

**APROXIMACIÓN AL PROBLEMA DE RUTEO DE VEHÍCULOS CON MÚLTIPLES  
DEPÓSITOS Y FLOTA HETEROGÉNEA UTILIZANDO UN ALGORITMO  
METAHEURÍSTICO**

**ANDRÉS FERNANDO PANTOJA VERGARA**

**TRABAJO PARA OPTAR POR EL TÍTULO DE MAGÍSTER EN INGENIERÍA INDUSTRIAL**

**DIRECTOR DE TESIS DE MAESTRÍA  
NUBIA MILENA VELASCO RODRÍGUEZ, Ph.D.**

**UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL  
BOGOTÁ, COLOMBIA  
2014**

## Tabla de contenido

1. Introducción .....	5
2. Descripción del problema .....	8
2.1. Objetivos del MDHFVRP .....	9
2.2. Formulación matemática .....	9
3. Método de solución .....	12
3.1 Búsqueda armónica.....	12
3.2 Intercambios 2-OPT.....	15
3.3 Método de solución aplicado.....	15
4. Test computacionales .....	22
5. Conclusiones.....	24
Bibliografía .....	25

## Lista de gráficos

Gráfico 1. Ejemplo de ruteo de vehículos multi-depósito. Fuente: (Surekha, 2011).....	6
Grafico 2. Modelo matemático. Adaptado de (Salhi, Imran, & Wassan, 2013).....	11
Gráfico 3. Búsqueda armónica. Fuente: (Geem, Lee, & Park, 2005).....	13
Gráfico 4. Comparación improvisación musical con problema de optimización.....	13
Grafico 5. Ejemplo datos de entrada del modelo.....	16
Gráfico 6. Estructura de la matriz armónica.....	17
Gráfico 7. Pseudocódigo inicialización de la memoria armónica.....	18
Grafico 8. 2-OPT, intercambio de arcos.....	18
Gráfico 9. Pseudocódigo de intercambios aleatorios 2-OPT.....	19
Gráfico 10. Pseudocódigo generación de nuevos vectores.....	20
Gráfico 11. Búsqueda armónica en problema de ruteo.....	21
Gráfico 12. Ejemplo de evolución del costo por iteración.....	22

## Lista de tablas

Tabla 1. Revisión bibliográfica.....	7
Tabla 2. Resultados del modelo .....	23

# APROXIMACIÓN AL PROBLEMA DE RUTEO DE VEHÍCULOS CON MÚLTIPLES DEPÓSITOS Y FLOTA HETEROGÉNEA UTILIZANDO UN ALGORITMO METAHEURÍSTICO

B. Sc. Pantoja Andrés, Ph.D. Velasco Nubia

Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia.

## Resumen:

En este trabajo se presenta una aproximación a la resolución del problema de ruteo de vehículos con múltiples depósitos y flota heterogénea con una distribución fija de vehículos mediante el uso de la metaheurística de la búsqueda armónica combinada con un método de afinamiento de rutas por medio de intercambios 2-OPT. El algoritmo generado en esta tesis se compara en términos de eficiencia computacional con un modelo exacto utilizando catorce instancias.

**Palabras clave:** Problema de ruteo de vehículos con múltiples depósitos y flota heterogénea, búsqueda armónica, VRP, TSP, 2-OPT, optimización.

**Keywords:** Multi depot and heterogeneous fleet vehicle routing problem, harmony search, VRP, TSP, 2-OPT, optimization.

## 1. Introducción

Las organizaciones se encuentran enmarcadas en un contexto cada vez más competitivo y globalizado, donde los clientes tienen mayores expectativas con respecto a la calidad de los productos y servicios que adquieren. El flujo cambiante del entorno empresarial hace que las decisiones que tomen las empresas sean cada vez más complejas y un problema que deben resolver es el del transporte de los productos, servicios, personas e información (Yepes, 2002). Además, el transporte es un rubro fundamental en la planeación económica de cualquier empresa puesto que representa una parte importante del costo final del producto comercializado o del servicio brindado (Zabala, 2006).

Lo expuesto anteriormente evidencia la importancia que tiene en términos económicos y de competitividad empresarial la optimización de las rutas de transporte, sin tener en cuenta el impacto medioambiental que genera la utilización de combustibles fósiles.

El problema de ruteo de vehículos (VRP, Vehicle Routing Problem) es uno de los más estudiados problemas de optimización combinatoria y a este le comprende diseñar óptimamente las rutas que serán utilizadas por una flota de vehículos que servirá a un conjunto de clientes (Baldacci, 2014).

Es conocido que el problema de ruteo de vehículos es de complejidad NP completa porque el esfuerzo computacional requerido para resolver este problema aumenta exponencialmente con su tamaño (Salah & Badr, 2012). Archetti (2011) prueba por medio de estructuras graficas especiales que el VRP con flota limitada e ilimitada tiene una complejidad computacional completa.

El problema que se abordara en esta tesis es una de las variaciones del VRP clásico, específicamente el problema de ruteo de vehículos con múltiples depósitos y flota heterogénea, con vehículos fijos por deposito (MDHFVRPFD, Multi Depot Heterogeneous Fleet Vehicle Routing Problem with fixed distribution of vehicles). En este problema existe más de un centro de distribución o depósito, los cuales tienen la función de satisfacer la demanda de los clientes. Cada depósito tiene una flota fija o cantidad de vehículos limitada y de diferente tipo, la cual se usa para repartir los productos demandados por los clientes, teniendo en cuenta que cada uno de ellos tiene una posición espacial y una demanda conocida de antemano. Los vehículos pueden presentar diferencias en el tamaño de su capacidad de carga, en los costos fijos por su utilización y en el valor del costo variable por distancia recorrida. Cada vehículo debe comenzar y terminar su ruta en el mismo depósito, y cada cliente debe ser visitado una única vez.

En el gráfico 1 se puede apreciar un ejemplo de un problema con diez clientes, dos depósitos, dos vehículos en cada depósito y dos tipos de vehículo. Se puede ver que los clientes son visitados una sola vez y que hay vehículos con distinta capacidad de carga por lo que pueden satisfacer mayor o menor demanda.

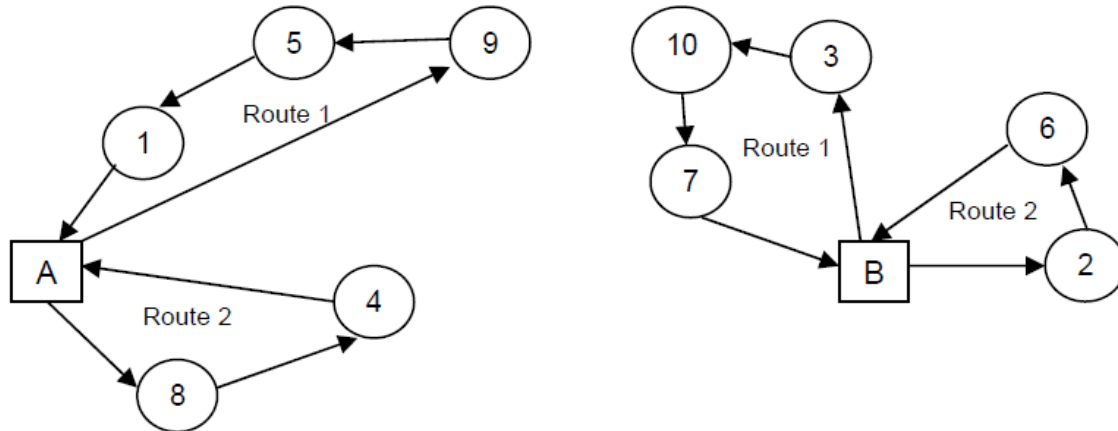


Gráfico 1. Ejemplo de ruteo de vehículos multi-depósito. Fuente: (Surekha, 2011)

En la tabla 1 se pueden ver los trabajos de investigación realizados para este problema o similares, donde utilizan diferentes estrategias de resolución incluyendo la metaheurística de la búsqueda armónica.

REFERENCIA	PROBLEMA	DETALLE
(Wei, Zhixiang, Qingquan, Shih-Lung, & Biyu, 2014)	Ruteo de vehículos con múltiples depósitos	Utilizan una metaheurística de dos niveles basada en diagramas de Voronoi, en el que primero asignan los clientes creando vecindarios con formas de anillos de Voronoi para posteriormente realizar el ruteo.
(Dondo, Méndez, & Cerdá, 2003)	Ruteo de vehículos con ventanas de tiempo, múltiples depósitos y flota heterogénea.	Resuelven el problema con métodos exactos y una estrategia de reducción del tamaño del problema con restricciones.
(Lim & Zhu, 2006)	Ruteo de vehículos con múltiples depósitos y vehículos fijos por depósitos. (MDVRPFD)	Inicialmente construye una solución con una heurística de inserción y posteriormente la mejora utilizando una metaheurística de recocido simulado.
(Yu, Yang, & Xie, 2011)	Ruteo de vehículos con múltiples depósitos.	Abordan el problema con una metaheurística de colonia de hormigas, utilizando un depósito virtual central al cual asignan a todos los vehículos.
(Salhi, Imran, & Wassan, 2013)	Ruteo de vehículos con múltiples depósitos y flota heterogénea ilimitada.	Los autores resuelven con una formulación matemática exacta y con un método metaheurístico de búsqueda por vecindarios variable (VNS)
(Geem, Lee, & Park, 2005)	Ruteo de transporte escolar con ventanas de tiempo utilizando búsqueda armónica.	Utilizan la metaheurística de búsqueda armónica para la resolución. Prueban una instancia con diez paradas de buses y un bus por depósito.
(Pichpibul & Kawtummachai, 2013)	Ruteo de vehículos capacitado	Inicialmente obtienen una solución por medio de la heurística de Clarke and Wright, esta alimenta la memoria armónica para posteriormente resolver el problema por medio de la metaheurística de búsqueda armónica
(Taha, Ayob, Ahmad, & Ahmad, 2013)	Ruteo de vehículos con ventanas de tiempo (VRPTW)	Investigan el rendimiento de la búsqueda armónica para resolver el VRPTW obteniendo resultados promisorios.

Tabla 1. Revisión bibliográfica. Fuente: Autor

Aunque se encuentran bastantes estudios de investigación sobre la resolución del problema de ruteo de vehículos, no se encuentra una referencia puntual donde trabajen en conjunto con las características de tener múltiples depósitos, de contar con una flota de vehículos heterogénea y con una distribución fija de vehículos para cada depósito. Entre los estudios existentes se encuentra la aplicación de la búsqueda armónica pero solamente ha sido utilizada en problemas básicos y nunca con las variaciones como las que se presentan en esta tesis, en toda la bibliografía consultada. Lo anterior genera el interés de evaluar la utilidad de usar esta metaheurística en problemas de ruteo de vehículos con mayor complejidad computacional.

## **2. Descripción del problema**

EL MDHFVRPFD es un problema no determinístico de tiempo polinómico completo. La complejidad NP-completa del MDHFVRPFD aumenta exponencialmente a medida que lo hace el número de clientes, depósitos y tipos de vehículo que tenga la instancia, por lo que se dificulta el desarrollo de métodos que resuelvan el problema de manera óptima en un tiempo razonable.

Este problema cuenta con las mismas cuestiones importantes por resolver que un MDVRP, esto es la asignación de clientes a los depósitos y el ruteo de los vehículos. La asignación de los clientes es fundamental ya que si se logra definir inicialmente que depósito atenderá a cada cliente, la complejidad del problema se convierte en la resolución de múltiples VRP de un único depósito. Para Wei et al. (2014) existen diferentes estrategias de asignación de los clientes en un MDVRP entre las cuales esta asignar cada cliente al depósito más cercano, formar clusters de clientes basados en su posición espacial y en organizar los clientes según se encuentren dentro o fuera de las fronteras de un radio de cercanía a los dos depósitos más cercanos. Para el ruteo de vehículos se busca utilizar técnicas que disminuyan el esfuerzo y el tiempo computacional del proceso de exploración de los vecindarios.

El problema de ruteo de vehículos con múltiples depósitos y flota heterogénea no ha sido muy estudiado a pesar de que tiene aplicabilidad en la industria, pues se ve reflejado en todas las empresas que transporten productos o servicios a través de diferentes tipos de vehículo y que cuenten con más de un centro de distribución. Además el problema abordado en este documento tiene en cuenta una flota fija de vehículos lo cual es más realista y de mayor aplicación en la toma de decisiones operativas puesto que se busca darle el mejor uso a la capacidad existente, lo anterior es diferente a una flota de vehículos ilimitada que es útil cuando se quiere dimensionar el tamaño de una nueva flota de vehículos. Entre las diversas aplicaciones se encuentran empresas comercializadoras de productos farmacéuticos, componentes electrónicos, empresas de transporte masivo, en el funcionamiento de los operadores logísticos, etc.



## 2.1. Objetivos del MDHFVRP

El objetivo principal del problema de ruteo de vehículos con múltiples depósitos y flota heterogénea es satisfacer la demanda de todos los clientes al menor costo posible.

La resolución del problema debe cumplir los siguientes requerimientos:

- Satisfacer la demanda de todos los clientes.
- Establecer cual depósito atenderá a cada cliente.
- Decidir cuáles clientes visitará cada vehículo y en qué orden.
- Disminuir la cantidad de vehículos utilizados.
- Definir rutas con la menor distancia recorrida.

El problema supone que siempre se debe satisfacer la demanda de todos los clientes y aunque podrían manejarse costos por insatisfacción, en el modelo propuesto en esta tesis inicialmente se revisa que la capacidad total de los vehículos sea mayor a la demanda total de los clientes.

Después de ejecutar el algoritmo de esta tesis se puede visualizar para todos los depósitos cual es la ruta de cada vehículo, lo que incluye los clientes que visita y el orden del ruteo. Además se establecen dos tipos de costos que genera cada vehículo que realice una ruta, estos son:

- Costos fijos: debido a que la utilización de cada vehículo según su tipo incurre en un costo, el algoritmo propenderá en buscar siempre las soluciones más baratas y dejara a un lado las que castiguen innecesariamente los costos fijos.
- Costos variables: por cada tipo de vehículo se tiene un costo por unidad de movimiento recorrido, esto es clave porque siempre se buscaran las soluciones que tengan rutas con menor recorrido total.

## 2.2. Formulación matemática

El problema está definido por un grafo dirigido de nodos y arcos  $G(N, A)$ , donde  $N = C \cup D$ , siendo  $C = \{1, \dots, n\}$  el conjunto de clientes y  $D = \{n + 1, \dots, n + m\}$  el conjunto de depósitos. Por tanto el conjunto de nodos se establece como  $N = \{1, \dots, n, n + 1, \dots, n + m\}$ .

Existe un conjunto de tipos de vehículos  $K = \{1, \dots, p\}$ , donde cada tipo tiene asociado una capacidad  $Q_k$ , costos fijos  $cf_k$  y un costo por distancia recorrida  $cv_k$ . En este estudio cada depósito tiene una cantidad definida de vehículos por tipo  $Veh_{dk}$ . El modelo tiene dos variables de decisión que son  $X_{ijkd}$  y  $Y_{ij}$ , la primera da información sobre si el arco  $(i,j)$  será activado, el tipo de vehículo que lo atendería y su depósito de origen. La segunda brinda la información de la cantidad de carga disponible en el vehículo antes de visitar el nodo  $j$ .

## Parámetros

$n$ : Número de clientes

$m$ : Numero de depósitos

$p$ : Tipos de vehículos

$q_i$ : Demanda por cliente  $i = 1, \dots, n$

$Q_k$ : Capacidad por tipo de vehículo  $k = 1, \dots, p$

$cf_k$ : Costos fijos por tipo de vehículo  $k = 1, \dots, p$

$cv_k$ : Costos por unidad de distancia recorrida según el tipo de vehículo  $k = 1, \dots, p$

$d_{ij}$ : Distancia asociada al arco (i,j)  $i, j = 1, \dots, n + m$

$veh_{kd}$ : Vehículos disponibles tipo k en el depósito d  $k = 1, \dots, p; d = n + 1, \dots, n + m$

## Variables de decisión

$X_{ijkd} = \begin{cases} 1, & \text{si un vehículo tipo } k \text{ que sale del depósito } d \text{ atraviesa el arco } (i, j) \\ 0, & \text{dnc} \end{cases}$

$Y_{ij} = \text{carga disponible del vehículo antes de visitar el nodo } j, \forall i, j = 1, \dots, n + m$

## Formulación

$$\text{Min } Z = \sum_{d=n+1}^{n+m} \sum_{k=1}^p cf_k \sum_{i=n+1}^{n+m} \sum_{j=1}^n X_{ijkd} + \sum_{d=n+1}^{n+m} \sum_{k=1}^p \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (cv_k * d_{ij} * X_{ijkd}) \quad (1)$$

Sujeto a:

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^p \sum_{i=1}^{n+m} X_{ijkd} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^p \sum_{j=1}^{n+m} X_{ijtd} = 1, \quad i = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^{n+m} X_{ijkd} = \sum_{i=1}^{n+m} X_{jikd}, \quad j = 1, \dots, n + m; \quad d = n + 1, \dots, n + m; \quad k = 1, \dots, p \quad (4)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^n Y_{ij} \geq \sum_{j=1}^n q_j \quad (5)$$

$$\sum_{i=1}^{n+m} Y_{ij} - \sum_{i=1}^{n+m} Y_{ji} = q_j \quad , \forall j = 1, \dots, n \quad (6)$$

$$Y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^p Q_k * X_{ijkd} \quad , \quad \forall i = 1, \dots, n+m; \quad \forall j = 1, \dots, n \quad (7)$$

$$X_{d_1 t d_2} = 0, \quad \forall i = 1, \dots, n; \quad \forall k = 1, \dots, p; \quad \forall d_1 \neq d_2 = n+1, \dots, n+m \quad (8)$$

$$X_{i d_1 t d_2} = 0, \quad \forall i = 1, \dots, n; \quad \forall k = 1, \dots, p; \quad \forall d_1 \neq d_2 = n+1, \dots, n+m \quad (9)$$

$$Y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^p (Q_k - q_i) * X_{ijkd} \quad , \quad \forall i = 1, \dots, n+m; \quad \forall j = 1, \dots, n \quad (10)$$

$$Y_{ij} \geq \sum_{d=n+1}^{n+m} \sum_{k=1}^p q_j * X_{ijkd} \quad , \quad \forall i, j = 1, \dots, n \quad (11)$$

$$\sum_{j=1}^c X_{ijk_i} \leq veh_{ti} \quad , \quad \forall k = 1, \dots, p; \quad \forall i = n+1, \dots, n+m \quad (12)$$

$$X_{ijkd} \in \{0,1\} \quad , \quad \forall i, j = 1, \dots, n+m; \quad \forall k = 1, \dots, p; \quad \forall d = n+1, \dots, n+m \quad (13)$$

$$Y_{ij} \geq 0 \quad , \quad \forall i, j = 1, \dots, n+m \quad (14)$$

**Grafico 2. Modelo matemático. Adaptado de (Salhi, Imran, & Wassan, 2013)**

El modelo tiene la función objetivo (1) diseñada para disminuir el costo total que se incurre al satisfacer la demanda de todos los clientes, lo que incluye los costos fijos por utilización de los vehículos y los costos variables asociados a la distancia que recorre cada vehículo. La restricción (2) estipula que los clientes sean visitados una sola vez, (3) y (4) aseguran la conservación de flujo. La restricción (5) obliga a que la cantidad que sale de los depósitos sea por lo menos igual a la cantidad demandada, (6) indica que la cantidad de carga del vehículo después de visitar un cliente es igual a la carga antes de visitarlo menos su respectiva demanda y (7) garantiza que se respete la capacidad del vehículo. Las restricciones (8) y (9) obligan a que las rutas que salen de un depósito terminen en el mismo. La restricción (10) controla que la carga que vaya entre el arco  $(i, j)$  no sea mayor a la existente antes de atender al cliente  $i$  y (11) permite que la carga que va entre  $(i, j)$  sea mayor a la demanda de  $j$ . La cantidad de vehículos por tipo en cada depósito está dada por la restricción (12). Por último (13) y (14) especifican el tipo de variable de decisión.

Tomando como base la formulación matemática, en esta tesis se tendrá en cuenta la siguiente nomenclatura:

Una solución del problema cualquiera  $S_i$ , está compuesta de un conjunto de rutas como se aprecia a continuación:

$$S_i = \{r_1^i, r_2^i, r_3^i, \dots, r_z^i\}$$

El número total de rutas y de vehículos es igual a  $z$ . Sabiendo que cada vehículo tiene una única ruta asociada en la solución y está sujeto a un solo depósito, se tiene que:

$$z = \sum_{k=1}^p \sum_{d=n+1}^{n+m} veh_{kd}$$

Por tanto se tiene un conjunto de todos los vehículos  $V = \{1, 2, \dots, z\}$ . Cada ruta está compuesta por un arreglo de clientes organizados en el orden en que se visitan y es realizada por un único vehículo, esta se expresa de la forma:

$$r_j^i = \{\ell_{1j1}^i, \ell_{4j2}^i, \ell_{j3}^i, \dots, \ell_{bjh}^i, \dots, \ell_{njt}^i\}$$

Donde  $\ell_{bjh}^i$  es el cliente  $b$  visitado en la  $h$ -ésima posición de la ruta  $j$  de la solución  $i$  y cada ruta puede estar compuesta por una cantidad como máximo del total de clientes  $n$ .

### 3. Método de solución

En este trabajo de investigación se propone inicialmente por cada depósito realizar una agrupación de clientes según la cercanía que tengan a estos. Después de esto se implementa la metaheurística de la búsqueda armónica para generar soluciones  $S_i$  interesantes en términos de costo y se genera un afinamiento de cada ruta de las soluciones  $r_j^i$  por medio de un algoritmo de intercambio de arcos 2-OPT.

#### 3.1 Búsqueda armónica

La eficiencia de un algoritmo matemático varía para cada tipo de problema que pretenda resolver, por lo que no se puede decir que se cuente con el mejor algoritmo para hallar el óptimo global de cualquier problema; lo anterior en concordancia con los teoremas para optimización de "No Free Lunch" mostrados en (Wolpert, 1997). Sin embargo, el algoritmo de la búsqueda armónica ha mostrado un gran potencial y eficiencia en comparación con otras metaheurísticas en un amplio espectro de aplicaciones (Manjarres, y otros, 2013) (Kumar, 2014).

La búsqueda armónica es un algoritmo de optimización metaheurístico que imita el comportamiento de una orquesta cuando pretende generar una melodía más armoniosa. Cuando se compara el proceso de improvisación de los músicos de la orquesta con un problema de optimización podemos asemejar a cada músico con su instrumento a una variable de decisión; el rango de tonos del instrumento musical corresponde al rango de valores que puede tomar la

variable de decisión; en cierto momento la armonía musical de la improvisación corresponde al vector de solución en cada iteración; y la estética de la armonía se compara con el desempeño que tiene la función objetivo del problema de optimización.

Así como la melodía de la música se mejora vez tras vez, el vector solución mejora iteración tras iteración. En el gráfico 3 se aprecia el símil entre la generación de música por parte de los músicos hasta llegar al oído del público y las decisiones que pueden tomar las variables hasta cambiar la función objetivo. En el gráfico 4 se muestra el paralelo punto por punto entre el proceso de improvisación musical y el problema de optimización.

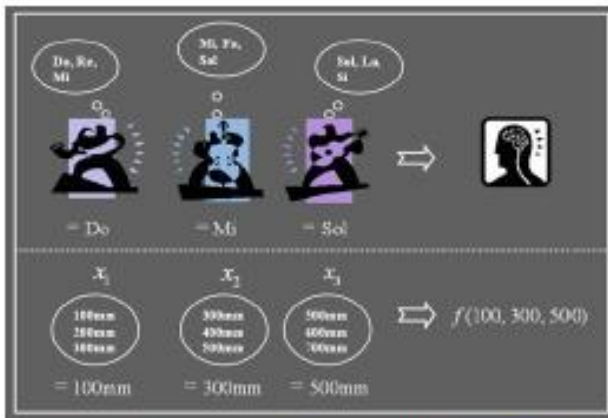


Gráfico 3. Búsqueda armónica. Fuente: (Geem, Lee, & Park, 2005)

El problema de optimización que se pretende resolver se podría definir como:

$$\begin{aligned} & \text{Minimizar } f(S), \\ & \text{Sujeto a } \ell_{bjh} \in G_j, \quad \forall b = 1, \dots, n; \forall j = 1, 2, \dots, z; \forall h = 1, \dots, \mu \end{aligned}$$

Donde  $f(S)$  es la función objetivo,  $\ell_{bjh}$  es el cliente  $b$  de la ruta  $j$  en la posición  $h$  y  $G_j$  es el conjunto de clientes que por cercanía puede atender la ruta o vehículo  $j$ .

Proceso de Improvisación Musical	Problema de Optimización
Instrumentos musicales (músicos)	Las rutas $r_j$
Rango del tono	Grupo de posibles clientes $G_j; \ell_{bjh} \in G_j, \ell_{bjh} \in r_j$
Armonía	Vector solución $S = \{r_1, r_2, r_3, \dots, r_z\}$
Estética	Función objetivo $f$
Práctica	Iteración
Experiencia o memoria armónica	Matriz donde se guardan las soluciones $S_i$

Gráfico 4. Comparación improvisación musical con problema de optimización

En el gráfico anterior se presenta un símil entre el proceso de improvisación musical descrito en la búsqueda armónica y los componentes del problema de optimización utilizados en la metodología.

La búsqueda armónica tiene los siguientes parámetros:

- HMS (Harmony Memory Size): este es el tamaño de la memoria armónica. La memoria armónica empieza vacía y se llena de una cantidad HMS vectores armónicos de solución. Esto quiere decir que se tienen soluciones  $S_i$ ,  $\forall i = \{1, \dots, HMS\}$
- HMCR (Harmony Memory Consideration Rate): este valor es la tasa de consideración de la memoria armónica, la cual representa la probabilidad con la que la variable tomará en cuenta algún valor alojado en su memoria armónica. Toma valores entre cero y uno,  $HMCR \in [0,1]$ .
- PAR (Pitch Adjusting Rate): la tasa de ajuste del tono establece la probabilidad de que la variable de decisión  $r_j$  sea modificada de forma cercana a la anterior. Este parámetro toma valores entre cero y uno,  $PAR \in [0,1]$ .
- NI: es el número de vectores armónicos nuevos que se generan después de construir la memoria armónica.

Si se analizan los parámetros de la búsqueda armónica, se puede identificar que la tasa de consideración de la memoria armónica (HMCR) y la tasa de ajuste del tono (PAR) están ofreciendo el balance entre la diversificación e intensificación de la metaheurística y esto influye en el rendimiento del algoritmo (Manjarres, y otros, 2013). La diversificación está siendo controlada por ambos parámetros debido a que el proceso de aleatorización permite examinar el espacio de búsqueda de una forma exploratoria y pretende encontrar mejores valores que los ya obtenidos. La intensificación está dada por la HMCR puesto que esta tasa representa que tanto serán consideradas las anteriores soluciones, por lo que un valor alto permite que la solución converja rápidamente. Debido a que la intensificación está dada por la HMCR es necesario establecer un equilibrio entre la PAR y la HMCR para evitar caer en la trampa de ubicarse en óptimos locales.

Como cualquier metaheurística poblacional, la búsqueda armónica funciona utilizando un conjunto de soluciones que puede ser explotada para mejorar la eficiencia del modelo. Por otra parte, la tasa de convergencia y la calidad de las soluciones que produce esta metaheurística no se ven dramáticamente afectadas por los vectores armónicos que conforman la memoria armónica inicial. El gradiente probabilístico que utiliza este algoritmo produce mejores soluciones iteración tras iteración, donde los operadores conducen a que el comportamiento del algoritmo guíe a la memoria armónica hacia regiones del espacio de solución con mejor ajuste (Manjarres, y otros, 2013).

Para Geem (2009), cuando se analiza el algoritmo se pueden identificar las siguientes características que a su vez se presentan como ventajas sobre otros modelos de optimización:

- Evita óptimos locales gracias a los ajustes del tono y a la improvisación de notas aleatorias.
- Puede manejar variables discretas y continuas.
- Permite resolver problemas de optimización lineal y no lineal.
- Un grupo de múltiples armonías pueden ser usadas en paralelo.

- No requiere cálculos complejos, lo cual da un procesamiento más rápido.

### **3.2 Intercambios 2-OPT**

Este algoritmo es probablemente el más utilizado para resolver el problema del agente viajero (Engels & Manthey, 2009), el cual consiste en encontrar la ruta más corta que visite una vez todos los nodos y retorne al nodo origen. El algoritmo 2-OPT realiza todos los intercambios de a dos arcos posibles en una ruta existente hasta encontrar la combinación que genere la menor distancia (Verhoeven, Aarts, & Swinkels, 1995). El algoritmo de búsqueda armónica ira generando rutas iteración tras iteración y cada una de ellas tiene el comportamiento de un TSP. El uso de 2-OPT propende a disminuir los costos variables asociados a la distancia total recorrida por cada vehículo. En esta tesis no se verifican todos los posibles intercambios de arcos si no que se establece un parámetro de cantidad de realización de intercambios aleatorios, lo cual otorga flexibilidad para diferentes tamaños de los problemas y se pretende realizar un método rápido computacionalmente.

### **3.3 Método de solución aplicado**

En este estudio la variable de decisión corresponde a la ruta que realiza cada vehículo, la cual es un arreglo de clientes en el que la posición donde estén ubicados es el orden en el que es visitado en esa ruta y cuando el arreglo es nulo significa que no se utilizó ese vehículo en la solución del problema. Como cada vehículo está sujeto a un depósito, esa ruta sale y regresa del mismo depósito.

A continuación se proceden a explicar los seis pasos utilizados en este trabajo de investigación para el acercamiento a la resolución del MDHFVRPFD por medio de un híbrido entre la metaheurística de la búsqueda armónica junto al algoritmo de intercambios 2-OPT. Estos pasos toman en cuenta lo expuesto por Geem (2009) en su metaheurística de la búsqueda armónica, junto a las particularidades adicionales del problema.

- a. Inicialización de parámetros de la búsqueda armónica:** en un archivo de texto se incluyen todos los parámetros del problema, tales como las características de cada tipo de vehículo, la ubicación y demanda de cada cliente, la cantidad de vehículos por tipo y la ubicación que tiene cada depósito. Además, se incluyen los parámetros concernientes a la búsqueda armónica y la cantidad de intercambios que realizara el 2- OPT en cada ruta cuando sea utilizado para mejorar la distancia.

```

1 HMCR, PAR, HMS, NI e intercambios 2-OPT
2 variables,0.5,0.8,50,50,si,50
3 costos,700-120-1.2,600-110-1.1,500-100-1,400-90-0.9,300-80-0.8
4 cliente1,6,-99,-97
5 cliente2,72,-59,50
6 cliente3,93,0,14
7 cliente4,28,-17,-66
8 cliente5,5,-69,-19
9 cliente6,43,31,12
10 cliente7,1,5,-41

```

**Grafico 5. Ejemplo datos de entrada del modelo**

En el gráfico anterior se ve un ejemplo del archivo de entrada para un problema con siete clientes, dos depósitos y tres tipos de vehículos. En la primera fila se está denotando los valores del *HMCR*, *PAR*, *HMS*, *NI* y la cantidad de intercambios del *2 – Opt*.

Cada fila dentro de la matriz de la memoria armónica y cada vector armónico de improvisación cuentan con un costo asociado que se calcula de la siguiente forma:

$$f(S) = \left( \sum_{k=1}^K cf_k * uv_k \right) + \left( \sum_{k=1}^K \sum_{i=1}^{N+D} \sum_{j=1}^{N+D} cv_k * d_{ij} * x_{ijk} \right)$$

La función objetivo está arrojando el costo total por visitar a todos los clientes y satisfaciendo todas las restricciones. La variable  $uv_k$  es la cantidad de vehículos utilizados del tipo  $k$ , la variable  $x_{ijk}$  es la variable binaria que indica si se recorrió el arco  $(i - j)$  con un vehículo tipo  $k$ ,  $d_{ij}$  es la distancia euclidiana entre los nodos  $i$  y  $j$  del arco. Los costos fijos y variables por tipo de vehículo se expresan con los parámetros  $cf_k$  y  $cv_k$  respectivamente.

- b. Asignación del conjunto de clientes a atender por depósito:** se relaciona a cada cliente el depósito que lo pueda atender según se encuentre con mayor grado de proximidad.

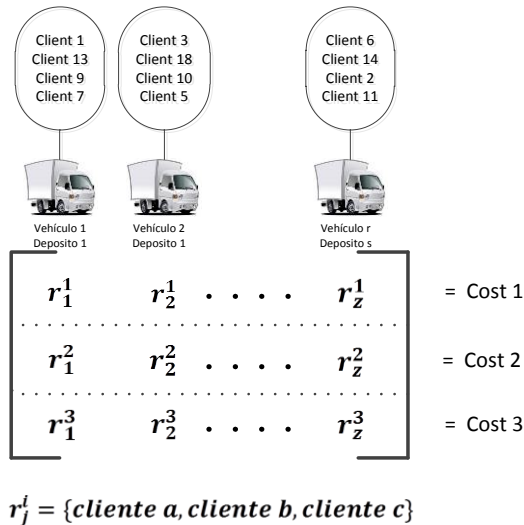
Posterior a eso, para cada depósito se procede a revisar los clientes que tiene asociados por proximidad y con estos se establece el conjunto de clientes o clúster que puede atender puntualmente cada depósito.

Debido a que todos los vehículos se encuentran asociados a un depósito de origen, cada uno de ellos tendrá como espacio de búsqueda al conjunto de los posibles clientes a atender  $G_j$ . Las rutas están conformadas por algunos de esos clientes en su respectivo orden de atención.

- c. Inicialización de la memoria armónica:** acá se van a construir las *HMS* filas iniciales de la memoria armónica. En el siguiente gráfico se representa un ejemplo de la estructura de la memoria o matriz armónica, cada posición de una fila representa un vehículo y el contenido



en cada valor es un arreglo de clientes, el cual representa la ruta que recorre ese vehículo. La dimensión de las filas o vectores está dada por la cantidad de vehículos que tenga la instancia, sabiendo que cada fila de la matriz representa una solución  $S_i$ .



**Gráfico 6. Estructura de la matriz armónica**

Cada vehículo asociado a un depósito cuenta con un conjunto de posibles clientes que atender tal como un músico tiene unas tonadas musicales en las cuales tiene predilección por tocar. Dentro de la memoria o matriz armónica están las soluciones particulares que ha tenido cada vehículo y sumadas a las de los demás nos están dando una solución al problema general que se aprecia en cada fila de esa matriz, junto a su respectivo costo. Una ruta  $r_j^i$ , hecha por un vehículo  $j$  en un vector solución  $i$ , es un arreglo de clientes organizado en el orden de visita. Ahora se procede a explicar cómo se realiza el llenado inicial de la memoria armónica. Para cada fila o vector solución se selecciona aleatoriamente un vehículo  $j$  y se empiezan a escoger aleatoriamente clientes del conjunto  $G_j$  de posibles clientes que puede atender ese vehículo hasta llenar su capacidad de tal forma que cree el arreglo de clientes  $r_i$ . El anterior proceso se repetirá hasta que todos los clientes hayan sido atendidos.

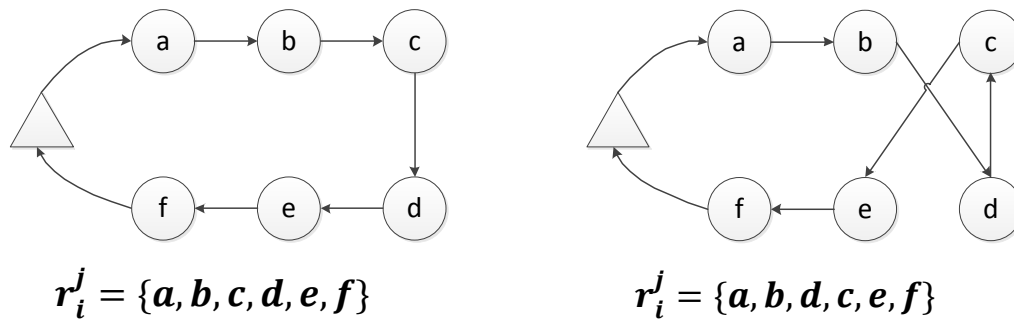
- 1: Para  $h = 1$  hasta  $HMS$
- 2: Mientras clientes sin atender  $> 0$  hacer
- 3:  $j = j \sim U[1, 2, \dots, z]$
- 4:  $i = 0$
- 5: Mientras capacidad  $r_j > dem_i$  hacer
- 6:  $r_j = r_j + \{i\}$
- 7: capacidad  $r_j = capacidad r_j - dem_i$
- 8:  $i = i \sim U[G_j]$
- 9: Fin mientras
- 10:  $r_j = r_j - \{0\}$

11: *Fin mientras*  
 12:  $h = h + 1$   
 13: *Fin para*

**Gráfico 7. Pseudocódigo inicialización de la memoria armónica**

El anterior pseudocódigo muestra que se genera la matriz con una cantidad de HMS filas (1)(12) y cada fila se construirá hasta que todos los clientes hayan sido atendidos (2). Se escoge un vehículo aleatoriamente (3) y mientras su capacidad le permita atender la demanda (5) se seleccionan aleatoriamente y adicionan clientes del conjunto de posibles a atender (6) (8), actualizando la capacidad disponible (7).

Debido a que el orden de visita de los nodos está haciéndose de forma aleatoria podemos no tener rutas eficientes en términos de costos variables asociados a la distancia recorrida, por lo anterior se ha planteado aplicar un algoritmo de intercambios 2-OPT que en combinación contribuya a mejorar la solución.



**Gráfico 8. 2-OPT, intercambio de arcos**

En el gráfico 8 se aprecia cómo puede darse un intercambio de arcos en alguna ruta solución  $r_j^i$ , de las que se encuentran en la matriz. El vehículo sale del depósito y hace su recorrido pero esta vez la ruta cambia ya que se intercambian los arcos  $b - c$  y  $d - e$  por los arcos  $b - d$  y  $c - e$  consiguiendo una menor distancia total.

Esta aplicación del algoritmo realiza una cantidad de intercambios aleatorios dado como parámetro inicial del modelo y en el siguiente pseudocódigo se aprecia la ejecución del mismo:

1: *ruta-existente = ruta sin 2OPT*  
 2: *Para t = 1, ..., parametro*  
 3: *mejor-distancia = calculoDistancia(ruta-existente)*  
 4:  $i = i \in \sim U[1, \text{Numero de nodos en la ruta}]$   
 5:  $k = k \in \sim U[1, \text{Numero de nodos en la ruta}]$ ,  $k > i$   
 6: *intercambio2OPT(ruta-existente, i, k)*  
 7: *Realizar la ruta desde el origen hasta i - 1*

```

8:      Tomar la ruta de  $i$  hasta  $k$  y adicionarla en orden inverso
9:      Tomar la ruta de  $k + 1$  hasta el final y adicionarla
10:     Retorna nueva-ruta
11:     nueva-distancia = calculoDistancia(nueva-ruta)
12:     si nueva-distancia < mejor-distancia
13:       ruta-existente = nueva-ruta
14:       mejor-distancia = nueva-distancia
15:     Fin si
16:      $t = t + 1$ 
17:     siguiente

```

**Gráfico 9. Pseudocódigo de intercambios aleatorios 2-OPT**

El pseudocódigo inicia con la ruta del vehículo (1) y calcula la distancia total en visitar a todos los clientes y volver al depósito (3); posteriormente se seleccionan aleatoriamente dos nodos de la ruta llamados  $i$  y  $k$  tales que en el orden inicial de la ruta  $k > i$  (4) (5). Se efectúa el proceso de intercambio de arcos 2-OPT (6)(7)(8)(9) y se calcula la distancia de la nueva ruta(11), en caso de ser la menor encontrada (12) se toma esa ruta como la mejor encontrada hasta el momento (13) y se actualiza la distancia mínima (14). Este proceso se repetirá el número de veces que se haya definido con el parámetro de entrada del modelo.

Este procedimiento a seguir para intercambiar arcos es una adaptación del registrado en Croes (1958), donde inicialmente se cuenta con una ruta ordenada  $[0, 1, 2, \dots, i - 1, i, i + 1, \dots, k - 1, k, k + 1, \dots, N]$  en el que se escogen dos nodos aleatoriamente  $i$  y  $k$ , donde  $i$  va antes que  $k$  en la ruta. La nueva ruta estaría conformada, en su respectivo orden, por: el trayecto del origen al nodo  $i - 1$ , el trayecto de  $i$  hasta  $k$  en orden inverso y la ruta de  $k + 1$  hasta el final; la ruta queda en el orden  $[0, 1, 2, \dots, i - 1, k, k - 1, \dots, i + 1, i, k + 1, \dots, N]$ .

- d. Improvisación de una nueva armonía:** después de tener la matriz inicial, se va a proceder a generar nuevos vectores de solución los cuales son construidos teniendo en cuenta tanto las soluciones alojadas en la matriz armónica como en valores totalmente nuevos, tal como se expresa en el siguiente pseudocódigo:

```

1: Mientras clientes sin atender > 0  hacer
2:    $j = j \sim U[VD]$  (vehículos disponibles)
3:   aleatorio1  $\sim U[0,1]$ 
4:   Si aleatorio1 < HMCR entonces
5:      $r_j' = r_j^i$ , donde  $i \sim U[1, \dots, HMS]$ 
6:     aleatorio2  $\sim U[0,1]$ 
7:     Si aleatorio2  $\leq PAR$  entonces
8:        $r_j' = 2OPT[r_j', (aleatorio2 * 10 * parametro)]$ 
9:     Fin si
10:  Si no
11:     $h = 0$ 
12:    Mientras capacidad  $r_j > dem_h$   hacer
13:       $r_j = r_j + \{h\}$ 
14:      capacidad  $r_j = capacidad r_j - dem_h$ 
15:       $h = h \sim U[G_j]$ 
16:    Fin mientras
17: Fin mientras

```

**Gráfico 10. Pseudocódigo generación de nuevos vectores**

El anterior pseudocódigo muestra como el modelo propuesto en esta tesis genera los nuevos vectores de solución. El vector se termina de crear hasta el momento en que se hayan atendido a todos los clientes (1); se empieza escogiendo aleatoriamente un vehículo disponible (2) y se genera un número aleatorio distribuido uniformemente entre cero y uno (3) que se compara con el parámetro de consideración de la memoria armónica *HMCR* (4), en caso de ser menor se va a asignar a ese vehículo aleatoriamente alguno de los arreglos de clientes que se encuentren en la memoria armónica (5). Posteriormente se va a generar otro aleatorio uniformemente distribuido entre cero y uno (6), el cual se compara con el parámetro de ajuste del tono *PAR* (7) y en caso de ser menor se generara un procedimiento de 2-OPT que tomara el arreglo obtenido en (5) y el número de intercambios que realizara es igual al primer decimal del aleatorio multiplicado por el parámetro de intercambios de 2-OPT (8). Cuando el primer aleatorio comparado contra *HMCR* es mayor, el vehículo llena su capacidad con los clientes definidos del conjunto que puede atender (12)(13)(14)(15).

Cuando se decida tener en consideración la memoria armónica para algún valor en particular, puede suceder que existan clientes de la ruta escogida que ya estuvieran asignados, por tal razón se verifica si ese vehículo tiene capacidad para atender más clientes y en caso afirmativo busca aleatoriamente clientes para visitar hasta llenar la capacidad. Si la capacidad no permite atender más clientes, la ruta se conserva como está.

En el siguiente gráfico se muestra la comparación del proceso de generación musical de un conjunto de músicos según la búsqueda armónica con la formación de soluciones para el

problema de ruteo de vehículos abordado en esta tesis. Todos los músicos participan con su tonada para componer una armonía que puede ser buena o mala, mientras que para este problema cada vehículo sujeto a un depósito genera una ruta donde atiende a un arreglo de clientes y en conjunto la flota de vehículos satisface la demanda de todos los clientes con un costo total que puede ser menor a los encontrados en la matriz armónica.

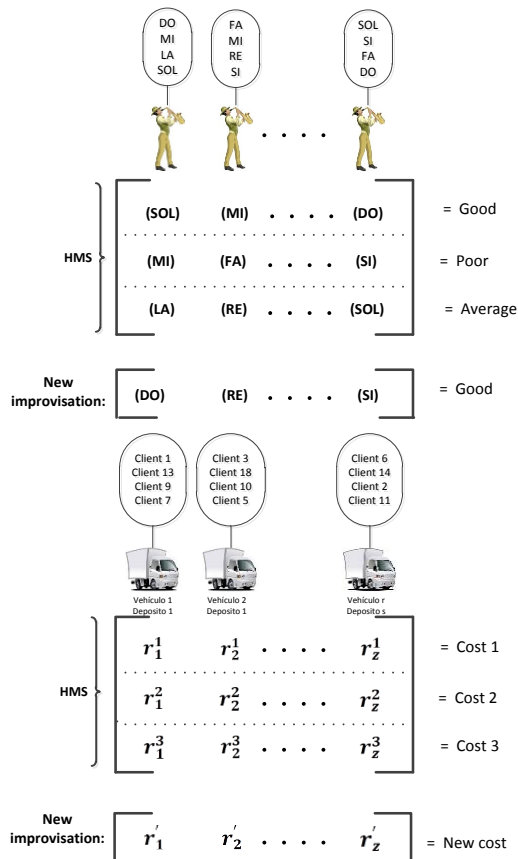


Gráfico 11. Búsqueda armónica en problema de ruteo

- e. **Actualizar la memoria armónica:** Después de obtener el nuevo vector armónico de improvisación se evalúa su valor en la función objetivo preestablecida, dando el costo de esa solución del problema, el cual se compara contra la peor solución existente dentro de la memoria armónica y en caso de ser menor, se procede a incluir ese nuevo vector en la matriz de la memoria armónica y se desecha la peor fila de la matriz.
- f. **Verificar el criterio de parada:** Se repite el proceso de improvisación de nuevos vectores y su comparación con la memoria armónica hasta completar el número de iteraciones  $NI$  que se estableció en el archivo de entrada. Cuando se hayan efectuado el total de iteraciones se efectúa una última revisión de cuál es el vector que genera el menor costo total. De este

vector se extrae la información que arroja el archivo de salida, como lo es: orden de las rutas de los vehículos por cada depósito, tiempo computacional y costo total de resolución del problema. El siguiente gráfico permite visualizar cómo cada iteración afecta el costo total de resolver el problema, donde se aprecia que la tendencia es a ir disminuyendo.

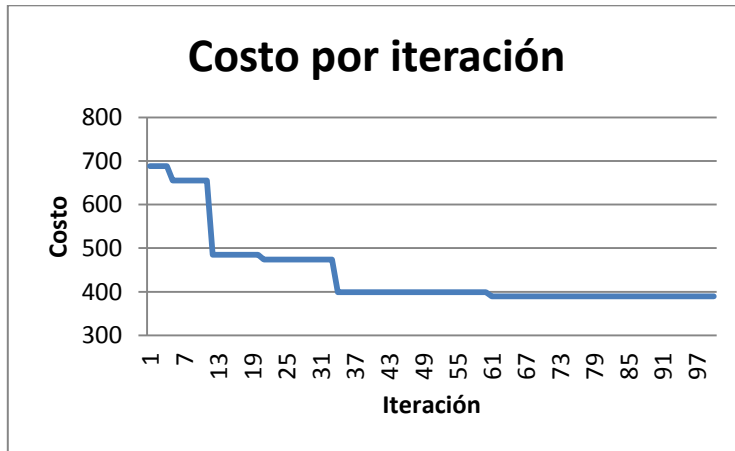


Gráfico 12. Ejemplo de evolución del costo por iteración

#### 4. Test computacionales

Se utilizan catorce instancias diseñadas por Salhi (2013) para resolver el problema de ruteo de vehículos con múltiples depósitos y flota heterogénea pero no se efectúa comparación con sus resultados ya que en esta tesis se está añadiendo una restricción al modelo y esto hace que varíen los resultados. El no trabajar con el supuesto de tener una flota ilimitada de vehículos en todos los depósitos le añade mayor dificultad (Prins, 2002).

El computador utilizado para ejecutar las pruebas tiene un procesador Intel Core i3, memoria de 4 GB y sistema operativo de 32 bits. El algoritmo metaheurístico propuesto está modelado en lenguaje de programación JAVA y es compilado en la plataforma NetBeans IDE 8.0; el cual es ejecutado diez veces por cada instancia para extraer el valor promedio del costo y la mejor solución.

Después de realizar diferentes pruebas para definir los valores que se adecuen a todas las instancias con buenos resultados, los parámetros de la búsqueda armónica que se decidieron utilizar son:

$$HMS = 15$$

$$HMCR = 0,5$$

$$PAR = 0,8$$

$$NI = 100$$

Se efectuaron tres tipos de prueba para probar el impacto del 2-opt y del tamaño de los clusters en el resultado. El primer método utiliza los intercambios 2-OPT y los clúster son formados asociando únicamente los clientes más cercanos; el segundo método no realiza intercambios 2-OPT y también conserva en su clúster los clientes más cercanos; el tercer método realiza los intercambios 2-OPT pero cada cliente puede ser atendido por los dos depósitos más cercanos y los clúster se forman con esta relación.

Method	2-OPT	Clúster Size
1	1000	1
2	No	1
3	1000	2

La formulación matemática del problema en programación lineal entera mixta es ejecutada en Xpress-MP, donde cada instancia corre durante tres horas y se extraen las cotas superiores e inferiores. Los resultados del problema aplicado a estas instancias se muestran a continuación:

Problem details			MILP		Method 1			Method 2			Method 3		
K = 5			Xpress - MP		Total Cost		Time	Total Cost		Time	Total Cost		Time
#	n	m	LB	UB	Best value	Mean		Best value	Mean		Best value	Mean	
1	55	4	1451,13	1844,01	1809,45	1944,31	1,3	1896,28	1921,13	0,2	2042,41	2059,77	1,3
2	85	3	2441,49	NF	3012,23	3066,68	3,1	3065,21	3109,76	0,3	3491,35	3524,81	3,0
3	85	3	1878,31	NF	2302,42	2359,89	2,8	2471,03	2530,66	0,3	2856,63	2869,66	2,7
4	50	4	785,25	NF	913,58	1024,35	1,1	959,59	995,73	0,1	1074,29	1092,77	1,2
5	50	4	758,00	1045,25	928,75	944,02	1,3	975,63	991,91	0,1	1080,62	1099,60	1,2
6	75	5	862,99	NF	1088,42	1162,52	1,5	1183,29	1208,91	0,2	1322,32	1394,20	1,4
7	100	2	1160,25	1752,58	1652,88	1817,46	2,6	1685,05	1894,54	0,3	2063,64	2132,08	2,5
8	100	2	1322,80	NF	1661,21	1682,66	2,4	1692,86	1697,32	0,3	2136,13	2158,16	2,4
9	100	3	1342,69	NF	1611,15	1630,66	2,1	1768,39	1786,45	0,2	1926,98	1944,11	2,1
10	100	4	NF	NF	1685,12	1749,34	3,1	1734,22	1783,28	0,3	1829,13	1890,00	3,0
11	249	3	NF	NF	18019,27	18248,11	11,2	18019,27	18159,90	1,8	23295,30	24651,56	10,6
12	80	2	3550,54	NF	4162,38	4465,59	1,8	4335,48	4707,61	0,2	6444,50	6498,32	1,8
13	160	4	NF	NF	9088,12	9500,31	5,2	9088,12	9553,00	1,1	13266,02	13470,51	5,1
14	240	6	NF	NF	14312,54	14708,91	9,4	14312,54	14726,58	0,9	20624,34	20631,87	9,3

LB, UB: lower and upper bounds using 3 h of CPU with Xpress - MP

Time: in minutes

NF: no solution (upper or lower bound) was found during the allowed CPU time

The models are executed in a computer equipped with 4 GB of RAM and a Intel Core i3 processor

**Tabla 2. Resultados del modelo**

Como es de esperar el programa de optimización lineal entera mixta no brindó una solución óptima para ninguna de las instancias ya que este problema tiene un tamaño considerable. Se aprecia también que el tamaño del clúster formado únicamente por los clientes más cercanos resultó en un mejor resultado en la función de costos que cuando se crean clústers observando para cada cliente sus dos depósitos más cercanos.

El algoritmo de intercambios 2-Opt está mejorando en promedio un 5% la solución con respecto a si no se utilizara y la mejora depende de la instancia en particular que se esté probando.

## 5. Conclusiones

El problema de ruteo de vehículos con múltiples depósitos y flota heterogénea cuenta con bastante aplicabilidad en el sector industrial pero en toda la bibliografía consultada no se encontró que se haya trabajado un número tan elevado de veces como con otras variaciones del problema. En este trabajo de investigación se aborda el problema a través de una metodología que mezcla el uso de una metaheurística poblacional, como lo es la búsqueda armónica, con un algoritmo de intercambio de arcos aleatorios de 2-OPT.

Para el problema abordado, con la metodología implementada y con las instancias ejecutadas, se puede evidenciar que la utilización de un método de intercambios aleatorios 2-OPT funciona adecuadamente para mejorar la solución. Por otra parte, para las instancias probadas se puede notar que la forma en que se generen los clústers de clientes por depósito impacta en la calidad de las soluciones.

El problema de ruteo de vehículos con múltiples depósitos y flota heterogénea, con vehículos fijos por depósito, puede ser tan grande que en algunas instancias no se consigue encontrar solución factible en tres horas de ejecución del modelo de programación lineal. Sin embargo, la metaheurística implementada en esta tesis puede encontrar soluciones factibles rápidamente.

Los futuros trabajos se deben concentrar en verificar diferentes formas de abordar la característica de los múltiples depósitos e implementar otros métodos de solución. Por otra parte, se recomienda también trabajar este problema asumiendo vehículos infinitos ya que este se enfoca en decisiones empresariales de tipo estratégico como lo es el diseño de toda la flota de vehículos.



## Bibliografía

- Archetti, C., Feillet, D., Gendreau, M., & Speranza, M. (Agosto de 2011). Complexity of the VRP and SDVRP. *Freight Transportation and Logistics*, Vol. 15, pp. 741-750.
- Baldacci, R. B. (2014). *The Vehicle Routing Problem, latest advanced and new challenges*. Canada: Springer.
- Cobos, C. P. (Junio de 2011). Una revisión de la búsqueda armónica. *Revista Avances en Sistemas e Informática*, Vol. 8, pp. 67 - 80.
- Croes, G. A. (Diciembre de 1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, Vol. 6(No. 6), pp. 791-812.
- Dondo, R., Méndez, A., & Cerdá, J. (2003). An optimal approach to the multiple-dept heterogeneous vehicle routing problema with time window and capacity constraints. *Latin American Applied Research*, pp. 129 - 134.
- Engels, C., & Manthey, B. (2009). Average-case approximation ratio of the 2-opt algorithm for the TSP. *Operations Research Letters*, Vol. 37, pp. 83-84.
- Geem, Z. (2009). *Music-Inspired Harmony Search Algorithm - Theory and Applications*. Chennai, India: Springer.
- Geem, Z., Lee, K., & Park, Y. (2005). Application of Harmony Search to Vehicle Routing. *American Journal of Applied Sciences*, Vol 2, 1552 - 1557.
- Kumar, V., Kumar, J., & Kumar, D. (Marzo de 2014). Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *Journal of Computational Science*, Vol. 5, pp. 144-155.
- Lim, A., & Zhu, W. (2006). A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach. *Advances in Applied Artificial Intelligence*, Vol. 4031, pp. 282–291.
- Manjarres, D., Landa, I., Gil, S., Del Ser, J., Bilbao, M., Salcedo, S., y otros. (Mayo de 2013). A survey on applications of the harmony search algorithm. *Engineering Applications of Artificial Intelligence*, Vol. 16, pp. 1818-1831.
- Pichpibul, T., & Kawtummachai, R. (March de 2013). Modified Harmony Search Algorithm for the Capacitated Vehicle Routing Problem. Proceedings of the International MultiConference of Engineers and Computer Scientists. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. II.
- Prins, C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. *Journal of Mathematical Modelling and Algorithms*, Vol. 1, pp. 135 - 150.

- Salah, N., & Badr, A. (Febrero de 2012). Complexity of Capacitated Vehicles Routing Problem Using Cellular Genetic Algorithms. *International Journal of Computer Science and Network Security*, Vol. 12, pp. 5-11.
- Salhi, S., Imran, A., & Wassan, N. (11 de Mayo de 2013). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, pp.1-11.
- Surekha, P. &. (Agosto de 2011). Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms. *World Applied Programming*, Vol. 1, pp. 118 - 131 .
- Taha, E., Ayob, M., Ahmad, M., & Ahmad, Z. (2013). Harmony Search Algorithm for Vehicle Routing Problem with Time Windows. *Journal of Applied Sciences*, Vol. 13, pp. 633-638.
- Verhoeven, M., Aarts, E., & Swinkels, P. (1995). A parallel 2-opt algorithm for the Traveling Salesman Problem. *Future Generation Computer Systems*, Vol. 11, pp. 175-182.
- Wei, T., Zhixiang, F., Qingquan, L., Shih-Lung, S., & Biyu, C. (Enero de 2014). A bi-level Voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem. *Transportation Research Part E*, Vol. 61, pp. 84-97.
- Wolpert, D. &. (Abril de 1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on evolutionary computation*, Vol. 1, pp. 67-82.
- Yepes, V. (2002). *Optimización heurística económica aplicada a las redes de transporte del VRPTW*, Tesis Doctoral. Universidad Politécnica de Valencia.
- Yu, B., Yang, Z., & Xie, J. (2011). A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society*, Vol. 62, pp. 183–188.
- Zabala, P. (2006). *Problemas de Ruteo de Vehículos*, Tesis Doctoral. Universidad de Buenos Aires.