

UN ALGORITMO METAHEURÍSTICO PARA LA SOLUCIÓN DEL PROBLEMA DE
RUTEO DE VEHÍCULOS CON MÚLTIPLES DEPÓSITOS Y FLOTA HETEROGÉNEA

RUBÉN IVÁN BOLAÑOS

FACULTAD DE INGENIERÍAS
PROGRAMA DE MAESTRÍA EN INGENIERÍA ELÉCTRICA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA

PEREIRA

Agosto de 2014

UN ALGORITMO METAHEURÍSTICO PARA LA SOLUCIÓN DEL PROBLEMA DE
RUTEO DE VEHÍCULOS CON MÚLTIPLES DEPÓSITOS Y FLOTA HETEROGÉNEA

RUBÉN IVÁN BOLAÑOS

Proyecto de Grado
Para optar al título de Magíster en Ingeniería Eléctrica
Línea de Investigación de Operaciones

Director: Ph.D. Ramón Alfonso Gallego Rendón
Codirector: Ph.D John Willmer Escobar Velásquez

FACULTAD DE INGENIERÍAS
PROGRAMA DE MAESTRÍA EN INGENIERÍA ELÉCTRICA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA
Agosto de 2014

A Andrea, mi madre y hermanas.

AGRADECIMIENTOS

- Al director de este proyecto Ph.D Ramón Alfonso Gallego Rendón por guiarme durante todo este proceso y haberme dado a conocer este maravilloso mundo de la investigación, por su paciencia y sus sabios consejos. A él un gran aprecio y profundo respeto.
- Al codirector de este proyecto Ph.D John Willmer Escobar por su valioso aporte en este proyecto y por creer en mis capacidades.
- A mi madre Luz Marina Bolaños por su entrega, dedicación y por haberme enseñado el camino correcto de la vida, por que en todo momento ha estado a mi lado brindándome todo el apoyo, por sus consejos y educación. A ella todo mi amor y aprecio.
- A mis hermanas Gloria, Laddy y Carolina, de quiénes he recibido todo el apoyo para culminar una etapa más en mi vida. A ellas todo mi amor.
- A mi abuela por su gran labor y sabios consejos.
- A Andrea Velásquez quien se ha convertido en la luz de mi vida y fuente de inspiración. A ella todo mi amor y respeto.
- A mis amigos Sebastián Russy, Edward Andrés Angulo, Camilo Acosta, Oscar Danilo Montoya, Alejandro Grajales, Luis Fernando Grisales, Luis Miguel Escobar, quiénes me han brindado su amistad y apoyo durante este proceso.
- A cada uno de los integrantes de los grupos de investigación Planeamiento en Sistemas Eléctricos, Dinop y Gaope. Especialmente a Mauricio Granada, Ricardo Hincapié y Eliana Toro, quienes hicieron un gran aporte a mi formación personal y profesional.

Rubén Iván.

RESUMEN

En este trabajo se presenta un algoritmo metaheurístico para resolver el problema de ruteo de vehículos con múltiples depósitos y flota heterogénea (MDHFVRP), la formulación matemática del problema se basa en un modelo de programación lineal entera mixta de cuatro índices propuesto en la literatura especializada. Para solucionar el problema MDHFVRP se utiliza una técnica de solución basada en un algoritmo genético de Chu-Beasley modificado.

La metodología presenta un enfoque híbrido que permite generar una población inicial de buena calidad, para ello se utiliza como punto de inicio el procedimiento heurístico de “*Lin-Kernighan-Helsgaun*” (LKH), apoyado con la solución del modelo de programación lineal para el problema de la ruta más corta. Dos métodos de recombinación que permiten obtener descendientes diversos son implementados y utilizados de forma conjunta. Adicionalmente, se presenta una etapa de mejoramiento utilizada como operador de mutación, donde seis estrategias de búsqueda local inter-rutas y dos estrategias de búsqueda local intra-rutas son utilizadas, de tal manera que cada descendiente sea mejorado en cada ciclo generacional tal que sea posible entrar a formar parte de la población.

Para validar la metodología propuesta, se emplean instancias de prueba cuyo número de clientes varía entre 50 y 360. Las instancias han sido adaptadas y utilizadas en la literatura especializada como prueba de las metodologías existentes para el MDHFVRP.

Índice general

| | |
|--|----------|
| Índice general | I |
| Índice de figuras | IV |
| Índice de tablas | v |
| 1. Introducción | 1 |
| 1.1. Objetivos | 2 |
| 1.1.1. Objetivo general | 2 |
| 1.1.2. Objetivos específicos | 2 |
| 1.2. Antecedentes | 3 |
| 1.3. Aportes del proyecto | 7 |
| 1.4. Productos derivados de la investigación | 8 |
| 1.4.1. Ponencia en congreso | 8 |
| 1.4.2. Artículos en revistas | 8 |
| 1.5. Estructura del documento | 8 |
| 2. Descripción y modelado matemático | 9 |

| | | |
|-----------|---|-----------|
| 2.1. | Formulación matemática | 10 |
| 2.1.1. | Modelo matemático MDHFVRP | 10 |
| 2.1.2. | Modelo matemático para el problema de la ruta más corta | 12 |
| 3. | Metodología: Algoritmo genético de Chu-Beasley modificado | 14 |
| 3.1. | Generalidades | 14 |
| 3.2. | Descripción del algoritmo propuesto | 15 |
| 3.3. | Población inicial: Proceso híbrido | 16 |
| 3.3.1. | Secuencias del Gran TSP | 16 |
| 3.3.2. | División y evaluación de cada secuencia | 17 |
| | División para el VRP clásico | 17 |
| | División propuesta para el MDHFVRP | 18 |
| 3.4. | Selección | 20 |
| 3.5. | Recombinación | 21 |
| 3.5.1. | Recombinación PMX | 21 |
| 3.5.2. | Recombinación OBX | 21 |
| 3.6. | Mutación | 22 |
| 3.6.1. | Búsqueda local inter-rutas | 22 |
| | Shift (1,0) | 23 |
| | Shift (2,0) | 23 |
| | Shift (3,0) | 24 |
| | Swap (1,1) | 25 |

| | |
|--|-----------|
| Swap (2,1) | 25 |
| Swap (2,2) | 26 |
| 3.6.2. Búsqueda local intra-rutas | 27 |
| Rotación | 27 |
| 2-Opt | 28 |
| 3.7. Ingreso a la población | 28 |
| 3.8. Algoritmo general de la metodología propuesta | 29 |
| 4. Resultados | 30 |
| 4.1. <i>SS97</i> Vs <i>AGCBM</i> | 31 |
| 4.2. <i>SIW13-VNS1-VNS2</i> Vs <i>AGCBM</i> | 32 |
| 4.3. <i>HGSADC</i> Vs <i>AGCBM</i> | 33 |
| 5. Conclusiones, recomendaciones y trabajos futuros | 35 |
| 5.1. Conclusiones | 35 |
| 5.2. Recomendaciones y trabajos futuros | 36 |
| Bibliografía | 37 |
| Apéndices | 40 |
| A. Instancia | 42 |
| A.1. Encabezado | 42 |
| A.2. Coordenadas y datos de carga | 43 |
| A.3. Tipos de vehículos, Costos fijos y variables | 44 |

Índice de figuras

| | |
|---|----|
| 3.1. Gran TSP o tour gigante | 16 |
| 3.2. División aplicada al VRP | 18 |
| 3.3. Estrategia Shift (1,0) | 23 |
| 3.4. Estrategia Shift (2,0) | 24 |
| 3.5. Estrategia Shift (3,0) | 24 |
| 3.6. Estrategia Swap (1,1) | 25 |
| 3.7. Estrategia Swap (2,1) | 26 |
| 3.8. Estrategia Swap (2,2) | 26 |
| 3.9. Rotación | 27 |
| 3.10. Estrategia 2opt | 28 |

Índice de tablas

| | |
|--|----|
| 4.1. Comparación de las metodologías SS97 y AGCBM | 31 |
| 4.2. Comparación de las metodologías SIW13-VNS1-VNS2 y AGCBM | 33 |
| 4.3. Comparación de las metodologías HGSADC y AGCBM | 34 |
| A.1. Datos iniciales | 42 |
| A.2. Coordenadas | 43 |
| A.3. Tipos de vehículos | 44 |

Capítulo 1

Introducción

El ruteamiento de vehículos es un problema clásico de la optimización combinatorial de gran interés en el campo de la investigación de operaciones debido a su elevada influencia económica y dificultad para ser resuelto. La solución de este problema juega un papel importante en la cadena de suministro de las empresas cuyas actividades involucran transporte de bienes o personas. Generalmente las empresas se ven obligadas a enfrentar este tipo de problemas dentro de su propio sistema de distribución, para lo cual se requiere una adecuada gestión en los procesos logísticos, lo cual conlleva ahorros económicos que van desde el 10 % hasta el 20 % del costo de los bienes [1].

En este sentido, durante las últimas 5 décadas, diferentes investigadores en el mundo, han realizado un gran esfuerzo por proponer nuevas formulaciones que permitan resolver diferentes variantes del problema de ruteo de vehículos, incorporando en cada modelo características que describan de una forma más precisa los problemas de la vida real y a su vez puedan ser resueltos de una manera eficiente. En esta medida la investigación de operaciones ha experimentado un crecimiento exitoso, que sin duda alguna, se debe al desarrollo de los sistemas informáticos y a sus bajos costos, con los cuales se ha logrado un gran avance en la implementación de algoritmos cada vez más potentes, precisos y rápidos.

Las situaciones reales presentes en la logística, han sido las principales fuentes de motivación para el estudio de nuevas variantes del VRP (Problema de Ruteo de Vehículos) que se ajustan a las necesidades de los clientes, como lo son las ventanas de tiempo, servicios de entregas y recogidas o

propiedades adicionales, tal como duración de ruta, múltiples depósitos, flota heterogénea, costos fijos etc. Sin embargo, independiente de cual sea la variante del problema, el objetivo principal siempre estará enmarcado en la optimización de los recursos de transporte y la satisfacción de la demanda de los clientes.

El caso específico del problema de ruteo de vehículos multi-depósito que considera una flota heterogénea y adicional a eso relaciona costos fijos y variables de la flota (MDHFVRP por sus siglas en inglés), es una variante de difícil solución debido a la explosión combinatorial que generan las diferentes opciones de vehículos y depósitos disponibles para encontrar las rutas óptimas que deben atender al conjunto de clientes. La literatura especializada lo clasifica en los problemas de tipo NP-hard, lo cual implica que no es posible hallar una solución exacta en un tiempo de cómputo polinomial a medida que el tamaño del problema aumenta. Razón por la cual, las pocas metodologías que se han propuesto en la literatura son basadas en las técnicas heurísticas y metaheurísticas, que aunque no aseguran una solución óptima global, obtienen soluciones de muy buena calidad con un esfuerzo computacional muy bajo en tiempos de cómputo razonables.

Bajo estas condiciones, en esta propuesta de trabajo se busca determinar un modelo matemático que describa de forma precisa el MDHFVRP para ser solucionado mediante la implementación de técnicas heurísticas y metaheurísticas de tal manera que se cumplan los siguientes objetivos:

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar un algoritmo metaheurístico que permita solucionar el problema de ruteo de vehículos multi-depósito considerando flota heterogénea con costos fijos y variables mediante la implementación de una técnica metaheurística de optimización.

1.1.2. Objetivos específicos

- Realizar una revisión del estado del arte del problema de ruteamiento de vehículos (VRP).

- Realizar una revisión sobre la variante específica del MDHFVRP.
- Definir los modelos matemáticos que mejor representen el problema de ruteo de vehículos con múltiples depósitos y flota heterogénea.
- Elaborar una metodología que permita desarrollar el problema a partir de la implementación de técnicas heurísticas y metaheurísticas en el ambiente de programación de C++.
- Validar la metodología con base en diferentes instancias de prueba disponibles en la literatura especializada.
- Publicar y difundir resultados.

1.2. Antecedentes

Aunque el problema general de ruteamiento de vehículos ha sido estudiado ampliamente desde la primera formulación aplicada a la distribución de combustible realizada por Dantzig y Ramser (1959) [2], la versión que involucra múltiples depósitos y diferentes tipos de vehículos ha sido estudiada en menor medida. Entre las primeras investigaciones de ruteamiento que involucran estas características, se encuentra el trabajo realizada por Cassidy y Bennett en [3], en el cual los autores proponen un algoritmo iterativo que busca mejoras progresivas a partir de una solución inicial aleatoria o ingresada de forma manual para solucionar el problema de programación de rutas de vehículos aplicado a la entrega de comidas escolares. Para la formulación del problema se tienen en cuenta múltiples depósitos, flota de vehículos heterogénea y sus principales restricciones son el tamaño y duración de cada ruta. El objetivo consiste en minimizar el tiempo de transporte de las comidas. Las instancias utilizadas son reales e involucran 600 escuelas donde deben entregarse las comidas, 300 centros de preparación y 100 vehículos.

Wren y Holliday [4], presentan un algoritmo iterativo para la programación de rutas de vehículos para atender un conjunto de clientes desde uno o varios depósitos. La solución inicial es creada por medio de una adaptación del algoritmo de los ahorros propuesto en Clarke y Wright [5], el cual es aplicado a cada depósito de forma separada. Se cuenta con una flota homogénea y limitada para atender a los usuarios de acuerdo a las prioridades específicas de cada uno. Los resultados obtenidos

son comparados con las soluciones existentes para un único depósito.

Renaud *et al* [6], implementan un algoritmo metaheurístico Búsqueda Tabú para solucionar el problema de entrega de bienes a un conjunto de usuarios desde varios depósitos, bajo las restricciones de capacidad y tamaño de rutas. La solución inicial es creada mediante la asignación de clientes al depósito mas cercano. Para las etapas de mejoramiento los autores usan las estrategias denominadas 1-ruta, 2-ruta y 3-ruta. La metodología es probada en 23 instancias de la literatura especializada

En Salhi y Sari [7], se propone una heurística multinivel para solucionar el problema de ruteo de vehículos multi-depósito con flota heterogénea. El método en la primera parte utiliza una composición heurística denominada *borderline customers*, la cual busca crear soluciones iniciales factibles. En la segunda parte, se realiza una estrategia de búsqueda local que permite deteriorar o infactibilizar las soluciones, esto con el fin de evitar óptimos locales. Se realizan pruebas sobre instancias de hasta 360 clientes, con una cantidad de depósitos que varían entre 2 y 9 con vehículos de diferentes capacidades. Los resultados obtenidos igualan a las mejores soluciones reportadas en la literatura con un tiempo de cómputo mucho menor.

Nagy y Salhi [8], exponen diferentes algoritmos heurísticos que permiten solucionar el problema de ruteo de vehículos con entregas y recogidas de bienes. La metodología es capaz de resolver aquellos casos donde se considera un depósito y también cuando se consideran múltiples depósitos. Los autores usan el concepto de inserción y permiten que las entregas y recogidas de bienes se realicen de forma integrada. Los resultados obtenidos son alentadores con respecto a los reportados en la literatura.

Crevier *et al* [9], exponen una extensión al problema de múltiples depósitos, en la cual se considera completar la carga de los vehículos en depósitos intermedios a lo largo de la ruta. La metodología combina una técnica heurística de inicialización denominada algoritmo del barrido y un modelo exacto de programación entera (*Set Partitioning Model*). La etapa de mejoramiento se realiza a través de un algoritmo de memoria adaptativa basado en Búsqueda Tabú. Las pruebas se realizan sobre un conjunto de instancias generadas de forma aleatoria.

Un algoritmo genético híbrido para solucionar el MDVRP es presentado por Ho *et al* [10], en la primera parte del algoritmo se crean soluciones iniciales de forma aleatoria, mientras que en la

segunda etapa la heurística del vecino mas cercano y el algoritmo de los ahorros son los encargados de encontrar la población inicial. Las pruebas se realizan sobre instancias de diferentes tamaños. Los resultados obtenidos muestran un mejor desarrollo para el algoritmo genético que combina heurísticas en la etapa de inicialización.

Mirabi *et al* [11], desarrollan heurísticas híbridas para solucionar el MDVRP, el método combina técnicas determinísticas, estocásticas y un algoritmo de Recocido Simulado (SA), con lo cual se busca minimizar el tiempo de entrega de cada vehículo. Experimentos computacionales sobre un conjunto de pruebas generado aleatoriamente con diferentes tamaños de clientes y depósitos, muestran la eficiencia de la metodología propuesta.

Una adaptación del MDVRP al problema de diseño de una red urbana de fibra óptica para telefonía es presentada por Baldacci y Dell'Amico [12]. El problema es abordado desde la teoría de grafos y diferentes algoritmos heurísticos son implementados como métodos de solución. La metodología es probada sobre casos de prueba reales y algunos casos propuestos en la literatura especializada.

En [13], Los autores muestran una extensión del MDVRP, en la cual consideran un ganancia bruta y un precio de adquisición por cada cliente visitado. Dos modelos matemáticos de programación lineal entera mixta son formulados. Sin embargo, debido a la complejidad que presenta el problema, una Búsqueda Tabú también es propuesta para solucionar instancias de tamaño mediano y grande. La Búsqueda Tabú presenta un tiempo de ejecución ligeramente mejor que el obtenido por los modelos exactos resueltos mediante un solver comercial.

Un modelo de programación entera es presentado por Gulczynski *et al* [14] para resolver el MDVRP con entregas divididas. El objetivo es minimizar la distancia recorrida por los vehículos con entregas divididas, considerando un solo depósito y múltiples depósitos. Los resultados son obtenidos sobre un conjunto de 30 casos de prueba. Adicionalmente, nuevas instancias de prueba de alta calidad son creadas.

Un algoritmo genético de agrupamiento basado en la forma geométrica, es presentado en [15]. El método busca asignar clientes a cada depósito, de tal forma que la distancia de cada cliente a cada depósito dentro del grupo sea mínima. El algoritmo propuesto presenta un mejor desempeño con respecto a la heurística del vecino mas cercano en cuanto al tiempo de cómputo requerido.

Xu *et al* [16], presentan una nueva formulación matemática para el MDHFVRP con ventanas de tiempo. El método de solución involucra un algoritmo modificado de búsqueda en vecindario variable, junto con los operadores de inserción e intercambio, con los cuales se busca un balance entre calidad de solución y tiempo de ejecución. El algoritmo es examinado en instancias de prueba de la literatura especializada, demostrando que la metodología propuesta es competitiva.

Kuo y Wang [17], implementan una estrategia de búsqueda local para solucionar la variante del MDVRP con costos de carga. La estrategia de solución considera 3 etapas: la primera considera un método estocástico para generar soluciones iniciales, la segunda considera cuatro operadores seleccionados de forma aleatoria para encontrar soluciones vecinas y finalmente en la tercera etapa, un método similar al Recocido Simulado es utilizado para seleccionar una lista reducida de soluciones vecinas. Los resultados obtenidos demuestran la eficiencia y efectividad del algoritmo para resolver este tipo de problemas.

Una nueva variante del problema, la cual incluye localización de múltiples depósitos (LRP), es abordada por Wu *et al* [18]. Los autores proponen un método secuencial e iterativo junto a un algoritmo de Recocido Simulado como método de solución, considerando una flota homogénea e ilimitada de vehículos. Las soluciones encontradas son de buena calidad y el desarrollo del algoritmo se realiza en un tiempo de cómputo considerable.

Un algoritmo híbrido heurístico de dos etapas, que involucra una primera fase de construcción mediante diferentes estrategias de diversificación y una fase de mejoramiento por medio de un algoritmo de Búsqueda Tabú Granular junto a una estrategia de perturbación aleatoria utilizada para evitar óptimos locales en la variante del LRP capacitado es propuesta por Escobar *et al* [19]. Los resultados computacionales demuestran la eficiencia del algoritmo en tiempos computacionales y nuevas soluciones para las instancias de la literatura especializada.

Andrea *et al* [20], proponen un método exacto de *Branch-Cut-Price* para el MDHFVRP con ventanas de tiempo. La metodología se basa en tres estrategias de búsqueda y una generación de columnas. Resultados experimentales sobre el modelo exacto son presentados.

Trabajos similares al realizado en este documento, en el cual los autores proponen una formulación que describe el MDHFVR con costos fijos y variables son presentados por Salhi *et al* [21] y Vidal

et al [22]. Salhi *et al* [21], presentan una implementación eficiente de una búsqueda en vecindad variable (VNS), la cual incorpora diferentes operadores de vecindad y búsqueda local. La estrategia incluye un esquema de pre-procesamiento para identificar “clientes límite” (*borderline customers*), un mecanismo que agrega y desagrega rutas entre depósitos y una prueba para reducir las soluciones vecinas, lo cual permite disminuir el tiempo de cómputo en un 80 % , especialmente en instancias de gran tamaño. El algoritmo es probado en 26 instancias de la literatura especializada encontrando 23 nuevas mejores soluciones, lo cual refleja la efectividad y eficiencia de la metodología propuesta. Por otra parte Vidal *et al* [22], introducen una metodología de programación dinámica para evaluar eficientemente los vecindarios combinando movimientos basados en secuencias con una elección óptima de vehículos y depósitos, además de una estrategia denominada “Rotación” que permite determinar de manera óptima el primer cliente a ser visitado en cada ruta. Estos conceptos son aplicados mediante dos metaheurísticas: una búsqueda local iterativa y un algoritmo genético híbrido. Diferentes pruebas computacionales demuestran el desempeño notable de la metodología propuesta sobre instancias de la literatura especializada. Nuevas soluciones son encontradas para el MDVRP y el MDHFVRP. La metodología es bastante general y puede ser aplicada a muchas variante del VRP.

1.3. Aportes del proyecto

El desarrollo de esta investigación tiene como aportes los siguientes aspectos:

- Generalmente las metodologías existentes en la literatura, que involucran técnicas metaheurísticas, se basan en algoritmos de trayectoria. En este trabajo, se considera un algoritmo genético de Chu-Beasley como método de solución, el cual esta basado en poblaciones, con lo cual es posible extender el problema a una variante multi-objetivo.
- Un procedimiento híbrido adaptado a la variante multi-depósito del VRP, el cual involucra un proceso heurístico y la solución del modelo del problema de la ruta mas corta, con lo cual se obtiene la poblacion inicial de buena calidad y totalmente diversa.
- En la etapa de mejoramiento, ocho estrategias de búsqueda local inter-rutas e intra-rutas son utilizadas, las cuales permiten explorar el espacio de solución en busca de mejoras sobre las

soluciones encontradas.

- La metodología general puede ser usada para solucionar diferentes variantes del problema de ruteo de vehículos que van desde múltiples agentes viajeros, VRP clásico y la versión multi-depósito incluyendo flota heterogénea.

1.4. Productos derivados de la investigación

A continuación se presentan los productos derivados de este proyecto.

1.4.1. Ponencia en congreso

1. **Solution to the Capacited Vehicle Routing Problem by Branch & Bound Algorithm.** VIII ALIO/EURO Workshop on Applied Combinatorial Optimization, Montevideo, Uruguay, December 8-10, 2014. Estado de la ponencia: *revisión*.

1.4.2. Artículos en revistas

1. **Solución del problema de múltiples agentes viajeros resuelto mediante técnicas heurísticas.** Scientia et Technica . Estado del artículo: *Aceptado*.
2. **A population-based algorithm for the multi Travelling salesman Problem.** Pensée Journal. Estado del artículo: *en revisión*.

1.5. Estructura del documento

En el Capítulo 2 de este documento se presenta la descripción y formulación matemática del problema de ruteo de vehículos multi-depósito con flota heterogénea, costos fijos y variables. Luego en el Capítulo 3 se propone una metodología basada en el algoritmo genético de Chu-Beasley para solucionar el problema tratado. En el Capítulo 4 se describe la aplicación del algoritmo en diferentes instancias de prueba y los resultados obtenidos. Finalmente en el capítulo 5 se presentan las conclusiones y recomendaciones obtenidas de este trabajo.

Capítulo 2

Descripción y modelado matemático

El problema de ruteamiento de vehículos VRP (*Vehicle Routing Problem*) en su forma más simple, busca diseñar rutas que permitan atender la demanda de un conjunto de clientes que se encuentran dispersos geográficamente, desde un depósito con vehículos de capacidad idéntica [23]. Este problema ha sido estudiado ampliamente por mas de 50 años, sin embargo hoy en día se considera que es un campo de investigación creciente en el área operativa debido al impacto económico que representa en aplicaciones de la vida real dentro de las compañías que ofrecen servicios y/o productos. Precisamente, los problemas reales en el campo de la logística han llevado a que diferentes investigadores agreguen nuevas variantes al VRP en busca de modelos que representen cada una de las situaciones de una forma mas precisa. Diversas aplicaciones prácticas conducen a una variedad de atributos que complementan el modelo clásico del VRP, de tal manera que se aproxime a los requerimientos de los diferentes clientes (horarios, asignación), las características de la red de suministro y de los vehículos (flota heterogénea, múltiples depósitos, costos fijos) entre otros.

En esta investigación se consideran especialmente dos atributos que buscan acercar el VRP hacia una perspectiva más real: múltiples depósitos y flota heterogénea. Estas características dan origen a una nueva variante conocida en la literatura como MDHFVRP (*Multi Depot Heterogeneous Fleet Vehicle Routing Problem*), la cual se clasifica como un problema combinatorial de tipo NP-difícil al igual que la versión general del VRP.

El MDHFVRP se define mediante un grafo completo no dirigido $G = (V, E)$, donde $V = I \cup$

$J = \{1, 2, \dots, n + m\}$ representa el conjunto de vértices, el subconjunto $I = \{1, \dots, m\}$ hace referencia a los depósitos, cada uno con capacidad ilimitada, desde donde se deben atender n clientes representados por el subconjunto $J = \{m + 1, \dots, m + n\}$. Cada cliente contiene una cantidad de demanda no negativa q_i . Todas las posibles aristas que permiten una conexión entre los vértices se representan mediante el conjunto $E = \{(i, j) : i \in V, j \in V\}$, a cada arco se le asocia un costo de viaje c_i . Todos los depósitos cuentan con una flota heterogénea de K tipos de vehículos, cada uno con capacidad Q_k , costo fijo F_k que intenta representar el costo de depreciación de cada vehículo y un costo variable por unidad de distancia recorrida α_k . Además se supone que la cantidad de cada tipo de vehículo es ilimitada.

A partir de los parámetros que se consideran en el MDHFVRP, el objetivo de este problema consiste en minimizar la distancia recorrida por el conjunto de rutas que visitan todos los clientes, de tal manera que cada ruta termine en el mismo depósito donde inició y cada cliente sea visitado por una única ruta teniendo en cuenta las capacidad del vehículo y la máxima duración de ruta establecida (D).

2.1. Formulación matemática

En esta sección se presentan dos modelos matemáticos, el primero corresponde a la formulación matemática que describe el MDHFVRP y el segundo corresponde al problema de la ruta mas corta en una red.

2.1.1. Modelo matemático MDHFVRP

Matemáticamente el MDHFVRP con costos fijos y variables, puede ser formulado mediante el modelo de programación lineal entera mixta de cuatro índices presentado por Salhi *et al* [21], mediante las ecuaciones (2.1) a (2.11) y cuyos parámetros se describen a continuación:

- n : Número de clientes.
 m : Número de depósitos.
 q_i : Demanda del nodo i .
 K : Número de tipos de vehículos.
 Q_k : Capacidad del vehículo de tipo k .
 F_k : Costo fijo del vehículo de tipo k .
 α_k : Costo por unidad de distancia recorrida con el vehículo de tipo k .
 D_{ij} : Distancia entre los nodos i y j .
 X_{ijkd} : Es una variable de decisión binaria, la cual toma el valor de 1 si el vehículo de tipo k recorre el arco (i, j) desde el depósito d y cero en caso contrario.
 Y_{ij} : Es una variable continua no negativa, la cual indica la carga total que permanece en el vehículo antes de visitar el nodo j mientras atraviesa el arco (i, j) .

$$\text{mín } Z = \sum_{d=n+1}^{n+m} \sum_{k=1}^K F_k \sum_{i=n+1}^{n+m} \sum_{j=1}^n X_{ijkd} + \sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \alpha_k D_{ij} X_{ijkd} \quad (2.1)$$

s.a.

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} X_{ijkd} = 1; \quad \forall j = 1, \dots, n \quad (2.2)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{j=1}^{n+m} X_{ijkd} = 1; \quad \forall i = 1, \dots, n \quad (2.3)$$

$$\sum_{i=1}^{n+m} X_{ijkd} = \sum_{i=1}^{n+m} X_{jikd}; \quad \forall k = 1, \dots, K; j = 1, \dots, n+m; d = n+1, \dots, n+m \quad (2.4)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^n Y_{ij} = \sum_{j=1}^n q_j \quad (2.5)$$

$$\sum_{i=1}^{n+m} Y_{ij} - \sum_{i=1}^{n+m} Y_{ji} = q_j; \quad \forall j = 1, \dots, n \quad (2.6)$$

$$Y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^K Q_k X_{ijkd}; \quad \forall i = 1, \dots, n+m; j = 1, \dots, n \quad (2.7)$$

$$X_{d_1 i k d_2} = 0; \quad \forall i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n + 1, \dots, n + m \quad (2.8)$$

$$X_{i d_1 k d_2} = 0; \quad \forall i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n + 1, \dots, n + m \quad (2.9)$$

$$X_{i j k d} \in \{0, 1\}; \quad \forall i, j = 1, \dots, n + m; k = 1, \dots, K; d = n + 1, \dots, n + m \quad (2.10)$$

$$Y_{i j} \geq 0; \quad \forall i, j = 1, \dots, n + m \quad (2.11)$$

La ecuación (2.1) representa el costo total. El cual incluye costos fijos de los vehículos y el costo por unidad de distancia recorrida. Las ecuaciones (2.2) y (2.3) garantizan que cada cliente sea visitado una sola vez. La expresión (2.4) asegura la conservación de flujo. La restricción (2.5) muestra que la cantidad de carga que sale de todos los depósitos es exactamente igual a la demanda de todos los clientes y (2.6) garantiza que la cantidad de carga que permanece en el vehículo después de visitar el cliente j , es igual a la cantidad de carga antes de visitar este usuario menos su demanda. La restricción (2.7) asegura que la capacidad de cualquier vehículo usado no sea violada. Las ecuaciones (2.8) y (2.9) imponen que cada vehículo debe retornar al mismo depósito de donde salió. Finalmente la ecuación (2.10) representa el conjunto de variables de decisión binarias y (2.11) representa el conjunto de variables continuas no negativas.

2.1.2. Modelo matemático para el problema de la ruta más corta

Este modelo permite determinar el camino mas corto entre dos vértices cualquiera de una red. Sea n el conjunto de nodos de una red, en la cual ingresa una unidad externa de flujo en el nodo s y sale en el nodo t , siendo s y t los nodos entre los cuales se desea encontrar la ruta mas corta. También se define a x_{ij} como la cantidad de flujo en el arco (i, j) y c_{ij} como el costo del arco (i, j) . El modelo matemático esta descrito por las ecuaciones (2.12) a (2.14).

$$\text{mín } Z_2 = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (2.12)$$

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^n x_{ki} = \begin{cases} 1; & \text{si } i = s \\ -1 & \text{si } i = t \\ 0 & \text{en caso contrario} \end{cases} ; \forall i = 1, \dots, n \quad (2.13)$$

$$x_{ij} \in \{0, 1\} \quad (2.14)$$

La ecuación (2.12) representa la función objetivo, dado por el costo de las aristas utilizadas en la ruta que va desde el vértice s al t . El conjunto de ecuaciones (2.13) representa la conservación del flujo en cada una de las aristas y finalmente la ecuación (2.14) representa el conjunto de variables binarias que representan el flujo por cada aristas, las cuales toman el valor de uno si el flujo atraviesa la arista (i, j) y cero en caso contrario.

Capítulo 3

Metodología: Algoritmo genético de Chu-Beasley modificado

3.1. Generalidades

Los algoritmos genéticos hacen parte de las denominadas técnicas evolutivas, originalmente propuestas en los años 50, las cuales tienen una estructura básica común: realizan reproducción, variaciones aleatorias, promueven la competencia y ejecutan selección de individuos de una población dada, obteniendo como resultado la evolución. En la simulación computacional, los algoritmos genéticos simulan un proceso de selección natural para dar solución a problemas del campo de la optimización. En este caso, el problema a ser resuelto hace el papel de medio ambiente y cada individuo de la población está asociado a una solución candidata. Se dice que un individuo está mejor adaptado al ambiente, siempre que este sea una solución de mejor calidad para el problema. El objetivo del proceso evolutivo consiste en encontrar soluciones de mejor calidad en cada generación, sin embargo, esto no asegura la solución óptima del problema.

El algoritmo genético de Chu-Beasley pertenece a las denominadas técnicas metaheurísticas poblacionales, útiles cuando se busca resolver problemas de optimización para los cuales las técnicas exactas no resultan eficientes o no son aplicables. Esta técnica utiliza los conceptos del algoritmo genético básico propuesto por Holland [24]. Sin embargo posee unas características que lo convierten

en un algoritmo más eficiente y competitivo a la hora de evaluar sistemas de gran tamaño y complejidad. A continuación se describen algunas de las características más importantes:

- Utiliza la función objetivo para identificar el valor de la solución de mejor calidad y maneja la infactibilidad para el proceso de reemplazo de una solución por otra que se encuentra en la población.
- Sólo genera y sustituye un individuo a la vez en la población, en cada ciclo generacional.
- Es un algoritmo elitista, ya que un individuo ingresa a la población sólo si este es de mejor calidad que el peor de los individuos existentes.
- Cada individuo que ingresa a la población debe ser diferente a todos los que conforman la población actual, lo que evita la convergencia prematura del algoritmo a óptimos locales y además mantiene la diversidad, condición necesaria para lograr soluciones de alta calidad.
- Incluye un criterio de aspiración que consiste en que un individuo puede ingresar a la población actual, a pesar de no cumplir con el criterio de diversidad, si este es de mejor calidad que la solución incumbente.
- Puede incluir una etapa de mejoramiento después de aplicar los operadores genéticos de selección, recombinación y mutación. Esto permite explorar las regiones vecinas de la solución descendiente antes de evaluar si puede o no ingresar a la población actual.

3.2. Descripción del algoritmo propuesto

En esta sección se propone un algoritmo genético de Chu-Beasley modificado (AGCBM) como técnica de solución para el MDHFVRP. El AGBCM precisa de una población diversa obtenida mediante un procedimiento híbrido. Para el proceso de selección, dos torneos son llevados a cabo, cada uno con dos individuos elegidos de forma aleatoria. Los individuos ganadores de cada torneo son usados para llevar a cabo un proceso de recombinación especial adaptado para el MDHFVRP. El descendiente resultante, es dividido en rutas y seguidamente pasa a la etapa de mutación, la cual es diferente a la mutación tradicional. En particular se consideran estrategias de mejoramiento

con mecanismos de búsqueda local inter-rutas e intra-rutas. Finalmente, un nuevo individuo es presentado a la población actual y es ingresado bajo los criterios de diversidad y calidad. El proceso se realiza de forma iterativa hasta alcanzar un criterio de parada.

3.3. Población inicial: Proceso híbrido

3.3.1. Secuencias del Gran TSP

La primera etapa de la construcción de la población inicial es obtenida de soluciones aproximadas del tradicional TSP considerando: *a)* el conjunto completo de clientes sin incluir los depósitos, *b)* diferentes vértices de inicio en cada etapa del proceso de solución y *c)* un vehículo de capacidad infinita. Esta estrategia permite encontrar N secuencias o permutaciones de los clientes, donde cada una se denomina gran TSP o tour gigante. Con cada secuencia es posible obtener individuos de alta calidad después de un proceso de división, debido a que cada ciclo incorpora arcos altamente sensibles en la función objetivo.

Las N secuencias son construidas considerando las heurística del vecino más cercano [25], el método heurístico de *Lin-Kernighan-Helsgaun* (LKH) [26] y permutaciones de clientes aleatorias. Cada una de las técnicas mencionadas anteriormente determina una secuencia de TSP gigante como la que se presenta la Figura 3.1.

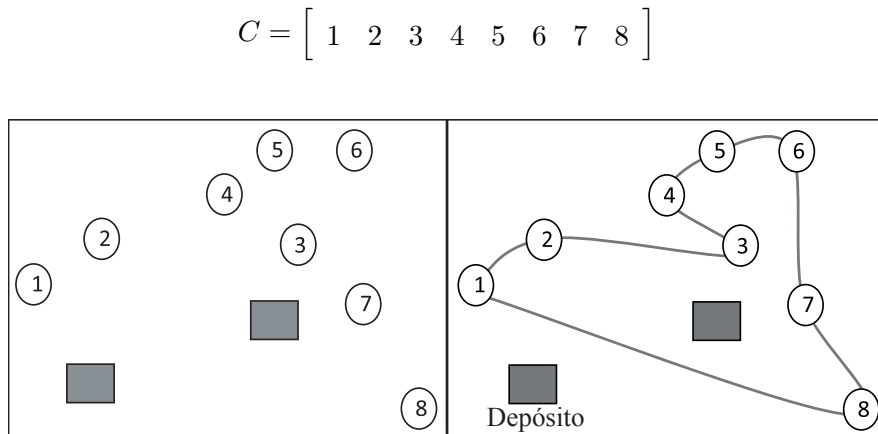


Figura 3.1: Gran TSP o tour gigante

3.3.2. División y evaluación de cada secuencia

Como punto de partida, se hace referencia al proceso de división para el VRP clásico presentado por Prins [27]. Seguidamente, se realiza una generalización para el MDHFVRP.

División para el VRP clásico

De acuerdo al orden de atención brindado por la secuencia del TSP gigante, se construye un grafo acíclico $H = (X, A)$, con $n + 1$ nodos, de esta manera X contiene un nodo estático o ficticio 0 más n vértices que representa el conjunto de clientes en el orden establecido por cada una de las secuencias $C = (C_1, \dots, C_n)$. Al conjunto A se añade un arco $(i - 1, j)$ por cada subsecuencia $(C_i, C_{i+1}, \dots, C_j)$ de C siempre y cuando la ruta que atiende a dichos clientes es un viaje factible. Es decir, si la demanda W_{ij} de todos los clientes de la subsecuencia no excede la capacidad del vehículo, tal como se muestra en la ecuación (3.1).

$$W_{ij} = \sum_{p=i}^j q(C_p) \leq Q \quad (3.1)$$

El costo de cada subsecuencia Z_{ij} esta dado por el costo de las aristas L_{ij} que representan el viaje asociado, el cual esta dado por las ecuación (3.2).

$$Z_{ij} = L_{ij} = D(0, C_i) + \sum_{p=i}^{j-1} D(C_p, C_{p+1}) + D(C_j, 0) \quad (3.2)$$

Para llevar a cabo el proceso de división de cada secuencia C de forma óptima, es necesario hallar la ruta más corta entre el nodo 0 y el nodo n del grafo H , para lo cual se resuelve el problema de programación lineal (PL) descrito en la sección ??ásCortamodeloRutamásCortae un grafo auxiliar que contiene todos las subsecuencias factibles.

La Figura 3.2 ilustra un ejemplo pequeño para el caso del VRP clásico, donde el gran tour esta dado por la secuencia $C = [a \ b \ c \ d \ e]$. En la Figura 3.2a se muestra el grafo acíclico H que representa la secuencia C junto con los posibles arcos que deben salir y llegar al depósito. La demanda de cada cliente se encuentra entre paréntesis y el costo asociado al recorrido se encuentra sobre cada arista. Cada uno de los viajes factibles extraídos de C son modelados por un arco sobre el

grafo auxiliar de la Figura 3.2b. Los arcos resaltados corresponden a la partición óptima, al aplicar el procedimiento híbrido, tomando como vértice de inicio el 0 y vértice final el e . La capacidad de los vehículos utilizados se asume igual a 10. La Figura 3.2c muestra las tres rutas obtenidas al realizar la decodificación de cada viaje con un costo total de 205 [27].

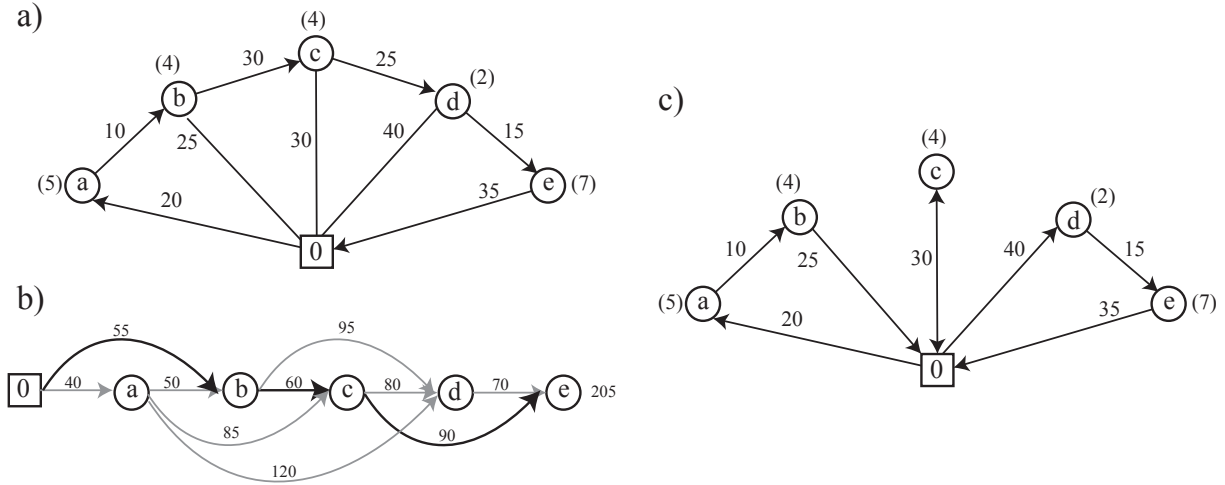


Figura 3.2: División aplicada al VRP

División propuesta para el MDHFVRP

Al igual que el caso del VRP, a partir de la secuencia C se construye el grafo acíclico H , sin embargo nuevas condiciones son consideradas debido a las características propias del MDHFVRP. Se considera que un viaje es factible, si la carga de cada subsecuencia $(C_i, C_{i+1}, \dots, C_j)$ no excede la capacidad del vehículo disponible de mayor tamaño, tal como se muestra en la ecuación 3.3.

$$W_{ij} \leq Q_K \quad (3.3)$$

Donde K corresponde al vehículo de mayor capacidad disponible.

El costo de cada subsecuencia Z_{ijkd} esta influenciada por los costos fijos (F_k) y variables (α_k) que dependen del tipo de vehículo k , con $k = 1, 2, \dots, K$, además se hace necesario la selección de el mejor depósito para cada subsecuencia. Para ello primero se realiza el cálculo del valor de los arcos

de la subsecuencia por medio de la ecuación (3.4).

$$L_{ij} = \sum_{p=i}^{j-1} D(C_p, C_{p+1}) \quad (3.4)$$

El siguiente paso consiste en encontrar, por medio de la ecuación (3.5), el tipo de vehículo más económico que permita suplir la demanda de cada subsecuencia.

$$Z_{ijk} = \text{mín}(F_k + \alpha_k L_{ij}); \quad \forall k = 1, 2, \dots, K; \quad W_{ij} \leq Q_k \quad (3.5)$$

Finalmente el mejor depósito es seleccionado por medio de la siguiente relación:

$$Z_{ijkd} = Z_{ijk} + \text{mín}(\alpha_k [D(d, C_i) + D(C_j, d)]); \quad \forall d = 1, 2, \dots, m \quad (3.6)$$

Es claro que cada viaje obtenido del grafo H , se describe por una arista en el digrafo auxiliar mediante una tripleta de la siguiente manera:

$$A_l = (Z_{ijkd}, d, k); \quad \forall l | l = (i, j) \text{ factible}$$

Como se puede observar, el proceso para llevar a cabo la división de cada secuencia C de forma óptima en el MDHFVRP, es similar al del VRP, con lo cual se resuelve el mismo PL descrito en la sección ??ásCortamodeloRutamásCorta Después de resolver el problema de la ruta más corta sobre el grafo H , la codificación del MDHFVRP puede representarse mediante una estructura de datos, la cual contiene los siguientes atributos: Un vector del TSP gigante, una matriz de las rutas que resultan después del proceso híbrido, un vector con el tipo de vehículo asignado a cada ruta, un vector con cada depósito desde donde se atiende a cada ruta, un vector que almacena el costo de cada ruta y una variable que almacena el valor de la función objetivo. A continuación muestra una estructura de datos que representa la codificación para el MDHFVRP a partir de un secuencia de TSP de 8 clientes.

$$Sol = \left\{ \begin{array}{l} Sec = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8] \\ \\ R = \left[\begin{array}{ccc} 1 & 2 & \\ 3 & 4 & 5 \\ 6 & & \\ 7 & 8 & \end{array} \right] \\ \\ C = [c_1 \ c_2 \ c_3 \ c_4] \\ \\ Dep = [d_1 \ d_2 \ d_3 \ d_4] \\ \\ V = [v_1 \ v_2 \ v_3 \ v_4] \\ \\ Fobj \end{array} \right.$$

3.4. Selección

Este proceso consiste en la selección de dos secuencias (individuos) de la población inicial, para ello se realizan dos torneos, cada uno con dos individuos elegidos de forma aleatoria. El criterio para elegir el ganador de cada torneo, se basa el valor de función objetivo que tenga la estructura asociada a cada secuencia elegida. A continuación se ilustra el proceso de selección.

$$\begin{array}{l} \textit{Torneo 1} \\ Sec_i = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8] \\ Sec_j = [8 \ 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1] \\ \textit{Torneo 2} \\ Sec_m = [4 \ 5 \ 6 \ 7 \ 8 \ 1 \ 2 \ 3] \\ Sec_n = [3 \ 2 \ 1 \ 8 \ 7 \ 6 \ 5 \ 4] \end{array} \quad \left\{ \begin{array}{l} \textit{Ganadores} \\ Sec_i = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8] \\ Sec_m = [4 \ 5 \ 6 \ 7 \ 8 \ 1 \ 2 \ 3] \end{array} \right.$$

3.5. Recombinación

Por tratarse de una codificación basada en permutaciones, llevar acabo el proceso de recombinación requiere de ciertos cuidado para evitar infactibilidades. Una de las restricciones que presenta el MDHFVRP es que cada cliente debe ser visitado una única vez y que todos los clientes deben ser atendidos, esto implica que dos genes dentro de un cromosoma no pueden tener el mismo valor. Diferentes estrategias pueden ser aplicadas para evitar secuencias infactibles. Entre ellas se encuentran los métodos PMX [28], OBX [29], OX [30] y CX [30]. En este trabajo se aplican las dos primeras estrategias de forma conjunta.

3.5.1. Recombinación PMX

A partir de una secuencia i y otra j , se generan dos puntos de corte (aleatoriamente) sobre la secuencia i y se extraen los genes que se encuentran dentro de dicho corte. Estos genes son copiados en un vector *hijo* respetando cada posición y luego se llenan los demás genes con la información de la secuencia j en su orden. Si existen repeticiones, estos se reemplazan por los genes correspondientes al corte aplicado en la secuencia j . El proceso se realiza de la siguiente manera:

$$\begin{array}{ccccccc} & & \downarrow & & \downarrow & & \\ Sec_i = & \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \right] \\ Sec_j = & \left[\begin{array}{cccccccc} 4 & 5 & 6 & 7 & 8 & 1 & 2 & 3 \end{array} \right] \\ Hijo = & \left[\begin{array}{cccccccc} 4 & 5 & 3 & 4 & 5 & 1 & 2 & 3 \end{array} \right] \rightarrow inf \\ Hijo = & \left[\begin{array}{cccccccc} 7 & 8 & 3 & 4 & 5 & 1 & 2 & 6 \end{array} \right] \rightarrow fac \end{array}$$

3.5.2. Recombinación OBX

A partir de dos secuencias i y j se genera un punto de corte sobre i y se copian en un vector hijo los genes ubicados a la izquierda de dicho corte. Luego se utilizan todos los genes de la secuencia j que aún no han sido empleados (respetando el orden de la secuencia). El proceso se realiza de la

siguiente manera:

$$\begin{array}{c}
 \downarrow \\
 Sec_i = \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \right] \\
 Sec_j = \left[\begin{array}{cccccccc} 8 & 7 & 2 & 3 & 6 & 5 & 1 & 4 \end{array} \right] \\
 Hijo = \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \end{array} \right]
 \end{array}$$

Una vez realizado el proceso de recombinación, el descendiente se divide mediante el proceso híbrido descrito en la sección 3.3.2. De esta manera se obtiene conjunto de rutas junto con los demás atributos pertenecientes a la codificación del MDHFVRP.

3.6. Mutación

El descendiente obtenido en el proceso anterior, es sometido a un proceso de mutación especial, el cual se basa en la aplicación de un proceso de estructuras de vecindad que busca mejorar cada una de las rutas obtenidas al realizar la división al descendiente que resulta del proceso de recombinación. Las etapas de mejoramiento se conocen en la literatura especializada como búsqueda local inter-rutas e intra-rutas [31].

3.6.1. Búsqueda local inter-rutas

El proceso inicia con una lista de rutas aleatoria, entre las cuales se va a llevar acabo el proceso de mejora. Seis estructuras de vecindad que involucran movimientos entre un par de rutas son aplicadas. El espacio de solución de las seis estrategias de mejora es explorado de forma exhaustiva, es decir, todas las posibles combinaciones son examinadas y sólo se considera el mejor de los movimientos en cada estrategia. Las seis estructuras de vecindad inter-rutas se describen a continuación:

Shift (1,0)

Un cliente C_i de una ruta R_k es transferido a una ruta R_l . En el siguiente ejemplo, se muestra como el cliente 6 que se encuentra en la primera posición de la ruta 3, es transferido al final de la ruta 2.

$$R = \begin{bmatrix} 1 & 2 \\ 3 & 4 & 5 \\ 6 \\ 7 & 8 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 \end{bmatrix}$$

La Figura 3.3 muestra gráficamente la aplicación el método Shift (1,0) sobre las rutas dadas anteriormente.

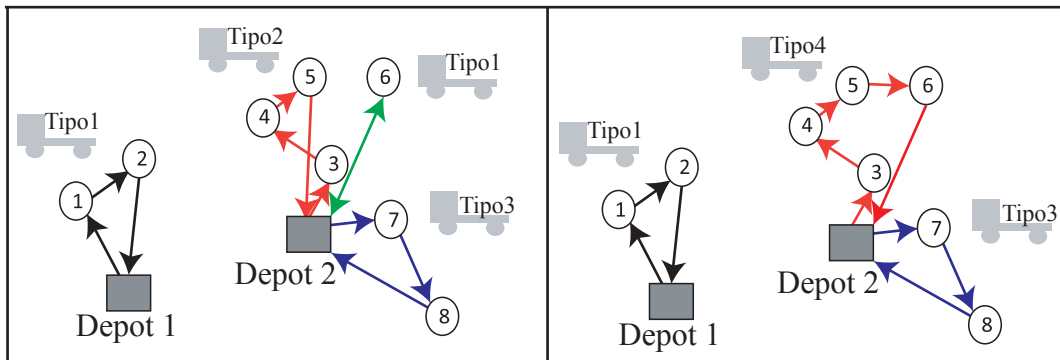


Figura 3.3: Estrategia Shift (1,0)

Shift (2,0)

Dos clientes consecutivos C_i y C_{i+1} de una ruta R_k son transferidos a una ruta R_l . Para el siguiente caso, los clientes 3 y 6 que se encuentran al inicio de la ruta 2, son transferidos al inicio de la ruta 3.

$$R = \begin{bmatrix} 1 & 2 \\ 3 & 6 & 4 & 5 \\ 7 & 8 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 3 & 6 & 7 & 8 \end{bmatrix}$$

La Figura 3.4 ilustra gráficamente el procedimiento Shift(2,0).

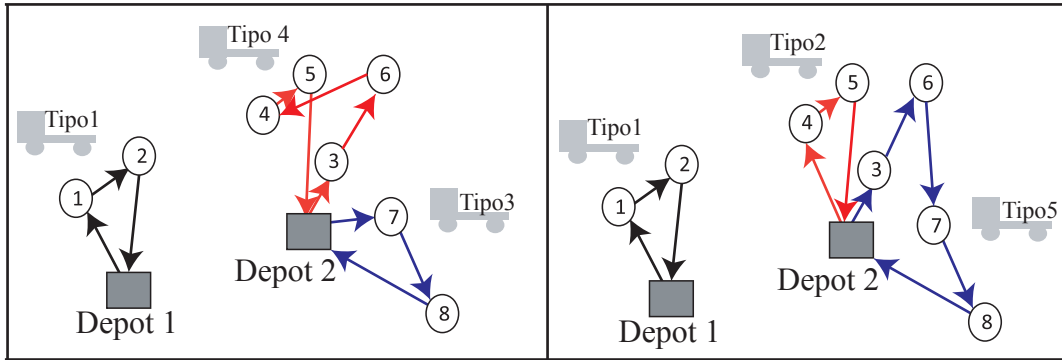


Figura 3.4: Estrategia Shift (2,0)

Shift (3,0)

Tres clientes consecutivos C_i , C_{i+1} y C_{i+2} son transferidos desde una ruta R_k a otra R_l . A continuación se muestra como los clientes 3, 6 y 7 que se encuentran al inicio de la ruta 3, son transferidos al final de la ruta 2.

$$R = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 3 & 6 & 7 & 8 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 4 & 5 & 3 & 6 & 7 \\ 8 \end{bmatrix}$$

El proceso Shift (3,0) se ilustra de forma detallada en la figura 3.5.

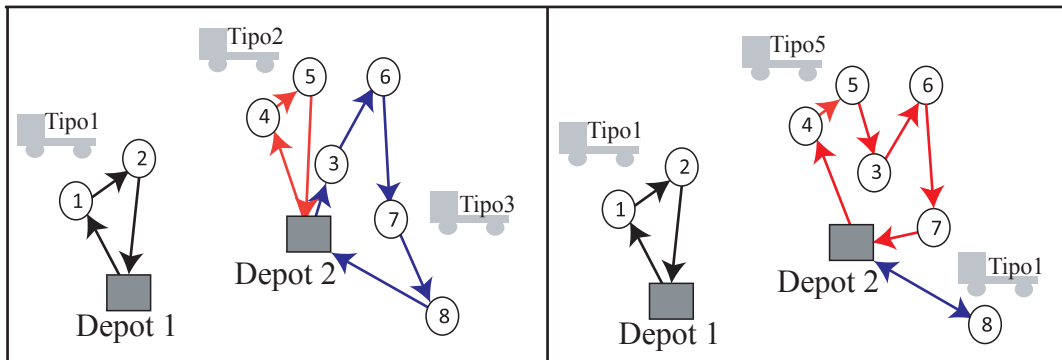


Figura 3.5: Estrategia Shift (3,0)

Swap (1,1)

Un cliente C_i de una ruta R_k es intercambiado con un cliente C_j de una ruta R_l . Para el siguiente caso, el cliente 4 quien pertenece a la ruta 1, es intercambiado con el cliente 2 de la ruta 2.

$$R = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 & 7 & 8 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 3 & 6 & 7 & 8 \end{bmatrix}$$

La Figura 3.6 ilustra de manera gráfica las modificaciones sobre cada ruta al realizar el procedimiento Swap (1,1).

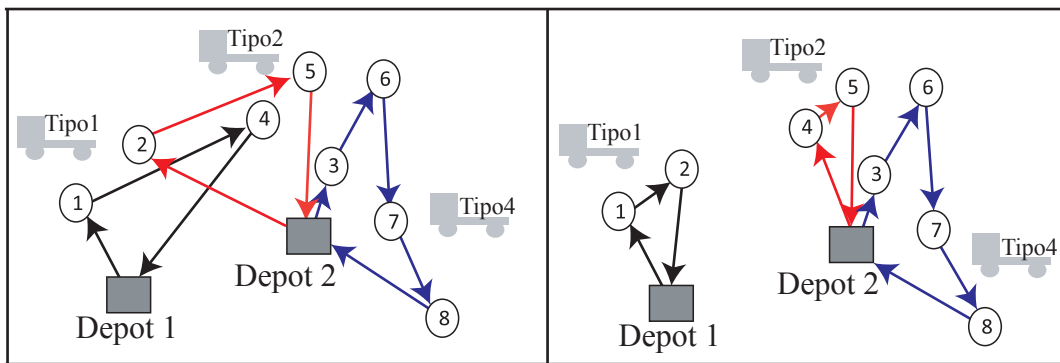


Figura 3.6: Estrategia Swap (1,1)

Swap (2,1)

Dos clientes consecutivos C_i y C_{i+1} de una ruta R_k son intercambiados por un cliente C_j de otra ruta R_l . El ejemplo muestra como los clientes 3 y 6 de la ruta 2 son intercambiados por el cliente 5 de la ruta 3.

$$R = \begin{bmatrix} 1 & 2 \\ 3 & 6 & 4 \\ 5 & 7 & 8 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 5 & 4 \\ 3 & 6 & 7 & 8 \end{bmatrix}$$

La Figura 3.7 ilustra el movimiento Swap (2,1) realizado en el ejemplo anterior.

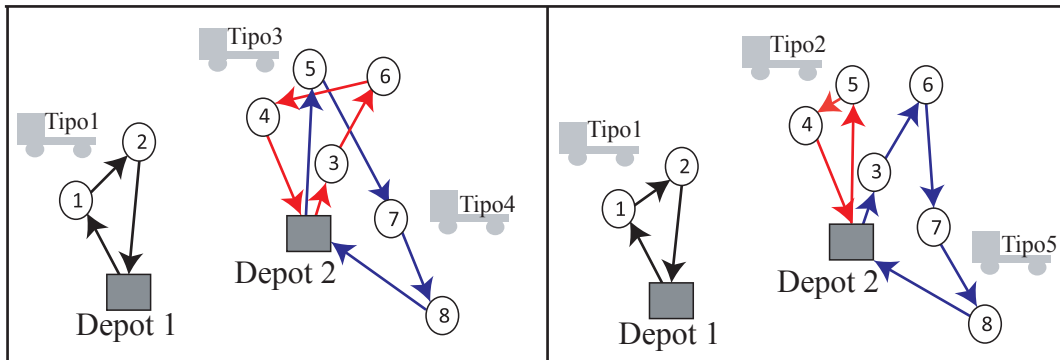


Figura 3.7: Estrategia Swap (2,1)

Swap (2,2)

Dos clientes consecutivos C_i y C_{i+1} de una ruta R_k son intercambiados con dos clientes consecutivos C_j y C_{j+1} pertenecientes a otra ruta R_l . El siguiente caso muestra que los clientes 2 y 5 de la ruta 2 son intercambiados con los clientes 3 y 6 de la ruta 3.

$$R = \begin{bmatrix} 1 \\ 2 \ 5 \ 7 \ 8 \\ 3 \ 6 \ 4 \end{bmatrix} \rightarrow R = \begin{bmatrix} 1 \\ 3 \ 6 \ 7 \ 8 \\ 2 \ 5 \ 4 \end{bmatrix}$$

A continuación se presenta gráficamente el movimiento Swap (2,2) para el caso presentado anteriormente.

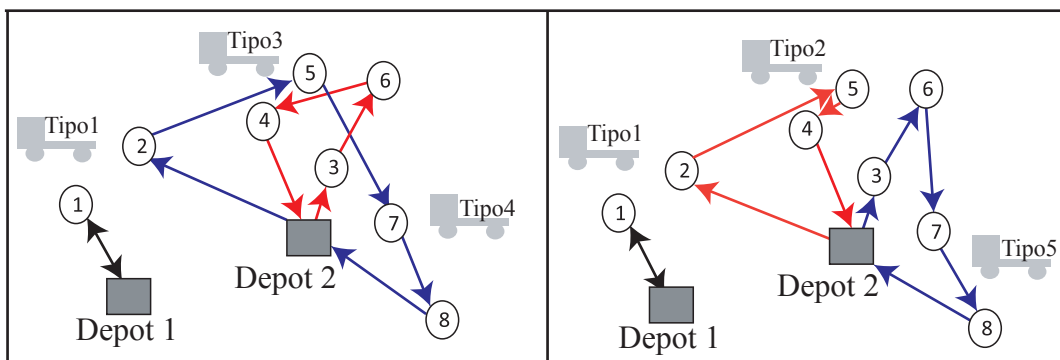


Figura 3.8: Estrategia Swap (2,2)

3.6.2. Búsqueda local intra-rutas

Dos estrategias de búsqueda local son consideradas como estructuras de vecindad para realizar movimientos en una misma ruta, cada una de ellas se describe a continuación:

Rotación

Como su nombre lo indica, esta estructura consiste en rotar la secuencia que representa una ruta R_K buscando encontrar una nueva ruta cuyo cliente inicial y final estén cerca a uno de los depósitos, es decir, la estrategia permite encontrar la mejor secuencia para cada ruta y a su vez el mejor depósito desde donde se deben atender los clientes que pertenecen a la ruta R_K . Para ello es necesario evaluar todas las secuencias resultantes de dicha rotación. En el siguiente ejemplo, el proceso de rotación se lleva a cabo sobre la ruta 2, de la cual se obtienen cuatro nuevas secuencias, suponiendo que la mejor secuencia para dicha ruta es $R_2 = [4 \ 5 \ 6 \ 3]$, también es posible determinar que el depósito 2 es el más conveniente para atender la nueva ruta. La figura 3.9 ilustra gráficamente los movimientos realizados sobre la ruta 2 mediante el proceso de rotación.

$$R = \begin{bmatrix} 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 \end{bmatrix} \quad \begin{matrix} 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 3 \\ 5 & 6 & 3 & 4 \\ 6 & 3 & 4 & 5 \end{matrix} \rightarrow R = \begin{bmatrix} 1 & 2 \\ 4 & 5 & 6 & 3 \\ 7 & 8 \end{bmatrix}$$

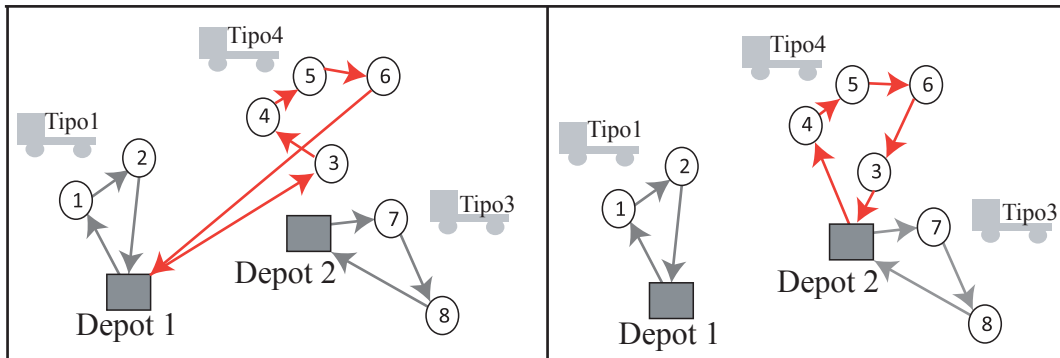


Figura 3.9: Rotación

2-Opt

Este movimiento busca que cada ruta tenga una forma radial, es decir, evita los cruces entre los arcos de una misma ruta. El método consiste en eliminar dos aristas no consecutivas $A(i, i + 1)$ y $A(j, j + 1)$ y agregar dos aristas nuevas $A(i', i' + 1)$ y $A(j', j' + 1)$, de tal manera que se construya una nueva ruta. En la Figura 3.10 se observa que las aristas $A(6, 4)$ y $A(5, D_2)$ son eliminadas y dos nuevas aristas son agregadas $A(D_2, 4)$ y $A(5, 6)$ formando una nueva ruta radial.

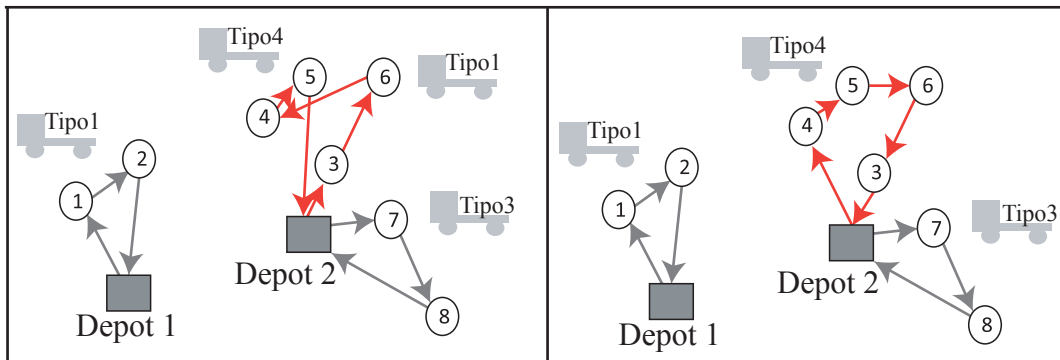


Figura 3.10: Estrategia 2opt

3.7. Ingreso a la población

Cuando un nuevo individuo es obtenido a través de los operadores genéticos de selección, recombinación y mutación, es necesario determinar si este nuevo individuo puede hacer parte de la población actual. Los pasos para realizar la sustitución son los siguientes:

- i* Verificar que el valor de la función objetivo del nuevo individuo sea menor que la función objetivo del peor individuo en la población actual.
- ii* Verificar que el tour gigante que representa el individuo no exista en la población actual; es decir se requiere como mínimo un grado de diversidad en cada secuencia de la población.

Si las dos condiciones descritas anteriormente se cumplen, entonces, el nuevo individuo entra a la población reemplazando al individuo de peor calidad.

3.8. Algoritmo general de la metodología propuesta

El algoritmo 1 presenta de manera general cada uno de los pasos de la metodología propuesta para solucionar el problema de ruteo de vehículos con múltiples depósitos y flota heterogénea, el cual se denominada Algoritmo Genético de Chu-Beasley Modificado.

Data: clientes n , depósitos m , tipos de vehículos K , demandas q_i , Capacidades Q_k , Tamaño de la población N , $P \leftarrow \{\emptyset\}$, $Sol \leftarrow \{\emptyset\}$, cantidad de iteraciones.

Result: Conjunto de rutas para el MDHFVRP

while $|P| \neq N$ **do**

$P \leftarrow P \cup$ Secuencias obtenidas de la heurística LHK, vecino más cercano y aleatorias;

end

for $i \leftarrow 1 \dots N$ **do**

$Sol \leftarrow Sol \cup$ División de cada secuencia mediante el método híbrido;

end

while $iter \leq (cantidad\ de\ iteraciones)$ **do**

$p_1, p_2 \leftarrow$ Selección por torneo (P);

if $((iter \bmod 2) == 0)$ **then**

$h \leftarrow$ Recombinación PMX(p_1, p_2);

else

$h \leftarrow$ Recombinación OBX(p_1, p_2);

end

$h \leftarrow$ Búsqueda local inter-ruta(h);

$h \leftarrow$ Búsqueda local intra-ruta(h);

if (h cumple criterio de diversidad) **then**

 Ingresar h a la población actual;

else

 Rechazar h ;

end

$iter ++$

end

return The best solution (P);

Algorithm 1: AGCBM para el MDHFVRP

Capítulo 4

Resultados

Gran cantidad de experimentos computacionales fueron realizados para analizar el comportamiento y la contribución del AGCBM, comparándolo con los métodos existentes en la literatura especializada para el problema específico MDHFVRP.

Las pruebas son realizadas en las mismas instancias utilizadas por Salhi y Sari [7], Salhi *et al* [21] y Vidal *et al* [22]. Estos casos de pruebas presentan una combinación de clientes distribuidos gráficamente de forma uniforme y usuarios en forma de clúster. La cantidad de clientes en cada instancia varía entre 50 y 360. La cantidad de depósitos por cada instancia va desde 2 hasta 9. Las instancias fueron adaptadas de las instancias típicas para MDVRP, tal como lo presenta Salhi y Sari [7], se toma la misma localización de los usuarios y depósitos, la misma demanda y se generan cinco tipos de vehículos v_k de la siguiente manera: $Q_k = (0,4 + 0,2k) Q^*$, $F_k = 70 + 10k$ y $\alpha_k = 0,7 + 0,1k$ con $k \in \{1, 2, \dots, 5\}$. Q^* hace referencia la capacidad del vehículo en la instancia original. En todas las pruebas realizadas se considera una flota de vehículos ilimitada.

En el anexo A se presenta una instancia original de 50 clientes y 4 depósitos para el MDVRP en el formato requerido por la metodología propuesta, además se muestra el cálculo de los tipos 5 tipos de vehículos junto con los costos fijos y variables asociados para transformar la instancia en un MDHFVRP.

La codificación del AGCBM fue realizada en el lenguaje de programación *C++* bajo el entorno *Visual-Studio 2010* junto con el solver *CPLEX 12.5* bajo la librería *CONCERT*. El sistema operativo

utilizado fue *Windows 7*, en un computador con procesador *Core 2 Duo* de *2.7 GHz* y memoria RAM de *4 GB*. A continuación se muestra la comparación de los resultados obtenidos con tres métodos existentes en la literatura especializada (Salhi y Sari en [7], Salhi *et al* en [21] y Vidal *et al* en [22]).

4.1. *SS97 Vs AGCBM*

La Tabla 4.1 presenta la comparación entre la metodología *SS97* presentada por Salhi y Sari en [7] y el algoritmo propuesto en este trabajo (*AGCBM*) para 14 instancias de la literatura especializada. Las tres primeras columnas corresponden a las características de las instancias, en su orden se encuentra: número de la instancia, la cantidad de clientes (n) y la cantidad de depósitos de cada caso de prueba (m). Las siguientes dos columnas presentan los resultados obtenidos con cada una de las metodologías. La columna final presenta el Gap de la solución encontrada con la metodología propuesta con respecto del algoritmo *SS97*. El signo negativo indica que la solución encontrada por el *AGCBM* es mejor que la reportada, mientras que el signo positivo indica lo contrario.

Claramente se puede observar que en 11 de las 14 instancias la metodología propuesta encuentra mejores resultados en comparación con los reportados por el algoritmo *SS97*. Las 3 instancias restantes presentan un Gap menor al 2.2%.

Tabla 4.1: Comparación de las metodologías *SS97* y *AGCBM*

| Instancia | n | m | SS97 | Propuesta (AGCBM) | Gap (%) |
|-----------|-----|---|--------|-------------------|---------|
| 1 | 50 | 4 | 1526,7 | 1513,12 | -0,8895 |
| 2 | 50 | 4 | 992,8 | 981,809 | -1,1071 |
| 3 | 75 | 2 | 1611,1 | 1605,77 | -0,3308 |
| 4 | 100 | 2 | 2361,9 | 2338,43 | -0,9937 |
| 5 | 100 | 2 | 1498,4 | 1491,83 | -0,0044 |
| 6 | 100 | 3 | 2277,5 | 2252,54 | -1,0959 |
| 7 | 100 | 4 | 2297,1 | 2294,79 | -0,1006 |

continúa ...

Tabla 4.1: Comparación de las metodologías SS97 y AGCBM
(continuación...)

| Instancia | n | m | SS97 | Propuesta (AGCBM) | Gap (%) |
|-----------|-----|---|---------|-------------------|---------|
| 8 | 80 | 2 | 2108,2 | 2079,98 | -1,3386 |
| 9 | 80 | 2 | 2126,8 | 2116,78 | -0,4711 |
| 10 | 80 | 2 | 2160,12 | 2160,46 | +0,0157 |
| 11 | 160 | 4 | 4116,2 | 4056,93 | -1,4399 |
| 12 | 240 | 6 | 6217 | 6193,38 | -0,3799 |
| 13 | 249 | 3 | 6211,4 | 6347,27 | +2,1874 |
| 14 | 360 | 9 | 9184,6 | 9353 | +1,8335 |

4.2. *SIW13-VNS1-VNS2 Vs AGCBM*

La Tabla 4.2 presenta la comparación entre la metodología *SIW13-VNS1-VNS2* presentada por Salhi *et al* en [21] y el algoritmo propuesto en este trabajo (*AGCBM*) para 14 instancias de la literatura especializada. El método *SIW13-VNS1-VNS2* involucra dos algoritmos diferentes, el primero denominado VNS1 y el segundo VNS2. Las tres primeras columnas corresponden a los datos de cada instancias, en su orden se encuentra número de la instancia, la cantidad de clientes (n) y la cantidad de depósitos de cada caso de prueba (m). Las siguientes tres columnas presentan los resultados obtenidos con cada una de las metodologías. La columna final presenta el Gap de la solución encontrada con la metodología propuesta con respecto a los dos algoritmos del trabajo presentado por Shali. El signo negativo indica que la solución encontrada por el *AGCBM* es mejor que la reportada, mientras que el signo positivo indica lo contrario.

En la Tabla 4.2 se observa que en 4 de las 14 instancias, las soluciones encontradas son mejores que las reportadas por los algoritmos *SIW13-VNS1-VNS2*. Para el resto de las instancias los Gaps varían entre 0.25 % y 4.7 %.

Tabla 4.2: Comparación de las metodologías SIW13-VNS1-VNS2 y AGCBM

| Instancia | n | m | SIW13 VNS1 | SIW13 VNS2 | Propuesta | Gap (%) |
|-----------|-----|---|---------------|---------------|-----------|---------|
| 1 | 50 | 4 | 1503,3 | 1499,3 | 1513,12 | +0,6532 |
| 2 | 50 | 4 | 974,9 | 984,5 | 981,809 | -0,2733 |
| 3 | 75 | 2 | 1571,2 | 1588,4 | 1605,77 | +1,0935 |
| 4 | 100 | 2 | 2321,9 | 2313,7 | 2338,43 | +0,7119 |
| 5 | 100 | 2 | 1459,3 | 1466,9 | 1491,83 | +1,6995 |
| 6 | 100 | 3 | 2224 | 2246,9 | 2252,54 | +0,2510 |
| 7 | 100 | 4 | 2249,2 | 2256,4 | 2294,79 | +1,7014 |
| 8 | 80 | 2 | 2085,8 | 2076,2 | 2079,98 | -0,2790 |
| 9 | 80 | 2 | 2118,6 | 2096,4 | 2116,78 | -0,0859 |
| 10 | 80 | 2 | 2188,6 | 2193,3 | 2160,46 | -1,2857 |
| 11 | 160 | 4 | 4001,8 | 4024,9 | 4056,93 | +0,7958 |
| 12 | 240 | 6 | 5994,6 | 5970,5 | 6193,38 | +3,3159 |
| 13 | 249 | 3 | 6094,7 | 6068,8 | 6361,85 | +4,1441 |
| 14 | 360 | 9 | 8935,6 | 8883,1 | 9353 | +4,6712 |

4.3. HGSADC Vs AGCBM

La Tabla 4.3 presenta la comparación entre la metodología *HGSADC* presentada por Vidal *et al* en [22]) y el algoritmo propuesto en este trabajo (*AGCBM*) para 14 instancias de la literatura especializada. Las tres primeras columnas corresponden a los datos de cada instancias, en su orden se encuentra número de la instancia, la cantidad de clientes (n) y la cantidad de depósitos de cada caso de prueba (m). Las siguientes dos columnas presentan los resultados obtenidos con cada una de las metodologías. La columna final presenta el Gap de la solución encontrada con la metodología propuesta con respecto a la metodología presentada por Vidal.

La Tabla 4.3 muestra que la metodología propuesta encuentra soluciones cercanas a las reportadas

por la metodología HGSADC, cuyos Gaps varían entre 0,0157% y 7,4%

Tabla 4.3: Comparación de las metodologías HGSADC y AGCBM

| Instancia | n | m | HGSADC | Propuesta | Gap (%) |
|-----------|-----|---|---------|-----------|---------|
| 1 | 50 | 4 | 1477,73 | 1513,12 | 2,3949 |
| 2 | 50 | 4 | 957,73 | 981,809 | 2,5142 |
| 3 | 75 | 2 | 1569,67 | 1605,77 | 2,2998 |
| 4 | 100 | 2 | 2292,64 | 2338,43 | 1,9973 |
| 5 | 100 | 2 | 1453,64 | 1491,83 | 2,6272 |
| 6 | 100 | 3 | 2208,66 | 2252,54 | 1,9867 |
| 7 | 100 | 4 | 2198,91 | 2294,79 | 4,3603 |
| 8 | 80 | 2 | 2072,18 | 2079,98 | 0,3764 |
| 9 | 80 | 2 | 2096,39 | 2116,78 | 0,9726 |
| 10 | 80 | 2 | 2160,12 | 2160,46 | 0,0157 |
| 11 | 160 | 4 | 3973,47 | 4056,93 | 2,1004 |
| 12 | 240 | 6 | 5887,43 | 6193,38 | 5,1967 |
| 13 | 249 | 3 | 5998,7 | 6361,85 | 6,0538 |
| 14 | 360 | 9 | 8709,26 | 9353 | 7,3914 |

Capítulo 5

Conclusiones, recomendaciones y trabajos futuros

5.1. Conclusiones

- En este trabajo se presentó la implementación de un Algoritmo Genético de Chu-Beasley Modificado para solucionar el problema de ruteamiento de vehículos con múltiples depósitos y flota heterogénea que incluye costos fijos y variables. La metodología combina operadores genéticos con estrategias de búsqueda local, que permiten explorar el espacio de solución de manera adecuada para encontrar soluciones atractivas al problema de MDHFVRP.
- Como etapa de inicialización se presenta un procedimiento híbrido que permite encontrar soluciones de buena calidad y con un buen grado de diversidad en la población inicial. El procedimiento combina heurísticas para poblaciones basadas en permutaciones junto con un proceso de división adaptado desde el problema clásico de VRP, con lo cual una nueva codificación para el problema de MDHFVRP es considerada.
- Diferentes estrategias de vecindad inter-rutas e intra-rutas permiten realizar una intensificación en el espacio de búsqueda, de tal manera que el algoritmo avance hacia regiones de mejor calidad en cada generación de la metodología propuesta.
- Para validar la metodología propuesta, 14 instancias de prueba de la literatura con tamaños

entre 50 y 360 clientes son utilizadas para realizar comparaciones con otras metodologías existentes. Los resultados obtenidos muestran la competitividad de este método para solucionar el problema de MDHFVRP. Los Gaps obtenidos con respecto a los mejores resultados reportados en la literatura especializada demuestran la notable contribución de este trabajo en futuras investigaciones dentro del grupo de investigación.

5.2. Recomendaciones y trabajos futuros

- Por ser este un algoritmo basado en poblaciones, nuevos objetivos pueden ser incluidos y dar paso a una variante Multi-Objetivo del problema de ruteo de vehículos. Uno de los nuevos objetivos a considerar es la reducción de los efectos ambientales provenientes de los vehículos de carga.
- Adaptar los datos de entrada, para hacer esta una metodología general, la cual permita solucionar diferentes versiones del problema de ruteamiento de vehículos, como VRP, CVRP, MDVRP entre otros.
- Se recomienda combinar las estrategias de búsqueda local presentadas en este trabajo con algoritmos metaheurísticos basados en trayectoria, como Búsqueda Tabú, con el fin de realizar comparaciones que permitan determinar que tipos de algoritmos resultan ser más adecuados a la hora de resolver este tipo de problemas.
- Nuevas estrategias de búsqueda local pueden ser incluidas, con el fin de explorar nuevas regiones en el espacio de solución, de tal forma que se encuentren mejores soluciones de manera eficiente.

Bibliografía

- [1] P. Toth and D. Vigo., *The Vehicle Routing Problem*. SIAM, 2002, ch. An overview of vehicle routing problems, pp. 1–26.
- [2] G. Dantzig and J. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, pp. 80–91, 1959.
- [3] P. J. Cassidy and H. S. Bennett, “A multi-depot vehicle scheduling system,” *Operational Research Quarterly*, vol. 23 (2), pp. 151–163, 1972.
- [4] A. Wren and A. Holliday, “Computer scheduling of vehicles from one or more depots to a number of delivery points,” *Operational Research Quarterly*, vol. 23 (3), pp. 333–344, 1972.
- [5] G. Clarke and J. Wright, “Scheduling of vehicles from a central depot to a number of delivery point,” *Operation Research*, vol. 12, pp. 568–581, 1964.
- [6] J. Renaud, G. Laporte, and F. F. Boctor, “A tabu search heuristic for the multi-depot vehicle routing problem,” *Computers Operations Researchs*, vol. 23 (3), pp. 229–235, 1996.
- [7] S. Salhi and M. Sari, “A multi-level composite heuristic for the multi-depot vehicle fleet mix problem,” *European Journal of Operational Research*, vol. 103, pp. 95–112, 1997.
- [8] G. Nagy and S. Salhi, “Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries,” *European Journal of Operational Research*, vol. 162, pp. 126–141, 2005.
- [9] B. Crevier, J.-F. Cordeau, and G. Laporte, “The multi-depot vehicle routing problem with inter-depot routes,” *European Journal of Operational Research*, vol. 176, pp. 756–773, 2007.

- [10] W. Ho, G. T. Ho, P. Jib, and H. W. Laub, “A hybrid genetic algorithm for the multi-depot vehicle routing problem,” *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 548–557, 2008.
- [11] M. Mirabi, S. F. Ghomi, and F. Jolai, “Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem,” *Robotics and Computer-Integrated Manufacturing*, vol. 26, pp. 564–569, 2010.
- [12] R. Baldacci and M. Dell’Amico, “Heuristic algorithms for the multi-depot ring-star problem,” *European Journal of Operational Research*, vol. 203, pp. 270–281, 2010.
- [13] N. Aras, D. Aksen, and M. T. Tekin, “Selective multi-depot vehicle routing problem with pricing,” *Transportation Research Part C*, vol. 19, pp. 866–884, 2011.
- [14] D. Gulczynski, B. Golden, and E. Wasil, “The multi-depot split delivery vehicle routing problem: An integer programming-based heuristic, new test problems, and computational results,” *Computers & Industrial Engineering*, vol. 61, pp. 794–804, 2011.
- [15] G. N. Yücenur and N. C. Demirel, “A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem,” *Expert Systems with Applications*, vol. 38, pp. 11 859–11 865, 2011.
- [16] Y. Xu, L. Wang, and Y. Yang, “A new variable neighborhood search algorithm for the multi depot heterogeneous vehicle routing problem with time windows,” *Electronic Notes in Discrete Mathematics*, vol. 39, pp. 289–296, 2012.
- [17] Y. Kuo and C.-C. Wang, “A variable neighborhood search for the multi-depot vehicle routing problem with loading cost,” *Expert Systems with Applications*, vol. 39, pp. 6949–6954, 2012.
- [18] T.-H. Wu, C. Low, and J.-W. Bai, “Heuristic solutions to multi-depot location-routing problems,” *Computers & Operations Research*, vol. 29, pp. 1393–1415, 2002.
- [19] J. W. Escobar, R. Linfati, and Paolo.Toth, “A two-phase hybrid heuristic algorithm for the capacitated location-routing problem,” *Computers & Operations Research*, vol. 40, pp. 70–79, 2013.

- [20] A. Bettinelli, A. Ceselli, and G. Righini, “A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows,” *Transportation Research Part C*, vol. 19, pp. 723–740, 2011.
- [21] S. Salhi, A. Imran, and N. A. Wassan. (2013) The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*. [Online]. Available: <http://dx.doi.org/10.1016/j.cor.2013.05.011>
- [22] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, “Implicit depot assignments and rotations in vehicle routing heuristics,” *European Journal of Operational Research*, vol. 237, pp. 15–28, 2014.
- [23] P. Toth and D. Vigo, *The Vehicle Routing Problem*, M. on Discrete Mathematics and Applications, Eds. SIAM, 2002.
- [24] R.A.Gallego, A.H.Escobar, and E.M.Toro, *Técnicas Metaheurísticas de Optimización*, segunda ed. Universidad Tecnológica de Pereira, 2008, ch. Algoritmo Genético, pp. 138–150.
- [25] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP,” *Proceedings of the romanian academy, series A*, vol. 117, pp. 81–86, 2002.
- [26] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, pp. 498–516, 1973.
- [27] C. Prins, “Two memetic algorithms for heterogeneous fleet vehicle routing problem,” *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 916–928, 2009.
- [28] D. Goldberg and R. L. a. Lingle, “The traveling salesman problem,” *Proc. First Int. Conf. Genetic Algorithms and their Applications*, pp. 154–159, 1985.
- [29] G. Syswerda, “Schedule optimization using genetic algorithms,” *In A Handbook of Genetic Algorithms (Edited by L. Davis)*, pp. 332–349, 1991.

- [30] I. M. Oliver, D. J. Smith, and J. R. C. Holland, “A study of permutation crossover operators on the traveling salesman problem,” *Proc. Second Int. Conf. Genetic Algorithms and their Applications*, pp. 224–230, 1987.
- [31] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, “A hybrid algorithm for the heterogeneous fleet vehicle routing problem,” *European Journal of Operational Research*, vol. 221, no. 2, pp. 285–295, 2012.

Apéndices

Apéndice A

Instancia

En este anexo se muestra el formato necesario para ingresar los datos de cada instancia de prueba en la metodología propuesta, para ello se muestra una instancia de MDHVRP que consta de 50 clientes y 4 depósitos y su respectiva modificación. Cada caso se encuentra en un archivo de texto plano *.txt*, el cual consta de las siguientes partes.

A.1. Encabezado

La primera fila del archivo de texto consta de la siguiente información: Las tres primeras columnas corresponden información original de la instancia. La primera columna indica la cantidad de clientes, la segunda muestra el número de depósitos y la tercera hace referencia a la capacidad de cada vehículo para el caso homogéneo (MDVRP). Finalmente se muestra la cantidad de tipo de vehículos que serán empleados en el caso MDHFVRP. La Tabla A.1 muestra la información descrita anteriormente.

Tabla A.1: Datos iniciales

| n | m | Q^* | Tipo de Vehículos |
|-----|-----|-------|-------------------|
| 50 | 4 | 80 | 5 |

A.2. Coordenadas y datos de carga

Las siguientes filas del archivo, corresponden a la identificación de cada cliente, las coordenadas “x” e “y” y la demanda respectiva de cada cliente, tal como se muestra en la Tabla A.2.

Tabla A.2: Coordenadas

| id | x | y | Demanda | id | x | y | Demanda | id | x | y | Demanda |
|----|----|----|---------|----|----|----|---------|----|----|----|---------|
| 1 | 37 | 52 | 7 | 19 | 13 | 13 | 9 | 37 | 32 | 22 | 9 |
| 2 | 49 | 49 | 30 | 20 | 57 | 58 | 28 | 38 | 45 | 35 | 15 |
| 3 | 52 | 64 | 16 | 21 | 62 | 42 | 8 | 39 | 59 | 15 | 14 |
| 4 | 20 | 26 | 9 | 22 | 42 | 57 | 8 | 40 | 5 | 6 | 7 |
| 5 | 40 | 30 | 21 | 23 | 16 | 57 | 16 | 41 | 10 | 17 | 27 |
| 6 | 21 | 47 | 15 | 24 | 8 | 52 | 10 | 42 | 21 | 10 | 13 |
| 7 | 17 | 63 | 19 | 25 | 7 | 38 | 28 | 43 | 5 | 64 | 11 |
| 8 | 31 | 62 | 23 | 26 | 27 | 68 | 7 | 44 | 30 | 15 | 16 |
| 9 | 52 | 33 | 11 | 27 | 30 | 48 | 15 | 45 | 39 | 10 | 10 |
| 10 | 51 | 21 | 5 | 28 | 43 | 67 | 14 | 46 | 32 | 39 | 5 |
| 11 | 42 | 41 | 19 | 29 | 58 | 48 | 6 | 47 | 25 | 32 | 25 |
| 12 | 31 | 32 | 29 | 30 | 58 | 27 | 19 | 48 | 25 | 55 | 17 |
| 13 | 5 | 25 | 23 | 31 | 37 | 69 | 11 | 49 | 48 | 28 | 18 |
| 14 | 12 | 42 | 21 | 32 | 38 | 46 | 12 | 50 | 56 | 37 | 10 |
| 15 | 36 | 16 | 10 | 33 | 46 | 10 | 23 | 51 | 20 | 20 | 0 |
| 16 | 52 | 41 | 15 | 34 | 61 | 33 | 26 | 52 | 30 | 40 | 0 |
| 17 | 27 | 23 | 3 | 35 | 62 | 63 | 17 | 53 | 50 | 30 | 0 |
| 18 | 17 | 33 | 41 | 36 | 63 | 69 | 6 | 54 | 60 | 50 | 0 |

A.3. Tipos de vehículos, Costos fijos y variables

A partir de la capacidad Q^* dada en la Tabla A.1, se calculan 5 tipos de vehículos diferentes por medio de la siguiente ecuación:

$$Q_k = (0,4 + 0,2k) Q^* \quad (\text{A.1})$$

También se calculan los valores fijos (F_k) y variables α_k correspondiente a cada tipo de vehículo mediante las siguientes ecuaciones:

$$F_k = 70 + 10k \quad (\text{A.2})$$

$$\alpha_k = 0,7 + 0,1k \quad (\text{A.3})$$

$$k \in \{1, 2, \dots, 5\}$$

La Tabla A.3 muestra el calculo de los 5 tipos de vehiculos con sus valores fijos y variables.

Tabla A.3: Tipos de vehículos

| Vehiculo | Q | alpha | F |
|----------|-----|-------|-----|
| 1 | 48 | 0.8 | 80 |
| 2 | 64 | 0.9 | 90 |
| 3 | 80 | 1.0 | 100 |
| 4 | 96 | 1.1 | 110 |
| 5 | 112 | 1.2 | 120 |