

# Un algoritmo metaheurístico para el problema de localización y ruteo con flota heterogénea

Rodrigo Linfati <sup>1</sup>, John Willmer Escobar <sup>2</sup> y Gustavo Gatica <sup>3</sup>

Recepción: 04-06-2013, Aceptación: 26-09-2013, En línea: 30-01-2014

PACS:68Qxx, 68W40  
doi:10.17230/ingciencia.10.19.3

---

## Resumen

Este artículo considera el problema de localización y ruteo con flota heterogénea (LRPH, por sus siglas en inglés), en el cual se busca determinar los depósitos a ser abiertos, los clientes a ser asignados a cada depósito, y las rutas a ser construidas para satisfacer las demandas de los clientes, considerando una flota de vehículos con capacidad diversa y costos de utilización asociados. El objetivo es minimizar la suma de los costos asociados con la apertura de depósitos, los costos de los vehículos utilizados, y los costos variables directamente relacionados con las distancias recorridas. En este artículo, se propone un algoritmo metaheurístico basado en una búsqueda tabú granular para la resolución del problema. Experimentos computacionales en instancias adaptadas de la literatura, muestran que el algoritmo propuesto es capaz de obtener, en tiempos computacionales razonables, soluciones de alta calidad demostrando su efectividad.

---

<sup>1</sup> Ph. D. in Operations Research, rlinfati@ubiobio.cl, Universidad del Bío-Bío, Chile.

<sup>2</sup> Ph. D. in Operations Research, jwescobar@javerianacali.edu.co, Pontificia Universidad Javeriana, Cali, Colombia.

<sup>3</sup> Candidato a Doctor en Ciencias de la Ingeniería USACH, ggatica@unab.cl, Universidad Andrés Bello, Santiago, Chile.

**Palabras clave:** Problema de Localización y Ruteo; Flota Heterogénea; Búsqueda Tabú Granular; Algoritmos Metaheurísticos.

---

## A Metaheuristic Algorithm for the Location Routing Problem with Heterogeneous Fleet

---

### Abstract

This paper considers the Location-Routing Problem with Heterogeneous Fleet (LRPH), in which the aim is to determine the depots to be opened, the customers to be assigned to each open depot, and the routes to be performed to fulfill the demand of the customers by considering a heterogeneous fleet with different capacities and associated costs. The objective is to minimize the sum of the cost of the open depots, of the used vehicle costs, and of the variable costs related with the distance traveled by the performed routes. In this paper, it is proposed a metaheuristic algorithm based on a granular tabu search to solve the LRPH. Computational experiments on adapted benchmark instances from the literature show that the proposed approach is able to obtain, within short computing times, high quality solutions illustrating its effectiveness.

**Key words:** Location Routing Problem; Heterogeneous Fleet; Granular Tabu Search; Metaheuristic Algorithms.

---

## 1 Introducción

La logística tiene como misión, proveer los productos o servicios apropiados, en el lugar correcto, en el momento adecuado, en la condición deseada, a un precio competitivo, contribuyendo a la obtención de la máxima utilidad y proporcionando el mayor nivel de servicio [1]. La logística comprende la integración de las actividades de abastecimiento, producción, transporte, distribución, inventario, almacenamiento, manipulación de materiales y empaque, así como el flujo de información entre ellas, convirtiéndose en uno de los factores fundamentales para alcanzar y mejorar las posiciones competitivas de las compañías.

La logística considera tres niveles de planeación de acuerdo con el horizonte de tiempo del proceso de toma de decisiones: estratégico, táctico y operacional; con horizontes superiores a un año, menor a un año y otras escalas de menor tiempo, es decir, semanas, días e incluso horas, respectivamente. Uno de los problemas abiertos en la literatura, corresponde al

ruteo de vehículos, que puede dividirse en aquellos con flota heterogénea (LRPH, por sus siglas en inglés) y aquellos que utilizan flotas homogéneas [2]. El presente artículo está enfocado en el desarrollo de un algoritmo metaheurístico para la solución de la problemática de integración de decisiones de localización de instalaciones (decisión estratégica) y ruteo de vehículos (decisión operacional) considerando flota heterogénea.

El LRPH es un problema nuevo y nuestro objetivo es atraer a otros investigadores a considerar su estudio. El único trabajo cercano a la problemática es propuesto por [3], quienes presentan una formulación matemática y un algoritmo de búsqueda de vecindario variable para el problema de ruteo de vehículos con múltiples depósitos (MDHVRP, por sus siglas en inglés). Sin embargo, en el trabajo presentado por [3] no se consideran decisiones de apertura y cierre de depósitos. Debido a lo anterior, no se cuentan con instancias reales, razón por la cual se han realizado pruebas computacionales con set de instancias adaptadas de la literatura.

En la práctica la problemática de localización y ruteo han sido estudiadas de manera independientes. Sin embargo, recientemente se han publicado algunos trabajos que demuestran que la integración de estas decisiones proporciona mejores resultados en términos de eficiencia en la cadena de suministro ([4], [5],[6] y [7]). Numerosos trabajos han considerado la problemática conjunta de localización y ruteo con flota homogénea, a pesar de ser inusual en la práctica ([2]).

El problema de localización de instalaciones está asociado a decisiones de apertura, ubicación y asignación de clientes a depósitos. Por otro lado, el ruteo de vehículos, responde al desarrollo de un conjunto de rutas, minimizando el costo de la distancia recorrida para satisfacer la demanda de los clientes. El problema de Localización y Ruteo (LRP, por sus siglas en inglés) integra ambas problemáticas. Algoritmos exactos han sido capaces de demostrar optimalidad para instancias pequeñas (menos de 100 clientes) para el LRP. Por esta razón algoritmos heurísticos y metaheurísticos han sido propuestos para resolver instancias de mayor tamaño, como las consideradas en este trabajo ([7]). En [2] se define una clasificación para el LRP de acuerdo a diferentes criterios como son la función objetivo, método de solución y periodo de planeación.

Para la mayoría de problemas de localización y ruteo de vehículos, el

número de vehículos es variable considerando una flota homogénea. De ahí que la literatura es escasa en cuanto a problemas con flota heterogénea y con decisiones de localización y ruteo simultáneamente. Algunas consideraciones de flota heterogénea son propuestas por [8], [9], [10], [11] y [12].

En [8] se aplica una descomposición Bender para dividir el LRP en dos subproblemas: localización-asignación y ruteo, se considera una flota heterogénea. Ambos subproblemas son resueltos de manera óptima en un esquema interactivo.

En [9] se considera un algoritmo de dos fases. En la fase de localización se consideran como entrada todas las posibles parejas de clientes. En la función objetivo se considera la longitud de los tours como costo un costo variable relacionado con la distancia recorrida. Estos costos son ajustados considerando una flota heterogénea de vehículos. En la fase de localización se utilizan movimientos de inserción y swap. Mientras que la fase de ruteo considera una heurística multinivel. El método propuesto en [10] es similar al anterior, excepto que ambas fases son resueltas en un algoritmo combinado de búsqueda tabú y simulado recocido considerando un solo vecindario.

En [11] se considera un problema LRP de 4 niveles, donde el nivel 1 son las plantas de fabricación, el nivel 2 los centros de distribución, el nivel 3 los puntos de transferencia y el nivel 4 los clientes finales. En este problema se incluyen decisiones de inventario con una flota heterogénea de vehículos. El problema es formulado como un modelo de programación entera mixta el cual es resuelto de manera óptima para instancias pequeñas utilizando software comercial, entre los cuales podemos nombrar CPLEX® de IBM®, y GUROBI®. Finalmente, en [12] se considera un problema de traslado desde hubs hasta clientes, considerando una flota heterogénea para realizar viajes cortos y largos.

El problema de localización y ruteo con flota heterogénea (LRPH) es considerado un problema *NP-Hard*[13] debido a que corresponde a una generalización del problema de localización y ruteo con restricciones de capacidad (CLRP, por sus siglas en inglés) y el problema de ruteo de vehículos con múltiple depósitos, considerando flota heterogénea (HMDVRP, por sus siglas en inglés). En este artículo se propone un algoritmo metaheurístico basado en una búsqueda tabú granular para resolver el LRPH. La descrip-

ción del LRPB es detallada y el marco general del algoritmo propuesto es presentado en la Sección 2. La Sección 3 muestra los resultados computacionales en instancias adaptadas de la literatura. Finalmente, se presentan las conclusiones.

## 2 Metodología Propuesta

### 2.1 Formulación del problema

El LRPB puede ser formalizado como: Sea  $G = (V, E)$  un grafo no dirigido donde  $V = \{1, \dots, m + n\}$  corresponde al conjunto de vértices y  $E$  al conjunto de arcos. Los vértices  $J = 1, \dots, m$  corresponden a los clientes con demanda  $d_j$  y los vértices  $I = m + 1, \dots, m + n$  corresponden a los depósitos potenciales, cada uno con capacidad  $w_i$  y costo de apertura  $o_i$ . Un costo  $c_{ij}$  es asociado a cada par de nodos  $(i, j) \in E$ . Un conjunto de vehículos heterogéneos  $K$ , cada uno con capacidad  $q_k$  está disponible en cada depósito  $i \in I$ . Asimismo, cada vehículo cuando realiza una ruta genera un costo fijo  $v_k$ .

El objetivo principal del LRPB es minimizar los costos totales de logística, los cuales incluyen costo fijo de apertura de depósitos, costo variable asociado a rutas establecidas y costo fijo de uso de los vehículos. El problema LRPB es mono objetivo, con restricciones, y discreto. Una solución para el LRPB es factible si las siguientes restricciones son satisfechas: i) cada ruta debe comenzar y terminar en el mismo depósito, ii) cada cliente debe ser visitado exactamente una vez por una sola ruta, iii) la suma de las demandas de los clientes visitados por cada vehículo  $k$ , no puede exceder su capacidad  $q_k$ , iv) la suma de las demandas de los clientes asignados a cada depósito abierto  $i \in I$ , no debe exceder su capacidad  $w_i$ , y v) no se permite flujo entre depósitos.

Se propone un modelo de programación lineal entera mixta de tres índices para el LRPB. El modelo extiende la formulación presentada por [4] para el problema de localización y ruteo con restricciones de capacidad y vehículos homogéneos (CLRP, por sus siglas en inglés), incluyendo las consideraciones de vehículos heterogéneos presentada en [9]. La formulación utiliza las variables binarias:

$y_i = 1$  si el depósito es abierto

$f_{ij} = 1$  si el cliente  $j \in J$  es asignado al depósito  $i \in I$

$x_{ijk} = 1$  si se utiliza el arco desde  $i \in V$  a  $j \in V$  en la ruta asignada al vehículo  $k \in K$

$$\min z = \sum_{i \in I} o_i y_i + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} v_k x_{ijk} \quad (1)$$

Sujeto a:

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{j \in J} \sum_{i \in V} d_j x_{ijk} \leq q_k \quad \forall k \in K \quad (3)$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0 \quad \forall i \in V, \forall k \in K \quad (4)$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq J, \forall k \in K \quad (6)$$

$$\sum_{u \in J} x_{iuk} + \sum_{u \in v \setminus \{j\}} x_{ujk} \leq 1 + f_{ij} \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (7)$$

$$\sum_{j \in J} d_j f_{ij} \leq w_i y_i \quad \forall i \in I \quad (8)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in V, \forall j \in V, \forall k \in K \quad (9)$$

$$y_i \in \{0, 1\} \quad \forall i \in V \quad (10)$$

$$f_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (11)$$

Note que el modelo no permite el flujo entre depósitos, por lo tanto, el costo de transporte  $c_{ij}$  no está definido para  $(i, j) \in I \times I$ . La función objetivo (1) minimiza la suma de tres costos distintos, el primero asociado a los depósitos abiertos, el segundo tiene relación con los arcos visitados por las rutas y el último con los costos fijos asociados al uso de los vehículos. Los términos de la función objetivo (1) son una extensión de los costos considerados en el CLRP por [4]. El grupo de restricciones (2) garantiza que cada cliente  $j \in J$  debe ser atendido por un solo depósito  $i \in I$  en una ruta asignada a un solo vehículo  $k \in K$ . El grupo de restricciones (3) y (8) son asociadas con las capacidades de los vehículos y los depósitos respectivamente. Restricciones (4) y (5) aseguran la continuidad de cada ruta y determina que cada una de ellas inicie y termine en el mismo depósito. La eliminación de los subtours es satisfecha a través del conjunto de restricciones (6). La restricción (7) especifica que un cliente puede ser asignado a un depósito solamente si existe una ruta que lo une. Finalmente, las restricciones (9), (10) y (11) representan las variables binarias usadas en el modelo matemático. Más detalle acerca de la notación usada en esta formulación puede ser encontrada en [14] y [15].

Al intentar resolver el modelo de manera óptima utilizando algún software comercial como CPLEX<sup>®</sup> se puede incurrir en elevados tiempo computacionales, y el consumo de memoria crecerá de forma exponencial con el número de nodos, dada la dificultad del problema [16]. Lo anterior afirma que desde el punto de vista computacional, el uso de modelos matemáticos, sin el uso de técnicas de descomposición, o algoritmos de ramificación y acotamiento, es prohibitivo. En ese sentido, aparecen las heurísticas y metaheurísticas como alternativa para la solución del LRPH. Dado que el problema considerado es relativamente nuevo, los resultados obtenidos por la metaheurística propuesta se han comparado con la solución obtenida del modelo (1) - (11), relajando las restricciones de eliminación de subtour (6). De esta forma, estas restricciones solo son añadidas en cada nueva solución entera encontrada durante el proceso de Branch-Cut desarrollado en CPLEX<sup>®</sup>.

## 2.2 Descripción del algoritmo propuesto

En esta sección, se presenta el algoritmo propuesto para resolver el LRPH. El algoritmo propuesto extiende la idea de búsqueda tabú granular (GTS, por sus siglas en inglés), introducida por [17] para el problema capacitado de ruteo de vehículos con un solo depósito, al problema LRPH. La idea general de la GTS es encontrar soluciones de gran calidad en tiempos computacionales reducidos, para ello se define una implementación efectiva de una lista de candidatos, la cual contiene los mejores vecindarios a visitar, reduciendo el tiempo computacional al utilizar solo vecindades que pueden ser examinados en un tiempo menor al requerido en la forma tradicional, preservando la calidad de las soluciones de una búsqueda tabú tradicional. En particular, buenos vecindarios son visitados, acotando la región de búsqueda mediante el descarte de una gran cantidad de soluciones vecinas “no prometedoras” y explorando sólo un subconjunto que contiene soluciones vecinas “prometedoras”.

En el algoritmo propuesto, se construye una solución inicial factible mediante un procedimiento híbrido que combina técnicas de clústeres con métodos exactos. El procedimiento inicial está fundamentado en la generación de clústeres de clientes con asignación de vehículos aleatoriamente, que permite obtener una solución factible con un tiempo de cómputo reducido. De esta manera se reduce el espacio de búsqueda del algoritmo propuesto de un problema de localización y ruteo con flota heterogénea, a un problema de múltiple depósitos con vehículos heterogéneos. El procedimiento de búsqueda tabú propuesto no considera decisiones de cierre y apertura de centros de distribución. Cuando el algoritmo es atrapado en un óptimo local por un número determinado de iteraciones, un procedimiento de perturbación es aplicado. Un óptimo local para un vecindario se alcanza cuando no existe una solución mejor que la actual, es decir, no se puede realizar una mejora moviéndose a otra solución dentro del mismo vecindario, según lo expresado por [18]. A continuación se describe el pseudocódigo del Algoritmo 1 propuesto.

---

**Algoritmo 1** Pseudocódigo del algoritmo propuesto

---

```

1: Entrada: instancia_lrph
2: Salida: rutas_lrph
3: Inicio
4:   Cargar instancia_lrph
5:   Resolver un TSP con todos los clientes
6:   Generar clústeres factibles por capacidad de vehículo del TSP anterior
7:   Asignación óptima de depósitos a clústeres
8:   Mientras no ocurra la condición de termino hacer
9:     Aplicar mejor movimiento no tabú de las vecindades
10:    Si solución actual es óptimo local, entonces
11:      Ejecutar perturbación
12:    Fin Si
13:    Actualizar lista tabú y parámetros
14:  Fin Mientras
15:  Escribir rutas
16: Fin

```

---

### 2.3 Espacio de búsqueda granular

El grafo original  $G$  se sustituye por un grafo que considera todos los arcos que tienen un costo menor a un umbral de granularidad  $\vartheta$ , los arcos incidentes a los depósitos y los arcos que pertenecen a la mejor solución encontrada durante la búsqueda. Sea  $\vartheta = \beta \bar{z}^*$  donde  $\bar{z}^*$  es el costo promedio de los arcos que pertenecen a las mejores soluciones encontradas y  $\beta$  un factor de esparsificación que controla los mecanismos de diversificación e intensificación de la búsqueda tabú. Valores grandes de  $\beta$  inducen procesos de diversificación, mientras que valores pequeños de  $\beta$  permiten procesos de intensificación.

### 2.4 Procedimiento de solución inicial

En primera instancia se construye un tour del vendedor viajero (TSP, por sus siglas en inglés) para todos los clientes utilizando la heurística Lin-

Kernighan (LKH, por sus siglas en inglés) ([19],[20]), que permite resolver el TSP de manera efectiva mediante la aplicación de diferentes estrategias de búsqueda local. A partir de dicho tour, se procede a determinar grupos de clientes (clústeres) que pueden ser atendidos en una misma ruta por un vehículo determinado. Dado el inicio con flota finita y heterogénea, el tour gigante es dividido en diferentes grupos de clientes de acuerdo a las capacidades de los vehículos disponibles mediante los siguientes pasos:

1. Seleccionar un vehículo libre  $k \in K$  de manera aleatoria para atender a un grupo de clientes (clúster).
2. Determinar un primer cliente para ser asignado a dicho vehículo  $k \in K$ .
3. Agregar uno a uno clientes nuevos siguiendo la trayectoria del TSP.
4. Una vez completada la capacidad del vehículo  $k \in K$ , se reinicia el proceso con el paso 1 hasta asignar todos los clientes a un vehículo de la flota disponible.

Luego para cada depósito  $i \in I$  y cada clúster  $g \in G$ , se aplica nuevamente el procedimiento LKH para determinar la longitud del tour ( $l_{ig}$ ) al asignar el depósito  $i$  al clúster  $g$ . Finalmente, los depósitos se asignan a los grupos de clientes mediante la solución del modelo de programación lineal entera mixta correspondiente al problema de localización de instalaciones con restricciones de *single sourcing* (FLPS, por sus siglas en inglés) ([21] y [22]). En este problema se consideran los siguientes conjuntos de variables binarias:

$x_{ig} = 1$  Si el depósito  $i \in I$  es asignado al clúster  $g \in G$

$y_i = 1$  Si se abre el depósito  $i \in I$

Donde  $G$ , corresponde al conjunto de clústeres resultantes de dividir el tour gigante en subtours y  $dc_g$  a la demanda global del clúster  $g$ , es decir, la suma de las demandas de los clientes asignados a  $g \in G$ . Así, el FLPS se puede formular mediante el modelo (12)-(16):

$$\min u = \sum_{i \in I} o_i y_i + \sum_{i \in I} \sum_{g \in G} l_{ig} x_{ig} \quad (12)$$

Sujeto a:

$$\sum_{i \in I} x_{ig} = 1 \quad \forall g \in G \quad (13)$$

$$\sum_{g \in G} dc_g x_{ig} \leq w_i y_i \quad \forall i \in I \quad (14)$$

$$x_{ig} \in \{0, 1\} \quad \forall i \in I, \forall g \in G \quad (15)$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (16)$$

La función objetivo (12), corresponde a la suma de los costos totales de apertura de los depósitos y los costos de la distancia recorrida al asignar los depósitos a los clústeres. El grupo de restricciones (13) garantiza que cada grupo de clientes sea asignado exactamente a un solo depósito. El grupo de restricciones (14) determina la capacidad de cada uno de los depósitos. Por último, las restricciones (15) y (16) determinan el conjunto de variables binarias usadas en el modelo. Note que el procedimiento inicial es repetido  $n$  veces (donde  $n$  es el número de clientes), considerando cada cliente como inicio para la división del TSP *tour*. En este procedimiento la mejor solución factible es conservada.

## 2.5 Movimientos

La cardinalidad de vecindarios es la cantidad de soluciones vecinas a la solución  $S$ . En [17] se menciona que el tiempo de ejecución de cada iteración de un algoritmo de búsqueda local es proporcional a la cardinalidad de los vecindarios y al tiempo necesario para verificar factibilidades y evaluar costos de las nuevas soluciones. En la mayoría de los casos el tiempo de cada iteración está asociado a una función que depende del tamaño de la instancia. El número de iteraciones a ser ejecutadas para alcanzar un mínimo local puede ser grande, dependiendo directamente del tamaño de la instancia dado por el número nodos.

Una forma de reducir el tiempo computacional es restringir la búsqueda a ciertas vecindades “prometedoras”, que normalmente pertenecen a

las mejores soluciones encontradas para el problema de ruteo de vehículos (VRP, por sus siglas en inglés), denominadas “vecindarios granulares”. El algoritmo propuesto utiliza los siguientes vecindarios:

- *Shift*: Un cliente es transferido desde su posición actual a otra posición, ya sea en la misma ruta o en una ruta diferente (que pertenece al mismo depósito o a otro diferente).
- *Swap*: Dos clientes de la misma o de diferente ruta intercambian su posición, ya sea en la misma ruta o en una ruta diferente (que pertenece al mismo depósito o a otro diferente).
- *Double-Shift*: Dos clientes consecutivos son transferidos de sus actuales posiciones a otras posiciones diferentes, manteniendo el arco que los conecta. Los clientes pueden ser insertados en su ruta actual o en una diferente (que pertenece al mismo depósito o a otro diferente).
- *Double-Swap*: Este movimiento es una extensión del Swap, obtenido considerando dos pares de clientes consecutivos. El arco que conecta a cada par de clientes es mantenido. Este movimiento es ejecutado entre dos rutas diferentes que pertenecen al mismo depósito o a otro diferente.

Los movimientos de cada una de las vecindades son ejecutados solo si los nuevos arcos se encuentran en la lista de candidatos.

## 2.6 Procedimiento de Búsqueda Tabú Granular

La solución inicial  $S_0$  obtenida por el procedimiento descrito en la sección 2.4 es mejorada mediante un procedimiento de búsqueda tabú granular, en el cual no se consideran vecindarios que modifiquen los depósitos abiertos. En ese sentido se busca mejorar las rutas determinadas por el procedimiento de solución inicial, permitiendo durante la Búsqueda Tabú soluciones infactibles con respecto a las capacidades de los depósitos y de los vehículos, utilizando las vecindades descritas en la Sección 2.5 y el espacio de búsqueda descrito en la Sección 2.3. La lista tabú está compuesta por los arcos eliminados de la ruta actual, permaneciendo 7 iteraciones antes de

ser descartados de la lista tabú. Los vecindarios son explorados de forma secuencial, seleccionando el mejor vecino entre todos los vecindarios.

La búsqueda tabú granular considera dos estrategias de diversificación. La primera, está relacionada con la diversificación granular. Inicialmente el factor de esparsificación  $\beta$  es ajustado a un valor pequeño  $\beta_0$ . Si la mejor solución factible encontrada no ha sido actualizada durante  $N_{imp}$  iteraciones, el valor de  $\beta$  se incrementa a un valor  $\beta_n$ . De esta manera una nueva lista de candidatos es calculada. Luego, después de  $N_{back}$  iteraciones, el valor del factor de esparsificación  $\beta$  se restablece al valor inicial de  $\beta_0$  y la búsqueda continua desde la mejor solución factible encontrada. Los valores de  $\beta_0$ ,  $N_{imp}$ ,  $\beta_n$  y  $N_{back}$  son parámetros dados en el desarrollo del algoritmo los cuales se pueden observar en la 3. La segunda estrategia de diversificación, se basa en un enfoque de penalización propuesto por [23] y [24] para el problema de ruteo de vehículos (VRP). En particular, para cada solución  $S$  se considera un conjunto de rutas  $R_1, \dots, R_k$ . Cada ruta  $R_s$ ,  $s \in \{1, \dots, K\}$ , puede ser denotada por  $(v_{s0}, v_{s1}, v_{s2}, \dots, v_{s0})$ , donde  $v_{s0}$  representa el depósito asignado a la ruta, y cada  $v_{sj}$ ,  $j \in J$ , representan los clientes visitados. Note que la solución  $S$  puede ser factible o infactible con respecto a la capacidad de los vehículos o la capacidad de los depósitos. Sea  $T$  el subconjunto de depósitos abiertos y  $D_i$  el conjunto de clientes asignados al depósito  $i$ ; la siguiente notación es usada:  $v \in R_s$ , si un cliente  $v$  pertenece a la ruta  $R_s$  y  $(u, v) \in R_s$ , si  $u$  y  $v$  son dos vértices consecutivos de la ruta  $R_s$ . La siguiente función objetivo  $f(S)$  está asociada a cada solución factible  $S$ :

$$f(S) = \sum_{i \in T} o_i + \sum_{k \in K} \sum_{(u,v) \in R_s} c_{uv} + \sum_{k \in K} v_k \quad (17)$$

De igual manera la ecuación (18), indica la función objetivo  $g(S)$  asociada con cualquier solución infactible  $S$ :

$$g(S) = f(S) + \gamma_d \sum_{i \in T} \left[ \sum_{v \in D_i} d_v - w_i \right]^+ + \gamma_r \sum_{k \in K} \left[ \sum_{v \in R_s} d_v - q_k \right]^+ \quad (18)$$

Donde  $[x]^+ = \max(0, x)$ ,  $\gamma_d$  y  $\gamma_r$  son dos pesos positivos utilizados para

incrementar el valor de la solución  $S$ , adicionando la suma de los excesos de carga de los depósitos y la suma de los excesos de demanda en las rutas, respectivamente.

Los dos pesos son calculados de la siguiente forma:  $\gamma_d = \alpha_d \times f(S_0)$  y  $\gamma_r = \alpha_r \times f(S_0)$ , donde  $f(S_0)$  es el valor de la función objetivo de la solución inicial obtenida mediante el procedimiento descrito en la sección 2.4 y  $\alpha_d$  y  $\alpha_r$  son dos parámetros ajustados durante la búsqueda tabú granular dentro del rango  $[\alpha_{min}, \alpha_{max}]$ . En particular, si no se han encontrado soluciones infactibles respecto a la capacidad de los depósitos durante  $N_{search}$  iteraciones, el valor de  $\alpha_d$  se establece como  $max\{\alpha_{min}, \alpha_d \times i_p\}$ , donde  $0 < i_p < 1$ . Por otro lado, si las soluciones encontradas durante  $N_{search}$  iteraciones son infactibles, entonces el valor de  $\alpha_d$  se establece como  $min\{\alpha_{max}, \alpha_d \times r_p\}$ , donde  $r_p > 1$ . Una regla similar se aplica para modificar el valor de  $\alpha_r$ ,  $\alpha_{min}$ ,  $\alpha_{max}$ ,  $N_{search}$ ,  $i_p$  y  $r_p$  son parámetros dados.

Finalmente, en la selección del mejor movimiento infactible para una solución  $S$  se considera el siguiente criterio: Si el valor de  $g(S)$  es menor que el costo de la mejor solución encontrada, se asigna un valor  $h(S) = g(S)$ . De lo contrario, como estrategia de diversificación se introduce una penalidad extra al valor de  $g(S)$  igual a  $\Delta_{max} \times t \times \sqrt{k}$ .  $\Delta_{max}$  corresponde a la diferencia absoluta entre dos valores sucesivos de la función objetivo  $g(S)$ ,  $t$  es un factor de escala y  $k$  es igual al número de rutas de la solución  $S$ . De esta manera se define  $h(S) = g(S) + \Delta_{max} \times t \times \sqrt{k}$ . Note que si la solución  $S$  es factible  $h(S) = f(S)$ . El movimiento con el menor valor de  $h(S)$  es ejecutado.

## 2.7 Procedimiento de perturbación

El algoritmo propuesto aplica el procedimiento propuesto en [7] para el CLRP, utilizando “distancia entre rutas” como medida de proximidad. La evaluación de la “distancia entre rutas” depende de las características de las instancias consideradas. En este nuestro caso, cada vértice del grafo  $G$  es asociado con un punto en el plano Cartesiano y el costo  $c_{ij}$  es calculado para el nodo  $(i, j)$  como la distancia Euclidiana entre los puntos asociados con los vértices  $i$  y  $j$ . Dado lo anterior, para medir la cercanía entre las rutas se utiliza sus “Centros de Gravedad”.

En particular si después de  $N_{pert}$  iteraciones el algoritmo está atrapado en un óptimo local, es aplicado el siguiente procedimiento:

- Seleccionar una primera ruta  $k_1$  aleatoriamente;
- Seleccionar la ruta más cercana a  $k_1(k_2)$ ;
- Seleccionar la ruta más cercana a  $k_2(k_3)$ , donde  $k_1 \neq k_3$ ;
- Seleccionar un cliente de la ruta  $k_1(i_1)$  y un arco de la ruta  $k_2(h_2, j_2)$  de acuerdo con el mínimo costo de insertar  $i_1$  en  $(h_2, j_2)$ ;
- Seleccionar un cliente de la ruta  $k_2(i_2)$  y un arco de la ruta  $k_3(h_3, j_3)$  de acuerdo con el mínimo costo de insertar  $i_2$  en  $(h_3, j_3)$ . En este caso  $i_2 \neq h_2$  y  $i_2 \neq j_2$ ;
- Obtener una nueva solución  $S$  realizando los siguientes pasos:
  - Remove el cliente  $i_1$  de la ruta  $k_1$  e insertar entre los vértices  $(h_2, j_2)$  de la ruta  $k_2$ .
  - Remove el cliente  $i_2$  de la ruta  $k_2$  e insertar entre los vértices  $(h_3, j_3)$  de la ruta  $k_3$ .

### 3 Resultados

El algoritmo propuesto ha sido implementado en C++, en ambiente Windows 7 Professional. Los experimentos computacionales han sido ejecutados en un Intel Core i5 (2.80 GHz) con 8 GB de memoria RAM.

Para probar la efectividad del algoritmo propuesto se han utilizado dos conjuntos de instancias adaptadas de la literatura. El primer conjunto, corresponde a instancias de [25] para el problema de ruteo de vehículos con flota heterogénea (HFVRP, por sus siglas en inglés). En HFVRP todos los depósitos son considerados para determinar el número de rutas que permitan satisfacer la demanda de los clientes. Por el contrario, en LRPH los depósitos son considerados como “potenciales” y por ende se debe decidir cuáles depósitos se deben abrir y qué clientes se deben asignar a cada uno de ellos.

**Tabla 1:** Información de las instancias adaptadas de [26]

Instancia	n	Tipo de Vehículos											
		A		B		C		D		E		F	
		$q_A$	$n_A$	$q_B$	$n_B$	$q_C$	$n_C$	$q_D$	$n_D$	$q_E$	$n_E$	$q_F$	$n_F$
13	50	40	6	60	1	70	4	120	2	200	1		
14	50	120	4	160	2	300	1						
15	50	50	4	100	3	160	2						
16	50	40	2	80	4	140	3						
17	75	50	4	120	4	200	2	350	1				
18	75	20	4	50	4	100	2	150	2	250	1	400	1
19	100	100	4	200	3	300	3						
20	100	60	6	140	4	200	3						

El primer conjunto de 8 instancias tiene las siguientes características: el número de depósitos potenciales  $m$  toma valores entre 5 y 10, el número de clientes  $n$  es de 50, 75 o 100. Las demandas y coordenadas fueron originalmente tomadas de [26], la demanda de los clientes es uniformemente distribuida entre [1, 40] y la capacidad ( $q_k$ ) de los vehículos usados, el costo de uso de los vehículos ( $v_k$ ) y el número de vehículos  $k$  es un parámetro determinado en [25]. La Tabla 1 muestra el número de clientes, el número de vehículos de cada tipo y la capacidad de cada uno de ellos.

El segundo conjunto de instancias ha sido adaptado de [4] para el CLRP. En particular se ha cambiado las características de la flota homogénea (capacidad y costo de uso del vehículo) por las características de los vehículos propuestos en [25]. Este grupo considera 16 instancias con las siguientes características: el número de depósitos potenciales toma valores de 5 y 10, el número de clientes es 20, 50 o 100, la demanda de los clientes presenta una distribución uniforme entre [11, 20] y la capacidad ( $q_k$ ) de los vehículos usados, el costo de uso de los vehículos ( $v_k$ ) y el número de vehículos  $k$  es un parámetro adaptado de [25]. La Tabla 2 muestra el número de clientes, el número de vehículos de cada tipo y la capacidad de cada uno de ellos.

### 3.1 Ajuste de parámetros

El éxito del algoritmo propuesto depende directamente de los parámetros utilizados, es por ello que al cabo de una sintonización fina de ellos, se ha

**Tabla 2:** Información de las instancias adaptadas de [4]

Instancia	n	Tipo de Vehículos											
		A		B		C		D		E		F	
		$q_A$	$n_A$	$q_B$	$n_B$	$q_C$	$n_C$	$q_D$	$n_D$	$q_E$	$n_E$	$q_F$	$n_F$
1_1a	20	40	6	60	1	70	4	120	2	200	1		
1_2a	20	40	6	60	1	70	4	120	2	200	1		
1_3a	20	40	6	60	1	70	4	120	2	200	1		
1_1b	20	20	4	30	2	40	3	70	4	120	2	200	1
1_2b	20	20	4	30	2	40	3	70	4	120	2	200	1
1_3b	20	20	4	30	2	40	3	70	4	120	2	200	1
2_1a	50	40	6	60	1	70	4	120	2	200	1		
2_2a	50	40	6	60	1	70	4	120	2	200	1		
2_1b	50	120	4	160	2	300	1						
2_2b	50	120	4	160	2	300	1						
3_1a	100	100	4	200	3	300	3						
3_1b	100	60	6	140	4	200	3						
4_1a	100	100	4	200	3	300	3						
4_2a	100	100	4	200	3	300	3						
4_1b	100	60	6	140	4	200	3						
4_2b	100	60	6	140	4	200	3						

logrado establecer que los valores presentes en la Tabla 3, permiten resolver todas las instancias consideradas en esta investigación. Los parámetros  $\beta_o$  y  $\beta_n$  permiten al algoritmo alternar entre fases de intensificación y diversificación respectivamente. El proceso de intensificación se realiza durante  $N_{imp}$  iteraciones, mientras que el proceso de diversificación se realiza durante  $N_{back}$  iteraciones. Los parámetros  $\alpha_d$ ,  $\alpha_r$ ,  $\gamma_{min}$ ,  $1/F_1(S_0)$ ,  $\gamma_{max}$ ,  $N_{search}$ ,  $i_p$ ,  $r_p$  y  $t$ , están relacionados con el esquema de penalización que permite considerar soluciones infactibles durante la búsqueda. Finalmente, el parámetro  $N_{pert}$  ayuda al algoritmo a escapar de óptimos locales. El proceso para la sintonización fina de parámetros se ha realizado de acuerdo al procedimiento propuesto en [27]. El algoritmo ha sido ejecutado por un número máximo de iteraciones igual a  $10 \times n$ .

### 3.2 Resultados sobre el conjunto de instancias

En las Tablas 4 y 5, se presentan los resultados computacionales con ambos conjuntos de instancias, la primera columna indica el problema analizado,

**Tabla 3:** Valor de los parámetros obtenidos para el conjunto de instancias

Parámetro	Valor Obtenido
$\beta_o$	1.25
$\beta_n$	2.20
$N_{imp}$	2.00 * n
$N_{back}$	1.00 * n
$\alpha_d$	0.01
$\alpha_r$	0.01
$\gamma_{min}$	$1/F_1(S_0)$
$\gamma_{max}$	0.05
$N_{search}$	10
$i_p$	2.00
$r_p$	0.50
$t$	0.01
$N_{pert}$	1.00 * n

las columnas sucesivas describen el número de rutas obtenido en la solución final ( $k'$ ), el número de depósitos abiertos ( $m'$ ), el costo de la distancia recorrida por las rutas ( $cost1$ ), el costo de los depósitos abiertos ( $cost2$ ) y el costo de los vehículos usados ( $cost3$ ). De igual manera se describe el costo total ( $total\_cost$ ) y el tiempo de ejecución (CPU time) en segundos empleado por el algoritmo propuesto ( $time$ ), respectivamente. Adicionalmente se reporta los resultados obtenidos mediante CPLEX® para el modelo relajado (1)-(11) dados por la cota inferior de la función objetivo (LB), la cota superior o mejor solución factible (UB), el tiempo de ejecución (T.CPLEX) con un límite de 15 horas y la información sobre el estado de CPLEX® al terminar (Info). TimeOut significa que no fue posible para CPLEX® encontrar la solución óptima dentro del límite de tiempo establecido, OutOfMemory significa que CPLEX® agotó la memoria disponible y termino antes del tiempo límite sin encontrar la solución óptima. Las cifras no reportadas corresponden a instancias donde CPLEX® no fue capaz de encontrar una cota, ya sea por agotamiento de la memoria o por exceso del límite de tiempo establecido.

**Tabla 4:** Resumen de resultados encontrados para el set de instancias adaptadas de [26]

Inst	k'	m'	cost1	cost2	cost3	total_cost	Time (s.)	LB	UB	T.CPLEX (s.)	Info
13	13	2	712.59	80,00	1.290,00	2.082,59	144,00	640,73	2.940,52	54.000,00	TimeOut
14	7	2	536,89	80,00	8.000,00	8.616,89	222,00	3.432,08	-	936,47	OutOfMemory
15	5	2	450,04	80,00	1.500,00	2.030,04	264,00	1.097,55	-	2.362,64	OutOfMemory
16	9	2	606,75	80,00	2.100,00	2.786,75	135,00	1.477,21	3.492,84	2.402,45	OutOfMemory
17	9	3	647,93	120,00	865,00	1.632,95	120,00	626,25	-	1.897,10	OutOfMemory
18	10	2	743,17	80,00	1.120,00	1.943,17	292,00	635,08	-	1.752,25	OutOfMemory
19	8	2	801,53	80,00	10.200,00	11.081,53	294,00	3.827,46	-	3.039,38	OutOfMemory
20	12	4	778,10	160,00	3.000,00	3.939,10	380,00	840,52	-	1.472,11	OutOfMemory

**Tabla 5:** Resumen de resultados encontrados para el set de instancias adaptadas de [4]

Inst	k	m'	cost1	cost2	cost3	total_cost	Time (s.)	LB	UB	T.CPLEX (s.)	Info
1_1a	5	2	271,85	13.911,00	400,00	14.582,85	67,00	6.124,70	22.859,99	838,72	OutOfMemory
1_1b	9	2	464,67	13.911,00	340,00	14.715,67	69,00	6.074,82	15.063,89	1.042,32	OutOfMemory
1_2a	6	3	374,48	21.158,00	455,00	21.987,48	43,00	6.367,51	22.179,20	820,00	OutOfMemory
1_2b	14	3	640,38	21.158,00	455,00	22.253,38	21,00	6.314,37	22.378,34	1.186,15	OutOfMemory
1_3a	5	2	268,04	15.497,00	565,00	16.330,04	68,00	7.245,61	16.458,07	572,91	OutOfMemory
1_3b	7	2	297,72	15.497,00	550,00	16.344,72	43,00	662,20	16.530,72	54.000,00	TimeOut
2_1a	9	2	797,18	14.773,00	1.040,00	16.610,18	128,00	722,79	-	2.166,73	OutOfMemory
2_1b	6	2	598,05	14.773,00	7.000,00	22.341,05	111,00	8.264,48	26.280,71	54.000,00	TimeOut
2_2a	13	3	673,75	16.472,00	1.280,00	18.425,75	162,00	583,74	-	2.727,24	OutOfMemory
2_2b	7	3	511,94	16.472,00	7.500,00	24.483,94	112,00	8.502,98	-	54.000,00	TimeOut
3_1a	8	3	614,94	132.890,00	9.300,00	142.804,94	329,00	5.780,15	-	54.000,00	TimeOut
3_1b	10	2	768,49	87.192,00	2.400,00	90.360,49	344,00	5.863,51	-	1.925,78	OutOfMemory
4_2a	9	3	650,49	154.942,00	10.700,00	166.292,49	612,00	5.536,25	-	54.000,00	TimeOut
4_2b	10	3	565,21	145.000,00	2.800,00	148.365,21	638,00	6.332,13	-	947,54	OutOfMemory
4_1a	7	2	636,38	96.704,00	8.100,00	105.440,38	287,00	6.241,18	-	54.000,00	TimeOut
4_1b	10	2	561,40	88.613,00	2.200,00	91.374,40	335,00	7.042,17	-	2.562,43	OutOfMemory

## 4 Conclusiones

En este artículo se propone un algoritmo efectivo basado en la metaheurística búsqueda tabú granular para el problema de localización y ruteo con restricciones de capacidad usando una flota heterogénea (LRPH). En particular, el algoritmo propuesto utiliza dos estrategias de diversificación, diversos vecindarios y un algoritmo de perturbación cuando se obtiene un óptimo local en un número determinado de iteraciones.

El algoritmo propuesto ha sido evaluado considerando instancias adaptadas de la literatura. Los experimentos computacionales muestran que el algoritmo propuesto es capaz de obtener, dentro de razonables tiempo de computación, soluciones de gran calidad. Los resultados obtenidos sugieren que el algoritmo propuesto podría ser aplicado a otros problemas de logística similares como son: el problema de ruteo con múltiples depósitos

con flota heterogénea (HMDVRP, por sus siglas en inglés), el problema de localización y ruteo con restricciones de periodicidad considerando flota heterogénea (HPLRP, por sus siglas en inglés). Como trabajo futuro, se sugiere la implementación de procedimientos de búsqueda local considerando movimientos que permitan abrir o cerrar instalaciones, de manera que se pueda explorar de mejor forma el espacio de solución del LRPH.

## Agradecimientos

El trabajo de investigación ha sido parcialmente soportado por el MIUR (Ministero Istruzione, Università e Ricerca), Italia, la Pontificia Universidad Javeriana Cali, Colombia y las Universidades del Bío-Bío y Andrés Bello, Chile. Los autores agradecen enormemente a los referees por sus valiosos comentarios.

## Referencias

- [1] R. H. Ballou, *Business Logistics/Supply Chain Management*. Prentice Hall, 2004. 56
- [2] G. Nagy and S. Salhi, “Location-routing: Issues, models and methods,” *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007. 57
- [3] S. Salhi, A. Imran, and N. A. Wassan, “The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation,” *Computers & Operations Research*, 2013. 57
- [4] C. Prins, C. Prodhon, A. Ruiz, P. Soriano, and R. W. Calvo, “Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic,” *Transportation Science*, vol. 41, no. 4, pp. 470–483, 2007. 57, 59, 61, 70, 71, 73
- [5] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon, “A GRASP  $\times$  ELS approach for the capacitated location-routing problem,” *Computers & Operations Research*, vol. 37, no. 11, pp. 1912–1923, 2010. 57

- [6] V. F. Yu, S.-W. Lin, W. Lee, and C.-J. Ting, “A simulated annealing heuristic for the capacitated location routing problem,” *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010. 57
- [7] J. Willmer Escobar, R. Linfati, and P. Toth, “A two-phase hybrid heuristic algorithm for the capacitated location-routing problem,” *Computers & Operations Research*, vol. 40, no. 1, pp. 70–79, 2013. 57, 68
- [8] J. H. Bookbinder and K. E. Reece, “Vehicle routing considerations in distribution system design,” *European Journal of Operational Research*, vol. 37, no. 2, pp. 204–213, 1988. 58
- [9] S. Salhi and M. Fraser, “An integrated heuristic approach for the combined location vehicle fleet mix problem,” *Studies in Locational Analysis*, vol. 8, pp. 3–21, 1996. 58, 59
- [10] T.-H. Wu, C. Low, and J.-W. Bai, “Heuristic solutions to multi-depot location-routing problems,” *Computers & Operations Research*, vol. 29, no. 10, pp. 1393–1415, 2002. 58
- [11] D. Ambrosino and M. Grazia Scutellà, “Distribution network design: New problems and related models,” *European Journal of Operational Research*, vol. 165, no. 3, pp. 610–624, 2005. 58
- [12] H. Gunnarsson, M. Rönnqvist, and D. Carlsson, “A combined terminal location and ship routing problem,” *Journal of the Operational Research Society*, vol. 57, no. 8, pp. 928–938, 2005. 58
- [13] M. T. C. S. JIS, *Computers and intractability A Guide to the Theory of NP-Completeness*. WH Freeman and Company, 1979. 58
- [14] G. Laporte, “The vehicle routing problem: An overview of exact and approximate algorithms,” *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992. 61
- [15] P. Toth and D. Vigo, *The vehicle routing problem*. Siam, 2002, vol. 9. 61
- [16] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, 1998. 61
- [17] P. Toth and D. Vigo, “The granular tabu search and its application to the vehicle-routing problem,” *INFORMS Journal on Computing*, vol. 15, no. 4, pp. 333–346, 2003. 62, 65

- [18] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003. 62
- [19] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations research*, vol. 21, no. 2, pp. 498–516, 1973. 64
- [20] K. Helsgaun, “An effective implementation of the lin–kernighan traveling salesman heuristic,” *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000. 64
- [21] J. Barceló and J. Casanovas, “A heuristic lagrangean algorithm for the capacitated plant location problem,” *European Journal of Operational Research*, vol. 15, no. 2, pp. 212–226, 1984. 64
- [22] J. G. Klincewicz and H. Luss, “A lagrangian relaxation heuristic for capacitated facility location with single-source constraints,” *Journal of the Operational Research Society*, pp. 495–500, 1986. 64
- [23] M. Gendreau, A. Hertz, and G. Laporte, “A tabu search heuristic for the vehicle routing problem,” *Management science*, vol. 40, no. 10, pp. 1276–1290, 1994. 67
- [24] É. Taillard, “Parallel iterative search methods for vehicle routing problems,” *Networks*, vol. 23, no. 8, pp. 661–673, 1993. 67
- [25] B. Golden, A. Assad, L. Levy, and F. Gheysens, “The fleet size and mix vehicle routing problem,” *Computers & Operations Research*, vol. 11, no. 1, pp. 49–66, 1984. 69, 70
- [26] N. Christofides and J. E. Beasley, “A tree search algorithm for the p-median problem,” *European Journal of Operational Research*, vol. 10, no. 2, pp. 196–204, 1982. 70, 73
- [27] S. P. Coy, B. L. Golden, G. C. Runger, and E. A. Wasil, “Using experimental design to find effective parameter settings for heuristics,” *Journal of Heuristics*, vol. 7, no. 1, pp. 77–97, 2001. 71