



UNIVERSIDAD
NACIONAL
DE COLOMBIA

**Metodología basada en un algoritmo natural
metaheurístico para programar el ruteo de los
vehículos de la pastelería ubicada en la región
cafetera**

Irma Jhuliet Betancourth Arias

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Industrial
Manizales, Colombia

2020

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

Irma Jhuliet Betancourth Arias

Trabajo de investigación presentado como requisito para optar al título de
Magíster en Ingeniería - Ingeniería Industrial

Director

Ph.D. Santiago Ruiz Herrera

Línea de Investigación

Dirección y Producción de Operaciones

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Industrial
Manizales, Colombia

2020

	Pág.
Lista de figuras	v
Lista de tablas	vi
Resumen	vii
Abstract	viii
1 Introducción	9
2 Descripción del problema	10
3 Justificación	12
4 Objetivos	13
4.1 Objetivo general	13
4.2 Objetivos específicos	13
5 Marco teórico	14
5.1 Historia de la pastelería	14
5.2 Características de una pastelería	16
5.3 Algoritmos naturales metaheurísticos	16
5.4 Metodología para la programación de ruteo en productos perecederos	19
5.4.1 Programación de rutas	20
5.4.2 Problema de ruteo de vehículos (VRP)	20
5.4.3 Métodos de solución para el VRP	22
5.4.3.1 Métodos exactos	23
5.4.3.2 Heurística	24
5.4.3.3 Metaheurísticas	24
6 Metodología	33
7 Desarrollo y aplicación del modelo NSGA II para la pastelería ubicada en la región cafetera	37
7.1 Análisis de factores internos y externos de la pastelería	37
7.2 Estudio de caso	39

7.3	Validación de la metodología.....	44
7.3.1	Diseño del algoritmo en MATLAB.....	44
7.3.2	Desarrollo metodológico.....	51
7.4	Resultado del algoritmo.....	63
7.5	Discusión de los resultados validados.....	65
8	Conclusiones.....	67
	Referencias.....	69

Lista de figuras

	Pág.
Figura 1. Acercamiento al marco teórico	14
Figura 2. Estructura de los Métodos de Solución para VRP	23
Figura 3. Estructura de los Métodos Exactos de Solución para VRP	23
Figura 4. Modelo plan de distribución.....	39
Figura 5. Ejemplo de organización genética de vehículos y puntos de venta	41
Figura 6. Cromosoma de distribución.....	42
Figura 7. Función buscar frentes.....	45
Figura 8. Función Buscar individuos – 1.....	46
Figura 9. Función Buscar Individuos – 2	47
Figura 10. Función Buscar Individuos - 3	48
Figura 11. Función Cromosoma Factible.....	49
Figura 12. Función Fobj	50
Figura 13. Función FND Ordenado	51
Figura 14. Diagrama de flujo de selección de padres.....	54
Figura 15. Proceso de mutación	57
Figura 16. Cromosoma de distribución de productos para 10 puntos de venta con 4 vehículos	63
Figura 17. Cromosoma de distribución de producto para 20 puntos de venta con 4 vehículos	64
Figura 18. Cromosoma de distribución de productos para 30 puntos de venta con 4 vehículos	65

Lista de tablas

	Pág.
Tabla 1. Autores que abordaron el proceso de cálculo de las variables dependientes	31
Tabla 2. Distancias entre la fábrica y puntos de venta de la Pastelería ubicada en la región cafetera.....	40
Tabla 3. Capacidad de los vehículos (canastillas) y consumo de gasolina (km/gl) de la flota de la Pastelería ubicada en la región cafetera	40
Tabla 4. Cromosoma	42
Tabla 5. Recorrido hasta 6 puntos de venta	43
Tabla 6. Resultado de la propuesta inicial	44

Resumen

En este documento se presenta el diseño de una metodología soportada en algoritmos naturales metaheurísticos, con el fin de programar la ruta de los vehículos de la Pastelería ubicada en la región cafetera, contribuyendo en la reducción de costos de abastecimiento y distribución y en la disminución de desperdicios alimenticios generados.

Esta investigación de tipo interpretativa contiene un procedimiento que permite solucionar el problema de distribución de productos visto como un VRP (*Vehicle Routing Problem*), donde el objetivo es minimizar la distancia de recorrido de los vehículos al distribuir los productos de una pastelería ubicada en la región cafetera de Colombia, generando un impacto positivo en los costos. Este procedimiento se basa en el diseño de algoritmo genético multiobjetivo NSGA II (*Elitist Non-Dominated Sorting Genetic Algorithm II*) aplicando la herramienta sistemática MATLAB (The Math Works Inc., 2020).

El resultado es una propuesta cuya finalidad se centra en la reducción de los costos por medio del análisis de variables de entrada y salida (distancias, tiempos, etc.) permitiendo solucionar los problemas actuales.

Palabras clave

Algoritmo, metaheurística, ruteo de vehículos, costos, multiobjetivo, distribución de alimentos.

Methodology based on a natural metaheuristic algorithm to program the routing of the vehicles of the bakery located in the coffee region

Abstract

This document presents the design of a methodology supported on natural metaheuristic algorithms in order to program the route of the vehicles of the Pastry of Manizales, contributing to the reduction of supply and distribution costs and the reduction of food waste generated.

This interpretative research contains a procedure that helped to solve a product distribution problem, seen as a Vehicle Routing Problem VRP, where the objective is to minimize the distance traveled by the vehicles by distributing the ducts of a pastry shop located in the coffee region of Colombia. Its application has a positive impact on costs. The procedure is based on the design of NSGA II multi-target genetic algorithm (Elitist Non-Dominated Sorting Genetic Algorithm II) applying the systematic tool MATLAB (The Math Works Inc., 2020).

The result is a proposal whose purpose is to reduce costs by analyzing input and output variables, such as distances, times and other variables that allowed to solve the current problems.

Keywords

Algorithm, metaheuristics, vehicle routing, costs, multi-target, food distribution.

1 Introducción

El presente trabajo investigativo tiene el fin de diseñar una metodología para programar la ruta de los vehículos de la Pastelería de Manizales para contribuir en la reducción de costos de abastecimiento y distribución y en la disminución de desperdicios alimenticios generados, basada en algoritmos naturales metaheurísticos. De esta forma, el referente teórico se desarrolla a través de una temática que comprende como área central los algoritmos y los diferentes métodos y aplicaciones, así mismo, se realiza una descripción teórica de los conceptos claves de la investigación, seguido del desarrollo y diseño del algoritmo. Los objetivos específicos que ayudaron a responder el objetivo general consistieron en la construcción de un marco teórico que permita identificar las bases literarias y conceptuales para el diseño de la metodología y la aplicación en un caso real.

Posteriormente, se realiza el diseño de una metodología de programación de vehículos a partir de la metodología NSGA II (*Elitist Non-Dominated Sorting Genetic Algorithm II*) para la solución de problemas multiobjetivo. Finalmente, se procedió a aplicar la metodología en un caso real, (Pastelería ubicada en la región cafetera) para evaluar su desempeño.

Para el desarrollo de los objetivos, se llevó a cabo el diseño de un algoritmo genético multiobjetivo NSGA II por medio de la herramienta matemática MATLAB (The Math Works Inc., 2020), la cual se utilizó para resolver el problema considerado un VRP, donde el objetivo principal es minimizar los costos generados por la distancia recorrida por los vehículos al distribuir los productos de la pastelería ubicada en la región cafetera. La metodología de investigación que se empleó para el desarrollo del proyecto es de tipo interpretativa, las fuentes de recolección de información primaria se basan en el personal de la organización y las secundarias se sustentan en el estudio de los referentes bibliográficos.

2 Descripción del problema

Los alimentos son de vital importancia en la cadena productiva del país, siendo su abastecimiento una de sus principales dificultades al presentar alto riesgo de pérdida y desperdicio, o de incremento en los costos de abastecimiento y distribución debido a situaciones que pudieron haber sido previstas. En los países en desarrollo los retos del suministro de alimentos están principalmente relacionados con limitaciones económicas, técnicas, de gestión de sistemas de información, infraestructura, envasado y sistemas de comercialización (FAO, 2012).

La planeación y programación de la distribución de alimentos hace parte del problema enrutamiento de vehículos (VRP). Este problema fue presentado en la literatura por Dantzig y Ramser (Dantzig & Ramser, 1959), se trata de un agente que parte de un origen, se desplaza a diferentes lugares y regresa nuevamente al punto de partida. VRP es uno de los problemas más importantes y reconocidos en la planificación de la distribución de productos, donde se determinan las rutas para una flota de vehículos con el fin de satisfacer la demanda de los clientes dispersos geográficamente, con el objetivo de minimizar el coste total de viajar (Kumar et al., 2016).

Lo anterior permite identificar la importancia del estudio de la distribución de alimentos, siendo este un punto del eslabón de la cadena de abastecimiento que debe ser fortalecido, en busca de soluciones que permitan reducir los desperdicios de los alimentos y minimizar los costos de abastecimiento y suministros, o al menos evitar que estos se incrementen.

Para abordar este tipo de problemáticas se determinó adoptar un caso de estudio empresarial en la ciudad de Manizales, es el caso de la Pastelería, la cual cuenta con una flota propia de vehículos con sistema de refrigeración con los cuales lleva

a cabo la distribución de los productos en establecimientos en la ciudad de Manizales y Pereira, y los municipios de Chinchiná y Villamaría.

Los vehículos tienen limitaciones de capacidad, donde las decisiones en la planeación y programación de las rutas se realiza con base en la experiencia de los conductores, quienes determinan los horarios y secuencia de los establecimientos a distribuir. Aunque este método resulta flexible y proporciona capacidad de respuesta ante eventualidades, resulta ser poco acertado con la finalidad de cumplir objetivos empresariales como son: reducir costos, tiempos de transporte y entrega, distancias de recorrido, pérdidas y desperdicios de los productos, entre otros. Para lograr lo anterior se hace necesaria la implementación de herramientas matemáticas que faciliten la secuenciación adecuada de rutas de vehículos.

Con el fin de investigar la distribución de los alimentos en el caso de estudio y teniendo en cuenta el carácter propio del producto de la Pastelería ubicada en la región cafetera, se plantea la siguiente pregunta de investigación:

¿Cómo programar los vehículos de la empresa Pastelería ubicada en la región cafetera para distribuir alimentos en diferentes locales en el eje cafetero, de manera que mejore los tiempos de entrega reales y minimicen costos logísticos de distribución?

3 Justificación

Las panaderías y pastelerías desde el ámbito productivo, familiar y comercial hacen parte de la cultura colombiana, donde representan un sector de gran importancia para la economía familiar. Los alimentos, después de la vivienda, son el segundo rubro más representativo de la canasta básica familiar, con un peso del 28,21% (MADR, 2013). Además, los desperdicios de alimentos por segmento son: 28% en consumo, 28% en producción, 17% en mercado y distribución, 6% durante el procesamiento y, 22% en manejo y almacenamiento (FAO, 2014). La optimización de los procesos de distribución y comercialización de los alimentos se puede realizar por medio de herramientas computacionales que permiten el diseño de secuenciación y planeación de las rutas facilitando la toma de decisiones basado en algoritmos. En la gestión del transporte es necesario tener en cuenta las particularidades del producto a distribuir, en este punto se requiere analizar la singularidad de los alimentos y se hace necesario tener en cuenta factores como: fecha de vencimiento, tiempos de entrega y condiciones especiales de transporte.

Las empresas que se esmeran por fortalecer sus procesos de distribución acarrear mayores costos aparentes por inversión, pero estos se ven reflejados en menores costos de distribución, menores tiempos de entrega, puntualidad y satisfacción de las necesidades del cliente (Díaz Gómez et al., 2008). Conocer cómo se desarrolla el proceso de distribución de producto terminado en diferentes puntos de venta de la pastelería ubicada en la región cafetera y determinar cuál es el papel de la inteligencia artificial en el área y su aporte en dichos procesos permiten una toma de decisiones asertiva respecto al ruteo de vehículos de la pastelería ubicada en la región cafetera.

La comunidad académica y empresarial puede encontrar en este trabajo un caso de estudio que puede ser asociado para la solución de problemas de diseño de redes de distribución de otras empresas con características similares.

4 Objetivos

4.1 Objetivo general

- Diseñar una metodología soportada en algoritmos naturales metaheurísticos para programar la ruta de los vehículos de la pastelería ubicada en la región cafetera con el fin de contribuir en la reducción de desperdicios.

4.2 Objetivos específicos

- Construir un marco teórico que permita identificar las bases teóricas y conceptuales para el diseño de la metodología y la aplicación en un caso real.
- Diseñar una metodología de programación de vehículos con ayuda de metaheurísticas para aplicar en un caso real.
- Aplicar la metodología en un caso real para evaluar el desempeño de esta

5 Marco teórico

Para el desarrollo teórico y conceptual del caso de estudio de la pastelería se abordan los siguientes temas: historia de la pastelería, características de una pastelería, metodología para la programación de ruteo en productos perecederos, algoritmos naturales y programación de rutas. En la *Figura 1*, se observa el acercamiento teórico del marco referencial.



Figura 1. Acercamiento al marco teórico

Fuente: Elaboración propia

5.1 Historia de la pastelería

Desde épocas primitivas el hombre ha consumido alimentos azucarados extraídos de la savia del arce, la miel silvestre, los cereales, las frutas y las semillas, productos que en el neolítico ya se mezclaban en papillas de cereales sobre piedras calientes, permitiendo de esta manera las elaboraciones de los primeros “pasteles”. En Europa se incrementó el consumo de azúcar por la excelente combinación resultante, extendiéndose rápidamente por las cortes europeas. Aunque la producción de dulces y pasteles se realizaba principalmente en los países europeos a nivel familiar desde la época antigua, se asegura que el origen de las tiendas de pastelería y confitería actuales surgieron a partir de las farmacias.

Es de anotar que, el origen de muchos dulces y pasteles surgió de la necesidad de encontrar métodos para la conservación de alimentos. Así, por ejemplo, se observó que, al calentar la leche con azúcar, dándole vueltas y dejándola que se concentrara se obtenía un producto (la leche condensada) de agradable y dulce sabor, el cual se podía conservar sin problemas durante largos períodos, de igual manera ocurre con las mermeladas hechas a partir de frutas y azúcar sometidas a cocción.

La etimología de la palabra pastel, que genera a su vez la de pastelería, proviene de una arcaica palabra francesa *watel* que significa alimento, aunque algunos coinciden que esta proviene del latín tardío *pasta* que a su vez se deriva del griego *pasté* que era como se le llamaba en la antigüedad a la “mezcla de harina y salsa”. Las dulces tentaciones que hoy son la delicia de los paladares más exigentes son el resultado del refinamiento y conocimiento labrado durante varios siglos, la tradición como tal nace cuando un conjunto de técnicas fundamentales y propias de este oficio confluyen en esta área, que se conoce como pastelería.

En tiempos pasados las abuelas optaban por secretos en sus recetas, las cuales a pesar de los cambios para mejorar o dar nuevas tendencias de ingredientes, han sobrevivido, permitiéndonos disfrutar en la actualidad de los dulces que a todos gusta. Colombia es un país amante del dulce por tradición, lo que le ha permitido conservar la imagen de la repostería tradicional. Adicionalmente, en el país se ha trabajado en la tecnificación y profesionalización de esta, optando por la recuperación de la tradición de los conocimientos de los grandes pasteleros colombianos. La tendencia ha estado enfocada en el profesionalismo de la repostería contando productos como: tortas y postres dulces.

5.2 Características de una pastelería

Las pastelerías se pueden clasificar en tradicionales e industriales, las cuales se diferencian por su estructura en la cadena de abastecimiento. La pastelería tradicional se caracteriza por la elaboración de productos en forma artesanal con ayuda de pequeños equipos, y por poseer un solo punto de venta. Por su parte, la pastelería industrial se caracteriza por tener una producción masiva y distribuir a pequeños establecimientos. En las pastelerías industriales se hace necesario considerar su sistema de distribución, el cual debe contar con infraestructura adecuada considerando las propiedades del producto, siendo común el uso de vehículos con sistema de refrigeración (Polo Hernán & Sastre Méndez, 2014).

5.3 Algoritmos naturales metaheurísticos

Por la década de los años setenta surgieron nuevos tipos de algoritmos de aproximación que hacían uso de métodos heurísticos simples de datos de alto nivel, estas herramientas permitían la exploración eficiente y efectiva en un espacio de búsqueda, a este nuevo tipo de algoritmos se les llamó metaheurísticas. Se caracterizan por que realizan un procedimiento de búsqueda para encontrar soluciones de aceptable calidad, mediante la aplicación de operadores independientes del dominio que modifican soluciones intermedias guiadas por la idoneidad de su función objetivo.

Algunas descripciones de las metaheurísticas permiten ver algunas de sus propiedades, como son las estrategias utilizadas para guiar el proceso de búsqueda, su objetivo de explorar eficientemente el espacio de búsqueda con el fin de encontrar las soluciones óptimas o las más cercanas posibles, están constituidas por técnicas que van desde procedimientos de búsqueda locales simples hasta complejos procesos de aprendizajes estocásticos.

Las técnicas metaheurísticas son una rama de las ciencias computacionales que mezclan las matemáticas aplicadas, fórmulas probabilísticas y teorías de la complejidad computacional, las que son muy utilizadas en áreas como la inteligencia artificial, computación inteligente, cómputo suave, programación matemática e investigación de operaciones.

Estas técnicas se pueden encontrar en cualquier lugar en donde las actividades requieran ser optimizadas. En el mundo de la industria se tienen un gran número de procesos que tienen como propósito ahorrar recursos, la mayoría de éstos se pueden modelar como problemas de optimización, con capacidad de encontrar la solución a problemas altamente complejos. Con estos procesos de mejora se tienen tres propósitos principales: resolver problemas rápidamente, resolver problemas grandes, producir algoritmos estables que al mismo tiempo sean fáciles de implementar y con una gran flexibilidad.

Así se encuentran un gran número de problemas de optimización, de la misma manera se han propuesto y generado diferentes soluciones metaheurísticas. Algunas de las técnicas metaheurísticas más conocidas son:

- Recocido Simulado o *Simulated Annealing* (SA): muy empleado para resolver problemas combinatorios. Se inicia con una solución x , se selecciona a continuación una solución vecina x' dentro de cierta vecindad o región N , posteriormente se evalúa (Hernández González, 2012). La primera aplicación de este algoritmo fue hecha por Kirkpatrick y Gelatt en el año 1983 (Kirkpatrick et al., 1983), esta prueba buscaba encontrar soluciones al problema del vendedor viajero con un número relativamente grande de ciudades. Este método está inspirado en el proceso de recocido del acero y cerámicas, una técnica que consiste en calentamiento y enfriamiento lento del material para variar sus propiedades físicas.

- Algoritmo de Estimación de Distribución de Probabilidades o *Estimation of Distribution Algorithm* (EDA) son una clase de algoritmos basados en la computación evolutiva implementados por Larrañaga y Lozano (Larrañaga & Lozano, 2001). Los EDA sustituyen las funciones de cruce y mutación tradicionales de los algoritmos evolutivos por la generación de individuos obtenidos por simulación de una distribución de probabilidad. La distribución es estimada a partir del proceso iterativo de competencia de los individuos seleccionados en la generación anterior.
- Búsqueda Dispersa o *Scatter Search* (SS), donde la primera descripción del método fue publicada por Glover (Glover, 1977). Este es un método evolutivo basado en estrategias para combinar reglas de decisión, así como en la combinación de restricciones, donde la calidad determina un conjunto de soluciones que pueden mejorar mediante la combinación de estas. En resumen, cuando se obtengan dos soluciones, se puede obtener una nueva mediante su combinación de modo que esta sea superior a las que la originaron.
- Algoritmos Genéticos o *Genetic Algorithms* (GA) se le atribuyen a John Holland por su libro *Adaptation in Natural and Artificial Systems* publicado en el año 1975 (Holland, 1992). Esta metaheurística consiste en una población de soluciones candidatas al problema, cada una se evalúa para obtener un valor que mide su validez como solución al problema, en función de este valor se les otorgará una probabilidad para reproducirse, además se realizarán cruces y mutaciones entre estas soluciones (Olivera, 2004).
- Búsqueda Tabú o *Tabu Search* (TS) fue desarrollado por Fred Glover en sus publicaciones en los años 1989 y 1990 (Glover, 1989) - (Glover, 1990), la búsqueda tabú aumenta el rendimiento del método de búsqueda local utilizando estructuras de memoria. Cuando se determina una solución, esta es marcada como “tabú” para que el algoritmo no vuelva a visitar esa posible solución.

- Los Algoritmos de Hormigas: Optimización con Colonia de hormigas o *Ant Colony Optimization* (ACO), es un método orientado a la solución de problemas encontrando las mejores rutas o caminos (Dorigo & Stützle, 2004). Este método se basa en el comportamiento real de las hormigas en su búsqueda de alimento, las hormigas depositan una feromona en los caminos más utilizados, permitiendo que las demás hormigas encuentren el camino hacia el alimento. El funcionamiento de este algoritmo permite fácilmente modelar y estudiar problemas que buscan encontrar los recorridos más cortos entre diversos puntos, contiene funciones que calculan, optimizan y evalúan diferentes recorridos para intentar encontrar el de menor distancia.

5.4 Metodología para la programación de ruteo en productos perecederos

En el ruteo de productos perecederos requieren un interés particular, ya que se consideran variables específicas y condiciones de manejo especiales en las operaciones de distribución y transporte. Dado que este tipo de vehículos requieren sistemas de refrigeración generan un mayor costo y requieren más combustible que los vehículos de tipo general; por lo tanto, requiere un control y cuidado para minimizar desperdicios (Pan et al., 2014). Una cadena de ruta de productos perecederos consta de tres partes: el suministro, la fabricación y la distribución.

La parte del suministro se concentra en el tiempo de recorrido, cómo, dónde y cuándo se abastecen las materias primas para la producción. La fabricación convierte estas materias primas en productos terminados y la distribución asegura de que dichos productos finales lleguen al consumidor a través de una red de distribuidores, almacenes y comercios minoristas.

5.4.1 Programación de rutas

La distribución de productos desde centros a clientes finales actúa como un papel central en la coordinación de los sistemas logísticos y su planificación para la disminución de desperdicios. En consecuencia, se obtendrán ahorros, los cuales explican en gran medida la utilización de técnicas computacionales y de investigación operativa. La programación de rutas es la parte de la cadena de suministros que planifica, implementa y controla el flujo efectivo y eficiente; el almacenamiento de artículos y servicios, también la información relacionada desde un punto de origen hasta un punto de destino con el objetivo de satisfacer a los clientes.

5.4.2 Problema de ruteo de vehículos (VRP)

Los problemas de ruteo pertenecen a problemas de combinatoria que requieren herramientas como el uso de inteligencia artificial para optimizarlos. Los algoritmos genéticos pertenecen a esta rama y aunque no suelen ofrecer la mejor solución, sí permiten acercarse a una solución óptima. La funcionalidad de los algoritmos genéticos en los problemas que requieren combinatorias se encuentra en su capacidad de adaptarse al inconveniente utilizando operadores de cruce y mutación (Rocha Medina et al., 2011).

El problema de ruteo de vehículos (*Vehicle Routing Problem – VRP*) consiste en un conjunto de clientes con demandas específicas, un depósito y una flota de vehículos con una capacidad determinada, cuyo objetivo es encontrar la ruta que minimice costos, determine el inicio y fin en el depósito (Secomandi, 2001). Así mismo, la demanda de los clientes solo se conoce al momento de atenderlos, razón por la que se puede presentar un fallo en la ruta, lo cual significa que el vehículo no tiene los productos necesarios para satisfacer la demanda del nodo actual; para esto se presentan distintas estrategias que evitan la modificación de la secuencia planteada.

En los años cincuenta del siglo pasado se creó la formulación de la programación matemática y el primer enfoque algorítmico para resolver el problema de la entrega de gasolina para estaciones de servicio (Dantzig & Ramser, 1959). Algunos años después, se propuso una técnica heurística para mejorar el problema de ruteo de vehículos denominado VRP, el cual es un problema de optimización para satisfacer demandas determinísticas a un costo mínimo (Clarke & Wright, 1964).

De esta manera se estableció que el VRP consiste en determinar un conjunto de rutas para una flota de vehículos que parten de uno o más depósitos o almacenes, la cual busca satisfacer la demanda de varios clientes dispersos geográficamente en una región. Es claro que, en el caso práctico, una flota de vehículos posee restricciones tales como la capacidad de carga, el tamaño del vehículo, la velocidad de traslado entre puntos, el costo de operación de la flota, etc.

El objetivo principal es entregar la demanda a todos los clientes minimizando el costo total involucrado que generan las rutas, disponiendo de una flota de vehículos con una cierta capacidad de transporte. Cada ruta es realizada por un solo vehículo que inicia y termina en el depósito, de tal forma que se satisfacen los requerimientos de los clientes y las restricciones operacionales. El tiempo durante el cual los productos están en tránsito se refleja en el número de envíos que pueden hacerse con un vehículo en un período dado, así como en los costos totales de transporte para todos los envíos. Un problema frecuente en la toma de decisiones es reducir los costos de transporte y mejorar el servicio al cliente encontrando los mejores caminos que debería seguir un vehículo en una red de carreteras, líneas ferroviarias, líneas de embarque o rutas de navegación aérea que minimicen el tiempo o la distancia.

Aunque hay muchas variaciones dentro de los problemas de diseño de rutas, se logran reducir a unos cuantos tipos básicos. Para Xiao y Jiang-qing (Xiao & Jiang-qing, 2012), un VRP tiene como propósito, minimizar el costo de viaje de los

vehículos, al mismo tiempo que se tienen en cuenta las limitantes de capacidad de estos, este autor tiene presente que VRP es un problema del tipo NP-complejo, el cual puede plantearse así:

Dado uno o más depósitos, una flota de vehículos homogéneos o no y con un conjunto de clientes cuyas demandas son conocidas o previstas; se busca un conjunto de rutas cerradas que originan y generalmente, terminan en uno de los depósitos, cumpliendo con todos los clientes al menor costo, también con las condiciones propias del vehículo y el depósito (Xiao & Jiang-qing, 2012). Según el mismo autor, VRP puede contemplar igualmente características del servicio, tiempo de los viajes, combinaciones de productos en los depósitos, variables relativas a los tiempos de viaje y pendientes.

En lo que respecta a los métodos de solución, a partir de la revisión bibliográfica se concluyó que se han abordado tres grandes categorías, las cuales pueden ser agrupadas de la siguiente manera: métodos exactos, heurísticas y metaheurísticas. (Rocha Medina et al., 2011). Ballou (Ballou, 2004) expone que los costos logísticos por realizar actividades de trasportación corresponden a una tercera parte de los costos totales de la logística. Se reconoce que la prestación del servicio de transporte incluye factores de calidad, compromiso, confianza e inversión de capital.

5.4.3 Métodos de solución para el VRP

Los métodos de solución para los problemas de ruteo de vehículos se abordan desde 3 categorías principales basándose en los trabajos realizados por Laporte (Laporte, 1992) y Rocha, González, y Orjuela (Rocha Medina et al., 2011), agrupados de la siguiente manera: métodos exactos, heurísticas y metaheurísticas. En la *Figura 2* se muestra la estructura mencionada y a partir de ella se desprenden las demás clasificaciones que se explican a continuación.

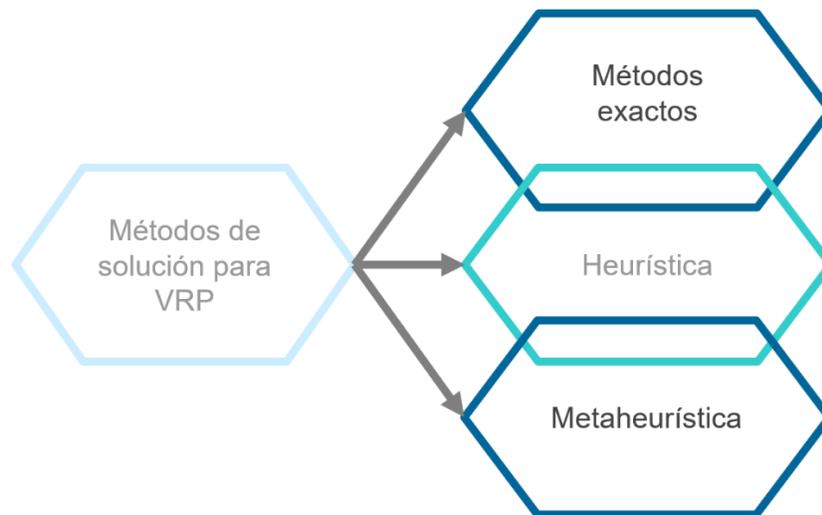


Figura 2. Estructura de los Métodos de Solución para VRP

Fuente: Elaboración propia

5.4.3.1 Métodos exactos

Los métodos exactos son eficientes en problemas de hasta 50 clientes, debido a restricciones de tiempo computacional. Los métodos exactos se pueden clasificar en 3 grupos: búsqueda directa de árbol, programación dinámica, programación lineal y entera. En la *Figura 3* se muestra la clasificación de estos métodos.

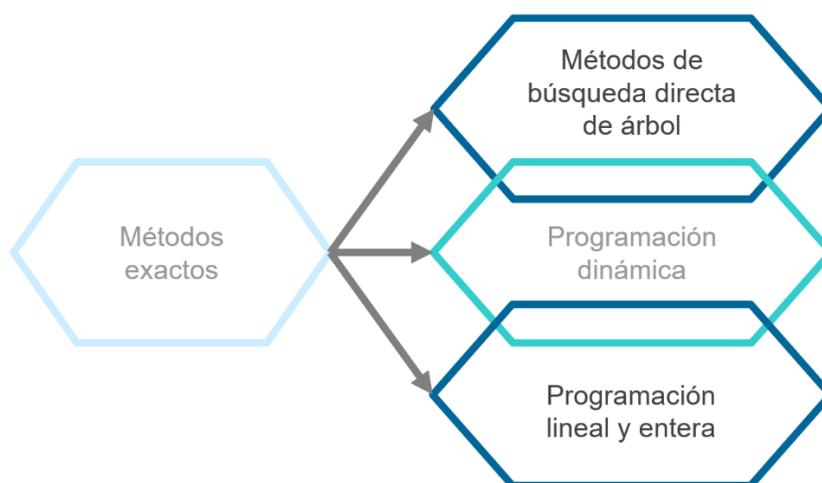


Figura 3. Estructura de los Métodos Exactos de Solución para VRP

Fuente: Elaboración Propia

Métodos de búsqueda directa de árbol

La búsqueda se realiza sobre todos los nodos de un árbol de acuerdo con criterios específicos propios de cada método.

5.4.3.2 Heurística

Las heurísticas son procedimientos que proporcionan soluciones de calidad aceptable mediante una exploración limitada del espacio de búsqueda. Clarke y Wright (Clarke & Wright, 1964) propusieron el primer algoritmo que resultó efectivo para resolver el VRP en 1964. La mayoría de las heurísticas clásicas para resolver el VRP fueron desarrolladas entre 1960 y 1990. Estos métodos parten de rutas que contienen un único nodo para encontrar el mejor par (nodo-ruta) que representa la mejor intersección. Los métodos heurísticos se pueden clasificar de diversas maneras, la más común es métodos constructivos, métodos de dos fases y heurísticas de mejora.

5.4.3.3 Metaheurísticas

Fueron desarrolladas hacia finales de la década de los noventa y se caracterizan por que realizan un procedimiento de búsqueda para encontrar soluciones de aceptable calidad, mediante la aplicación de operadores independientes del dominio que modifican soluciones intermedias guiadas por la idoneidad de su función objetivo.

Una de las propuestas más interesantes dentro del grupo de soluciones metaheurísticas al VRP se encuentra en las adaptaciones de métodos inspirados en los comportamientos de colonias de hormigas, cuyo origen se reconoce a los estudios en *colonias de hormigas* realizados por Dorigo y Stützle (Dorigo & Stützle, 2004), citados por Xiao y Jiang-qing (Xiao & Jiang-qing, 2012); cuyo interés se despertó en la dinámica propia de las colonias naturales de hormigas, las cuales

logran trazar los caminos más cortos a partir de la acumulación de feromonas, haciendo lectura de cambios y condiciones ambientales adversas.

En casos como el trabajo de Xiao y Jiang-qing (Xiao & Jiang-qing, 2012) se propone una optimización del algoritmo *híbrido de colonia de hormiga* (HAA), denominado por estos autores como *Ant Colony Optimization* (ACO), el cual consiste en una mejora del HAA, empleado en problemas de optimización combinatoria. ACO es entonces un planteamiento que tiene como base el algoritmo natural HAA mediante su aplicación computacional, el cual se sostiene también en el empleo de elementos del método heurístico de vecinos más cercanos o *Nearest Neighbor*; para Xiao y Jiang-qing (Xiao & Jiang-qing, 2012), ACO logra mejorar HAA para su empleo en VRP, bajo la combinación de la mejora planteada en la heurística y el acomodamiento dinámico de los parámetros vinculados en dicho algoritmo.

Xiao y Jiang-qing (Xiao & Jiang-qing, 2012) plantean que ACO simula de forma precisa el trabajo de estas especies de insectos formícidos, los cuales buscan optimizar su búsqueda de alimento desde sus nidos a las distintas fuentes, depositando feromonas en los senderos por donde transitan, atrayendo a sus compañeras, dicho depósito de feromonas es importante, pues guía a las demás de manera que se produce acumulación en las rutas más cortas. Por el contrario, la presencia de la feromona se difumina en los senderos más largos, proceso que además permite una adaptación de las rutas a las condiciones del medio, puesto que un obstáculo o condición adversa disminuye la acumulación. La ACO, como lo muestran Xiao y Jiang-qing (Xiao & Jiang-qing, 2012) es empleada para obtener de forma parcial o global la optimización en situaciones donde ocurre VRP y como antecedentes de la investigación de estos autores se encuentran estudios realizados a finales de la década del noventa y los primeros años del siglo XXI, concretamente, los dos autores chinos señalan como primera aparición de ACO, en situaciones marcadas por el VRP.

Un aspecto crucial dentro de la solución brindada por Xiao y Jiang-qing (Xiao & Jiang-qing, 2012), consiste en el desarrollo analógico con base en el caso de las colonias de hormigas, por lo que en el planteamiento de la solución de ruteo se inicia teniendo en cuenta que las unidades de transporte deben emplear el trazado señalado por el empleo de anteriores rutas, al modo como ocurre en las colonias. Además, se deben servir de la información heurística para construir soluciones factibles (Xiao & Jiang-qing, 2012).

De manera general la formulación matemática del CVRP puede ser establecida como se expone a continuación. En el caso de la maximización de la mayor cantidad de cargas con servicio, esto se logra aislando la falla a través de aquellos EM, más cercanos a la falla. Los otros dos objetivos son parte del proceso de optimización propuesto según las ecuaciones (5.1) y (5.2).

$$\text{Min } P(x) = \text{Pérdidas } (x) \quad (5.1)$$

$$\text{Min } NM(x) = \sum_{i=1}^s |sw_i - x_i| \quad (5.2)$$

Las restricciones consideradas en este problema son:

- Estructura radial
- Voltajes dentro límites permitidos

$$V_{min} < V_i < V_{max} \quad (5.3)$$

- Corrientes dentro límites permitidos

$$i_i < i_{max} \quad (5.4)$$

En esta nomenclatura x representa la topología de la red, s es el número de EM de la red, sw_i el estado del i -ésimo EM justo antes de la falla y x_i el estado del i -ésimo EM con la nueva topología. Por otra parte, la i_i es la corriente en la i -ésima línea e i_{max} el máximo valor de corriente permisible.

Función objetivo

El objetivo de estos algoritmos es encontrar un conjunto de soluciones eficientes bajo el concepto de dominancia de Pareto. La definición de dominancia de Pareto (asumiendo minimización) para dos vectores de decisión $x, y \in F$ (F se refiere a la región factible), indica lo siguiente:

Un vector $x = (x_1, x_2, \dots, x_k)$ se dice que domina (en el sentido de Pareto) a otro vector $y = (y_1, y_2, \dots, y_k)$ que se denota como $(x < y)$ sí y solo si:

$$\forall i \in (1, \dots, k), x_i \leq y_i \wedge \exists i \in (1, \dots, k): x_i < y_i \quad (5.5)$$

Donde,

k	Flota
x	Vector de decisión
y	Vector de decisión
F	Región factible

En otras palabras, un vector domina a otro, cuando este es menor o igual para todos los componentes y es estrictamente menor en al menos uno de ellos. Es importante notar que, si una solución x no domina a otra solución y , e y no domina a x , entonces ambos son no-dominados, es decir, no pueden ser comparados.

Las principales etapas del NSGA-II se describen a continuación:

1. Generar una población aleatoria P de tamaño N .
2. Identificar los frentes de dominancia: Para esto primero, por cada solución, se calculan 2 valores: (i) n_i , el número de soluciones que dominan a la solución i , y (ii) S_i , un conjunto de soluciones las cuales la solución i domina. Posteriormente se identifican todos los puntos que tienen $n_i = 0$ y se almacenan en una lista F_1 . Se llama F_1 a la frontera actual. Ahora, para cada solución en la frontera actual

se visita cada miembro (j) en su conjunto S_i y reduce su n_j uno a uno. De esta forma, si para cualquier miembro de j el valor se vuelve cero, se pone en una lista separada H . Cuando todos los miembros de la frontera actual han sido chequeados, se declara a los miembros de la lista F_1 como miembros de la primera frontera. Luego se continúa el proceso usando la nueva frontera H como la frontera actual.

3. Para mantener la diversidad en la frontera de Pareto, se calculan las distancias de apilamiento en cada frente. Para esto se debe calcular el perímetro del cuboide formado por las soluciones vecinas que poseen el mismo rango de dominancia que la alternativa i , lo cual se describe por medio de la siguiente ecuación.

$$d_i = \sum_{m=1}^M \left| \frac{fm^{i+1} - fm^{i-1}}{fm^{max} - fm^{min}} \right| \quad (5.6)$$

Donde I_m es un vector que indica la alternativa de solución vecina a la alternativa i , fm^{max} y fm^{min} son los valores máximos y mínimos sobre todo el espacio de soluciones de la función objetivo m , y M es el número de funciones objetivos optimizadas.

4. Usando selección, cruzamiento y mutación se genera una población descendiente del mismo tamaño P .
5. Reunir padres e hijos en un conjunto de tamaño $2N$, clasificar los frentes de dominancia y distancias de apilamiento.
6. Determinar el conjunto descendiente final seleccionando los frentes de mejor rango. Si se supera el límite de población N , eliminar las soluciones con menor distancia de apilamiento en el último frente seleccionado.
7. Si se cumple el criterio de convergencia, se finaliza el proceso. De lo contrario retornar al paso 4.

En el planteamiento de Xiao y Jiang-qing (Xiao & Jiang-qing, 2012) la aplicación de los elementos descritos permite generar un procedimiento de permanente mejoramiento del ruteo para solución del VRP, en el cual los pasos a seguir, propuestos por los autores, son: [1] se plantean los parámetros de funcionamiento de la flota de vehículos, [2] se genera una solución inicial a partir del uso del método heurístico de vecinos más cercanos.

Por su parte, Çatay (Çatay, 2010) presenta un algoritmo que busca solucionar el problema de la simultaneidad de entregas y recogida (agregando al modelo anteriormente mostrado la regla de recoger sólo en el momento que han realizado entrega dentro de su ruta), así como la base de ahorro con la que debe cumplir el sistema de transporte en el cual se aplica dicho algoritmo, distanciándose un poco del caso concreto planteado dentro del presente trabajo. Sin embargo, el propósito de los autores cumple, como se comentó con resaltar que dentro de este sistema “el objetivo es minimizar la distancia total atravesada” (Çatay, 2010). Por lo tanto, como lo señala el autor, se trata de un problema de VRP más complejo, aunque puede deducirse que sus logros pueden asegurar elementos útiles dentro de la estimación de límites de carga para cada vehículo y diversas variables propias de los productos transportados, así como el manejo apropiado para evitar su contaminación. Es esta la limitante de la propuesta que realiza Çatay (Çatay, 2010), el cual desarrolla un planteamiento numérico más detallado del ACO, apoyado por servidor.

Rincón García y otros (Rincon-Garcia et al., 2017) presentan un algoritmo metaheurístico híbrido para resolver el problema de enrutamiento del vehículo dependiente del tiempo con ventanas de tiempo, este algoritmo presentado hace uso de los enfoques de búsqueda de vecindarios grandes y las técnicas de búsqueda de vecindarios variables para guiar la búsqueda. Una primera etapa está diseñada específicamente para reducir la cantidad de vehículos requeridos en un espacio de búsqueda, esto mediante la reducción de las penalizaciones generadas

por las infracciones de la ventana de tiempo con los procedimientos de búsqueda de vecindarios grandes. Una segunda etapa minimiza la distancia de viaje y el tiempo de viaje en un espacio de búsqueda *siempre factible*.

Yu y otros (Yu et al., 2016) manejan el problema de entrega de alimentos al estilo occidental en la ciudad de Dalian China, que puede describirse como un problema de ruta de un vehículo con ventanas de tiempo. Desarrollan un modelo lineal entero para el problema, proponiendo un algoritmo mejorado de colonia de abejas artificiales, que posee una nueva táctica llamada estrategia adaptativa, la cual es una operación de cruce y otra de mutación para resolver el problema.

De Armas y Melián-Batista (De Armas & Melián-Batista, 2015) presentan un problema dinámico de enrutamiento de vehículos ricos en ventanas de tiempo, donde tuvieron en cuenta varias limitaciones reales, como la flota heterogénea de vehículos, las ventanas de tiempo múltiple y flexible, incluyendo las prioridades de los clientes. El uso de métodos exactos no es una solución adecuada para este tipo de problemas, dado que la llegada de una nueva solicitud debe ir seguida a una rápida fase de optimización para incluirla en la solución en cuestión. Por lo tanto, propusieron un procedimiento metaheurístico basado en la búsqueda de vecindad variable para resolver este problema en particular.

Kassem y Chen (Kassem & Chen, 2013) estudiaron un problema de enrutamiento del vehículo con la recogida y entrega simultáneas en la optimización de la red logística de circuito cerrado. Proponen un modelo de programación de enteros mixtos para formular el problema considerado, donde desarrollan un enfoque de solución heurística buscando resolver el modelo, debido a la naturaleza difícil de NP. La solución heurística se utiliza como una solución inicial de un procedimiento de recocido simulado para soluciones mejoradas.

A continuación, en la *Tabla 1*, se observan los autores que abordaron el proceso de cálculo de las variables dependientes:

Tabla 1. Autores que abordaron el proceso de cálculo de las variables dependientes

Artículo	Función objetivo - variable dependiente							
	Minimizar tiempo de entrega	Minimizar tardanza, retrasos o precocidad	Maximizar carga de trabajo	Minimizar tiempos de flujo	Minimizar costos de almacenamiento	Minimizar tiempo de preparación	Minimizar costo de recurso consumido	
Kassem & Chen, (2013)		1						1
De Armas & Melián-Batista, (2015)	1			1				
Yu et al., (2016)	1	1		1	1			
Rincon-Garcia et al., (2017)	1	1		1				
Onggo et al., (2019)	1					1		1
Li et al., (2020)	1	1	1	1				1
Zhang et al., (2017)	1		1	1				1
Çatay, (2010)	1	1						1
Bullnheimer et al., 1999)	1	1			1			1
López Ruiz, 2017)	1	1		1				1
Zulvia et al., (2020)	1	1		1				1
Asefi et al., (2019)			1					1
Sawik et al., (2017)	1					1		1
Simsir & Ekmekci, (2019)	1					1		1
Baradaran et al., (2019)	1	1		1				1
Coelho et al., (2016)	1	1		1	1			1
Kulkarni & Horn, (1996)	1			1				1
Ortega Calvo & Cayuela Domínguez, (2002)		1	1		1			1
Peduzzi et al., (1996)	1			1				1
Deb, (2001)		1	1		1	1		
Jaramillo, (1999)			1	1				1
Pelikan et al., (1999)		1			1			1
Farhang-Mehr & Azarm, (2002)			1		1	1		
Xiao & Jiang-qing, (2012)	1		1	1				1
Clarke & Wright, (1964)	1	1		1				
Dantzig & Ramser, (1959)	1	1			1			1
Secomandi, (2001)	1				1	1		

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

Artículo	Función objetivo - variable dependiente						
	Minimizar tiempo de entrega	Minimizar tardanza, retrasos o precocidad	Maximizar carga de trabajo	Minimizar tiempos de flujo	Minimizar costos de almacenamiento	Minimizar tiempo de preparación	Minimizar costo de recurso consumido
Rosen, (2003)	1	1	1		1		1
Prins, (2004)		1		1		1	
Dethloff, (2001)	1	1			1	1	1
Total	22	18	9	15	11	8	22

Fuente: Elaboración propia

6 Metodología

Para un adecuado desarrollo del trabajo fue necesario definir el tipo de investigación a efectuar, las técnicas de recolección de datos y los instrumentos utilizados para dar respuesta a la pregunta de investigación. Este estudio es de naturaleza cuantitativa y cualitativa dado que se analizarán diversas variables estadísticas y teóricas que allí se plasmarán. De igual forma, era necesario establecer de qué manera y con cuáles elementos e instrumentos se desarrollaría el objetivo.

El tema de interés fue estudiado de manera descriptiva ya que “se propone este tipo de investigación para la alineación de la información a la que se accederá a lo largo de la investigación, es describir de modo sistemático las características de una población, situación o área de interés” (Tamayo, 1999). El trabajo de investigación también se aborda de forma interpretativa, ya que se toman los datos en relación para posteriormente ser interpretados de acuerdo con el objeto de estudio planteado en la investigación. Para la investigación que se realizó, fue necesario utilizar dos técnicas para la recolección de información:

- *Revisión teórica o documental:* es una técnica que se basa en la recopilación de información por medio de estudios documentales y teorías planteados por los autores más representativos del tema a investigar.
- *La entrevista a semiestructurada:* que se realiza al encargado del área de logística de la pastelería ubicada en la región cafetera, esta entrevista “intenta ir al fondo de ciertas cosas, de ciertos aspectos generales necesarios para la investigación” (Aktouf, 2016).

En este trabajo, se utiliza un algoritmo genético multiobjetivo para resolver un problema de distribución de productos, visto como un VRP. Donde el objetivo, es minimizar la distancia recorrida por los vehículos, repercutiendo positivamente en

los costos. En este caso de estudio se enfoca al ruteo de vehículos al distribuir los productos de la pastelería ubicada en la región cafetera, ubicada en la ciudad de Manizales. Este problema en la actualidad está resuelto de manera empírica, lo cual impide que la distribución de los productos se realice de manera eficiente en tiempo y costo para la empresa.

Como se expresó anteriormente, la metaheurística NSGA-II (*Elitist Non-Dominated Sorting Genetic Algorithm*), es un algoritmo genético elitista que ha demostrado obtener buenos resultados en muchos problemas multiobjetivo en los últimos años. Tomando la estructura básica de este algoritmo, diseñamos operadores adecuados a nuestro problema para aproximar la curva de eficiencia. Primeramente, describimos la estructura básica del NSGA-II, en resumen, este es un algoritmo genético que opera sobre un conjunto de soluciones, una población de soluciones, también, usa operadores típicos de estos algoritmos: selección de padres, cruce, mutación, renovación o selección de individuos que pasan a la siguiente generación, etc.

En este punto es importante tener en cuenta que el problema de investigación consiste en diseñar las rutas en la planeación, buscando minimizar la distancia total recorrida y satisfaciendo la demanda de los distribuidores. Por lo cual, se evaluó el problema desde la perspectiva multiobjetivo, se determina un equilibrio entre los costos de transporte y los de almacenamiento, buscando buenas soluciones para ambos objetivos. El diseño de rutas alternativas que permitan la disminución de costos de transporte y reducción del recorrido, este se realizó a partir del análisis de los resultados obtenidos luego de la aplicación del VRP y el NSGA II, con lo cual se puede establecer la ruta más corta que atiende la mayor cantidad de clientes.

Este proceso de construcción puede entenderse como una toma secuencial de decisiones regida por una regla de transición estocástica. La interrelación de los factores que intervienen en la asignación de rutas y vehículos para la distribución

de los productos comercializados por el operador logístico y la definición de los parámetros a usar en la propuesta, la cual se llevó a cabo teniendo en cuenta el modelo actualmente implementado y estudiando su funcionamiento, es decir, teniendo en cuenta la información de entrada, como se analiza, se clasifica y como se determina que vehículos cubre una determinada zona de la ciudad.

De esta manera, para establecer grupos de clientes según la ubicación y distancia al operador logístico, se tuvo en cuenta el uso de la herramienta Google Maps (Google LLC, n.d.) para así tener un dato más exacto y facilitar el diseño de una red de transporte en la que se ve representada la ubicación de cada punto de venta, por medio de nodos y sus posibles conexiones por medio de arcos que representen sus distancias. La información heurística, también llamada visibilidad, mide lo deseable que es un nodo j para ser visitado desde i , con base en la información a priori del problema. En este caso, el rastro de feromona es modificado por dichos insectos virtuales. El parámetro τ refleja el rastro de feromona, mientras que η representa la información heurística (visibilidad). Ambos parámetros son utilizados en el algoritmo para calcular la probabilidad que permitirá a cada hormiga decidir cómo moverse.

Es importante destacar, que para la formulación y diseño del algoritmo se utilizó el programa MATLAB (The Math Works Inc., 2020), el cual básicamente es una herramienta de cálculo técnico y simulación que pueda emplearse sin necesidad de ser un programador para analizar o resolver un problema como sucedía antiguamente. Con esta herramienta tecnológica pero sencilla se determinan los cromosomas factibles o no factibles para la empresa donde el modelo de las hormigas en definitiva no se adapta de manera satisfactoria al modelo de negocio, puesto que no solo debe contar con una manera de seguimiento al proceso como lo determina un cromosoma tomado con el método de las hormigas. Es así, como se adopta la metaheurística NSGA II, con una metodología que desarrolla un multiobjetivo, propuesto para lograr los objetivos planteados.

De esta manera, se busca diseñar las rutas de sus vehículos para la distribución de sus productos buscando una reducción en los costos de transporte. Para resolver la problemática planteada se propuso permitir una cierta flexibilidad en la fecha de entrega, concretamente, permitir adelantar pedidos; esto es factible para la empresa ya que la entrega se hace a puntos de venta y no a usuarios finales. Como esta empresa en particular opera con productos perecederos, el adelanto permitido en la entrega fue de un día, sin embargo, el modelo y las metodologías de solución desarrolladas consideraron el caso general de adelantar los pedidos. Entonces el problema tratado fue diseñar de manera óptima las rutas de los vehículos que distribuyen los productos, satisfaciendo la demanda solicitada y realizando la entrega en la fecha inicialmente propuesta por las delegaciones o días antes.

Esta política de adelantar pedidos genera un costo asociado al almacenamiento, lo cual repercute en la economía de la empresa, por lo que también se abordó el problema desde un enfoque multiobjetivo buscando minimizar tanto el costo de transporte, medido por la distancia total recorrida por los vehículos, como el costo de dicha distribución.

7 Desarrollo y aplicación del modelo NSGA II para la pastelería ubicada en la región cafetera

7.1 Análisis de factores internos y externos de la pastelería

Para lograr una adecuada optimización en los tiempos de entrega del producto terminado es necesario tener en cuenta factores como:

Tipo y estado de las vías

Las vías por utilizar deben permitir el tránsito normal de los vehículos de la empresa como lo son autopistas, avenidas, viaductos, vías con pavimento rígido o asfaltadas para que los conductores no requieran realizar maniobras anormales durante el traslado. Esto es de gran importancia para el transporte del producto terminado ya que al transitar por una vía que se encuentre en malas condiciones el vehículo podría tener movimientos que comprometan la presentación de los productos.

Horas de alto tráfico

Son momentos en lo que no sería rentable para la empresa realizar la distribución de los productos, debido a que por el alto flujo vehicular que se presente en las zonas de distribución, los tiempos de entrega se retrasarían y esto puede tener un efecto negativo para los otros puntos de venta al no llegar a tiempo el inventario solicitado a la planta, además del alza en el costo del combustible por trayecto al tener un vehículo detenido con motor en marcha.

Conductores idóneos

Para esta operación es necesario contar con la experiencia de personas que tengan pleno conocimiento de la zona para que puedan realizar un desplazamiento acorde a las necesidades de la empresa, ya sea por las rutas que se programen previamente y que operen con normalidad o que sean capaces de tomar decisiones

respecto al cambio de itinerario según sea necesario en momentos de dificultades en las vías, además que cuenten con capacitación en manipulación de alimentos y conozcan los cuidados necesarios que se deben tener el momento de transporte de este tipo de productos.

Zona de descargue

La operación debe contar con zonas para el parqueo de los vehículos transportadores lo más cercanas posibles del punto de venta, donde no se ponga en riesgo la integridad física de las personas involucradas en el descargue de los productos y tampoco la calidad de estos por mala manipulación. Debe existir una coordinación entre el personal del punto de venta encargado de hacer el registro de ingreso del inventario y el personal que hace la correspondiente entrega, para darle dinamismo a este proceso de manera que se pueda continuar con la operación de entrega para otros puntos de venta.

Plan de distribución

El plan de distribución contiene un modelo con la cual se puede detallar las opciones disponibles para poner en marcha este proceso, conforme a esto determinar cuál de ellas es la que genera mayor rentabilidad a la hora de poner en marcha dicho plan (NSGA II).

Con este modelo determinamos el dinamismo necesario para aplicarle a la operación y de esta manera tener favorabilidad hacia los costos implícitos.

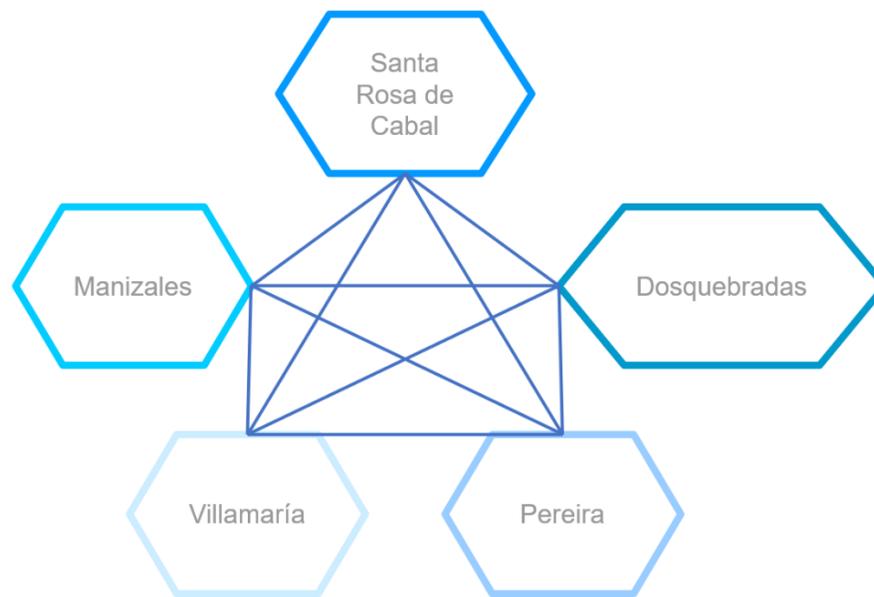


Figura 4. Modelo plan de distribución

Fuente: elaboración propia

Existen herramientas tecnológicas con las cuales se pueden hacer los cálculos necesarios para casos reales en soluciones de transporte. Para el caso de la Pastelería ubicada en la región cafetera se hicieron pruebas con el software MATLAB (The Math Works Inc., 2020) con miras a obtener resultados aplicando algoritmos, como se explica a continuación.

7.2 Estudio de caso

Después de analizar las variables anteriores, se procede a aplicar la metodología a la pastelería ubicada en la región cafetera, para aplicar esta metodología al caso real, se tiene en cuenta varios aspectos tales como puntos de venta, cantidad de vehículos, capacidad de carga y consumo de combustible por kilómetro. Para este tipo de programación, el estudio se hace mediante la organización *genética* de los vehículos según los puntos de venta a los que se les tenga que cumplir con entrega de inventario para cubrir su demanda de venta.

En la *Tabla 2* se muestran las distancias del proveedor a la pastelería, mientras que en la *Tabla 3* se indica la capacidad de los vehículos (canastillas) y consumo de gasolina (km/gl) de la flota de la Pastelería ubicada en la región cafetera.

Tabla 2. Distancias entre la fábrica y puntos de venta de la Pastelería ubicada en la región cafetera

	Fábrica	Enea	Estrella	Parque de las aguas	Centro	Villamaría	Palermo
Fábrica	0	3,9	8,2	12,9	12,2	9,5	7,7
Enea	3,9	0	7,3	11,4	11,3	10,6	6,7
Estrella	8,2	7,3	0	5,6	3,9	8,5	1,8
Parque de las aguas	12,9	11,4	5,6	0	1,5	7,9	6,2
Centro	12,2	11,3	3,9	1,5	0	7,2	4,6
Villamaría	9,5	10,6	8,5	7,9	7,2	0	6,7
Palermo	7,7	6,7	1,8	6,2	4,6	6,7	0

Fuente: Elaboración propia

Tabla 3. Capacidad de los vehículos (canastillas) y consumo de gasolina (km/gl) de la flota de la Pastelería ubicada en la región cafetera

Vehículos	Capacidad (canastillas)	km / gl
1	135	25
2	150	20
3	140	24
4	150	21

Fuente: Elaboración propia

En el ADN los cromosomas son los que determinan las características específicas del individuo. En este caso sería lo que determine como debe ser el orden o cuáles sean las opciones de ruta, de esta manera se toman las decisiones

que realmente sean adecuadas para el proceso de entrega y optimización de los costos de la operación. En la *Figura 5*, se puede observar el ejemplo:

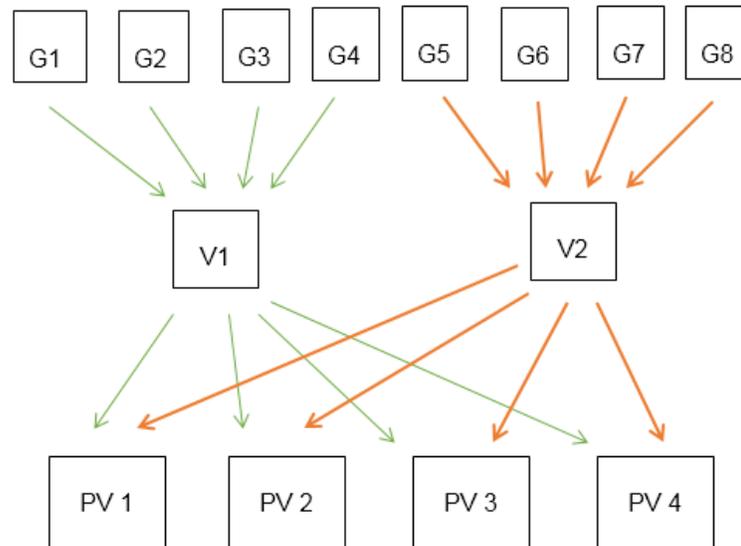


Figura 5. Ejemplo de organización genética de vehículos y puntos de venta

Fuente: Elaboración propia

Observando el anterior ejemplo de programación de los vehículos hacia los puntos de venta, se determina que no es factible dado que un solo vehículo puede cubrir la demanda de tortas de más de un punto de venta, mientras que este cromosoma muestra lo contrario y esto se traduce en aumento de recorridos y consumo de combustible, por lo que se considera como un cromosoma no factible.

Tomando otra opción para el análisis de la programación óptima de los vehículos se hace referencia al modelo multiobjetivo NSGA-II, el cual proporciona la guía para obtener las variables necesarias para lograr llegar al resultado esperado. En el caso de la pastelería ubicada en la región cafetera podemos tomar dos de ellos como lo son tiempos transcurridos para la entrega a cada punto de venta y el consumo de combustible de los carros o el recorrido. Según ese planteamiento, aquí es donde se evalúa cuáles de los recorridos hechos por los carros son los que se deben aplicar para lograr tener la ventaja real tanto para la operación, como el área

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

financiera de la empresa. Tomando como referencia 7 puntos de venta, la fábrica y 4 vehículos se establece el cromosoma, el cual se observa en la *Figura 6*.

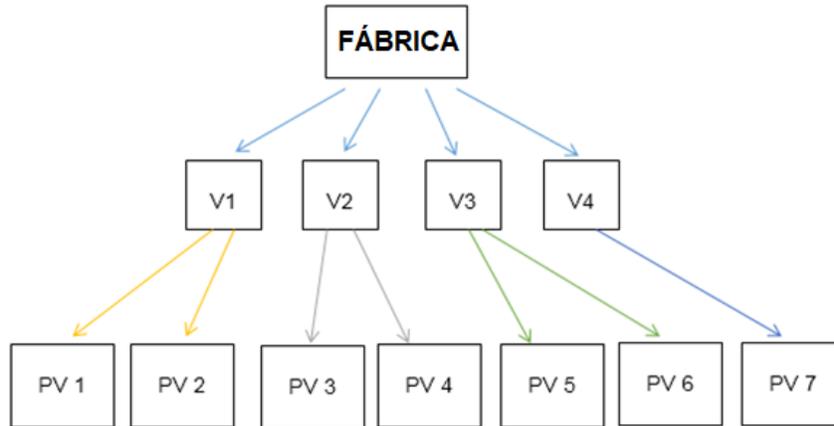


Figura 6. Cromosoma de distribución

Fuente: Elaboración Propia

En la *Tabla 4* se puede observar un ejemplo de cromosoma:

Tabla 4. Cromosoma

Genes	1	2	3	4	5	6	7	8	9
Vehículo	1	1	1	2	2	2	3	3	3
Punto de distribución	1	2	3	1	2	3	1	2	3

Fuente: Elaboración propia

A continuación, se describen los resultados de la tabla anterior:

Sí, el cromosoma es:

Ejemplo 1. Genes 1-6-8, significa que:

- El vehículo 1 va al punto 1
- El vehículo 2 va al punto 3
- El vehículo 3 va al punto 2

Ejemplo 2. Genes: 1-2-9, significa que:

- Vehículo 1 visita los puntos 1 y 2
- El vehículo 3 va al punto 3
- No se utiliza el vehículo 2

Con esta propuesta de distribución, se emplea el 1 vehículo en la entrega de las tortas necesarias para dos puntos de venta, así de esta manera, el cuarto vehículo entregaría al último punto y podría, si es necesario apoyar y/o reforzar en caso de que alguno de los otros puntos de venta se vea con agotados en su existencia de tortas o de apoyo en caso de alguna falla mecánica de los otros vehículos. En la *Tabla 5* se observa lo mencionado anteriormente.

Tabla 5. Recorrido hasta 6 puntos de venta

Vehículo	Origen de la ruta	Destino de la ruta	Recorrido en km	Consumo km/gl	Precio gl Acpm	Costo recorrido
V1	Fábrica	Enea	3,9	25	\$ 9.684	\$ 3.021
V1	Fábrica	Palermo	7,7	25	\$ 9.684	\$ 5.965
V1	Fábrica	Estrella	8,2	25	\$ 9.684	\$ 6.353
V2	Fábrica	Centro	12,2	20	\$ 9.684	\$ 11.814
V2	Fábrica	Parque de las Aguas	12,9	20	\$ 9.684	\$ 12.492
V3	Fábrica	Villamaría	9,5	24	\$ 9.684	\$ 7.667

Fuente: Elaboración Propia

Los resultados dados por esta tabla de cálculo fueron alimentados con combustible tipo ACPM y las distancias fueron apoyadas por resultado aproximados de la aplicación Google Maps (Google LLC, n.d.), la cual es una herramienta muy utilizada para la guía de rutas, esta indica el tiempo estimado de llegada y distancia en km del punto A (partida) al punto B (llegada).

Teniendo estos resultados, se verifican la cantidad de tortas que estos vehículos pueden entregar a cada punto de venta. Para este cálculo se toman como datos el transporte de hasta 300 tortas por viaje, cada una con un valor promedio de \$25.000. Estas tortas se repartirán según el cromosoma propuesto para el sistema de distribución desde la fábrica hasta el punto de venta. En la *Tabla 6*, se puede observar lo mencionado anteriormente:

Tabla 6. Resultado de la propuesta inicial

Vehículo	Origen de la ruta	Destino de la ruta	Cantidad cargada en fábrica	Cantidad entregada puntos de venta	Valor unitario	Valor entregado
V1	Fábrica	Enea		40	\$ 25.000	\$ 1.000.000
V1	Fábrica	Palermo	150	60	\$ 25.000	\$ 1.500.000
V1	Fábrica	Estrella		50	\$ 25.000	\$ 1.250.000
V2	Fábrica	Centro		80	\$ 25.000	\$ 2.000.000
V2	Fábrica	Parque de las Aguas	100	70	\$ 25.000	\$ 1.750.000
V3	Fábrica	Villamaría	90	90	\$ 25.000	\$ 2.250.000

Fuente: Elaboración Propia

7.3 Validación de la metodología

7.3.1 Diseño del algoritmo en MATLAB¹

Inicialmente se realizan los *scripts* y las funciones que se ejecutan en forma secuencial, luego se crean en ficheros *m* los cuales ingresan uno o varios argumentos de entrada, esto ejecuta una serie de operaciones que dan como resultado uno o varias variables, estas se conocen como las variables de salidas. El primer paso para crear una función es ingresar al formato predeterminado por MATLAB (The Math Works Inc., 2020) y escribir la función paso a paso. A continuación, se muestran las funciones creadas para el diseño del algoritmo:

¹ The Math Works Inc., (2020)

- **Buscar Frentes:** En la función buscar frentes se pretende obtener la información de todos los frentes, organizarlos y realizar un comparativo de variables como el número de población y el recorrido con el fin de identificar los elementos que quedan en la variable P los cuales no se ubicaron en ningún frente (*Figura 7*).

```

% >>
function[Frentes]=BuscarFrentes(V,N,M,TipoOpt);
% OBTENER TODOS LOS FRENTES
% V=Matriz de funciones objetivo;
% N=Número de individuos de Población
% M= # de funciones objetivo
% TipoOpt matriz donde 0 minimiza y 1 maximiza

P=1:1:N*2;
% P=[1 2 3 . . . . 2N];
% ORDENAR P y VAux ASCENDENTE O DESCENDENTE DEPENDIENDO SI ES DE
% MINIMIZAR O MAXIMIZAR LA FUNCION OBJETIVO 1

VAux=V;
if TipoOpt(1,1)==0
    for Cont1=1:2*N
        for Cont2=Cont1+1:2*N
            if VAux(Cont1,1)>VAux(Cont2,1)
                TempV=VAux(Cont1,:);
                TempP=P(1,Cont1);
                VAux(Cont1,:)=VAux(Cont2,:);
                P(1,Cont1)=P(1,Cont2);
                VAux(Cont2,:)=TempV;
                P(1,Cont2)=TempP;
            end
        end
    end
end

```

Figura 7. Función buscar frentes

Fuente: Elaboración propia

- **Función Buscar nuevos individuos:** En esta función, básicamente organizar los elementos de los frentes en forma descendente con relación a la distancia, es decir, los más distanciados se ubican en los primeros frentes, para esto se aplican los condicionales *While* e *If*, los cuales determinan los resultados arrojados con el fin de determinar los elementos de cada frente hasta completar el ultimo. En la *Figura 8*, se observa la función expresada en MATLAB (The Math Works Inc., 2020).

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
function[NuevosIndividuos]=BuscarNuevosIndividuos(MatrizFO,Frentes,M);  
  
%OJO LOS QUE ESTAN MAS DISTANCIADOS EN LOS PRIMEROS  
% FRENTES  
  
Tamano=size(Frentes); %Resulta vector (# de filas, # de columnas)  
N=Tamano(1,2)/2;  
NuevosIndividuos=[];  
L=1;  
i=1;  
  
% ACUMULAR ELEMENTOS DE LOS FRENTES EN NuevosIndividuos HASTA QUE  
%ELEMENTOS >= N  
while i<Tamano(1)+1 % HASTA NUMERO DE FILAS + 1  
    for j=1:Tamano(2) % HASTA NUMERO DE COLUMNAS  
        if Frentes(i,j)>0  
            NuevosIndividuos(L)=Frentes(i,j);  
            L=L+1;  
        end  
    end  
    if length(NuevosIndividuos)>=N  
        FilaUltimoFrente=i;  
        i=Tamano(1)+1;  
    end  
end  
end
```

Figura 8. Función Buscar individuos – 1

Fuente Elaboración propia

Posteriormente al ubicar los nuevos individuos, se procede a determinar si los elementos de los individuos cumplen las restricciones específicas, para esto se utiliza el condicional *If*, el cual establece los resultados aplicando la función *Length*, con el fin de escoger los elementos de los frentes extremos y ordenar de mayor a menor con respecto a la distancia en una matriz. En la *Figura 9*, se detallan las funciones y las restricciones mencionadas anteriormente.

```

% >>
%DETERMINAR NuevosIndividuos
if length(NuevosIndividuos)==N %SI LOS ELEMENTOS DE NuevosIndividuos SON IGUALES A N
    % nada hacer
else
    Descartar=[];
    %ESTE SERA EL VECTOR QUE TENDRA LOS ELEMENTOS A DESCARTAR DE NuevosIndividuos

    %SI HAY QUE ESCOGER UN ELEMENTO DEL ULTIMO FRENTE, ES DECIR EL PRIMER
    %EXTREMO
    if length(NuevosIndividuos)-length(UltimoFrente)+1==N %
        Descartar=UltimoFrente(1,2:length(UltimoFrente));
    else
        %SI HAY QUE ESCOGER DOS ELEMENTOS DEL ULTIMO FRENTE, ES DECIR LOS
        %EXTREMOS
        if length(NuevosIndividuos)-length(UltimoFrente)+2==N
            Descartar=UltimoFrente(1,2:length(UltimoFrente)-1);
        else
            %SI HAY QUE ESCOGER LOS EXTREMOS Y ALGUNOS ELEMENTOS INTERNOS
            %DEL ULTIMO FRENTE, ES DECIR HAY QUE ESCOGER MAS DE 2 ELEMENTOS

            % MATRIZ DE DISTANCIAS
            Distancias=zeros(length(UltimoFrente)-2,2);
            Distancias(:,1)=UltimoFrente(1,2:length(UltimoFrente)-1);
            for i=1:M
                for j=2:length(UltimoFrente)-1
                    Numerador=MatrizFO(UltimoFrente(1,j+1),i)-MatrizFO(UltimoFrente(1,
                    Denominador=MatrizFO(UltimoFrente(1,1),i)-MatrizFO(UltimoFrente(1,
                    if Denominador==0
                        Denominador=1.0000e-323;
                    end
                    Sumando=abs(Numerador/Denominador);
                    Distancias(j-1,2)=Distancias(j-1,2)+Sumando;
                end
            end
        end
    end
end

```

Figura 9. Función Buscar Individuos – 2

Fuente: Elaboración propia

Finalmente, la matriz se ordena de menor a mayor con respecto a la distancia, se escogen los de menor distancia de apilamiento aplicando la función *Length*, se descartan los nuevos individuos que no cumplan con la restricción propuesta; ($Distancias(1:length(NuevosIndividuos)-N,1)$), para luego ser borrados con la función *Borrar los descartados* para esto se aplican los ciclos *For* e *If* (Figura 10).

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
% ORDENAR MATRIZ DISTANCIAS DE MENOR A MAYOR
TamanoDistancias=size(Distancias);
for i=1:TamanoDistancias(1)-1;
    for j=i+1:TamanoDistancias(1);
        if Distancias(i,2)>Distancias(j,2)
            TemporalElemento=Distancias(i,1);
            TemporalDistancia=Distancias(i,2);
            Distancias(i,1)=Distancias(j,1);
            Distancias(i,2)=Distancias(j,2);
            Distancias(j,1)=TemporalElemento;
            Distancias(j,2)=TemporalDistancia;
        end
    end
end

%SE ESCOGEN LOS DE MENOR DISTANCIA DE APILAMIENTO
%Y SE COLOCAN EN Descartar
Descartar(1,:)=Distancias(1:length(NuevosIndividuos)-N,1);

end

end

% BORRAR LOS DESCARTADOS DE NuevosIndividuos
Posicion=1;
NuevosIndividuosAux=[];
for x=1:length(NuevosIndividuos)
```

Figura 10. Función Buscar Individuos - 3

Fuente: Elaboración propia

- **Función Filtrar cromosoma:** Esta es una función, en la cual se crean dos matrices:
 1. La primera matriz que se genera es la *MatrizInfactible*, la cual contiene el cromosoma infactible y se utiliza el bucle *For* el cual se usa para disponer un ciclo con un contador, que comprueba que se conserva dentro el límite propuesto.
 2. La segunda matriz generada es la *MatrizFactible* en la que por medio del ciclo *For*, en esta se determina la posición de la última bodega agregada

En la *Figura 11*, se detalla la función completa en el software MATLAB:

```

function [MatrizFactible]=FiltrarCromosoma(CromosomaInfactible, Genes, MatrizBase

% GENERAR MATRIZ INFACIBLE
MatrizInfactible=CromosomaInfactible;
for i=1:Genes
    MatrizInfactible(2,i)=MatrizBase(2,CromosomaInfactible(1,i));
    MatrizInfactible(3,i)=MatrizBase(3,CromosomaInfactible(1,i));
end

MatrizInfactible;

% GENERAR MATRIZ FACTIBLE
for i=1:Bodegas
    MatrizFactible(:,i)=MatrizInfactible(:,1);
    %EN QUE POSICION ESTA LA ULTIMA BODEGA AGREGADA
    P = find(MatrizInfactible(3,:)==MatrizFactible(3,i));
    [f, c] = size(P);
    for j=1:c
        MatrizInfactible(:,P(1,j)+1-j)=[];
    end
end
end

```

Figura 11. Función Cromosoma Factible

Fuente: Elaboración propia

- **Función Fobj:** Por medio de esta función se obtienen los datos iniciales de la carga, la posición, el recorrido y el consumo por cada vehículo. Para esto, inicialmente se utiliza el bucle *For* para verificar si el vehículo se va a utilizar, si el resultado es afirmativo, se identifica en qué posición está y se crea una matriz del vehículo usando el operador lógico *If*. Después se suman las variables por cada recorrido y se verifica si la cantidad de bodegas supera la cantidad del vehículo. Finalmente se suma el regreso de cada uno usando las funciones matemáticas. En la *Figura 12*, se observa lo mencionado anteriormente.

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
function [Kilometros,Consumo]=Fobj(MatrizFactible,Bodegas,Distancias,Demanda,Restricciones,v);  
  
% DATOS INICIALES = CARGA, POSICION, KILOMETROS, CONSUMO POR VEHICULO  
for i=1:v  
    Carga(i,1)=Restricciones(i,2);  
    Posicion(i,1)=Bodegas+1;  
    Kilometros(i,1)=0;  
    Consumo(i,1)=0;  
end  
  
Carga;  
Posicion;  
  
for i=1:v  
    % VERIFICAR SI EL VEHICULO SE VA A UTILIZAR  
    Auxiliar(1,:)=MatrizFactible(2,:);  
    cantidad=numel(Auxiliar(Auxiliar==i));  
    MatrizVehiculo=[];  
    if cantidad~=0  
  
        % EN QUE POSICIONES DE AUXILIAR ESTA I  
        P = find(Auxiliar==i);  
        [f, c] = size(P);  
  
        % CREAR MATRIZ DE VEHICULO :  
        for j=1:c  
            MatrizVehiculo(1,j)=MatrizFactible(2,P(1,j));  
            MatrizVehiculo(2,j)=MatrizFactible(3,P(1,j));  
        end  
        MatrizVehiculo;  
        [f, c] = size(MatrizVehiculo);  
  
        % SUMAR CARGA, POSICION, KILOMETROS, CONSUMO POR VEHICULO  
  
        % SUMAR PRIMER RECORRIDO CARGA, POSICION, KILOMETROS, CONSUMO POR VEHICULO  
        Kilometros(i,1)=Kilometros(i,1)+Distancias(Bodegas+1,MatrizVehiculo(2,1));  
        Carga(i,1)=Carga(i,1)+Demanda(MatrizVehiculo(2,1),1);  
    end  
end
```

Figura 12. Función Fobj

Fuente: Elaboración propia

- **Función FND Ordenado:** Esta función es la encargada de determinar la posición, si es izquierda o derecha, para esto se utiliza el condicional *If*, por medio de la cual genera restricción el largo de la variable P aplicando la función *Length*.

En la Figura 13 se observa la función completa en el MATLAB:

```

% Funcion FND_ordenado (V,P,TipoOpt,i)
function [FrenteAuxiliar]=FND_ordenado(V,P,TipoOpt,i);

N=length(P);
PosDividir=round(N/2);
if N>1
    Izq=FND_ordenado(V,P(1,1:PosDividir),TipoOpt,i);
    Der=FND_ordenado(V,P(1,PosDividir+1:N),TipoOpt,i);
    if TipoOpt(i)==0;
        if V(Der(length(Der)),i)<=V(Izq(length(Izq)),i)
            M=[Izq Der];
        else
            M=Izq;
        end
    else
        if V(Der(length(Der)),i)>=V(Izq(length(Izq)),i)
            M=[Izq Der];
        else
            M=Izq;
        end
    end
    FrenteAuxiliar=M;
else
    FrenteAuxiliar=P;
end

```

Figura 13. Función FND Ordenado

Fuente: Elaboración propia

7.3.2 Desarrollo metodológico

Cada punto de venta posee un valor d_{ij} que representa la distancia entre un punto i y el punto j . Se dispone de varios medios de trabajo o vehículos ($k=4$), y cada uno de ellos posee una capacidad Q_k de transportar carga y un consumo de combustible determinado. Adicionalmente se ha colocado las necesidades de carga para cada punto de venta y que los vehículos retornen a la fábrica al final de su ruta. Atendiendo a los teóricos sobre la conveniencia de poblaciones relativamente grandes, se trabaja con una población inicial de 100 individuos.

Detalle del algoritmo genético:

Paso 1: Leer las distancias entre puntos de ventas, los datos básicos de demanda de los distintos puntos. Así mismo cargar la información de los vehículos

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

disponibles, las restricciones impuestas al sistema, el número de especies (índices) o tamaño de la población y los parámetros constitutivos de la función de aptitud.

```
Distancias=[
0      7.3    11.4   11.3   10.6   6.7    7.9    3.9;
7.3    0      5.6    3.9    8.5    1.8    2      8.2;
11.4   5.6    0      1.5    7.9    6.2    3.6    12.9;
11.3   3.9    1.5    0      7.2    4.6    2.2    12.2;
10.6   8.5    7.9    7.2    0      6.7    7.1    9.5;
6.7    1.8    6.2    4.6    6.7    0      3      7.7;
7.9    2      3.6    2.2    7.1    3      0      9.3;
3.9    8.2    12.9   12.2   9.5    7.7    9.3    0];

Demanda=[55; 40; 35; 70; 40; 65; 60];

% TRANSPORTE    CAPACIDAD    CONSUMO
Restricciones=[
1              135          25;
2              150          20;
3              140          24;
4              150          21; ];

% Numero de individuos de la poblacion inicial, TIENE QUE SER PAR
N=100;

% Tasa de Mutacion
Tm=0.3;

% Determina el tipo de funciones y el número de funciones
TipoOpt=[0 0]; % 0 minimiza, 1 maximiza

% Número de Iteraciones
NumGeneraciones=150;
```

Paso 2: Para registrar la función de aptitud, se genera una matriz que representa una solución lógica, pero infactible, donde en la primera fila se representa el cromosoma infactible (números ordenados de 1 a 28), la segunda fila hace referencia al número asignado a cada uno de los vehículos disponibles (1 a 4) y la tercera fila hace referencia a cada punto de venta (1 a 7).

```

% UNA SOLUCION LOGICA PERO INFECTIBLE
% PRIMERA FILA REPRESENTA EL CROMOSOMA INFECTIBLE
% SEGUNDA FILA ES EL NUMERO DE VEHICULO
% TERCERA FILA ES EL PUNTO DE VENTA
cont=0;
for i=1:v % vehiculos
    for j=1:a % punto de venta
        cont=cont+1;
        MatrizBase(1,cont)=cont;
        MatrizBase(2,cont)=i;
        MatrizBase(3,cont)=j;
    end
end
end

```

Generando la matriz:

```

MatrizBase =
[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4
 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7]

```

Paso 3: Se genera una población de N individuos, donde cada individuo es una solución potencial al problema. Cada individuo tiene un cromosoma de tamaño 28 (número de vehículos x número puntos de venta) y cuyos genes son ubicados de manera aleatoria.

```

% GENERAR POBLACION DE PADRES INFECTIBLES
for i=1:N
    Poblacion(i,:)=randperm(Genes); % selección aleatoria
end

```

Por ejemplo, para el primer individuo, $i = 1$, se podría generar el siguiente cromosoma:

```

Poblacion(1,:)=
[17 10 20 16 14 9 18 8 2 21 25 15 4 22 7 11 1 5 12 3 27 28 24 23 6 26 19 13]

```

Después de formar la matriz población, de tamaño $[N$ (individuos) \times 28], el algoritmo evoluciona mediante tres operadores: selección, cruce y mutación, que en el programa se ejecutan gracias al siguiente código:

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
for i=1:N/2
% CRUZAR PAR DE INDIVIDUOS
Padre=Poblacion(i*2-1,:);
Madre=Poblacion((i*2),:);
% PMX - EXPLOTACION
[Hijo1,Hijo2]=PMX(Padre,Matre,Genes);

%PROCESOS DE MUTACION
if Tm>rand
[HijoMutado]=Mutacion(Hijo1,Genes);
Hijo1=HijoMutado;

[HijoMutado]=Mutacion(Hijo2,Genes);
Hijo2=HijoMutado;
end

%INGRESO A LA POBLACION DE PADRES E HIJOS
Poblacion((i*2-1)+N,:)=Hijo1;
Poblacion((i*2)+N,:)=Hijo2;
end
```

Paso 4: Se realiza la **Selección** de padres donde estos pasan sus genes a la próxima generación. Sobre la población conformada de tamaño N, las posiciones impares se seleccionan como padres y las posiciones pares como madres.

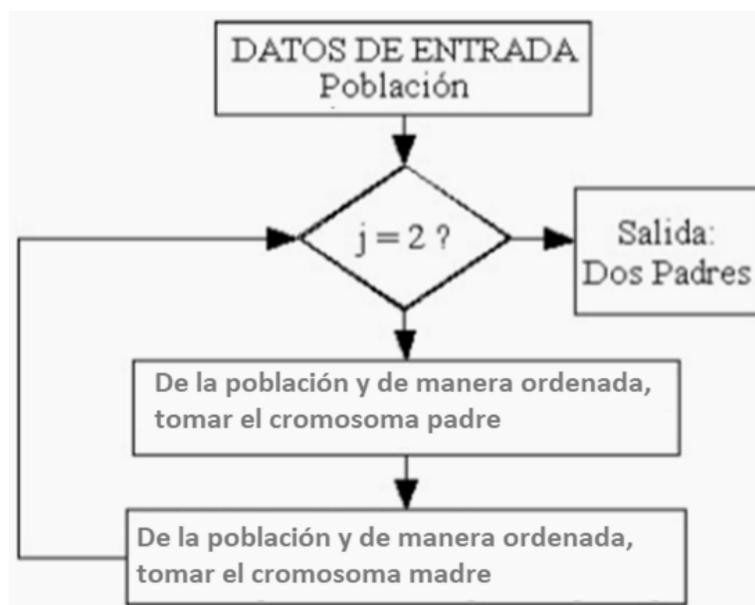


Figura 14. Diagrama de flujo de selección de padres

```

for i=1:N/2
    % CRUZAR PAR DE INDIVIDUOS
    Padre=Poblacion(i*2-1,:);
    Madre=Poblacion((i*2),:);

```

Paso 5: Se realiza el **Crossover** o **Recombinación** que representa la mezcla entre individuos. De los dos individuos seleccionados como padres, en dos puntos elegidos aleatoriamente dentro de sus cadenas de bits, se efectúa un corte y luego se realiza un cruzamiento de porciones del cromosoma. Se aplica la recombinación PMX (*Partially Mapped Crossover*), de manera que se garantice la legitimidad de las configuraciones obtenidas y que consiste en lo siguiente:

- a) Elegir aleatoriamente dos puntos de cruce (Pc1 y Pc2).

```

Pc1=floor(rand*Genes)+1;
Pc2=floor(rand*Genes)+1;
% Restricción para evitar tomar el mismo punto
while Pc2==Pc1
    Pc2=floor(rand*Genes)+1;
end
% Garantizar que el punto 2 sea mayor que el punto 1
if Pc1>Pc2
    Temp=Pc1;
    Pc1=Pc2;
    Pc2=Temp;
end

```

- b) Intercambiar estos dos (2) segmentos en los padres y generar dos (hijos).
c) En cada hijo eliminar el valor que se encuentre repetido y reemplazarlo por el bit (gen) que sustituyó tal valor del segmento en la cadena del padre.

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
% MAPEAR Padre para Hijo1
% PARA RECORRER AL PADRE
for i=Pc1:Pc2
    % PARA RECORRER AL HIJO 1 Y VERIFICAR QUE NO ESTARÁ REPETIDO
    for j=1:Genes
        if Madre(i)==Hijo1(j)
            Hijo1(j)=Padre(i);
            j=Genes;
        end
    end
end
Hijo1(Pc1:Pc2)=Madre(Pc1:Pc2);
```

Se sigue el mismo procedimiento para el hijo 2, luego tendremos que:

```
Padre=[ 17 8 12 25 1 18 14 15 5 7 24 9 19 3 4 6 2 22 11 16 23 13 21 28 20 27 10 26]
Madre=[ 3 9 27 4 10 16 18 22 6 21 2 20 13 26 25 14 12 1 11 15 23 8 19 24 17 28 5 7]
```

Pc1 = 2 y Pc2 = 11

```
Padre=[ 17 8 12 25 1 18 14 15 5 7 24 9 19 3 4 6 2 22 11 16 23 13 21 28 20 27 10 26]
Madre=[ 3 9 27 4 10 16 18 22 6 21 2 20 13 26 25 14 12 1 11 15 23 8 19 24 17 28 5 7]
Hijo1=[17 9 27 4 10 16 18 22 6 21 2 8 19 3 25 5 24 15 11 14 23 13 7 28 20 12 1 26]
Hijo2=[3 8 12 25 1 18 14 15 5 7 24 20 13 26 4 18 27 10 11 22 23 9 19 2 17 28 6 21]
```

En este ejemplo el gen de valor 9 de la madre reemplazó el gen 8 del padre y por ende en la cadena del hijo 1, donde se encontraba el gen 9 (que no hacía parte del segmento) y dado que no puede estar repetido se reemplaza por este valor 8.

Paso 6: El algoritmo de *Mutación* permite crear una gama de propuestas y su propósito radica en mantener la diversidad dentro de la población e inhibir la convergencia prematura. En esta etapa el cromosoma hijo obtenido en el proceso de recombinación tras un proceso de comparación entre un número aleatorio y la tasa de mutación puede ser sometido al proceso de mutación, en el cual después de escoger de manera aleatoria dos bits o genes, de manera comparativa se selecciona o bien si se toma una cadena de genes (limitada por los puntos

anteriores) para ser invertida en el cromosoma hijo o bien se realiza un intercambio entre los genes de estos puntos.

Como se muestra en la *Figura 15*, en el trabajo la permutación solamente se realiza una vez y sólo si un número que se selecciona aleatoriamente es menor que la tasa de mutación, sin embargo, el realizar el 100% de los casos una mutación (cambio de la T_m a valor de 1) parece llevar este problema en particular a un punto de saturación donde las permutaciones no aportan ninguna mejora a la secuencia evaluada pero sí aumentan el tiempo y recurso computacional empleado.

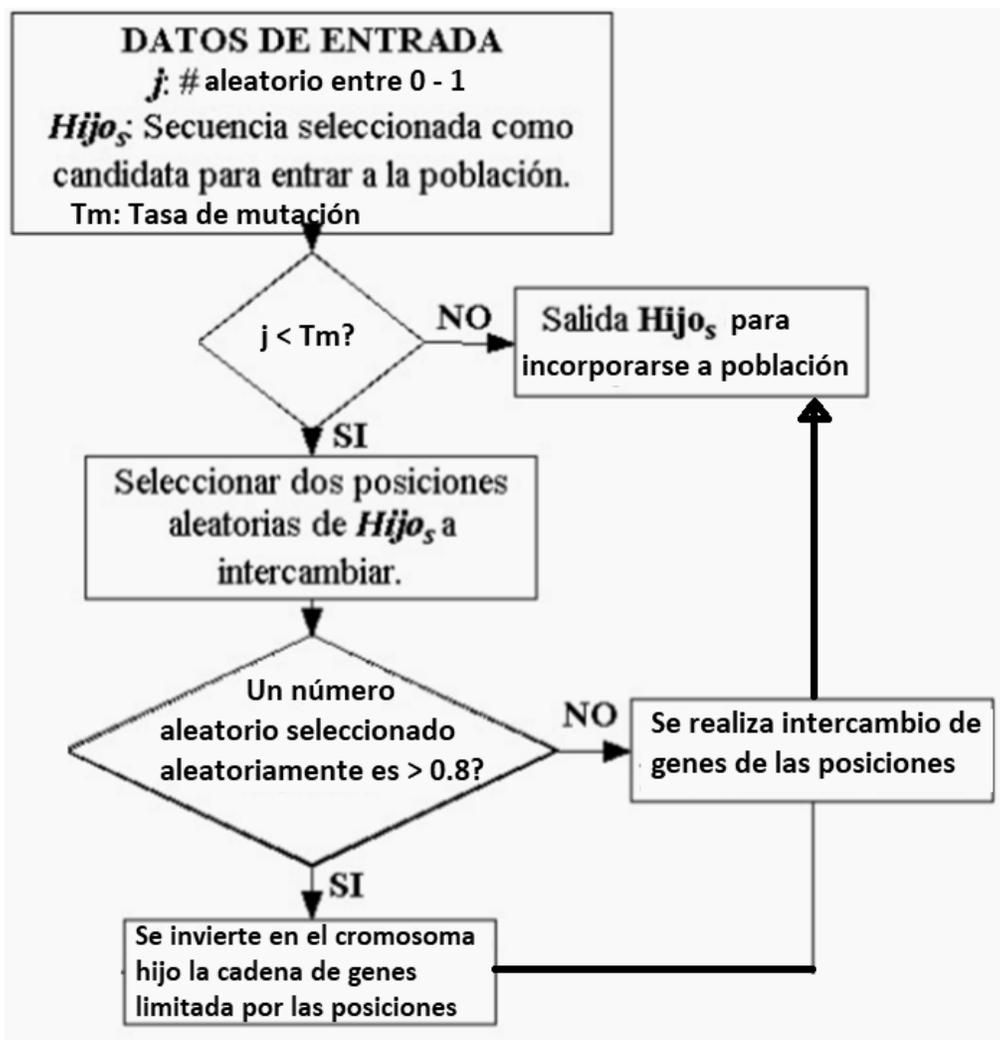


Figura 15. Proceso de mutación

Metodología basada en un algoritmo natural metaheurístico para programar el ruteo de los vehículos de la pastelería ubicada en la región cafetera

```
%PROCESOS DE MUTACION
if Tm>rand
    [HijoMutado]=Mutacion(Hijo1,Genes);
    Hijo1=HijoMutado;

    [HijoMutado]=Mutacion(Hijo2,Genes);
    Hijo2=HijoMutado;
end
```

Función *Mutación*

```
if rand>0.8
    % PARA ASEGURAR QUE Aleatorio1 es menor que Aleatorio2
    if Aleatorio1>Aleatorio2
        Temp=Aleatorio1;
        Aleatorio1=Aleatorio2;
        Aleatorio2=Temp;
    end

    % INVERTIR CADENAS DE GENES, Aleatorio1 y Aleatorio2
    Contador=0;
    for i=Aleatorio1:Aleatorio2
        HijoMutado(i)=Cromosoma(Aleatorio2-Contador);
        Contador=Contador+1;
    end
else
    % INVERTIR GENES, Aleatorio1 y Aleatorio2
    HijoMutado(Aleatorio1)=Cromosoma(Aleatorio2);
    HijoMutado(Aleatorio2)=Cromosoma(Aleatorio1);
end
```

Los dos nuevos hijos creados de esta mezcla se ubican en la próxima generación de la población, duplicando así el tamaño inicial de tal población.

```
%INGRESO A LA POBLACION DE PADRES E HIJOS
Poblacion((i*2-1)+N,:)=Hijo1;
Poblacion((i*2)+N,:)=Hijo2;
```

Paso 7: Se determinan las funciones (kilometro y consumo) para la población infactible del paso 5: población inicial + hijos: Para ello se genera una matriz con el

número de filas de la nueva población y con el número de columnas según las funciones a analizar.

```
% GENERAR FUNCIONES DE POBLACION INFECTIBLE
MatrizFO=zeros(N*2,M);
for i=1:N*2
    [MatrizFactible]=FiltrarCromosoma(Poblacion(i,:),Genes,MatrizBase,Bodegas);
    [Kilometros,Consumo]=FObj(MatrizFactible,Bodegas,Distancias,Demanda,Restricciones,v);
    MatrizFO(i,1)=max(Kilometros);
    MatrizFO(i,2)=sum(Consumo);
end
```

En el código de la función *FiltrarCromosoma*, se toma cada uno de los individuos de la población (padres e hijos), según el orden en que se hayan distribuidos los números en sus genes, se construye una matriz (matriz infactible) que, con base en la *MatrizBase* y acorde a los números de la primera fila, relaciona tanto vehículos como puntos de venta.

```
% GENERAR MATRIZ INFECTIBLE
MatrizInfactible=CromosomaInfactible;
for i=1:Genes
    j=CromosomaInfactible(i);
    MatrizInfactible(2,i)=MatrizBase(2,j);
    MatrizInfactible(3,i)=MatrizBase(3,j);
end
```

Así, por ejemplo:

```
Poblacion(1,:) = [6 18 20 16 11 5 14 12 10 22 21 4 28 2 9 15 13 19 1 8 7 3 26 17 27 24 23 25]
MatrizInfactible =[6 18 20 16 11 5 14 12 10 22 21 4 28 2 9 15 13 19 1 8 7 3 26 17 27 24 23 25
                  1 3 3 3 2 1 2 2 2 4 3 1 4 1 2 3 2 3 1 2 1 1 4 3 4 4 4 4
                  6 4 6 2 4 5 7 5 3 1 7 4 7 2 2 1 6 5 1 1 7 3 5 3 6 3 2 4]
```

Luego y con un tamaño acorde al número de puntos de venta, se genera una *MatrizFactible* así:

1. Inicio de ciclo
2. La matriz factible, toma la primera columna de la *MatrizInfactible* actual

3. Posteriormente y acorde al punto de venta recientemente agregado a la *MatrizFactible* se busca en la *MatrizInfactible* actual las posiciones en las que tal punto de venta se encuentre, por ejemplo, en el primer elemento de la infactible se hace referencia al punto de venta 6, luego con el algoritmo se encuentra que este punto de venta es referenciado en las posiciones 1, 3, 17 y 25.
4. Luego se retiran de la matriz infactible las otras combinaciones que vayan al punto de venta 6, de tal manera que no se repita el punto de venta que ya fue visitado y se garantiza que todos los puntos de venta serán visitados.

```
% GENERAR MATRIZ FACTIBLE
for i=1:Bodegas
    MatrizFactible(:,i)=MatrizInfactible(:,1);
    %EN QUE POSICION ESTA LA ULTIMA BODEGA AGREGADA
    P = find(MatrizInfactible(3,:)==MatrizFactible(3,i));
    c = length(P);
    for j=1:c
        MatrizInfactible(:,P(1,j)+1-j)=[];
    end
end
```

Así, para el mismo individuo, tendremos que se construye la siguiente *MatrizFactible*:

```
MatrizFactible =
     6    18    16     5    14    10    22
     1     3     3     1     2     2     4
     6     4     2     5     7     3     1
```

Una vez se cuenta con una matriz factible, se determina la función objetivo, para ello, por cada vehículo se hace lo siguiente:

1. Se verifica si en la propuesta (*MatrizFactible*) el vehículo analizado se va a utilizar, ello es sí en la fila 2 de la matriz se encuentra el número asignado al

vehículo bajo análisis, cuántas veces aparece y en qué posición de la *MatrizFactible* lo hace. En el ejemplo el vehículo 1 sí aparece, se utiliza 2 veces, en las posiciones 1 y 4.

2. Se construye una matriz de vehículo donde la primera fila hará referencia al número de vehículo (1 en este caso) y la segunda a los puntos de venta que visita, en el ejemplo 6 y 5.

$$\text{MatrizVehiculo} = \begin{matrix} 1 & 1 \\ 6 & 5 \end{matrix}$$

3. Luego de determinar la *MatrizVehiculo*, se calcula para su primer recorrido los kilómetros recorridos, los cuales corresponden a los kilómetros que existen desde la fábrica (punto de venta 8) hasta el primer punto de venta visitado, en el ejemplo, punto 6.

Por su parte, la carga remanente en el vehículo se determina restando a la carga inicial (considerando que el vehículo se carga a su máxima capacidad) la demanda del punto de venta visitado. Así también, el consumo de combustible se determina mediante el cociente de la distancia recorrida y el parámetro de consumo del respectivo vehículo. Finalmente, mediante el número que corresponde al punto de venta, se indica la posición donde quedó ubicado el vehículo.

```
% SUMAR PRIMER RECORRIDO CARGA, POSICION, KILOMETROS, CONSUMO POR VEHICULO
Kilometros(i,1)=Kilometros(i,1)+Distancias(Bodegas+1,MatrizVehiculo(2,1));
Carga(i,1)=Carga(i,1)-Demanda(MatrizVehiculo(2,1),1);
Consumo(i,1)=Consumo(i,1)+Distancias(Bodegas+1,MatrizVehiculo(2,1))/Restricciones(i,3);
Posicion(i,1)=MatrizVehiculo(2,1);
```

Kilómetros = 7.7000

Consumo = 0.3080

Carga = 135 - 65 = 70

Posición = 6

4. Cuando el vehículo visita más de un punto de venta, se deberá analizar si puede continuar su recorrido desde el punto de venta actual, o si por demanda del punto a visitar deberá regresar hacia la bodega para volver a cargar el vehículo en su totalidad. Independiente del caso, se determinará, sumando a los valores anteriores o acumulados, el kilometraje recorrido y el consumo respectivo.
5. Al visitar los puntos de venta determinados en la *MatrizVehiculo*, se calcula el total de kilometraje recorrido y el consumo total efectuado por cada vehículo de la *MatrizFactible* trabajada. En el ejemplo de la *MatrizFactible*, tendremos que los kilómetros recorridos por cada vehículo (1 al 4) son (1 al 4) son 23,9; 25,8; 24,3; 7,8 respectivamente, mientras que el consumo de combustible para cada vehículo es de 0,956; 1,290; 1,013; 0,371; respectivamente.
6. De los datos anteriormente obtenidos, se halla la distancia recorrida (25,8) y se suma el combustible utilizado por todos los vehículos, en el individuo analizado.

Una vez se ha calculado el valor de la función de aptitud para todos los individuos, se procede a seleccionar los descendientes, mediante el siguiente procedimiento:

1. Se carga la matriz en la que se han registrado los resultados de la función objetivo para cada individuo de la población.
2. Luego por comparación entre sí de cada uno de los resultados de la primera función objetivo, se va construyendo un vector *VAux* en el que se posicionan de manera ordenada (según se trate de minimización), así también el vector *P* indica cuál es la posición en la *MatrizOF* original.
3. Mientras los datos del vector sean mayores a cero, se obtendrá un frente con el método de *Kung*, se utiliza la función *FND ordenado*, con la cual, si se tiene más de un dato, se divide el vector *VAux* en dos partes iguales y a las cuales en forma cíclica se aplica la función *FND_ordenado*, hasta obtener un vector definitivo denominado *FrenteAuxiliar*.

7.4 Resultado del algoritmo

Después de ingresar los datos puntuales y reales al software MATLAB (The Math Works Inc., 2020), se obtuvo el siguiente cromosoma que tiene como punto de partida la fábrica y como destino los puntos de venta. Se observa que los recorridos más extensos son los que se hacen hasta los puntos de venta del parque (punto de venta 3) y centro (punto de venta 4) se realizan, según este planteamiento con el vehículo 2, el cual hace entrega de 75 unidades a cada punto consumiendo 44,8 y 44,4 galones de combustible respectivamente. Esta cantidad de galones consumidos por ese vehículo para la entrega a los dos puntos de venta está alrededor de \$863.000 (precio por galón \$9.684), esto es una muestra clara del costo elevado de la operación. Dado esto, se debe hacer otra planeación para esta ruta que resulte óptima y se vea reflejada en los costos operacionales. En la *Figura 14* se detalla el modelo propuesto:

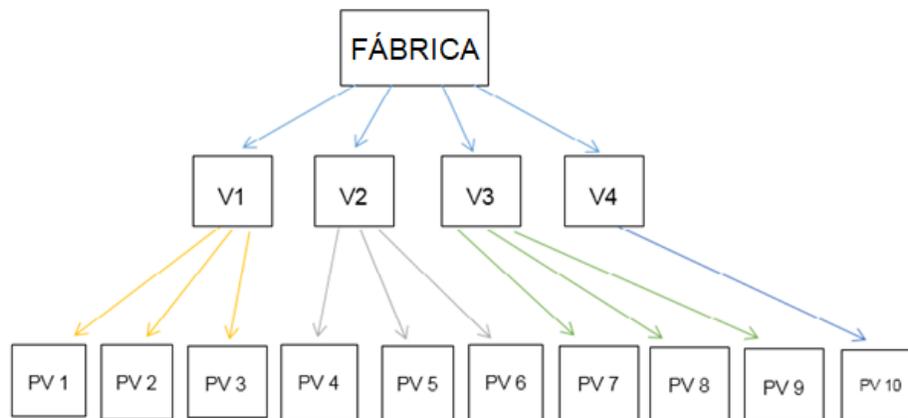


Figura 16. Cromosoma de distribución de productos para 10 puntos de venta con 4 vehículos

Fuente: Elaboración propia

Con este modelo de cromosoma se aumenta la carga a un punto de venta más con respecto al cromosoma anterior, manteniendo la opción de un vehículo para apoyo a la operación que es el V4 que hace entrega al punto de venta 10 pero hace el respaldo de las otras rutas.

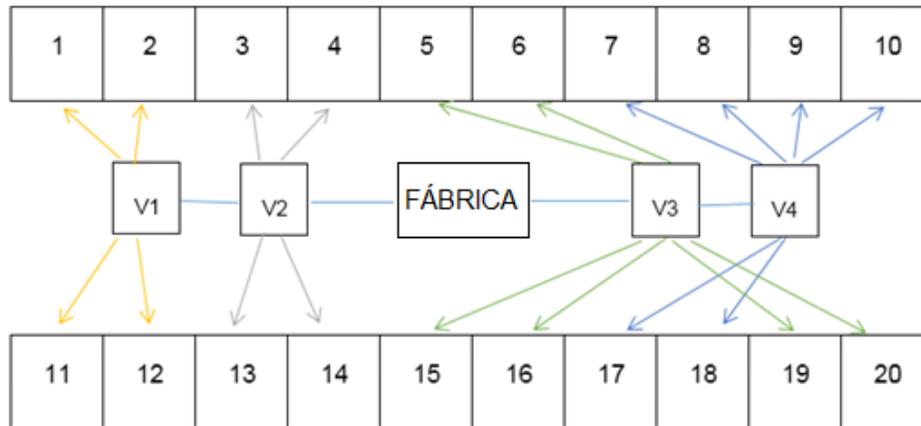


Figura 17. Cromosoma de distribución de producto para 20 puntos de venta con 4 vehículos

Fuente: Elaboración propia

Con el modelo de repartición para 20 puntos de venta, se observa que los puntos 9, 10, 19 y 20 serían programados en las rutas de los vehículos V3 y V4. Teniendo como respaldo los V1 y V2 como primera opción, como segunda sería la planeación de ruta semana a semana para que así los vehículos tengan rotación y el desgaste natural por su uso sea de manera equitativa.

En esta la distribución siguiente (Figura 18) se evidencia un aumento de rutas para cada uno de los vehículos lo que no sería factible con esta cantidad (4 vehículos), pues los tiempos de entrega van a ser mayor para la misma cantidad de tortas por punto lo que se traduciría en aumento de la relación combustible y costo de ruta. Con este cromosoma la conclusión es la necesidad de conseguir por lo menos un vehículo más, pues ya quedaría para la rotación semanal 6 puntos de venta para cada uno, o la programación entre los cuatro existentes y el quinto que sea el apoyo siendo un refuerzo para los imprevistos.

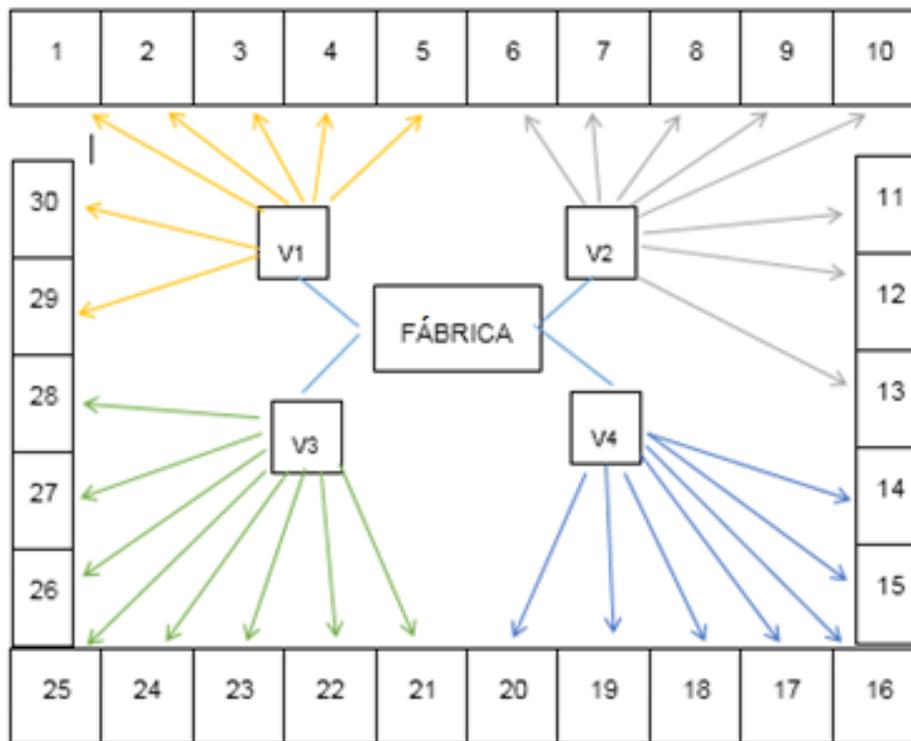


Figura 18. Cromosoma de distribución de productos para 30 puntos de venta con 4 vehículos

Fuente: Elaboración propia

7.5 Discusión de los resultados validados

En todo proceso empresarial es relevante hasta el más mínimo beneficio que resulte de las metodologías empleadas, con el fin de llevar a cabo su operación según la capacidad disponible para su modelo de negocio. En el caso de la pastelería ubicada en la región cafetera, se pudo determinar que la programación de sus vehículos para la entrega de la materia prima a los puntos de venta es diseñada de forma empírica en lugar de estar basada en un análisis logístico que evalúe aspectos técnicos y económicos. Teniendo en cuenta que son varios los puntos de la ciudad que se deben cubrir, la administración de todos los factores que participan en esta parte de la operación juega un papel importante en la productividad de toda la empresa. Buscando soluciones que sean adaptables e impliquen una mínima inversión, utilizando los recursos que ya existen, se

consultaron algunos modelos implementados y adaptados para este mismo tipo de negocio, como resultado se obtuvo el modelo de vida de las hormigas el cual logra de manera satisfactoria transportar la comida requerida desde la fuente hasta su nido. Pero para la productividad del negocio no resulta factible tenerlo en cuenta, porque esta es una solución mono objetivo la cual brinda sólo una opción de operación.

El modelo multiobjetivo NSGA-II resulta más adecuado para la operación de la logística de la pastelería, ya que esta opción involucra factores importantes y necesarios para el estudio de factibilidad como lo son tiempo transcurrido, distancia y consumo de combustible. Con estos datos, fácilmente se puede descubrir y seleccionar la guía de operación con la que se alcanzará la productividad que el modelo de negocio, logrando obtener mejores ganancias.

8 Conclusiones

- En este trabajo se desarrolló una metodología como opción para la designación de rutas de la pastelería ubicada en la región cafetera. Esta metodología se desarrolló en la herramienta computacional MATLAB (The Math Works Inc., 2020), la cual admite ejecutar una serie de acciones requeridas para la interactividad que demandan los procedimientos metaheurísticos.
- El buen desempeño del algoritmo depende de una buena calibración de los parámetros: tamaño de la población inicial, el valor de k para la elección de los descendientes, número de permutaciones (mutación). El valor asignado para cada parámetro depende exclusivamente de la complejidad del problema.
- Los cromosomas desarrollados evidencian la necesidad de conseguir por lo menos un vehículo más, pues ya quedaría para la rotación semanal 6 puntos de venta para cada uno, o la programación entre los cuatro existentes y el quinto que sea el apoyo siendo un refuerzo para los imprevistos.
- Con la metodología implementada fue posible encontrar múltiples soluciones garantizando su adaptabilidad a diferentes condiciones de demanda.
- Este método se proyecta como una alternativa en la solución de problemas de alta complejidad matemática y con el cual se podrían resolver problemas de mayor dificultad, simplemente modificando algunos parámetros.
- En el algoritmo planteado no fue necesario implementar una etapa de factibilidad, debido a que la población inicial se genera con secuencias factibles y en el operador de recombinación se utiliza el método *PMX* que garantiza secuencias legítimas.

- Teniendo en cuenta la metodología desarrollada se puede comprobar que empresas que tienen flota vehicular propia para el abastecimiento y distribución de productos, pueden desarrollar y aprovechar modelos matemáticos de VRP, para su posterior análisis y comparación de resultados con los métodos empíricos.
- Con el desarrollo de metodologías para el ruteo de vehículos se pueden optimizar los recursos de una empresa aplicando los modelos matemáticos. Además, el desarrollo de estas metodologías facilita la solución a gran número de problemas de ruteo en diferentes industrias y estas deben ser promovidas para su utilización.

Por lo anterior, de acuerdo con la argumentación expuesta, se cumplió el objetivo general y los específicos del trabajo final de la maestría.

Referencias

- Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1988). *Estructuras de datos y algoritmos* (Vol. 1). Addison-Wesley Iberoamericana.
- Aktouf, O. (2016). *La administración: entre tradición y renovación*.
- Andradóttir, S. (1995). A Stochastic Approximation Algorithm with Varying Bounds. *Operations Research*, 43(6), 1037–1048. <https://doi.org/10.1287/opre.43.6.1037>
- Asefi, H., Shahparvari, S., Chhetri, P., & Lim, S. (2019). Variable fleet size and mix VRP with fleet heterogeneity in Integrated Solid Waste Management. *Journal of Cleaner Production*, 230, 1376–1395. <https://doi.org/https://doi.org/10.1016/j.jclepro.2019.04.250>
- Ballou, R. H. (2004). *Logística: Administración de la cadena de suministro*. Pearson educación.
- Baradaran, V., Shafaei, A., & Hosseinian, A. H. (2019). Stochastic vehicle routing problem with heterogeneous vehicles and multiple prioritized time windows: Mathematical modeling and solution approach. *Computers and Industrial Engineering*, 131, 187–199. <https://doi.org/10.1016/j.cie.2019.03.047>
- Brooks, O. (2007). Solving discrete resource allocation problems using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm. *SpringSim* (3), 55–62.
- Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved Ant System algorithm for the Vehicle Routing Problem. *Annals of Operations Research*, 89(0), 319–328. <https://doi.org/10.1023/A:1018940026670>
- Çatay, B. (2010). A new saving-based ant algorithm for the Vehicle Routing Problem with simultaneous Pickup and Delivery. *Expert Systems with Applications*, 37(10), 6809–6817. <https://doi.org/10.1016/j.eswa.2010.03.045>
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581.
- Coelho, V. N., Grasas, A., Ramalinho, H., Coelho, I. M., Souza, M. J. F., & Cruz, R. C. (2016). An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. *European Journal of Operational Research*, 250(2), 367–376.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80–91.
- De Armas, J., & Melián-Batista, B. (2015). Variable neighborhood search for a Dynamic Rich Vehicle Routing Problem with time windows. *Computers and Industrial Engineering*, 85, 120–131. <https://doi.org/10.1016/j.cie.2015.03.006>

- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons.
- Deb, K., Anand, A., & Joshi, D. (2002). A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4), 371–395.
- Delyon, B. (1996). General results on the convergence of stochastic algorithms. *IEEE Transactions on Automatic Control*, 41(9), 1245–1255.
- Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1), 79–96.
- Díaz Gómez, H., García Cáceres, R., & Porcell Mancilla, N. (2008). Las Pymes: costos en la cadena de abastecimiento. *Revista Escuela de Administración de Negocios*, 0(63 SE-Artículos científicos). <https://doi.org/10.21158/01208160.n63.2008.438>
- Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. The MIT Press.
- FAO. (2012). *Pérdidas y desperdicio de alimentos en el mundo – Alcance, causas y prevención* (p. 42). <http://www.fao.org/3/i2697s/i2697s.pdf>
- FAO. (2014). *Pérdidas y desperdicios de alimentos en América Latina y el Caribe* (p. 10). <http://www.fao.org/3/a-i3942s.pdf>
- Farhang-Mehr, A., & Azarm, S. (2002). Diversity assessment of Pareto optimal solution sets: an entropy approach. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 1, 723–728.
- Glover, F. (1989). Tabu search—part I. *ORSA Journal on Computing*, 1(3), 190–206.
- Glover, F. (1990). Tabu search—part II. *ORSA Journal on Computing*, 2(1), 4–32.
- Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1), 156–166.
- Glover, F. W., & Kochenberger, G. A. (2006). *Handbook of metaheuristics* (Vol. 57). Springer Science & Business Media.
- Google LLC. (n.d.). *[Eje Cafetero]*. <https://www.google.com/maps>
- Hernández González, S. (2012). Experimentos con dos estrategias de búsqueda implementadas en el algoritmo recocido simulado para un problema de inventario. *Revista Facultad de Ingeniería Universidad de Antioquia*, 62, 33–43. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-62302012000100004&lang=pt
- Ho, W., Ho, G. T. S., Ji, P., & Lau, H. C. W. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4), 548–557.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

- Jaramillo, P. (1999). *Desarrollo de un sistema soporte a la decisión para la asignación de recursos naturales con satisfacción de múltiples objetivos y múltiples decisores*. Universidad Politécnica de Valencia.
- Kassem, S., & Chen, M. (2013). Solving reverse logistics vehicle routing problems with time windows. *The International Journal of Advanced Manufacturing Technology*, 68(1–4), 57–68.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Kulkarni, S. R., & Horn, C. S. (1996). An alternative proof for convergence of stochastic approximation algorithms. *IEEE Transactions on Automatic Control*, 41(3), 419–424.
- Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, A., Thakur, L. S., & Tiwari, M. K. (2016). Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Computers & Industrial Engineering*, 99, 29–40. <https://doi.org/https://doi.org/10.1016/j.cie.2015.07.003>
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3), 345–358. [https://doi.org/https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/https://doi.org/10.1016/0377-2217(92)90192-C)
- Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation* (Vol. 2). Springer Science & Business Media.
- Li, Y., Chu, F., Côté, J.-F., Coelho, L. C., & Chu, C. (2020). The multi-plant perishable food production routing with packaging consideration. *International Journal of Production Economics*, 221, 107472. <https://doi.org/https://doi.org/10.1016/j.ijpe.2019.08.007>
- López García, J. C. (2009). *Guía de algoritmos y programación para docentes* (2nd ed.). Fundación Gabriel Piedrahita. <http://eduteka.icesi.edu.co/pdfdir/AlgoritmosProgramacion.pdf>
- López Ruiz, J. (2017). *Metaheurísticas para el análisis de datos masivos en el ámbito del transporte por carretera* [Universidad Politécnica de Madrid]. http://oa.upm.es/47927/1/TFM_JAVIER_LOPEZ_RUIZ.pdf
- MADR. (2013). *Boletín de coyuntura económica IPC. Alimentos*. Ministerio de Agricultura y Desarrollo Rural.
- Martí, R. (2003). Procedimientos metaheurísticos en optimización combinatoria. *Matemáticas, Universidad de Valencia*, 1(1), 3–62.
- Martínez, C., & Mota, E. (2000). Del poliedro del agente viajero gráfico al de rutas de vehículos con demanda compartida. *Qüestió: Quaderns d'estadística i Investigació Operativa*, 24(3), 495–530.

- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097–1100.
[https://doi.org/https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/https://doi.org/10.1016/S0305-0548(97)00031-2)
- Olivera, A. (2004). Heurísticas para problemas de ruteo de vehículos. In *Reportes Técnicos 04-08*. UR. FI – INCO.
<https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/3508/1/TR0408.pdf>
- Onggo, B. S., Panadero, J., Corlu, C. G., & Juan, A. A. (2019). Agri-food supply chains with stochastic demands: A multi-period inventory routing problem with perishable products. *Simulation Modelling Practice and Theory*, 97, 101970.
<https://doi.org/https://doi.org/10.1016/j.simpat.2019.101970>
- Ortega Calvo, M., & Cayuela Domínguez, A. (2002). Regresión logística no condicionada y tamaño de muestra: una revisión bibliográfica. *Revista Española de Salud Pública*, 76, 85–93. <https://www.scielosp.org/article/resp/2002.v76n2/85-93/>
- Pan, T., Zhang, Z.-H., & Cao, H. (2014). Collaborative production planning with production time windows and order splitting in make-to-order manufacturing. *Computers & Industrial Engineering*, 67, 1–9.
<https://doi.org/https://doi.org/10.1016/j.cie.2013.10.006>
- Pardo, L., & Valdés, T. (1987). *Simulación: aplicaciones prácticas en la empresa*. Díaz de Santos.
- Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology*, 49(12), 1373–1379.
[https://www.jclinepi.com/article/S0895-4356\(96\)00236-3/abstract](https://www.jclinepi.com/article/S0895-4356(96)00236-3/abstract)
- Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, 1, 525–532.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.8687&rep=rep1&type=pdf>
- Polo Hernán, D., & Sastre Méndez, M. (2014). *Aprovisionamiento interno en pastelería. HOTR0109*. IC Editorial.
- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985–2002.
- Rincon-Garcia, N., Waterson, B., & Cherrett, T. (2017). A hybrid metaheuristic for the time-dependent vehicle routing problem with hard time windows. *International Journal of Industrial Engineering Computations*, 8(1), 141–160.
- Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. *Ingeniería*, 16(2), 35–55.

- Rosen, K. H. (2003). *Discrete Mathematics and Its Applications*, McGraw-Hill. *United State*.
- Sawik, B., Faulin, J., & Pérez-Bernabeu, E. (2017). A Multicriteria Analysis for the Green VRP: A Case Discussion for the Distribution Problem of a Spanish Retailer. *Transportation Research Procedia*, 22, 305–313.
<https://doi.org/https://doi.org/10.1016/j.trpro.2017.03.037>
- Secomandi, N. (2001). A Rollout Policy for the Vehicle Routing Problem with Stochastic Demands. *Operations Research*, 49(5), 796–802.
<https://doi.org/10.1287/opre.49.5.796.10608>
- Sedgewick, R. (1988). *Algorithms* (2nd Ed). Addison-Wesley Longman Publishing Co., Inc.
- Simsir, F., & Ekmekci, D. (2019). A metaheuristic solution approach to capacited vehicle routing and network optimization. *Engineering Science and Technology, an International Journal*, 22(3), 727–735.
<https://doi.org/https://doi.org/10.1016/j.jestch.2019.01.002>
- Tamayo, M. (1999). Aprender a investigar. In *Bogotá: Icfes* (1ra ed.). Arfo.
- The Math Works Inc. (2020). *MATLAB* (Version 2020a). [Computer software].
<https://www.mathworks.com/>
- Weiss, M. A., Moreno, J. tr L., Eguíluz, A. colab. téc, & Jacob, I. colab. téc. (1995). *Estructuras de datos y algoritmos*. Addison-Wesley Iberoamericana.
- Wilf, H. S. (1989). Algorithmes et complexité. In Masson (Ed.), *Volumen 2 de Logique mathématiques informatique*.
- Wilf, H. S. (2002). *Algorithms and Complexity*. Taylor & Francis.
- Xiao, Z., & Jiang-qing, W. (2012). Hybrid Ant Algorithm and Applications for Vehicle Routing Problem. *Physics Procedia*, 25, 1892–1899.
<https://doi.org/https://doi.org/10.1016/j.phpro.2012.03.327>
- Yu, S., Tai, C., Liu, Y., & Gao, L. (2016). An improved artificial bee colony algorithm for vehicle routing problem with time windows: A real case in Dalian. *Advances in Mechanical Engineering*, 8(8), 1687814016665298.
- Zhang, D., Cai, S., Ye, F., Si, Y.-W., & Nguyen, T. T. (2017). A hybrid algorithm for a vehicle routing problem with realistic constraints. *Information Sciences*, 394–395, 167–182. <https://doi.org/https://doi.org/10.1016/j.ins.2017.02.028>
- 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.
<https://doi.org/https://doi.org/10.1016/j.jclepro.2019.118428>