

## Desempeño de las técnicas de agrupamiento para resolver el problema de ruteo con múltiples depósitos

### Performance of clustering techniques for solving multi depot vehicle routing problem

Eliana M. Toro-Ocampo<sup>1</sup>, Andrés H. Domínguez-Castaño<sup>2</sup> y Antonio H. Escobar-Zuluaga<sup>3</sup>

Recibido: 17 de junio de 2015,  
Aceptado: 17 de noviembre de 2016

---

#### Cómo citar / How to cite

Eliana M. Toro-Ocampo, Andrés H. Domínguez-Castaño y Antonio H. Escobar-Zuluaga, "Desempeño de las técnicas de clusterización para resolver el problema de ruteo con múltiples depósitos", *Tecno Lógicas*, vol. 19, no. 36, pp. 49-62, 2016.



---

<sup>1</sup> MSc. en Investigación de Operaciones y Estadística, Facultad de Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira-Colombia, elianam@utp.edu.co  
<sup>2</sup> MSc. en Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Pereira-Colombia, ahdominguez@utp.edu.co  
<sup>3</sup> PhD. en Ingeniería Eléctrica, Escuela de Tecnología Eléctrica, Universidad Tecnológica de Pereira, Universidad Tecnológica de Pereira, Pereira-Colombia, aescobar@utp.edu.co

## Resumen

El problema de ruteo de vehículos considerando múltiples depósitos es clasificado como NP duro, cuya solución busca determinar simultáneamente las rutas de un conjunto de vehículos, atendiendo un conjunto de clientes con una demanda determinada. La función objetivo del problema consiste en minimizar el total de la distancia recorrida por las rutas, teniendo en cuenta que todos los clientes deben ser atendidos cumpliendo restricciones de capacidad de depósitos y vehículos. En este artículo se propone una metodología híbrida que combina las técnicas aglomerativas de clusterización para generar soluciones iniciales con un algoritmo de búsqueda local iterada, *iterated location search* (ILS) para resolver el problema. Aunque en trabajos previos se proponen los métodos de clusterización como estrategias para generar soluciones de inicio, en este trabajo se potencia la búsqueda sobre el sistema de información obtenido después de aplicar el método de clusterización. Además se realiza un extenso análisis sobre el desempeño de las técnicas de clusterización y su impacto en el valor de la función objetivo. El desempeño de la metodología propuesta es factible y efectivo para resolver el problema en cuanto a la calidad de las respuestas y los tiempos computacionales obtenidos, sobre las instancias de la literatura evaluadas.

## Palabras clave

Búsqueda local iterada, optimización combinatoria, ruteo con múltiples depósitos, red de distribución, técnicas de agrupamiento.

## Abstract

The vehicle routing problem considering multiple depots is classified as NP-hard. MDVRP determines simultaneously the routes of a set of vehicles and aims to meet a set of clients with a known demand. The objective function of the problem is to minimize the total distance traveled by the routes given that all customers must be served considering capacity constraints in depots and vehicles. This paper presents a hybrid methodology that combines agglomerative clustering techniques to generate initial solutions with an iterated local search algorithm (ILS) to solve the problem. Although previous studies clustering methods have been proposed like strategies to generate initial solutions, in this work the search is intensified on the information generated after applying the clustering technique. Besides an extensive analysis on the performance of techniques, and their effect in the final solution is performed. The operation of the proposed methodology is feasible and effective to solve the problem regarding the quality of the answers and computational times obtained on request evaluated literature.

## Keywords

Combinatorial optimization, clustering techniques, distribution network, iterated local search, Multi-Depot Vehicle Routing.

## 1. INTRODUCCIÓN

En el gerenciamiento óptimo de la cadena de suministros debe gestionarse de forma eficiente la distribución de materias primas y mercancías a través de la red de distribución. Esta eficiencia se mide a menudo en las entregas oportunas a los usuarios finales. El problema de ruteo considerando múltiples depósitos, *Multi-Depot Vehicle Routing Problem* (MDVRP) es formulado como una red logística, que pretende resolver, entre otros aspectos, la atención a los clientes en el menor tiempo posible. El MDVRP es una variante del problema de ruteo de vehículos, *Vehicle Routing Problem* (VRP), el cual corresponde al nombre genérico de una temática que involucra una gran variedad de problemas de optimización combinatorial [1], [2]. Estos problemas son de gran interés en el área académica y de negocios debido a sus múltiples aplicaciones a nivel empresarial, militar y gubernamental en el campo logístico.

Las principales contribuciones de este artículo son: 1) Comparación de diferentes técnicas y estrategias de clusterización para generar soluciones de inicio y su efecto en el desempeño del ILS aplicada en la solución del MDVRP. 2) La estructura de información, tipo árbol, obtenida de la técnica de clusterización que es utilizada para construir soluciones iniciales. Cada nivel el árbol es explorado de forma iterativa a fin de encontrar la mejor solución posible. 3) El uso de heurísticas como inicializadores, para identificar regiones promisorias que faciliten el desempeño de un algoritmo de optimización, de forma que se puedan encontrar soluciones diversas de buena calidad en tiempos computacionales cortos.

## 2. ESTADO DEL ARTE

El MDVRP se puede identificar en una gran variedad de contextos y su correcta

solución permite una reducción considerable en los costos de distribución. Estudios de casos documentados incluyen entrega de alimentos procesados [3], distribución de productos químicos [4], reparto de bebidas gaseosas, [5] distribución de productos en empresas multinacionales [6], distribución de gases industriales [7], despacho de vehículos cisterna con derivados de petróleo [8], distribución de alimentos lácteos [9].

Para la solución MDVRP se han aplicado técnicas heurísticas y técnicas exactas.

En cuanto a las técnicas exactas, Baldacci y Mingozzi [10] proponen un método basado en el procedimiento de generación sistemática de límites con base en la propuesta de Christofides et al. [11]. El algoritmo exacto utiliza tres tipos de procedimientos para generar límites inferiores, basados en relajación lineal y en relajación lagrangeana, realizadas a la formulación matemática. Los procedimientos propuestos reducen el número de variables del modelo matemático, lo que permite resolver el problema resultante de forma exacta utilizando un software comercial de programación lineal entera. Baldacci et al. [12] describen una formulación de programación entera del problema de ruteo de vehículos con visita periódicas, *Periodic Vehicle Routing Problem* (PVRP), que se utiliza para obtener diferentes límites inferiores. El MDVRP se puede formular como un PVRP debido a que los depósitos pueden modelarse como los múltiples periodos en el contexto de un PVRP. Por lo tanto, cualquier algoritmo que resuelve el PVRP también puede resolver el MDVRP. Contardo y Martinelli [13] presentan un algoritmo exacto, el cual utiliza dos formulaciones para el MDVRP, usadas en diferentes etapas del algoritmo. Se agregan varios conjuntos de restricciones válidas para fortalecer los modelos propuestos.

Los algoritmos heurísticos han sido adoptados para resolver los problemas de MDVRPs debido a que son metodologías que encuentran soluciones de forma rápi-

da, además de ser relativamente fácil de implementar. Salhi et al. [14] proponen una heurística multinivel con dos pruebas de reducción. Las soluciones iniciales factibles se construyen en el primer nivel, mientras que en los otros niveles son mejoradas mediante heurísticas inter e intra depósito [15].

Giosa et al. [16] implementó la estrategia de “agrupar primero y rutear después”, aplicada a una variante del MDVRP que considera ventanas de tiempo, *MDVRP with Time Windows* (MDVRPTW). En la etapa de clusterización se asignan los clientes a los depósitos, luego cada clúster se resuelve usando el algoritmo de ahorros. El artículo realiza un estudio computacional con seis diferentes técnicas de clusterización. Nagy et al. [17] resuelven una variante del MDVRP donde se consideran entregas y recogidas simultáneas, *Multi-depot VRP with mixed Pickup and Delivery* (MDVRPMPD). La metodología es desarrollada en varias etapas: i) Divide el conjunto de clientes en limítrofes y no limítrofes. ii) Asigna los clientes no limítrofes al depósito más cercano iii) Para cada depósito se encuentra una solución factible que se obtiene de la solución del problema de ruteo con entregas y recogidas considerando un depósito, *Vehicle Routing Problem with Pickups and Deliveries* (VRPPD). iv) Se insertan los clientes limítrofes en las rutas encontradas en el paso anterior. Luego se realizan procedimientos de mejoramiento a través de heurísticas como 2 y 3-OPT, reinserciones e intercambios.

Lee et al. [18] plantean el MDVRP como un problema de programación dinámica determinística, con estados finitos, luego usan el algoritmo heurístico de la ruta más corta para resolverlo. Crevier et al. [19] proponen una heurística que combina el método de búsqueda tabú con la programación lineal entera, en esta aproximación los vehículos pueden ser reabastecidos en depósitos intermedios a lo largo de la ruta. Jeon et al. [20] proponen un algoritmo

genético híbrido para el MDVRP, el cual considera el mejoramiento de una solución inicial mediante tres heurísticas y una tasa de mutación flotante para escapar de óptimos locales. Ho et al. [21] desarrollaron dos algoritmos genéticos híbridos (HGA1 y HGA2). La mayor diferencia entre los genéticos es la forma en que se generan las soluciones iniciales; en el HGA1 se generan de forma aleatoria. El algoritmo de ahorros de Clarke and Wright [22] y el algoritmo del vecino más cercano fueron incorporados en el proceso de inicialización del HGA2. El desempeño de ambos algoritmos es evaluado usando dos instancias generadas de forma aleatoria, donde se consideran 2 depósitos, 50 y 100 clientes.

Sureka y Sumathi [23] resuelven el problema a través de un algoritmo genético. Los clientes son agrupados con base en la distancia a los depósitos más cercanos y luego son ruteados con el algoritmo de ahorros de Clarke and Wright [22]. Luego se realiza el procedimiento de optimización usando el algoritmo genético. Cordeau y Maischberger [24] presentan una heurística paralela con conceptos de búsqueda tabú. La metodología resuelve diferentes variantes del VRP, entre ellas el MDVRP. La heurística combina el Búsqueda Tabú con un simple mecanismo de perturbación para garantizar la exploración del espacio de búsqueda y además se describe la implementación paralela de la heurística. Vidal et al. [25] proponen un algoritmo genético híbrido donde se permiten tanto soluciones factibles como infactibles, las cuales son separadas en dos grupos. Se aplica sucesivamente un número de operadores para seleccionar dos individuos padres y combinarlos. El nuevo individuo hijo, generado inicialmente, es mejorado usando procedimientos de búsqueda local y luego es incluido en la subpoblación apropiada con relación a su factibilidad. Este es un método muy eficiente y alcanza todos los óptimos de la literatura.

Subramanian et al. [26] proponen una técnica matheurística que combinan una

secuencia de modelos lineales enteros mixtos con columnas que corresponden a rutas, las cuales son generadas mediante una aproximación metaheurística. Esta metodología resuelve diferentes variantes del VRP, entre ellas el MDVRP y el MDVRPMPD. Escobar et al. [27] proponen un algoritmo búsqueda tabú granular híbrido, el cual considera diferentes vecindarios y estrategias de diversificación para mejorar la solución inicial obtenida, mediante un procedimiento híbrido. Los resultados obtenidos son competitivos en cuanto a calidad de las respuestas y tiempos computacionales. Finalmente, Montoya-Torres et al. [2] presentan un completo estado del arte del MDVRP donde relaciona 173 publicaciones publicadas desde 1988 hasta 2014. Hacen un recorrido por diferentes variantes del problema como: ventanas de tiempo, división de las entregas, recogidas y entregas simultáneas en la misma ruta y flota heterogénea. Otro aspecto importante es que el documento presenta tanto los abordajes mono-objetivo como multiobjetivo que se han desarrollado para resolver el problema, así como líneas futuras de investigación.

### 3. MODELO MATEMÁTICO

El MDVRP puede ser definido como un conjunto de Depósitos  $I$  y clientes  $J$ , donde el objetivo es encontrar una secuencia factible de entregas a través de un conjunto de rutas, las cuales deben minimizar la distancia o el tiempo, de forma que se atiendan todos los clientes de la red. A cada arco  $(i,j) \in I \cup J$  se le asocia un costo de viaje  $c_{ij}$ . Cada cliente  $j \in J$  tiene una demanda  $d_j$ , la cual debe ser atendida en su totalidad en una sola visita. Un conjunto de  $K$  vehículos idénticos con capacidad  $Q_k$  está disponible en cada depósito. El MDVRP es un problema NP-duro debido a que simultáneamente se deben determinar las rutas de una flota de vehículos que inician desde diferentes depósitos. El obje-

tivo es determinar un conjunto de rutas que minimice la distancia del recorrido, para realizar todas las entregas o el tiempo invertido en la atención de todos los clientes. Se deben cumplir de forma estricta las siguientes restricciones: *i)* La demanda de cada cliente debe ser atendida por un único vehículo. *ii)* Cada vehículo inicia y termina su ruta en el mismo depósito. *iii)* La totalidad de la demanda de cada ruta no debe exceder la capacidad del vehículo. El modelo que se presenta a continuación, corresponde al presentado por Surekha y Sumathi [23].

#### Nomenclatura

##### Conjuntos:

---

|     |                       |
|-----|-----------------------|
| $I$ | Conjunto de depósitos |
| $J$ | Conjunto de clientes  |
| $K$ | Conjunto de vehículos |

---

##### Parámetros:

---

|          |                                     |
|----------|-------------------------------------|
| $N$      | Número de vehículos                 |
| $c_{ij}$ | Distancia entre los nodos $i$ y $j$ |
| $W_i$    | Capacidad del depósito              |
| $d_i$    | Demanda del cliente $j$             |
| $Q_k$    | Capacidad del vehículo (ruta) $k$   |

---

##### Variables de decisión

---

|           |   |
|-----------|---|
| $x_{ijk}$ | Variable binaria que indica que el nodo $i$ precede al nodo $j$ en la ruta $k$ .                                    |
| $Z_{ij}$  | Variable binaria que define si el consumidor ubicado en el nodo $j$ es atendido por el centro de distribución $i$ . |
| $U_{lk}$  | Variable auxiliar usada en las restricciones de eliminación de sub-tours en la ruta $k$ .                           |

---

La función objetivo (8) minimiza la distancia total de todas las rutas. La ecuación (9) establece que cada cliente debe ser asignado a una ruta. La ecuación (10) representa la restricción de capacidad del conjunto de vehículos. El conjunto de restricciones de eliminación de sub-tours está representado por (11). Las restricciones de conservación de flujo son expresadas me-

diante (12). La ecuación (13) garantiza que cada vehículo atiende una única ruta. Las restricciones de capacidad de los depósitos están dadas por (14). La ecuación (15) especifica que un cliente puede ser asignado a un depósito, únicamente si hay una ruta que parte desde el depósito y transita a través del cliente. Las restricciones (16), (17) definen la naturaleza binaria de las variables  $x_{ijk}$ ,  $z_{ij}$ . Finalmente la restricción (18) define  $U_{lk}$  como una variable no negativa continua.

#### Formulación del problema

$$\min \sum_{i \in I \cup J} \sum_{j \in I \cup J} \sum_{k \in K} C_{ij} x_{ijk} \quad (8)$$

$$\sum_{k \in K} \sum_{i \in I \cup J} x_{ijk} = 1 \quad j \in J \quad (9)$$

$$\sum_{j \in J} d_j \sum_{i \in I \cup J} x_{ijk} \leq Q_k \quad k \in K \quad (10)$$

$$U_{lk} - U_{jk} + N x_{ijk} \leq N - 1 \quad l, j \in J, k \in K \quad (11)$$

$$\sum_{j \in I \cup J} x_{ijk} - \sum_{j \in I \cup J} x_{jik} = 0 \quad k \in K, i \in I \cup J \quad (12)$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1 \quad k \in K \quad (13)$$

$$\sum_{j \in J} d_j z_{ij} \leq V_i \quad i \in I \quad (14)$$

$$-z_{ij} + \sum_{u \in I \cup J} (x_{iuk} + x_{ujk}) \leq 1 \quad i \in I, j \in J, k \in K \quad (15)$$

$$x_{ijk} \in \{0, 1\} \quad i \in I, j \in J, k \in K \quad (16)$$

$$z_{ij} \in \{0, 1\} \quad i \in I, j \in J \quad (17)$$

$$U_{lk} \geq 0 \quad l \in J, k \in K \quad (18)$$

## 4. TÉCNICAS DE CLUSTERIZACIÓN

Los Métodos jerárquicos y otros algoritmos de agrupamiento representan un intento para encontrar "buenos" clústeres de una nube de datos. Generalmente no es posible examinar todas las posibilidades de agrupamiento, para conjuntos de datos de

gran volumen [28]. Varios autores han utilizado técnicas de agrupamiento en diferentes algoritmos para resolver diferentes variantes del VRP entre ellos, Giosa et al. [16] en el MDVRP y Barreto et al. [29] en el CLRP, ambos con resultados promisorios. Los algoritmos de agrupamiento jerárquico implican un proceso secuencial. En cada paso del enfoque jerárquico de agrupamiento, una observación o un conjunto de observaciones se va fusionado en otro grupo. En todos los métodos se inicia el procedimiento con base en la matriz de distancias entre clientes, definida como la distancia inicial entre cada clúster. Se debe tener en cuenta que al inicio del proceso cada cliente forma un clúster.  $C_i$  representa el clúster  $i$  con  $n_i$  clientes y  $C_j$  representa el clúster  $j$  con  $n_j$  clientes.

Las diferentes métricas de clusterización usadas en este estudio son: Distancia mínima, *single linkage (SL)*; Distancia máxima, *Complete Linkage (CL)*, Distancia promedio no ponderada, *unweight arithmetic average (UAD)*, Promedio ponderado, *weighted arithmetic (AL)*, Método del centroide ponderado, *Weighted centroid method (WCM)*, Método de la mediana, *Centroid method unweight (CMU)*, Método de ward, *Ward's method (W)*. [28] En la Tabla 1 se presentan Las ecuaciones de las métricas.

Adicionalmente a los métodos de clusterización, se implementó una heurística constructiva que se describe a continuación.

*Heurística constructiva (HC)*: esta heurística hace una selección aleatoria de clientes asignándolos al depósito más cercano. La localización de los clientes se realiza con base en la capacidad del depósito. Cuando se alcanza la capacidad del depósito, los clientes restantes serán asignados al depósito más cercano. El procedimiento se repite hasta cuando todos los clientes han sido ubicados a los depósitos.

Tabla 1. Métricas de clusterización usadas en este estudio. Fuente: Autores

| Método     | Métrica de distancia   |
|------------|--|
| <i>SL</i>  | $d(c_i, c_j) = \min_{\substack{x_l \in c_i \\ x_m \in c_j}} \{d(x_l, x_m)\}; l = 1, \dots, n_i, \quad m = 1, \dots, n_j \quad (1)$   |
| <i>CL</i>  | $d(c_i, c_j) = \max_{\substack{x_l \in c_i \\ x_m \in c_j}} \{d(x_l, x_m)\}; l = 1, \dots, n_i, \quad m = 1, \dots, n_j \quad (2)$   |
| <i>UAL</i> | $d(c_i, c_j) = \frac{d(c_{i1}, c_j) + d(c_{i2}, c_j)}{2} \quad (3)$  |
| <i>AL</i>  | $d(c_i, c_j) = \frac{n_{i1}d(c_{i1}, c_j) + n_{i2}d(c_{i2}, c_j)}{n_{i1} + n_{i2}} \quad (4)$  |
| <i>WCM</i> | $d(c_i, c_j) = \frac{n_{i1}d^2(c_{i1}, c_j)}{n_{i1} + n_{i2}} + \frac{n_{i2}d^2(c_{i2}, c_j)}{n_{i1} + n_{i2}} - \frac{n_{i1}n_{i2}d^2(c_{i2}, c_j)}{(n_{i1} + n_{i2})^2} \quad (5)$ |
| <i>CMU</i> | $d(c_i, c_j) = \frac{d^2(c_{i1}, c_j)}{2} + \frac{d^2(c_{i2}, c_j)}{2} - \frac{d^2(c_{i2}, c_j)}{4} \quad (6)$   |
| <i>W</i>   | $\text{Min } E_k = \sum_{i \in n_k} \sum_{j \in n} (x_{ij}^k - m_j^k)^2 \quad (7)$   |

## 5. METODOLOGÍA DE SOLUCIÓN

Para resolver el MDVRP se implementó el algoritmo ILS, [30] este es un método de búsqueda local aleatorio que genera una secuencia de soluciones obtenida mediante una heurística integrada. Su estructura se fundamenta en 3 pasos: *i)* Procedimiento de solución inicial. *ii)* Procedimiento de búsqueda local. *iii)* Perturbación.

### 5.1 Estructura de datos

La estructura de datos implementada es una estructura de árbol, donde los terminales corresponden a los clústeres. La distancia entre ellos corresponde a una métrica elegida de la Tabla 1. Supóngase que después de aplicar el método de clusterización se obtiene un arreglo de clústeres como el que se presenta en la Tabla 2. En la Tabla 2 en el nivel 1 se tienen 7 clústeres, debido a que cada individuo es un clúster en el paso inicial. A medida que se va ascendiendo de nivel se van disminuyendo la cantidad de clústeres, hasta llegar a 2 clústeres como se observa en el nivel 6.

### 5.2 Procedimiento para generar la solución inicial

La metodología para generar las soluciones iniciales se representa en los diagramas de flujo de las Fig. 1 y 2. Este procedimiento tiene como insumo la información de clústeres de la Tabla 2. El diagrama de flujo de la Fig. 1 es un procedimiento interno del diagrama de flujo de la Fig. 2 y entre ambos establecen una búsqueda exhaustiva hasta encontrar la configuración de menor costo posible. El resultado de este procedimiento corresponde a la solución inicial del algoritmo ILS.

Tabla 2. Estructura tipo árbol. Fuente: Autores

| Nivel | No de clústeres | cluster       | cluster | cluster | cluster | cluster | cluster | cluster |
|-------|-----------------|---------------|---------|---------|---------|---------|---------|---------|
| 1     | 7               | A             | B       | C       | D       | E       | F       | G       |
| 2     | 6               | (E,C)         | A       | B       | D       | E       | F       |         |
| 3     | 5               | (D,F)         | (C,E)   | A       | B       | G       |         |         |
| 4     | 4               | (C,E,D,F)     | A       | B       | G       |         |         |         |
| 5     | 3               | (A,C,E,D,F)   | B       | G       |         |         |         |         |
| 6     | 2               | (A,B,C,E,D,F) | G       |         |         |         |         |         |

### 5.3 Procedimiento de Búsqueda Local

El algoritmo de búsqueda contiene dos tipos de operadores de mejora, Intra-ruta e Inter-ruta, los cuales se enumeran en dos listas. A partir de la solución de inicio generada en el paso anterior, se da inicio a la búsqueda local, cuyo objetivo es mejorar la solución actual usando los operadores o estructuras de vecindad. El procedimiento se ejecuta de la siguiente manera: 1) Seleccionar un operador al azar de la lista de estructuras de vecindad inter-ruta. Si hay alguna mejora ir al paso 3, de lo contrario, ir al paso 2. 2) En caso de que la aplicación de la estructura de vecindad no pueda

mejorar la solución incumbente, prohibirla para la próxima selección. Si todas las estructuras de vecindad han sido prohibidas, la búsqueda local se detiene. En caso contrario ir al paso 1. 3) Seleccionar una estructura de vecindad de la lista de operadores intra-ruta. Si hay una mejora, ir al paso 5. De lo contrario, ir al paso 4. 4) Si la aplicación de la estructura de vecindad dentro de la ruta no puede mejorar la solución actual, la estructura queda prohibida para las selecciones posteriores. Ir al paso 1. Si se han prohibido todas las estructuras de vecindad intra-ruta. De lo contrario ir al paso 3. 5) Ir al paso 3.

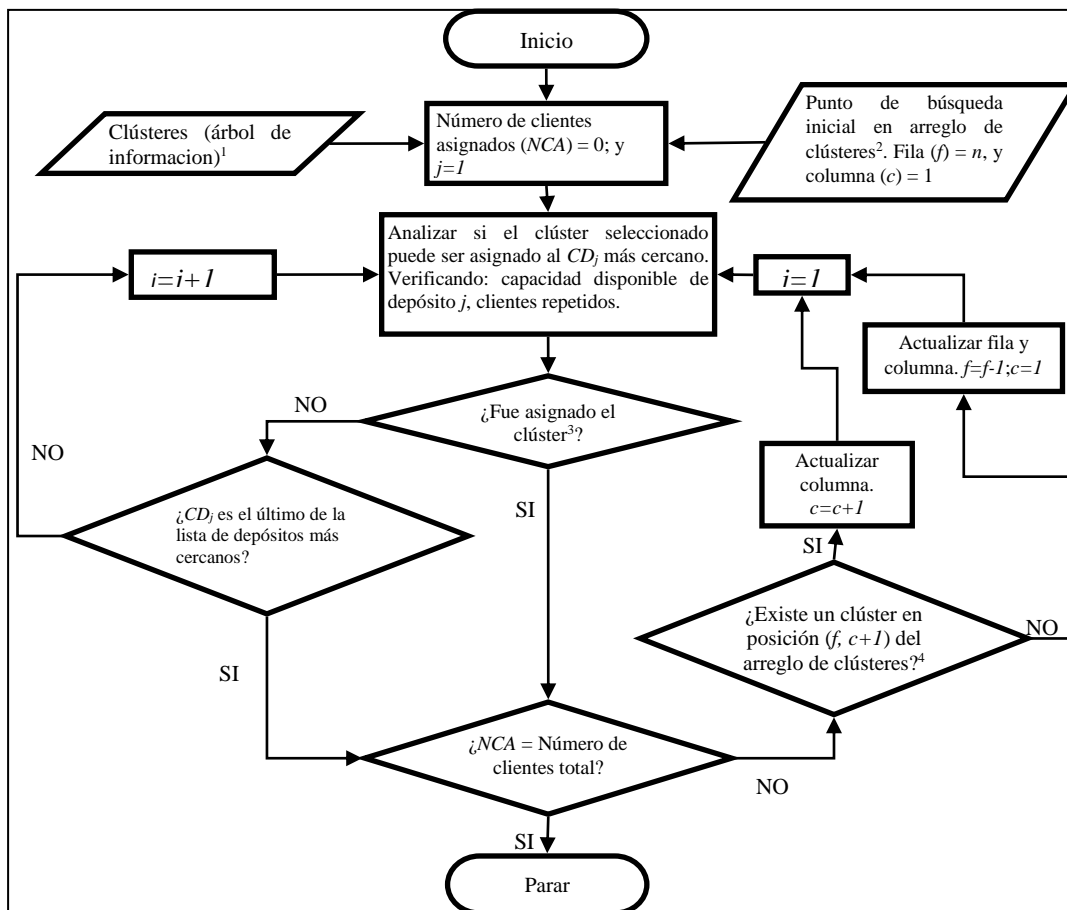


Fig. 1. Algoritmo de asignación de clientes a los depósitos. Fuente: Autores.

<sup>1</sup>Cada clúster tiene una estructura de datos formada por: centroide, demanda, lista de los depósitos más cercanos en orden ascendente de acuerdo con la distancia euclidiana respecto al centroide. <sup>2</sup>Punto de búsqueda inicial: es un clúster del arreglo de clústeres compuesto por uno o varios clientes, ubicado en la fila seleccionada  $n$  y columna número uno (tabla 2). <sup>3</sup>Cuando se asigna un clúster, se asignan todos los clientes del clúster al depósito analizado. <sup>4</sup>Recordar que en el arreglo de clústeres cada fila del arreglo tiene diferente número de columnas.



La lista de estructuras de vecindad inter-ruta usada es: shift(1,0), shift (2,0), shift(3,0), swap(1,1), swap(1,2), swap(2,2), 2-OPT, La lista intra-ruta está formada por: swap(1,1), shift (1,0),2-OPT [15].

### 5.4 Algoritmo de perturbación

El procedimiento de perturbación se lleva a cabo con clientes seleccionados aleatoriamente, los cuales deben pertenecer a diferentes rutas. Los operadores usados para tal fin son: swap (1, 1), swap (2, 1), swap (2, 2), shift (1, 0), shift (2, 0). El

operador seleccionado se aplica dos veces y de forma consecutiva. Para aplicar el operador se debe cumplir que la distancia entre los clientes seleccionados, no sobrepasen los límites de una distancia que es calculada a cada paso. Este valor se calcula como el promedio de la distancia del depósito asociado al primer cliente, respecto a los dos depósitos más cercanos. Esa distancia es afectada por un factor  $\beta$  seleccionado aleatoriamente y de acuerdo a pruebas preliminares debe estar entre [0.1 – 1.0].

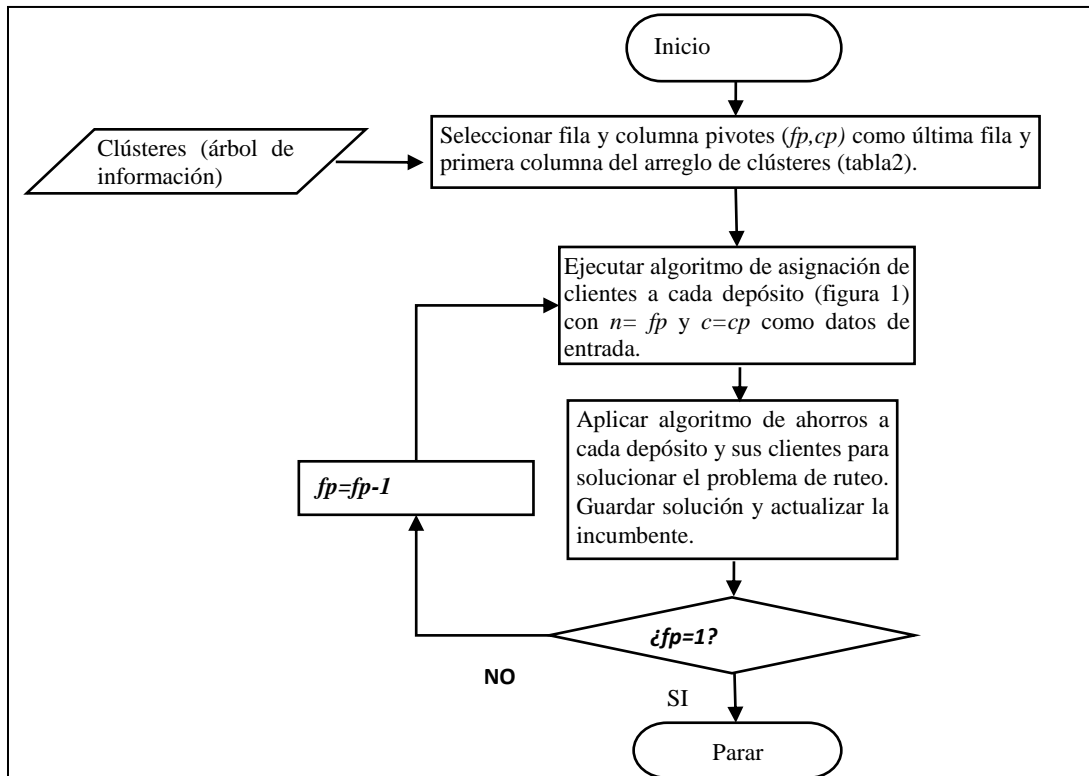


Figura 2. Algoritmo de generación de soluciones iniciales. Fuente: Autores

## 6. ANÁLISIS DE RESULTADOS

El algoritmo fue implementado en C++ y ejecutado en un computador Intel i5-3470 3,2 GHZ, 8 GB. La metodología es validada y comparada con las instancias propuestas por Cordeau et al.[19]. Se realizaron pruebas preliminares intensivas para encontrar los parámetros de mejor desempeño. El criterio de parada del algoritmo fue defini-

do en 800.000 iteraciones globales. El factor  $\beta$  de mejor desempeño se obtiene mediante la generación de un valor aleatorio entre 0,1 y 1. En la fase de perturbación se realizan hasta 50 intentos para identificar el cliente 2, debido a que se deben cumplir las condiciones de que pertenezca a una ruta diferente de la del cliente 1 y a que este ubicado dentro del radio permitido.

Los resultados de la Tabla 3 corresponden a las mejores soluciones encontradas después de realizar 10 corridas por métrica de clusterización. Como se observa, las soluciones iniciales encontradas tienen valores similares en cuanto a la función objetivo, a excepción de las obtenidas con la HC. En la Tabla 4 se presentan los tiempos computacionales, la técnica HC es la de menor consumo, por el contrario W es la de mayor consumo.

En las Tablas 5 y 6 se presenta el desempeño del algoritmo propuesto, se realizaron 10 corridas por método de clus-

terización, se reporta el promedio y la mejor solución obtenida. El tiempo de cálculo más bajo se obtiene cuando se utiliza CMU como estrategia de inicialización. En la mayoría de casos que utilizan esa técnica de clusterización como estrategia de inicio se puede ver que las soluciones son de buena calidad. La solución promedio de menor calidad en la etapa de optimización es la solución inicial utilizando HC, sin embargo, es la que utiliza menos tiempo de cálculo en la etapa inicial.

Tabla 3. Resultados del método de inicialización usado. Fuente: Autores

| Instancia | Paso de inicialización (%GAP) |       |      |       |      |       |       |       |       |      |
|-----------|-------------------------------|-------|------|-------|------|-------|-------|-------|-------|------|
|           | BKS                           | WCM   | CMU  | AL    | UAL  | SL    | CL    | W     | HC    | Min  |
| P01       | 576,86                        | 6,92  | 5,39 | 6,92  | 6,92 | 5,67  | 6,92  | 6,92  | 12,25 | 5,39 |
| P02       | 473,53                        | 6,66  | 6,25 | 6,66  | 5,72 | 6,66  | 6,03  | 6,66  | 9,08  | 5,72 |
| P03       | 641,19                        | 7,77  | 6,62 | 8,93  | 8,93 | 9,27  | 8,70  | 8,70  | 9,27  | 6,62 |
| P04       | 1001,04                       | 7,99  | 7,99 | 7,99  | 7,99 | 6,62  | 8,85  | 8,85  | 10,07 | 6,62 |
| P05       | 750,03                        | 9,22  | 8,12 | 10,37 | 9,22 | 11,02 | 8,73  | 10,37 | 11,02 | 8,12 |
| P06       | 876,5                         | 7,96  | 8,02 | 6,56  | 8,02 | 6,55  | 5,76  | 7,64  | 8,02  | 5,76 |
| P07       | 881,97                        | 6,2   | 6,2  | 6,2   | 6,2  | 6,2   | 6,2   | 6,2   | 11,87 | 6,2  |
| P12       | 1318,95                       | 3,54  | 3,54 | 2,12  | 1,61 | 3,54  | 1,07  | 3,1   | 14,36 | 1,07 |
| P15       | 2505,42                       | 6,36  | 8,13 | 8,15  | 7,1  | 9,02  | 7,71  | 6,21  | 15,72 | 6,21 |
| P18       | 3702,85                       | 9,49  | 7,33 | 10,08 | 7,45 | 10,65 | 10,47 | 8,94  | 19,57 | 7,33 |
| P21       | 5474,74                       | 10,95 | 9,27 | 10,44 | 9,63 | 12,25 | 11,22 | 11,85 | 21,62 | 9,27 |
| Media     |                               | 7,55  | 6,99 | 7,67  | 7,16 | 7,95  | 7,42  | 7,7   | 12,99 | 6,21 |

Tabla 4. Métodos de inicialización para cada instancia. Fuente: Autores

| Instancia | Tiempo computacional en la etapa de inicialización (segundos) |        |        |        |         |        |           |       |
|-----------|---|--------|--------|--------|---------|--------|-----------|-------|
|           | WCM   | CMU    | AL     | UAL    | SL      | CL     | W         | HC    |
| P01       | 0,070   | 0,071  | 0,075  | 0,076  | 0,127   | 0,108  | 0,711     | 0,000 |
| P02       | 0,097   | 0,095  | 0,102  | 0,112  | 0,169   | 0,148  | 0,718     | 0,015 |
| P03       | 0,268   | 0,299  | 0,352  | 0,281  | 0,603   | 0,471  | 7,988     | 0,000 |
| P04       | 0,624   | 0,624  | 0,734  | 0,655  | 1,279   | 0,951  | 40,170    | 0,000 |
| P05       | 0,627   | 0,671  | 0,849  | 0,697  | 1,659   | 1,154  | 40,139    | 0,015 |
| P06       | 0,593   | 0,608  | 0,655  | 0,609  | 1,233   | 0,936  | 40,092    | 0,000 |
| P07       | 0,764   | 0,752  | 1,036  | 0,832  | 1,886   | 1,142  | 46,254    | 0,016 |
| P12       | 0,343   | 0,312  | 0,359  | 0,375  | 0,905   | 0,546  | 8,799     | 0,015 |
| P15       | 2,777   | 2,668  | 2,762  | 2,636  | 12,939  | 4,845  | 526,126   | 0,016 |
| P18       | 13,039  | 12,496 | 14,619 | 9,033  | 70,832  | 17,737 | 6685,146  | 0,015 |
| P21       | 41,075  | 58,422 | 40,820 | 47,421 | 327,467 | 57,926 | 69878,787 | 0,031 |
| Media     | 5,48  | 7,00   | 5,67   | 5,70   | 38,10   | 7,81   | 7024,99   | 0,01  |

Tabla 5. Resultados del algoritmo ILS comparado con las BKS. Fuente: Autores

| instancia | Resultados con el algoritmo ILS (%GAP) |                   |                |                   |               |                   |                |                   |
|-----------|--|-------------------|----------------|-------------------|---------------|-------------------|----------------|-------------------|
|           | WCM<br>(mejor)                         | WCM<br>(promedio) | CMU<br>(mejor) | CMU<br>(promedio) | AL<br>(mejor) | AL<br>(ponderado) | UAL<br>(mejor) | UAL<br>(promedio) |
| P01       | 0,00                                   | 2,2               | 0,00           | 1,16              | 0,00          | 1,16              | 0,00           | 0,98              |
| P02       | 0,59                                   | 3,43              | 0,85           | 1,82              | 0,59          | 3,14              | 0,85           | 2,25              |
| P03       | 0,00                                   | 1,86              | 0,00           | 1,27              | 0,00          | 1,99              | 0,00           | 1,70              |
| P04       | 0,05                                   | 1,67              | 0,026          | 2,11              | 0,00          | 1,48              | 0,26           | 1,52              |
| P05       | 0,15                                   | 0,19              | 0,15           | 0,27              | 0,00          | 0,26              | 0,15           | 0,22              |
| P06       | 0,87                                   | 2,18              | 0,00           | 5,42              | 0,09          | 1,24              | 1,93           | 5,17              |
| P07       | 2,68                                   | 4,79              | 2,68           | 4,62              | 2,68          | 4,91              | 2,68           | 5,03              |
| P12       | 0,00                                   | 0,00              | 0,00           | 0,00              | 0,00          | 0,60              | 0,00           | 0,33              |
| P15       | 0,00                                   | 2,59              | 0,00           | 1,87              | 0,00          | 2,57              | 0,00           | 2,62              |
| P18       | 0,00                                   | 3,17              | 0,16           | 4,05              | 0,7           | 4,18              | 1,55           | 4,30              |
| P21       | 0,002                                  | 3,35              | 0,46           | 4,70              | 0,39          | 4,86              | 2,3            | 6,57              |
| Media     | 0,39                                   | 2,31              | 0,39           | 2,48              | 0,4           | 2,40              | 0,88           | 2,79              |

Tabla 6. Resultados del algoritmo ILS comparado con las BKS. Fuente: Autores

| Instancia | Resultados con el algoritmo ILS (%GAP) |                  |               |                  |              |                 |      |
|-----------|--|------------------|---------------|------------------|--------------|-----------------|------|
|           | SL<br>(mejor)                          | SL<br>(promedio) | CL<br>(mejor) | CL<br>(promedio) | W<br>(mejor) | W<br>(promedio) | CH   |
| P01       | 4,94                                   | 5,27             | 0,00          | 1,07             | 0,00         | 2,16            | 0,68 |
| P02       | 1,25                                   | 3,44             | 0,85          | 1,91             | 0,59         | 3,48            | 1,23 |
| P03       | 0,00                                   | 0,98             | 0,62          | 1,26             | 0,00         | 0,94            | 0,00 |
| P04       | 0,27                                   | 4,68             | 0,05          | 1,73             | 0,45         | 2,22            | 3,46 |
| P05       | 0,15                                   | 0,19             | 0,07          | 0,24             | 0,00         | 0,26            | 0,05 |
| P06       | 0,45                                   | 1,36             | 0,70          | 2,10             | 0,33         | 1,86            | 4,18 |
| P07       | 2,68                                   | 5,02             | 2,68          | 5,02             | 2,68         | 4,60            | 3,89 |
| P12       | 0,00                                   | 0,00             | 0,00          | 0,00             | 0,00         | 0,60            | 0,00 |
| P15       | 0,00                                   | 2,01             | 0,00          | 1,92             | 0,00         | 1,68            | 0,00 |
| P18       | 0,00                                   | 2,60             | 0,00          | 2,91             | 0,00         | 0,00            | 0,22 |
| P21       | 0,63                                   | 3,35             | 1,75          | 4,82             | 0,98         | 0,00            | 0,40 |
| Media     | 0,94                                   | 2,63             | 0,61          | 2,09             | 0,46         | 1,98            | 1,28 |

En la Tabla 7 se presenta el tiempo total requerido para el funcionamiento de la metodología completa que incluye la técnica de la clusterización y una versión básica del algoritmo de ILS. El hecho de la obtención de soluciones de diferente calidad en la fase de optimización utilizando el algoritmo ILS muestra que las heurísticas utilizadas como inicializadores, identifican soluciones en diferentes puntos del espacio de soluciones. Este aspecto es conveniente

para hacer una búsqueda diversa, en el espacio de soluciones.

Comparando los BKS contra las soluciones iniciales encontradas con técnicas de clusterización, se observa que estas técnicas localizan los clientes al depósito más cercano, en un porcentaje superior al 80 % en todos los casos, lo que demuestra la fortaleza de las mismas. El algoritmo de optimización presenta un buen rendimiento, ya que pasa de 6.21 % en la etapa ini-

cial hasta 0.30 % de GAP en la etapa de optimización. Para los 11 casos analizados se obtuvo el BKS en 6 de ellos. Para P21, la instancia más grande, se obtuvo una configuración óptima totalmente diferente a la reportada en la literatura. Se puede decir,

en general, que las técnicas de clusterización usadas como inicializadores facilitan el desarrollo de la técnica ILS para encontrar soluciones de alta calidad en tiempos computacionales razonables.

Tabla 7. Metodología completa. Fuente: Autores

| instancia | Técnica de clusterización+ ILS (segundos) |         |          |          |          |         |           |          |
|-----------|---|---------|----------|----------|----------|---------|-----------|----------|
|           | WCM                                       | CMU     | AL       | UAL      | SL       | CL      | W         | CH       |
| P01       | 0,279                                     | 0,223   | 0,28     | 0,296    | 0,437    | 0,318   | 0,96      | 0,156    |
| P02       | 103,172                                   | 39,421  | 99,752   | 102,141  | 2,545    | 2,486   | 100,792   | 0,094    |
| P03       | 1,048                                     | 18,691  | 13,6     | 13,369   | 1,651    | 2,6     | 9,485     | 2,746    |
| P04       | 122,363                                   | 74,007  | 65,625   | 136,781  | 60,544   | 53,43   | 236,341   | 220,713  |
| P05       | 51,448                                    | 7,582   | 193,63   | 43,597   | 140,447  | 382,044 | 164,174   | 1316,315 |
| P06       | 147,514                                   | 187,949 | 161,881  | 25,657   | 17,02    | 113,35  | 46,129    | 197,048  |
| P07       | 30,61                                     | 30,185  | 30,64    | 30,646   | 31,252   | 30,379  | 76,191    | 0,328    |
| P12       | 0,343                                     | 0,312   | 0,858    | 4,602    | 0,905    | 0,578   | 15,928    | 8,393    |
| P15       | 155,625                                   | 22,76   | 73,851   | 95,144   | 77,384   | 327,792 | 570,805   | 739,904  |
| P18       | 3534,845                                  | 677,649 | 2025,411 | 72,899   | 3463,423 | 620,011 | 7995,713  | 485,32   |
| P21       | 8567,631                                  | 846,753 | 7504,1   | 5576,696 | 6237,983 | 995,848 | 72951,869 | 903,24   |
| Media     | 1155,90                                   | 173,23  | 924,51   | 554,71   | 912,14   | 229,89  | 7469,85   | 352,21   |

## 7. CONCLUSIONES

Se propuso un mecanismo eficiente de exploración a la estructura tipo árbol, generada después de aplicar las técnicas de clusterización. Este entrega soluciones iniciales de buena calidad en tiempos computacionales muy cortos

Se proponen diferentes técnicas heurísticas que se utilizan como inicializadores para el MDVRP. Su análisis se realiza teniendo en cuenta el tiempo y la calidad de las soluciones tanto iniciales como finales. No se puede llegar a la conclusión de que específicamente una técnica de agrupamiento es la adecuada para la solución de todos los casos. Sin embargo, en promedio, la heurística con mejor desempeño es CMU en los casos de mayor tamaño.

Encontrar diferentes soluciones de inicio de buena calidad mediante la aplicación de diferentes técnicas de clusterización se constituyen en una ventaja cuando

se implementan algoritmos poblacionales, en las que el éxito se basa en la diversidad de la población inicial, la cual debe mantenerse durante todo el proceso con el fin de asegurar una búsqueda adecuada del espacio de soluciones. Se propone como trabajo futuro implementar una metaheurística poblacional empleando diferentes estrategias de clusterización para generar las soluciones de inicio.

## 8. AGRADECIMIENTOS

Los autores agradecen a la Universidad Tecnológica de Pereira por el apoyo al proyecto “Optimización multiobjetivo para resolver el problema de ruteo de vehículos asociados a múltiples depósitos considerando técnicas de clusterización”, cod. 7-15-1.

## 9. REFERENCIAS

- [1] K. F. Doerner and V. Schmid, "Survey: Matheuristics for rich vehicle routing problems," in *7th International Workshop on Hybrid Metaheuristics, HM 2010*, vol. 6373 LNCS, M. Blesa, C. Blum, G. Raidl, A. Roli, and M. Samples, Eds. Berlin: Springer, 2010, pp. 206–221.
- [2] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Comput. Ind. Eng.*, vol. 79, pp. 115–129, Jan. 2015.
- [3] P. Cassidy and H. Bennett, "Vehicle System Scheduling," *OR Soc.*, vol. 23, no. 2, pp. 151–163, 2015.
- [4] M. Ball, B. Golden, and A. Assad, "Planning for Truck Fleet Size in the Presence of a Common-Carrier Option," *Decis. Sci.*, vol. 14, no. 1, pp. 103–120, 2007.
- [5] B. L. Golden and E. a. Wasil, "OR Practice--Computerized Vehicle Routing in the Soft Drink Industry," *Oper. Res.*, vol. 35, no. 1, pp. 6–17, 1987.
- [6] M. Fisher, R. Greenfield, R. Jaikumar, and J. Lester, "A computerized Vehicle Routing Application," *Interfaces (Providence)*, vol. 4, pp. 42–52, 12AD.
- [7] W. J. Bell, L. M. Dalberto, M. L. Fisher, a. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman, "Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer," *Interfaces (Providence)*, vol. 13, no. 6, pp. 4–23, 1983.
- [8] A. G. G. Brown, C. J. Ellis, G. W. Graves, and D. Ronen, "of the dispatching are to minimize cost of delivered product, to balance the workload among the company trucks, and to load the maximum weight on a truck while adhering to all laws. The important," vol. 17, no. 1, pp. 2–4, 1987.
- [9] J. Pooley, "Integrated Production and Distribution Facility Planning at Ault Foods," *Interfaces (Providence)*, vol. 24, no. 4, pp. 113–121, 1994.
- [10] R. Baldacci and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Math. Program.*, vol. 120, no. 2, pp. 347–380, Sep. 2009.
- [11] N. Christofides, A. Mingozzi, and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Math. Program.*, vol. 20, no. 1, pp. 255–282, Dec. 1981.
- [12] R. Baldacci, E. Bartolini, a. Mingozzi, and a. Valletta, "An Exact Algorithm for the Period Routing Problem," *Oper. Res.*, vol. 59, no. 1, pp. 228–241, 2011.
- [13] C. Contardo and R. Martinelli, "A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints," *Discret. Optim.*, vol. 12, pp. 129–146, May 2014.
- [14] S. Salhi and M. Sari, "A multi-level composite heuristic for the multi-depot vehicle fleet mix problem," *Eur. J. Oper. Res.*, vol. 103, no. 1, pp. 95–112, Nov. 1997.
- [15] E. M. Toro O., A. H. Escobar Z., and M. Granada E., "Literature Review On The Vehicle Routing Problem In The Green Transportation Context," *Luna Azul*, vol. 42, no. 42, pp. 362–387, Dec. 2015.
- [16] I. D. Giosa, I. L. Tansini, and I. O. Viera, "New assignment algorithms for the multi-depot vehicle routing problem," *J. Oper. Res. Soc.*, vol. 53, no. 9, pp. 977–984, Sep. 2002.

- [17] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," *Eur. J. Oper. Res.*, vol. 162, no. 1, pp. 126–141, Apr. 2005.
- [18] C.-G. Lee, M. a. Epelman, C. C. White, and Y. a. Bozer, "A shortest path approach to the multiple-vehicle routing problem with split pick-ups," *Transp. Res. Part B Methodol.*, vol. 40, no. 4, pp. 265–284, May 2006.
- [19] B. Crevier, J.-F. Cordeau, and G. Laporte, "The multi-depot vehicle routing problem with inter-depot routes," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 756–773, Jan. 2007.
- [20] G. Jeon, H. R. Leep, and J. Y. Shim, "A vehicle routing problem solved by using a hybrid genetic algorithm," *Comput. Ind. Eng.*, vol. 53, no. 4, pp. 680–692, Nov. 2007.
- [21] W. Ho, G. T. S. Ho, P. Ji, and H. C. W. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Eng. Appl. Artif. Intell.*, vol. 21, no. 4, pp. 548–557, Jun. 2008.
- [22] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, Aug. 1964.
- [23] P. Surekha and S. Sumathi, "Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms," *World Appl. Program.*, vol. 1, no. 3, pp. 118–131, 2011.
- [24] J.-F. Cordeau and M. Maischberger, "A parallel iterated tabu search heuristic for vehicle routing problems," *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012.
- [25] T. Vidal and T. G. G. Crainic, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Oper. ...*, vol. 60, no. 3, pp. 611–624, 2012.
- [26] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Comput. Oper. Res.*, vol. 40, no. 10, pp. 2519–2531, Oct. 2013.
- [27] J. W. Escobar, R. Linfati, P. Toth, and M. G. Baldoquin, "A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem," *J. Heuristics*, vol. 20, no. 5, pp. 483–509, Oct. 2014.
- [28] A. C. Rencher, *A Review Of "Methods of Multivariate Analysis, "* vol. 37, no. 11. 2005.
- [29] S. Barreto, C. Ferreira, J. Paixão, and B. S. Santos, "Using clustering analysis in a capacitated location-routing problem," *Eur. J. Oper. Res.*, vol. 179, no. 3, pp. 968–977, Jun. 2007.
- [30] A. Subramanian, "Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems." Universidade Federal Fluminense, Niteroi, Sao Pablo, pp. 60–69, 2012.