\_cost* se actualiza y su comportamiento durante la corrida del algoritmo evidencia como los operadores construidos son lo suficientemente robustos para encontrar soluciones mejores dentro del espacio explorado.

## 8. Experimentación y resultados

Una vez diseñados tanto el método de solución exacta en GAMS como la metaheurística de MATLAB, se definen las instancias que conforman la experimentación de la presente investigación.

### 8.1. Instancias propuestas

Las instancias que se proponen para la evaluación del modelo matemático (2eIRPTW) se dividen en dos grupos. Las primeras 3 instancias definidas se diseñan para un modelo de tamaño moderado, debido a la capacidad que ofrece GAMS en su licencia comunitaria (hasta 5000 variables y 5000 restricciones). Estas instancias son evaluadas tanto en GAMS como en la metaheurística diseñada en MATLAB, con el objetivo de comparar ambas herramientas en términos de la calidad de la solución obtenida y el tiempo computacional requerido.

Además de esto, se evalúan 3 instancias adicionales para modelos notablemente más grandes, simulando así redes de distribución más realistas en comparación a las condiciones observadas en casos reales. Estas instancias son evaluadas únicamente en MATLAB, dada su flexibilidad a la hora de evaluar sistemas de mayor envergadura. Como se menciona en la redacción de los objetivos específicos, los datos empleados en todas las instancias evaluadas no derivan de investigaciones previas, sino que corresponden a datos sintéticos ajustados de manera intencional para garantizar la adaptabilidad del modelo, forzando la utilización integral de todas las variables consideradas (mantenimiento de inventarios, penalización por estado de frescura, ventanas de tiempo, entre otras), con el objetivo de evaluar el correcto funcionamiento y comportamiento del modelo en condiciones específicas. La solución exacta de las primeras tres instancias se determina haciendo uso de GAMS 48.5.0 (ver Apéndice A), el algoritmo propuesto se evalúa en MATLAB R2024b (ver Apéndice B) y el análisis estadístico se logra a través de

STATGRAPHICS 19.7.01 (ver Apéndice C). Todo esto haciendo uso de un computador portátil con un procesador Intel Core i5 de 2 GHz y 8 GB de RAM.

**Tabla 11**

*Parámetros de estructura – Instancias pequeñas*

Parámetro	Nomenclatura	Valor
Periodos	T	2
Planta	P	1
Distribuidores	D	$2 + \pi$ , donde $\pi \in \{0,2\}$
Clientes	C	$2 + \mu$ , donde $\mu \in \{0,1\}$
Flota 1e	K	2
Flota 2e	M	$4 + \beta$ , donde $\beta \in \{0,1\}$

**Tabla 12**

*Parámetros de estructura – Instancias grandes*

Parámetro	Nomenclatura	Valor
Periodos	T	4
Planta	P	1
Distribuidores	D	$4 + \pi$ , donde $\pi \in \{1\}$
Clientes	C	$8 + \mu$ , donde $\mu \in \{2,4\}$
Flota 1e	K	D
Flota 2e	M	C

**Tabla 13**

*Parámetros de demanda – Todas las instancias*

Parámetro	Nomenclatura	Valor (unidades)
Demanda clientes	$dem_u^t$ , con $u \in C$ y $t \in T$	Aleatoria y entera en el intervalo [60,100]

**Tabla 14**

*Parámetros - capacidad de flotas de vehículos*

Instancias	Nomenclatura	Valor (unidades / vehículo*periodo)
1 – 2 – 3	$Q_K$	$Q_K \in \{500,650,600\}$
	$Q_M$	$Q_M \in \{150,170,150\}$
4 – 5 – 6	$Q_K$	$Q_K \in \{750,1100,1200\}$
	$Q_M$	$Q_M \in \{250,350,500\}$

**Tabla 15**

*Parámetros – capacidad de almacenamiento en nodos*

Instancias	Nomenclatura	Valor (unidades / nodo*periodo)
1 – 2 – 3	$\Omega_i$ , con $i \in D$	$\Omega_i \in \{450,550,300\}$
	$\Omega_u$ , con $u \in C$	$\Omega_u \in \{120,150,170\}$
4 – 5 – 6	$\Omega_i$ , con $i \in D$	$\Omega_i \in \{300,600,750\}$
	$\Omega_u$ , con $u \in C$	$\Omega_u \in \{170,250,300\}$

**Tabla 16**

*Parámetros inventario inicial en nodos – Todas las instancias*

Parámetro	Nomenclatura	Valor (unidades / nodo)
Inventario inicial	$I_a^0, I_i^0, I_u^0$	0

Para las instancias pequeñas los tiempos de viaje son manejables, lo que permite imponer restricciones más estrictas en las ventanas de tiempo. Mientras que en las instancias grandes se toma la decisión estratégica de ampliar la flexibilidad de las TW con el fin de disminuir incertidumbres operacionales causadas por el tamaño del sistema.

**Tabla 17**

*Parámetros – capacidad de almacenamiento en nodos*

Instancias	Parámetro	Valor (unidades de tiempo)
1 – 2 – 3	Límite inferior $tw_u^l$	$tw_u^l \in \{0,1\}$
	Límite inferior $tw_u^s$	$tw_u^s \in \{3,4,5\}$
4 – 5 – 6	Límite inferior $tw_u^l$	0
	Límite inferior $tw_u^s$	20

La tabla 18 permite evidenciar el rango en el que oscilan los diferentes valores asociados al costo de transporte para cada subgrupo de instancias estudiado. A pesar de que no se define explícitamente la localización de los nodos de forma geográfica, se mantiene un orden lógico en donde se asume que entre mayor es el tiempo entre viajes para dos nodos, mayor será el costo del conjunto de aristas dirigidas que los conecta.

**Tabla 18**

*Parámetros – costos de envío*

Instancias	Nomenclatura	Valor (pesos / periodo)
1 – 2 – 3	$C_{a,i}$ , Planta → CD	En el intervalo [500,750]
	$C_{i,u}$ , CD → Cliente	En el intervalo [250,370]
4 – 5 – 6	$C_{a,i}$ , Planta → CD	En el intervalo [690,720]
	$C_{i,u}$ , CD → Cliente	En el intervalo [250,500]

**Tabla 19**

*Parámetros – costos de almacenamiento*

Instancias	Nomenclatura	Valor (pesos / periodo)
1 – 2 – 3	$Ch_a$ , $a \in P$	$Ch_a \in \{1,2,3\}$
	$Ch_i$ , $i \in D$	$Ch_i \in \{0.5,1,2\}$
	$Ch_u$ , $u \in C$	$Ch_u \in \{0.5,1,2\}$
4 – 5 – 6	$Ch_a$ , $a \in P$	0.2
	$Ch_i$ , $i \in D$	En el intervalo [0.5, 2]
	$Ch_u$ , $u \in C$	En el intervalo [2, 3.5]

A continuación, la tabla 20 permite ver de una forma más detallada y organizada cada una de las instancias en las que se evalúa el presente modelo de optimización.

**Tabla 20***Instancias evaluadas*

Instancia	D	C	Flotas	Capacidad vehículo	Capacidad almacenamiento	Nomenclatura
1	2	2	K = 2	$Q_K = 500$	$\Omega_i = 450$	1 – 2 – 2
			M = 4	$Q_M = 150$	$\Omega_u = 120$	
2	2	3	K = 2	$Q_K = 650$	$\Omega_i = 550$	1 – 2 – 3
			M = 4	$Q_M = 170$	$\Omega_u = 150$	
3	4	2	K = 2	$Q_K = 600$	$\Omega_i = 300$	1 – 4 – 2
			M = 5	$Q_M = 150$	$\Omega_u = 170$	
4	4	8	K = 4	$Q_K = 750$	$\Omega_i = 300$	1 – 4 – 8
			M = 8	$Q_M = 250$	$\Omega_u = 170$	
5	4	10	K = 4	$Q_K = 1100$	$\Omega_i = 600$	1 – 4 – 10
			M = 10	$Q_M = 350$	$\Omega_u = 250$	
6	5	12	K = 5	$Q_K = 1200$	$\Omega_i = 750$	1 – 5 – 12
			M = 12	$Q_M = 500$	$\Omega_u = 300$	

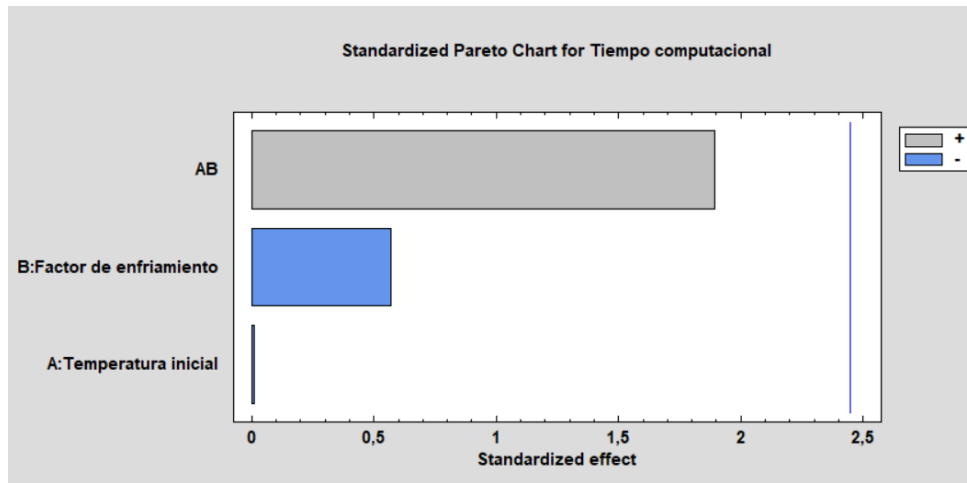
**8.2. Análisis de resultados**

Inicialmente, se plantea un diseño de experimentos en función de los parámetros relacionados a la temperatura del sistema (temperatura inicial y factor de enfriamiento) dentro del SA+RO. Este análisis procede a ser descartado debido a los resultados iniciales del diagrama de Pareto estandarizado, que evidencia que ni los efectos ni la interacción entre ellos son significativos, ya que ningún factor sobrepasa el umbral de significancia en la gráfica. A

continuación, la figura 26 muestra los efectos evaluados en la instancia 4 sobre el tiempo computacional.

**Figura 26**

*Diagrama de Pareto estandarizado*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

Con base en esto, el diseño de experimentos se realiza en función de la temperatura inicial del sistema y el número de iteraciones, con el objetivo de analizar como la amplitud de la búsqueda inicial y el tiempo total de afinación interactúan entre sí, y el efecto que tienen sobre la calidad de la solución final obtenida y el requerimiento computacional. Se presentan seguidamente los datos utilizados para los factores a evaluar, en donde cada escenario propuesto es replicado tres veces, con el objetivo de analizar variaciones en la función objetivo y en el tiempo computacional.

**Tabla 21**

*Factores diseño de experimentos*

	<b>Factor</b>	<b>Nivel bajo (-1)</b>	<b>Nivel alto (+1)</b>
<b>A</b>	Temperatura inicial	50000	100000
<b>B</b>	Número de iteraciones	100000	500000

Para evaluar el rendimiento del algoritmo se plantean dos estrategias diferenciadas en función del tamaño de las instancias propuestas. A partir de la tabla anterior se define un diseño factorial  $2^2$  acompañado de su respectivo análisis de la varianza, evaluado únicamente en las 3 instancias de mayor envergadura, que son en las que no se puede comparar el resultado exacto obtenido por PLEM con la mejor solución encontrada por la metaheurística. Por otro lado, para las 3 instancias más pequeñas se planea evaluar la brecha de optimización (pesos) y tiempo computacional (segundos), para justificar el desempeño del algoritmo en términos de precisión y eficacia.

### 8.2.1. Instancia 1 (1-2-2)

Para las primeras tres instancias se realizan tres réplicas tanto en GAMS como en MATLAB, y a partir de los resultados obtenidos se toma el menor tiempo computacional y la solución óptima (o la más cercana en su defecto) para hacer el análisis comparativo. Para las corridas en MATLAB se utiliza un valor de 1000 como temperatura inicial, se fija un máximo de 1000 iteraciones y la tasa de enfriamiento es 0,99 en todas las instancias pequeñas.

**Tabla 22**

*Resultados instancia 1*

Software	FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	R1	R2	R3	R1	R2	R3
<b>GAMS</b>	4461,50	4461,50	4461,50	0,670	0,740	0,700
<b>MATLAB</b>	4868,50	4917,75	4817,50	1,120	1,110	0,915

La solución exacta para esta instancia se fija en 4461.5 haciendo uso de PLEM en GAMS.

La comparación entre ambos métodos de solución demuestra que, para esta instancia, el algoritmo diseñado genera soluciones factibles en donde la diferencia porcentual en la calidad de la solución

es aproximadamente 7.98%. Sin embargo, hay un incremento en el tiempo computacional de 0.245 segundos, representando un 36.57% de diferencia porcentual.

### 8.2.2. Instancia 2 (1-2-3)

La siguiente tabla muestra los resultados obtenidos para la configuración 1-2-3 de la red de distribución estudiada.

**Tabla 23**

*Resultados instancia 2*

Software	FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	R1	R2	R3	R1	R2	R3
<b>GAMS</b>	3900,40	3900,40	3900,40	0,920	0,833	0,816
<b>MATLAB</b>	5125,90	4091,05	4107,45	1,220	1,470	1,320

Los resultados indican que GAMS obtiene consistentemente el valor óptimo de la función objetivo, mientras que el algoritmo propuesto presenta un mejor desempeño para esta segunda instancia evaluada, en donde el aumento en la función objetivo es de un 4.89%, pero incrementando en un 49.51% el tiempo computacional requerido.

### 8.2.3. Instancia 3 (1-4-2)

Seguidamente, se presentan los resultados obtenidos al ejecutar el algoritmo diseñado y la formulación matemática exacta en GAMS. Se encuentran reducciones importantes en el *optimality gap* existente entre los dos métodos de solución evaluados.

**Tabla 24***Resultados instancia 3*

Software	FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	R1	R2	R3	R1	R2	R3
<b>GAMS</b>	3924,25	3924,25	3924,25	1,230	1,240	1,330
<b>MATLAB</b>	5090,65	5081,65	5124,65	1,690	1,360	1,550

En esta instancia los resultados obtenidos son más prometedores en cuanto al tiempo computacional, demostrando mejoras valiosas en comparación con instancias con menos nodos, aunque sacrificando calidad en las soluciones encontradas. Para esta configuración, el aumento en la función objetivo fue de 29.49%, pero el incremento en el tiempo total de computación se redujo a un 10.57%.

#### **8.2.4. Instancia 4 (1-4-8)**

A partir de la instancia 4 empieza el análisis estadístico de los resultados, en donde para cada instancia se presenta una tabla ANOVA, acompañada de las gráficas de interacción y de efectos principales para las variables evaluadas. Para las instancias 4-6 se utiliza un valor de tasa de enfriamiento fijado en 0,9999.

**Tabla 25**

*Resultados instancia 4*

Escenario	FACTOR		FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	A	B	R1	R2	R3	R1	R2	R3
1	-1	-1	19231,10	19258,05	19325,10	190,4071	212,2670	188,8059
2	+1	-1	19268,45	19344,70	19266,25	213,3104	214,1847	210,5295
3	-1	+1	19210,15	19226,71	19258,95	921,7464	975,5788	1027,3919
4	+1	+1	19204,50	19252,75	19171,32	915,6610	963,9541	1365,9233

Con base en los resultados obtenidos se determina que el número de iteraciones es el único factor estadísticamente significativo sobre la solución final encontrada. A su vez, este mismo factor es abrumadoramente dominante sobre el tiempo computacional requerido para correr el algoritmo, siendo prácticamente el único factor relevante para esta variable de respuesta.

**Figura 27**

*ANOVA para TC – Instancia 4*

Análisis de Varianza para Tiempo computacional

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Temperatura inicial	11246,5	1	11246,5	0,70	0,4264
B:Número de iteraciones	2,03425E6	1	2,03425E6	126,99	0,0000
AB	6268,98	1	6268,98	0,39	0,5490
Error total	128148,	8	16018,5		
Total (corr.)	2,17992E6	11			

R-cuadrada = 94,1214 por ciento

R-cuadrada (ajustada por g.l.) = 91,917 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 28**

*ANOVA para FO – Instancia 4*

Análisis de Varianza para Función objetivo

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Temperatura inicial	0,364008	1	0,364008	0,00	0,9885
B:Número de iteraciones	11363,4	1	11363,4	6,85	0,0307
AB	1460,59	1	1460,59	0,88	0,3754
Error total	13262,1	8	1657,76		
Total (corr.)	26086,4	11			

R-cuadrada = 49,1609 por ciento

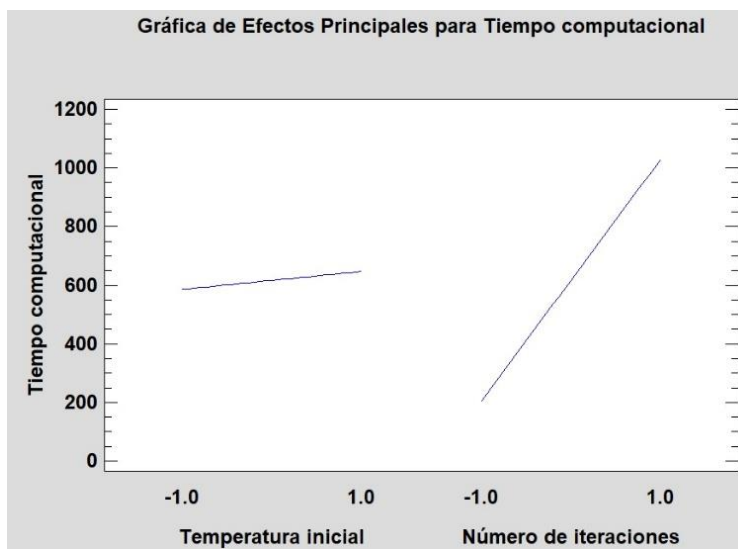
R-cuadrada (ajustada por g.l.) = 30,0963 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

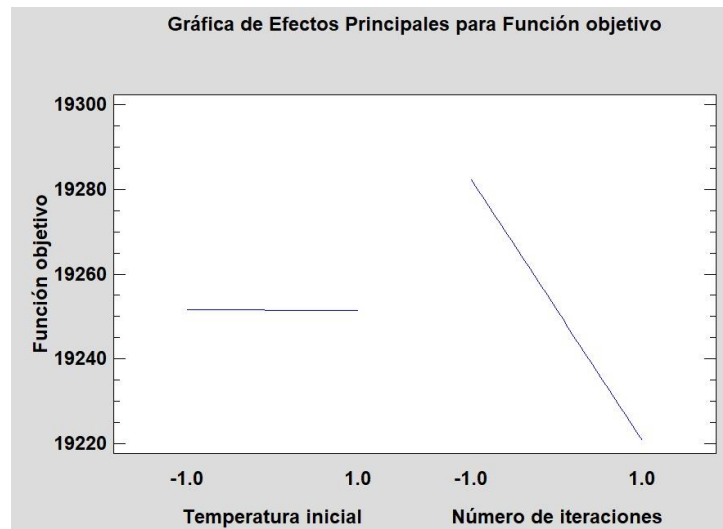
La ANOVA presentadas anteriormente, definidas para un nivel de confianza del 95%, sugieren que el factor B es determinante y su impacto es altamente significativo en ambas variables de respuesta. Además de esto, el R-cuadrado explica el 91.92% de la variabilidad computacional, lo que indica que los factores analizados en los niveles definidos capturan la gran mayoría de las variaciones observadas.

**Figura 29**

*Efectos principales TC – Instancia 4*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 30***Efectos principales FO – Instancia 4*

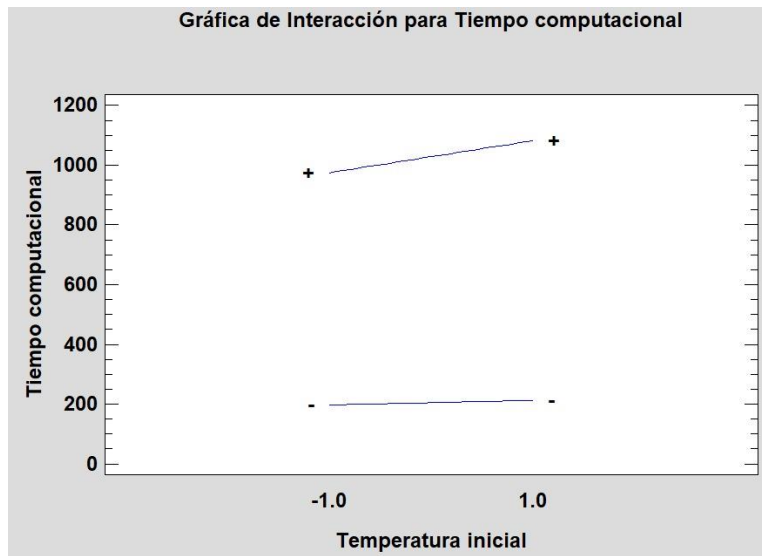
Nota: Tomado de Statgraphics Centurion Versión 19.7.01

Para ambas variables de respuesta, la línea para temperatura inicial es casi horizontal, esto quiere decir que cambiar este factor en los niveles evaluados no tiene un efecto práctico ni estadísticamente significativo. Por otra parte, ambas líneas asociadas al número de iteraciones cuentan con pendientes bastante pronunciadas, por lo que ambas variables están fuertemente dominadas por el número de iteraciones. Para el TC, aumentar el número de iteraciones incrementa el requerimiento computacional de forma predecible. Mientras que para FO, a mayor número de iteraciones, el algoritmo es capaz de encontrar soluciones de mayor calidad.

Acto seguido se presentan las gráficas de interacción que muestran como los dos factores combinados afectan (o no) a la variable de respuesta en cuestión. La figura 31 presenta un par de líneas que son casi paralelas, lo que indica que la interacción entre los factores A y B no es significativa en el tiempo computacional. Ahora bien, la figura 32 da a entender que la combinación de los factores tiene algún efecto sobre la calidad de la solución, pero el ANOVA indica que no es lo suficientemente fuerte como para ser significativo.

**Figura 31**

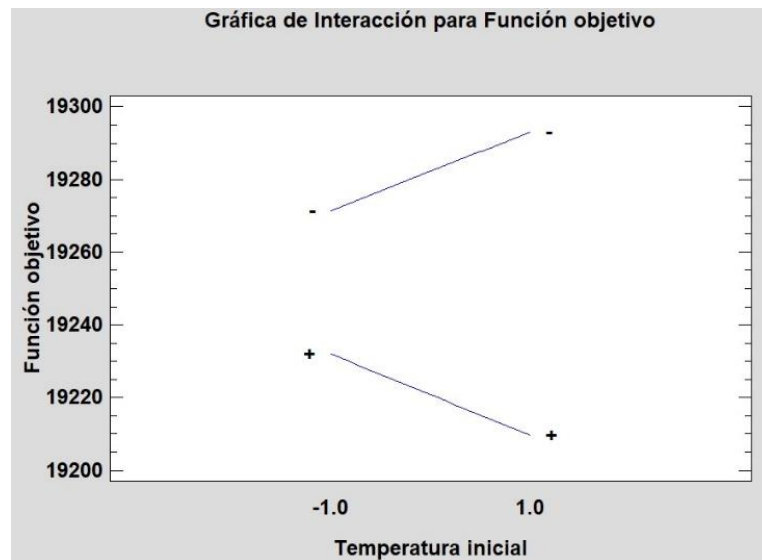
*Interacciones TC – Instancia 4*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 32**

*Interacciones FO – Instancia 4*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**8.2.5. Instancia 5 (1-4-10)**

A continuación se muestran los resultados obtenidos para la instancia 5.

**Tabla 26**

*Resultados instancia 5*

Escenario	FACTOR		FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	A	B	R1	R2	R3	R1	R2	R3
1	-1	-1	19901,20	19933,35	19937,00	244,985	324,0348	247,0427
2	+1	-1	19969,50	19995,55	19957,05	218,5348	222,4991	273,6625
3	-1	+1	19884,20	19906,65	19903,05	1066,421	1069,906	1072,109
4	+1	+1	19866,85	19912,70	19857,35	1077,137	1050,783	1193,059

En función de los resultados encontrados, se establece un nivel de confianza de 95% en el ANOVA para cada una de las variables de respuesta configuradas en el diseño factorial.

**Figura 33**

*ANOVA para TC – Instancia 5*

Análisis de Varianza para Tiempo computacional

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Temperatura inicial	10,4103	1	10,4103	0,00	0,9466
B:Número de iteraciones	2,08221E6	1	2,08221E6	955,92	0,0000
AB	3813,09	1	3813,09	1,75	0,2224
Error total	17425,9	8	2178,23		
Total (corr.)	2,10346E6	11			

R-cuadrada = 99,1716 por ciento

R-cuadrada (ajustada por g.l.) = 98,8609 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 34***ANOVA para FO – Instancia 5***Análisis de Varianza para Función objetivo**

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
A:Temperatura inicial	729,3	1	729,3	1,62	0,2382
B:Número de iteraciones	10971,7	1	10971,7	24,44	0,0011
AB	3589,75	1	3589,75	8,00	0,0222
Error total	3590,94	8	448,868		
Total (corr.)	18881,7	11			

R-cuadrada = 80,9819 por ciento

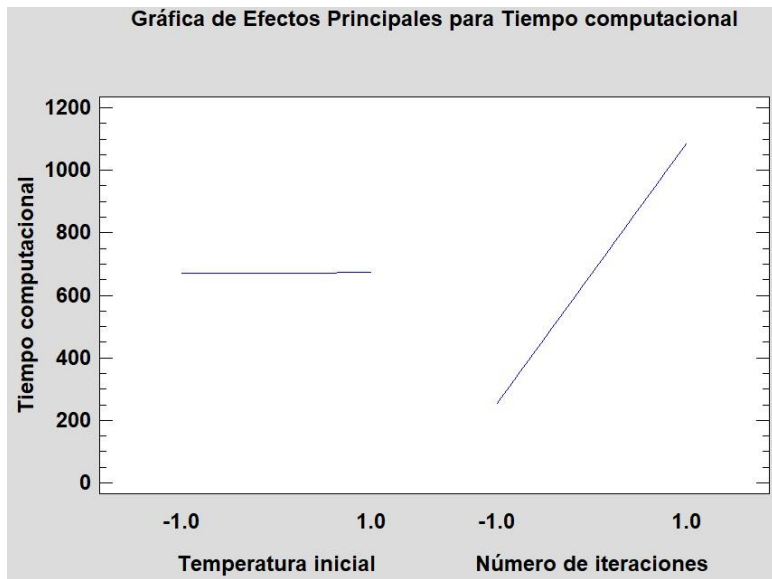
R-cuadrada (ajustada por g.l.) = 73,8501 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

Las dos figuras mostradas anteriormente indican que, en términos estadísticos, el factor B es sumamente significativo tanto para el tiempo de computación, como para el valor de función objetivo, obteniendo un valor- $p$  mucho menor a 0.05 en ambos casos. Por lo que para esta instancia, el número de iteraciones es el principal determinante de la calidad de la solución obtenida y del tiempo requerido para encontrarla, aunque la interacción entre A y B también es significativa únicamente para la calidad de la solución encontrada.

**Figura 35**

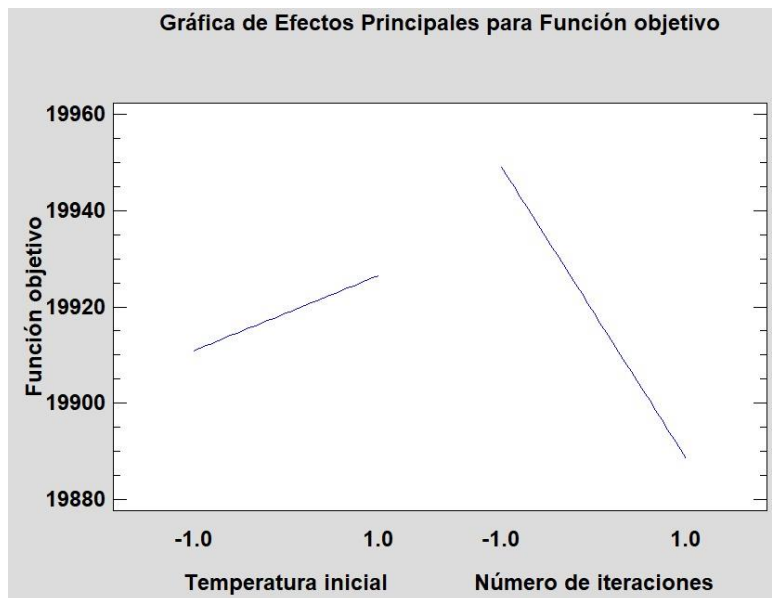
*Efectos principales TC – Instancia 5*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 36**

*Efectos principales FO – Instancia 5*

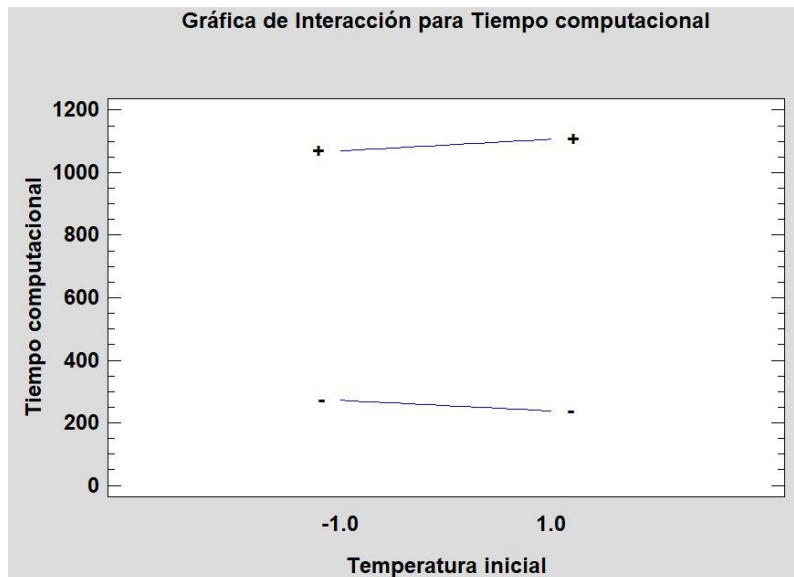


Nota: Tomado de Statgraphics Centurion Versión 19.7.01

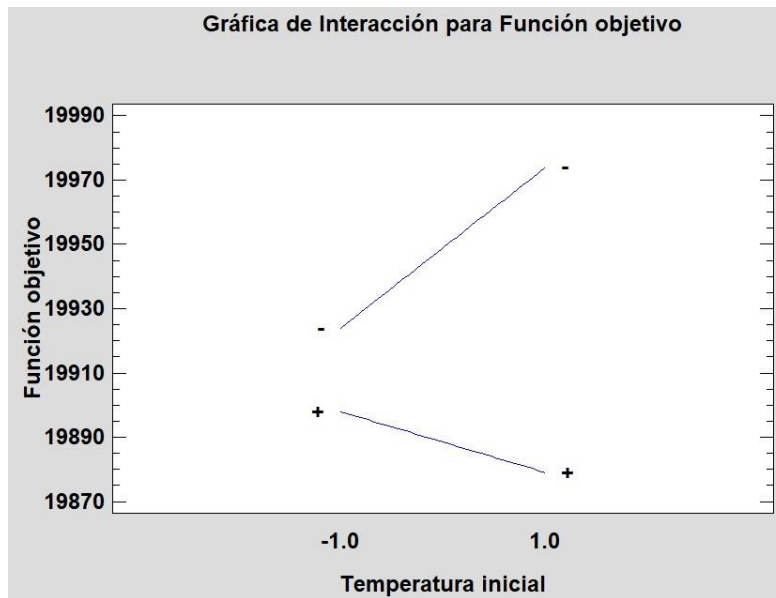
Analizando los efectos de la temperatura inicial en ambas variables de respuesta, se evidencia que para el tiempo computacional la línea es prácticamente horizontal, lo que indica que este factor es irrelevante para esta variable; mientras que respecto a la función objetivo, la pendiente positiva da a entender que un aumento en la temperatura inicial produce también un aumento en la FO, lo que es negativo en este contexto ya que el modelo busca minimizar costos. Agregando a lo anterior, el número de iteraciones presenta una pendiente positiva pronunciada para el tiempo computacional, y una pendiente negativa pronunciada para la calidad de la solución. Esto significa que un mayor número de iteraciones representa tiempos de computación más prolongados y mejoras en el valor de la función objetivo.

### Figura 37

*Interacciones TC – Instancia 5*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 38***Interacciones FO – Instancia 5*

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

La figura 37 muestra que la interacción de estos factores es irrelevante para el tiempo computacional. En cambio, para la función objetivo las líneas cuentan con pendientes más marcadas, con una evidencia visual de la interacción estadísticamente significativa existente, esto es sumamente valioso para el algoritmo construido, ya que permite entender que la optimización de los parámetros no es trivial, y es importante encontrar una combinación sinérgica entre ambos factores. En este caso, la mejor configuración constituye un nivel alto tanto para el número de iteraciones como para la temperatura inicial definida.

8.2.6. Instancia 6 (1-5-12)

Tabla 27

Resultados instancia 6

Escenario	FACTOR		FUNCIÓN OBJETIVO			TIEMPO COMPUTACIONAL		
	A	B	R1	R2	R3	R1	R2	R3
1	-1	-1	23886,05	23822,45	23890,05	329,4723	327,3221	328,8472
2	+1	-1	23957,10	23947,05	23875,25	332,4818	330,6329	323,9454
3	-1	+1	23795,65	23769,10	23697,40	2110,385	1847,541	1383,594
4	+1	+1	23803,85	23809,55	23741,30	1929,879	3269,089	1573,755

Los resultados obtenidos se recogen en la tabla 27, para los cuales se configura un diseño factorial con 95% de nivel de confianza.

Figura 39

ANOVA para TC – Instancia 6

Análisis de Varianza para Tiempo computacional

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Temperatura inicial	171034,	1	171034,	0,73	0,4171
B:Número de iteraciones	8,5709E6	1	8,5709E6	36,69	0,0003
AB	170357,	1	170357,	0,73	0,4180
Error total	1,86905E6	8	233631,		
Total (corr.)	1,07813E7	11			

R-cuadrada = 82,664 por ciento

R-cuadrada (ajustada por g.l.) = 76,163 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 40**

*ANOVA para FO – Instancia 6*

Análisis de Varianza para Función objetivo

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Temperatura inicial	6228,96	1	6228,96	3,35	0,1048
B:Número de iteraciones	48272,8	1	48272,8	25,93	0,0009
AB	649,741	1	649,741	0,35	0,5710
Error total	14896,1	8	1862,01		
Total (corr.)	70047,6	11			

R-cuadrada = 78,7343 por ciento

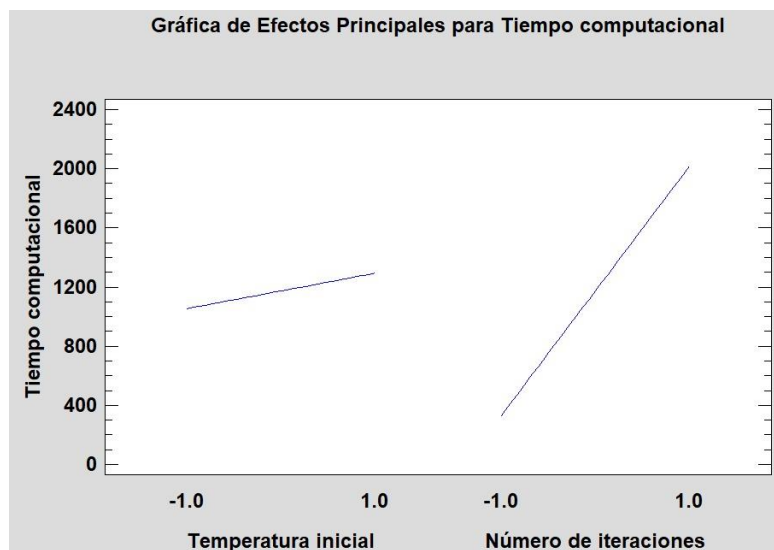
R-cuadrada (ajustada por g.l.) = 70,7597 por ciento

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

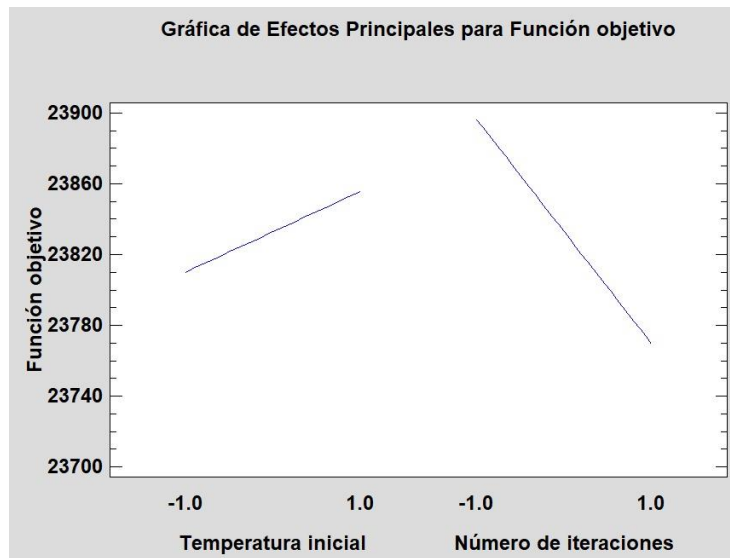
Los análisis de varianza realizados indican que el factor B es sumamente significativo para el comportamiento de ambas variables de respuesta, y para esta instancia de mayor complejidad se consolida como el factor más decisivo para la calidad de la solución confirmando la estabilidad del rendimiento del algoritmo. Se evidencia también un aumento considerable en el requerimiento computacional, en donde para algunas configuraciones, se documentaron corridas cercanas a los 55 minutos (3269 segundos).

**Figura 41**

*Efectos principales TC – Instancia 6*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

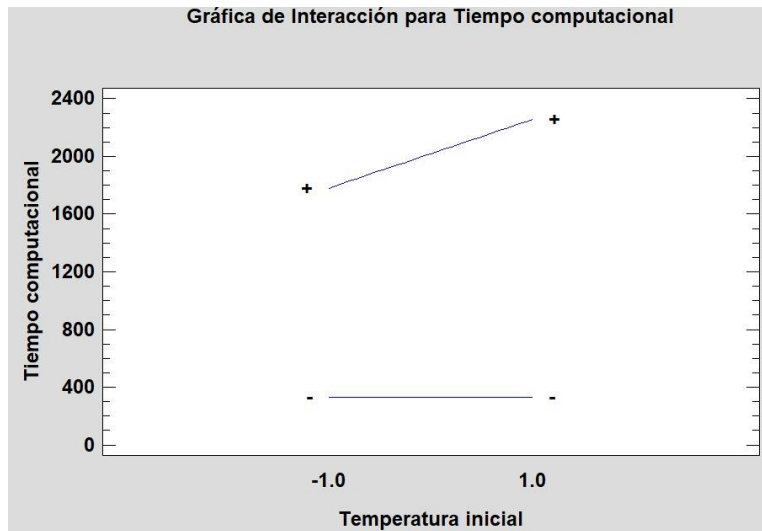
**Figura 42***Efectos principales FO – Instancia 6*

Nota: Tomado de Statgraphics Centurion Versión 19.7.01

Nótese que para el tiempo computacional, ambos factores presentan una pendiente positiva, por lo que un aumento en cualquiera de ellos representa un incremento en el tiempo computacional. Sin embargo, en este caso solo el número de iteraciones (pendiente más pronunciada) es estadísticamente significativo sobre la variable de estudio. Por otra parte, la figura 42 muestra visualmente que a mayor temperatura, el costo de solución empeora, pero el efecto no es lo suficientemente marcado para llegar a considerarse significativo. Caso contrario del factor B, el cual presenta una evidencia visual mucho más contundente.

**Figura 43**

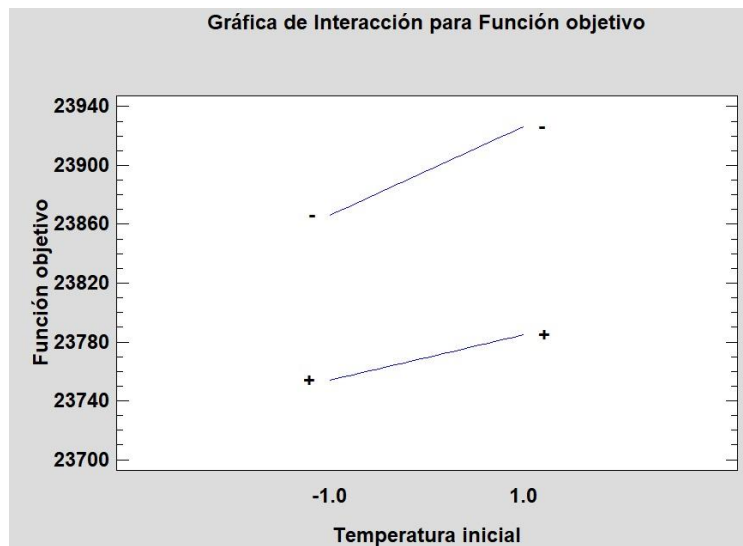
*Interacciones TC – Instancia 6*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

**Figura 44**

*Interacciones FO – Instancia 6*



Nota: Tomado de Statgraphics Centurion Versión 19.7.01

A partir de las gráficas de interacciones presentadas se entiende que hay una ausencia total de interacción entre los dos parámetros evaluados para ambas variables de respuesta. Esto debido

a que tanto en la figura 43 como en la 44, las líneas que representan la interacción son esencialmente paralelas, y este paralelismo está respaldado por el valor-p obtenido en el ANOVA.

Antes de pasar al análisis final de los resultados, es crucial abordar la validación de los supuestos estadísticos subyacentes. El supuesto de independencia se satisface ya que cada réplica obtenida al ejecutar el algoritmo es completamente independiente y no tiene influencia sobre las ejecuciones posteriores. Ahora bien, tanto para los supuestos de normalidad como para los de homocedasticidad, a pesar de que no se realizan pruebas formales que validen su comportamiento, la literatura estadística estudiada ratifica a él ANOVA como un procedimiento lo suficientemente robusto en cuanto a este tipo de desviaciones moderadas que se pueden obtener en los resultados (Montgomery, 2012). Por esto, dado que el principal objetivo de este análisis es definir la significancia y la magnitud de los efectos de los parámetros analizados sobre las 2 variables de respuesta, se considera que solidez y estructura del método ANOVA permite obtener conclusiones válidas para los propósitos de este proyecto de investigación.

De acuerdo con los resultados obtenidos en las 6 instancias evaluadas, se logra validar el comportamiento del algoritmo propuesto, encontrando también una serie de características particulares relacionadas con los factores estudiados. Se encuentra que, dentro del subgrupo de las instancias pequeñas, el algoritmo obtiene soluciones más cercanas al óptimo global del problema cuando el tamaño del sistema es menor, conforme aumenta el número total de nodos en el sistema es más fácil que el algoritmo no encuentre una estrategia similar a la óptima encontrada en GAMS. Sumado a esto, los resultados asociados al tiempo computacional requerido para generar una respuesta tanto para el método exacto como para la metaheurística indican que este se relaciona de forma directa con la complejidad y el tamaño del sistema que se evalúa. Así mismo, a pesar de que

el algoritmo no es capaz de reducir el requerimiento computacional en las instancias pequeñas, si garantiza soluciones de alta calidad, lo que valida la robustez de los operadores construidos.

El análisis estadístico propuesto permite identificar la relevancia que tienen los factores analizados respecto a las dos variables de respuesta analizadas en la configuración del experimento. Para el factor A, se entiende que en términos generales el impacto sobre el tiempo computacional es mínimo, en donde la temperatura inicial no demuestra tener un efecto principal significativo por sí sola en ningún caso. Mientras que el factor B resultó siendo el principal determinante tanto en el tiempo computacional como en la calidad de las soluciones obtenidas. Finalmente, la interacción entre A y B no es fuerte en la mayoría de las instancias, aunque llega a ser estadísticamente significativa en la instancia 5. Con base en esto, priorizando el tiempo computacional, es vital controlar el número de iteraciones y reducirlo si se busca disminuir el tiempo total de ejecución, mientras que la temperatura inicial es indiferente en los niveles evaluados. Por otra parte, priorizando la calidad de la solución, es recomendable permitir niveles altos de iteraciones y de temperatura inicial que permitan una exploración del espacio de búsqueda más exhaustiva.

## 9. Conclusiones

El análisis bibliométrico realizado permite evidenciar que la investigación académica de modelos de optimización matemática enfocados a problemas logísticos está en alza. Especialmente aquellos que involucran la rama de la investigación de operaciones con un enfoque en temas de localización, producción, ruteo y gestión de inventarios. Se determina también que la gran mayoría de estas investigaciones se hacen en países desarrollados, posiblemente por la disponibilidad de recursos, y el interés de grandes empresas multinacionales en reducir costos operacionales.

Se resalta también la relevancia que tiene el proceso del diseño y formulación de los métodos de solución evaluados, destacando la complejidad de estos, en donde GAMS permite encontrar la mejor solución posible para el escenario propuesto gracias a la programación lineal entera mixta, que permite un alto grado de detalle a la hora de modelar todas las características restrictivas del modelo. Mientras que la construcción del algoritmo SA+RO en MATLAB es un método aproximado, en donde se trabaja con una representación simplificada del problema, sacrificando precisión en la exploración del espacio de soluciones a cambio de eficiencia computacional, especialmente para las instancias de mayor complejidad. Por estas razones, la experiencia del investigador con los softwares a utilizar es determinante en la calidad de las soluciones obtenidas, y en la estructuración correcta de todas las características distintivas del modelo. Además, la comparación entre los métodos de solución utilizados revela un comportamiento común dentro del área de la investigación de operaciones. En donde el *gap* de optimización y el mayor tiempo computacional observados en las instancias pequeñas no deben ser interpretados necesariamente como una debilidad en el algoritmo propuesto, ya que el valor de la metaheurística reside en su capacidad de escalabilidad a la hora de abordar instancias de gran envergadura, obteniendo soluciones de muy buena calidad en tiempos de cómputo razonables.

El presente trabajo investigativo muestra que, para la red de distribución planteada, la amplitud de las ventanas de tiempo establecidas juega un papel importante en el costo total encontrado por los métodos de solución utilizados. Esto se prueba en las instancias evaluadas, en donde se evidencia que las ventanas de tiempo afectan de forma notoria a la flexibilidad del ruteo de vehículos, en algunos casos llegando impedir la capacidad de generar soluciones factibles. Del mismo modo, el modelo intenta en la medida de lo posible reducir al máximo o no almacenar unidades en los distribuidores, manteniendo así la penalización por el estado de frescura del producto en valores bajos en las instancias propuestas. Se concluye también que para todas las instancias, tanto en GAMS como en MATLAB, la estrategia final encontrada siempre incluye un ruteo de vehículos multi-parada en alguno de los periodos evaluados, garantizando así una reducción en los costos asociados al transporte.

El algoritmo de recocido simulado mejorado con un reparador de operación prueba ser robusto y completo a la hora de encontrar estrategias de ruteo y manejo de inventarios. Este algoritmo se diseñó específicamente para este problema (2eIRPTW), y prueba ser eficiente a la hora de encontrar soluciones factibles para las instancias evaluadas. Además de esto, la utilización del *repair operator* es de suma importancia dentro de su estructura, ya que sin este, el SA tradicional no lograba encontrar soluciones factibles en muchas corridas, para la mayoría de las instancias evaluadas. De la misma forma, la inclusión del mecanismo de diversificación propuesto para las instancias grandes prueba ser una estrategia que añade valor al algoritmo, cuya función principal es la de mitigar el riesgo de convergencia en óptimos locales. La evidencia empírica de las corridas realizadas durante la experimentación refleja que el algoritmo logra en muchas ocasiones encontrar mejores soluciones después de activarse el mecanismo, lo que sugiere que este logra reorientar la búsqueda hacia regiones prometedoras dentro del espacio de soluciones. Sin

embargo, no se puede garantizar que la solución final no sea en sí misma un óptimo local de alta calidad en lugar del óptimo global, especialmente con los resultados obtenidos en las instancias de menor envergadura. De acuerdo con el análisis de resultados del capítulo 8, se determina que el SA+RO, a pesar de no llegar a disminuir el tiempo de corrida en comparación a GAMS, es capaz de encontrar soluciones muy buenas, en donde el incremento en la función objetivo oscila entre un 4.89% a 29.49%, en comparación con el óptimo global.

Para las instancias más grandes, el número de iteraciones es significativo en la mayoría de las configuraciones planteadas, tanto para el tiempo computacional como para el valor de función objetivo obtenido, en donde de forma general, se puede concluir que a mayor número de iteraciones, se obtienen mejores soluciones requiriendo más recurso computacional. El comportamiento de la temperatura inicial también fue analizado, en donde en la mayoría de las instancias, no afecta el tiempo computacional requerido, y su impacto en la calidad de la solución depende de la instancia evaluada, y se recomienda un nivel inicial alto que fomente una exploración más completa del espacio de soluciones.

La interacción entre los factores tiene efectos significativos sobre las variables analizadas exclusivamente en la instancia 5, lo que revela un comportamiento sofisticado del algoritmo y a su vez sugiere que una calibración de la sinergia entre los parámetros evaluados es beneficiosa a la hora de ajustar el rendimiento del SA+RO. En base a los resultados obtenidos, la estrategia recomendada global es fijar la temperatura inicial en un nivel alto para fomentar una exploración más completa del espacio de soluciones, y teniendo al número de iteraciones como el parámetro de control principal dentro del algoritmo, ajustando el balance entre la calidad de la solución y el tiempo de ejecución deseado. Por estas razones, un balance adecuado entre la temperatura inicial y el número de iteraciones es clave para potenciar el desempeño del algoritmo propuesto.

## 10. Recomendaciones

Teniendo en cuenta los resultados obtenidos en la presente investigación, algunas propuestas de mejora para futuros proyectos incluyen:

Respecto a las ventanas de tiempo, se recomienda la utilización de no sólo ventanas de tiempo estrictas (*hard time windows*), sino también incluir flexibles (*soft time windows*) que otorguen mayor flexibilidad al modelo en la planificación de rutas, disminuyendo así la cantidad de soluciones no factibles.

En lo respectivo al modelamiento del deterioro por estado de frescura, para este trabajo se acudió a un modelamiento lineal con una tasa de deterioro constante. Por ende se recomienda explorar la opción de modelar las características percederas de forma no lineal, exponencial o estocástica, ya que posiblemente brinden un grado de realismo mayor al proceso de optimización, permitiendo tomar decisiones más eficientes, reduciendo así desperdicios y costos.

Con el objetivo de hacer el IRP más realista, se recomienda también la inclusión de clusterización de clientes, en donde cada distribuidor cuente con su propia flota de vehículos. Además, debido a que la cadena de suministro estudiada se desenvuelve en la industria agrícola, la inclusión de múltiples productos con diferentes tasas de deterioro podría agregar valor al modelo. También sería interesante la inclusión de tiempos de servicio o de descarga a través de toda la red de distribución, para un enfoque más preciso a la hora de atender a los clientes dentro de ventanas de tiempo establecidas. Por último, respecto al modelo se recomienda incluir criterios y costos asociados al cuidado del medio ambiente.

En la construcción y desarrollo del algoritmo, se propone explorar otros métodos heurísticos más avanzados para la generación de la solución inicial, que brinden más robustez al recocido simulado y ayuden a guiar más la exploración del espacio de soluciones, así mismo, la

generación de las soluciones vecinas también se puede mejorar con la utilización de otros operadores que permitan perturbaciones más grandes o métodos de enfriamiento más sofisticados. La hibridación del algoritmo con algún otro método también puede mejorar significativamente el desempeño del mismo, por ejemplo combinándolo con una búsqueda tabú que brinde ese componente de memoria estructurada durante la exploración de soluciones, o con un método como la optimización por enjambre de partículas que permita encontrar soluciones iniciales de mejor calidad. Finalmente, afrontar este tipo de modelos a partir de la teoría robusta de optimización puede ser beneficioso, esto debido a la complejidad del problema estudiado y de todas las características particulares atribuidas a esta red de distribución.

**Referencias bibliográficas**

- Abdelhalim, A., Eltawil, A., & Fors, M. N. (2015). The multiple vehicle inventory routing problem for perishable products. *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. doi: 10.1109/IEEM.2015.7385832.
- Acosta López, A., Rivas Trujillo, E., & Salcedo Parra, O. (2019). *Investigación de operaciones*. Ecoe Ediciones. Recuperado de: <https://www-digitaliapublishing-com.bibliotecavirtual.uis.edu.co/a/101619>
- Akbay M., Kalayci C.B., Blum C. y Polat O. (2022). Variable Neighborhood Search for the Two-Echelon Electric Vehicle Routing Problem with Time Windows. *Applied Sciences*. 12(3):1014. <https://doi.org/10.3390/app12031014>
- Alkaabneh, F., Diabat, A., & Gao, H. O. (2020). Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. *Computers & Operations Research*, 113(104751). <https://doi.org/10.1016/j.cor.2019.07.009>
- Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., & Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, 283(2), 511–529. <https://doi.org/10.1016/j.ejor.2019.11.015>
- Armas Costa, R., & Trujillo Cedeño, M. (2017). *Introducción al modelado matemático con MatLab*. Universidad de la Salle. <https://www-digitaliapublishing-com.bibliotecavirtual.uis.edu.co/a/65586>
- Baghel, M., Agrawal, Shikha, & Silakari, S. (2012). Survey of Metaheuristic Algorithms for Combinatorial Optimization. *International Journal of Computer Applications*, 58(19), 21–31. <https://doi.org/10.5120/9391-3813>

- Barragán, E., y Romero, L. J. (2015). *Estudio y desarrollo de un modelo matemático para el problema de inventario y ruteo (IRP)*. (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.
- Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8(1).  
<https://doi.org/10.1214/ss/1177011077>
- Carreño Solís, A. (2017). *Cadena de suministro y logística*. Pontificia Universidad Católica de Perú. <https://www-digitaliapublishing-com.bibliotecavirtual.uis.edu.co/a/51222>
- Cataño, J. M. (2023). *Solución al Problema de Ruteo de Vehículos Multi-Escalón Considerando Ventanas de Tiempo y Drones en la Entrega de Última Milla Para la Gestión De Operaciones Humanitarias*. (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.
- Córdoba, E. E. (2020). *Solución a un modelo de optimización para la gestión de inventario y ruteo de vehículos de dos escalones con transbordos laterales (2E-IRPT) a través de una metaheurística híbrida*. (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.
- Dellaert, N., Van Woensel, T., Crainic, T.D., y Saridarq, F. (2021). A multi-commodity two-Echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach. *Computers & Operations Research*, 127(105154).  
<https://doi.org/10.1016/j.cor.2020.105154>.
- Eglese, R. (1990). Simulated annealing: A tool for operational research. *European Journal Of Operational Research*, 46(3), 271-281. [https://doi.org/10.1016/0377-2217\(90\)90001-r](https://doi.org/10.1016/0377-2217(90)90001-r)
- Garay Candia, A. E. (2014). *Logística: conocimientos, habilidades y actitudes*: ( ed.). El Cid Editor.  
<https://elibro-net.bibliotecavirtual.uis.edu.co/es/lc/uis/titulos/98158>

- Henderson, D., Jacobson, S. H., & Johnson, A. W. (2006). The Theory and Practice of Simulated Annealing. En Kluwer Academic Publishers eBooks (pp. 287-319). [https://doi.org/10.1007/0-306-48056-5\\_10](https://doi.org/10.1007/0-306-48056-5_10)
- Hu, W., Toriello, A., & Dessouky, M. (2018). Integrated inventory routing and freight consolidation for perishable goods. *European Journal of Operational Research*, 271(2), 548–560. <https://doi.org/10.1016/j.ejor.2018.05.034>
- Iassinovskaia, G., Limbourg, S., & Riane, F. (2017). The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains. *International Journal of Production Economics*, 183, 570–582. <https://doi.org/10.1016/j.ijpe.2016.06.024>
- Jafari Nozar, F., & Behnamian, J. (2021). Hyper-heuristic for integrated due-window scheduling and vehicle routing problem for perishable products considering production quality. *Engineering Optimization*, 53(11), 1902–1921. <https://doi.org/10.1080/0305215x.2020.1837792>
- Ji, Y., Du, J., Han, X., Wu, X., Huang, R., Wang, S., & Liu, Z. (2020). A mixed integer robust programming model for two-echelon inventory routing problem of perishable products. *Physica A*, 548(124481), 124481. <https://doi.org/10.1016/j.physa.2020.124481>
- Ji, Y., Du, J., Wu, X., Wu, Z., Qu, D., & Yang, D. (2021). Robust optimization approach to two-echelon agricultural cold chain logistics considering carbon emission and stochastic demand. *Environment Development and Sustainability*, 23(9), 13731–13754. <https://doi.org/10.1007/s10668-021-01236-z>

- Jia, T., Li, X., Wang, N., & Li, R. (2014). Integrated inventory routing problem with quality time windows and loading cost for deteriorating items under discrete time. *Mathematical Problems in Engineering* 1–14. <https://doi.org/10.1155/2014/537409>
- Kritikos, M. N., & Lappas, P. Z. (2021). Computational intelligence and combinatorial optimization problems in transportation science. In *Learning and Analytics in Intelligent Systems* (pp. 325–367). Springer International Publishing.
- Kumar, A., Mangla, S. K., Kumar, P., & Karamperidis, S. (2020). Challenges in perishable food supply chains for sustainability management: A developing economy perspective. *Business Strategy and the Environment*, 29(5), 1809–1831. <https://doi.org/10.1002/bse.2470>
- Kumar, M., Kumar, D., Saini, P., & Pratap, S. (2022). Inventory routing model for perishable products toward circular economy. *Computers & Industrial Engineering*, 169(108220). <https://doi.org/10.1016/j.cie.2022.108220>
- Liu, P., Hendalianpour, A., Razmi, J., & Sangari, M. S. (2021). A solution algorithm for integrated production-inventory-routing of perishable goods with transshipment and uncertain demand. *Complex & Intelligent Systems*, 7(3), 1349–1365. <https://doi.org/10.1007/s40747-020-00264-y>
- Liu, Z., Zuo, X., Zhou, M., Guan, W., & Al-Turki, Y. (2023). Electric vehicle routing problem with variable vehicle speed and soft time windows for perishable product delivery. *IEEE Transactions on Intelligent Transportation Systems: A Publication of the IEEE Intelligent Transportation Systems Council*, 24(6), 6178–6190. <https://doi.org/10.1109/tits.2023.3249403>

- Mirzaei, S., & Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87, 213–227.  
<https://doi.org/10.1016/j.cie.2015.05.010>
- Montgomery, D. C. (2012). *Design and Analysis of Experiments*. Wiley.
- Niño, A. (2017). *Estudio del problema de ruteo de vehículos de dos escalones*. (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.
- Paterson, C., Kiesmüller, G., Teunter, R., & Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2), 125–136.  
<https://doi.org/10.1016/j.ejor.2010.05.048>
- Pérez Kaligari, E., & Guerrero Rueda, W. J. (2015). Métodos de optimización para el problema de ruteo de vehículos con inventarios y ventanas de tiempo duras. *Revista Ingeniería Industrial*, 14(3), 31-49.
- Prensa, A. (2024, May 6). Costos logísticos, dolor de cabeza para los actores del comercio exterior. *Analdex - Asociación Nacional de Comercio Exterior*.  
<https://www.analdex.org/2024/05/06/costos-logisticos-dolor-de-cabeza-para-los-actores-del-comercio-exterior/>
- Rahmanifar, G., Mohammadi, M., Sherafat, A., Hajiaghaei-Keshteli, M., Fusco, G. y Colombaroni, G. (2023). Heuristic approaches to address vehicle routing problem in the Iot-based waste management system. *Expert Systems with Applications*, 220(119708), 0957-4174.  
<https://doi.org/10.1016/j.eswa.2023.119708>.
- Rohmer, S. U. K., Claassen, G. D. H., & Laporte, G. (2019). A two-echelon inventory routing problem for perishable products. *Computers & Operations Research*, 107, 156–172.  
<https://doi.org/10.1016/j.cor.2019.03.015>

- Shaabani, H. (2022). A literature review of the perishable inventory routing problem. *The Asian Journal of Shipping and Logistics*, 38(3), 143–161. <https://doi.org/10.1016/j.ajsl.2022.05.002>
- Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. A. J. (2015). Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164, 118–133. <https://doi.org/10.1016/j.ijpe.2015.03.008>
- Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. A. J. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, 89, 168–182. <https://doi.org/10.1016/j.cor.2016.02.003>
- Violi, A., Laganá, D., & Paradiso, R. (2020). The inventory routing problem under uncertainty with perishable products: an application in the agri-food supply chain. *Soft Computing*, 24(18), 13725–13740. <https://doi.org/10.1007/s00500-019-04497-z>
- Wang, X. P., Wang, M., Ruan, J. H., & Li, Y. (2018). Multi-objective optimization for delivering perishable products with mixed time windows. *Advances in Production Engineering & Management*, 13(3), 321–332. <https://doi.org/10.14743/apem2018.3.293>
- Wang, Y., Luo, S., Fan, J., & Zhen, L. (2024). The multidepot vehicle routing problem with intelligent recycling prices and transportation resource sharing. *Transportation Research Part E: Logistics and Transportation Review*, 185(103503), 103503. <https://doi.org/10.1016/j.tre.2024.103503>

- Wang, Z., Wei, X., & Pan, J. (2021). Research on IRP of perishable products based on mobile data sharing environment. *International Journal of Cognitive Informatics and Natural Intelligence*, 15(2), 139–157. <https://doi.org/10.4018/ijcini.20210401.0a10>
- World Bank. (2023). Connecting to Compete 2023: Trade Logistics in an Uncertain Global Economy - The Logistics Performance Index and its Indicators. Worldbank.org. Retrieved August 6, 2024, from [https://lpi.worldbank.org/sites/default/files/2023-04/LPI\\_2023\\_report\\_with\\_layout.pdf](https://lpi.worldbank.org/sites/default/files/2023-04/LPI_2023_report_with_layout.pdf)
- Xiao, N., & Rao, Y. L. (2016). Multi-product multi-period inventory routing optimization with time window constraints. *International Journal of Simulation Modelling*, 15(2), 352–364. [https://doi.org/10.2507/IJSIMM15\(2\)CO8](https://doi.org/10.2507/IJSIMM15(2)CO8)
- Xin, X., Wang, X., Chen, Z., & Chen, K. (2021). Coastal shuttle tanker inventory routing model with a discrete loaded quantity. *Applied Economics*, 53(53), 6120–6137. <https://doi.org/10.1080/00036846.2021.1918625>
- Yang, X. (2010). *Engineering optimization : An introduction with metaheuristic applications*. John Wiley & Sons, Incorporated.
- Yu, V. F., Jodiawan, P., Schrottenboer, A. H., & Hou, M.-L. (2023). The two-echelon vehicle routing problem with time windows, intermediate facilities, and occasional drivers. *Expert Systems with Applications*, 234(120945), 120945. <https://doi.org/10.1016/j.eswa.2023.120945>
- Zhou, H., Qin, H., Zhang, Z., & Li, J. (2022). Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery. *Soft Computing*, 26(7), 3345–3360. <https://doi.org/10.1007/s00500-021-06712-2>

- Zhou, S., Zhang, D., Ji, B., & Li, S. (2024). Two-echelon vehicle routing problem with direct deliveries and access time windows. *Expert Systems with Applications*, 244(121150).  
<https://doi.org/10.1016/j.eswa.2023.121150>
- Zulvia, F. E., Kuo, R. J., & Nugroho, D. Y. (2020). A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products. *Journal of Cleaner Production*, 242(118428), 118428.  
<https://doi.org/10.1016/j.jclepro.2019.118428>