UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

DYNA
http://dyna.medellin.unal.edu.co/

A hybrid metaheuristic algorithm for the capacitated location routing problem

John Willmer Escobar [a], Rodrigo Linfati [b] & Wilson Adarme-Jaimes [c]

[a] *Departamento de Ingeniería Civil e Industrial, Pontificia Universidad Javeriana, Cali, Colombia. jwescobar@javerianacali.edu.co*
[b] *Departamento de Ingeniería Industrial, Universidad del Bío-Bío, Concepción, Chile. rlinfati@ubiobio.cl*
[c] *Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Bogotá, Colombia. wadarmej@unal.edu.co*

Received: June 22th, 2014. Received in revised form: June 17th, 2014. Accepted: June 18th, 2014.

Abstract
This paper addresses the Capacitated Location-Routing Problem (CLRP) 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. The objective is to minimize the sum of the cost of the open depots, of the used vehicle costs, and of the variable costs associated with the distance traveled by the performed routes. In this paper, a Granular Tabu Search (GTS) with different diversification strategies within a Iterated Local Search (ILS) is proposed to solve the CLRP. A shaking procedure is applied whenever the best solution found so far is not improved for a given number of iterations. Computational experiments on benchmark instances taken from the literature show that the proposed approach is able to obtain, within short computing times, high quality solutions illustrating its effectiveness.

Keywords: Location Routing Problem (LRP), Iterated Local Search (ILS), Granular Tabu Search (GTS), Metaheuristic Algorithms.

Un algoritmo metaheurístico híbrido para el problema de localización y ruteo con restricciones de capacidad

Resumen
Este artículo considera el problema de localización y ruteo con restricciones de capacidad (CLRP), en el cual la meta es determinar los depósitos a ser abiertos, los clientes a ser asignados a cada depósito abierto y las rutas a ser desarrolladas para satisfacer la demanda de los clientes. El objetivo es minimizar la suma de los costos fijos de los depósitos abiertos, el costo del uso de los vehículos, y los costos variables asociados con la distancia recorrida por las rutas. En este artículo, una búsqueda tabú granular (GTS) con diferentes estrategias de diversificación contenida en una búsqueda iterativa local (ILS) es propuesta para resolver el CLRP. Un procedimiento de perturbación es aplicado cuando la mejor solución encontrada no se puede mejorar por un número determinado de iteraciones. Experimentos computacionales sobre instancias de benchmarking tomadas de la literatura muestran que el algoritmo propuesto es capaz de obtener, con tiempos computacionales reducidos, soluciones de alta calidad mostrando su efectividad.

Palabras clave: Problema de Localización y Ruteo (LRP), Búsqueda Iterativa Local (ILS), Búsqueda Tabú Granular (GTS), Algoritmos Metaheuristicos.

1. Introducción

El problema de localización y ruteo con restricciones de capacidad (CLRP) puede ser modelado como el siguiente problema de grafos: Sea $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ un grafo indirecto, donde $\boldsymbol{V}$ es un conjunto de nodos que contienen un subconjunto $\boldsymbol{I} = \{\boldsymbol{1}, \dots, \boldsymbol{m}\}$ de depósitos potenciales y un subconjunto $\boldsymbol{J} = \{\boldsymbol{1}, \dots, \boldsymbol{n}\}$ de clientes. Cada depósito potencial $\boldsymbol{i} \in \boldsymbol{I}$ tiene una capacidad $\boldsymbol{w_i}$ y un costo de apertura $\boldsymbol{o_i}$. Cada cliente $\boldsymbol{j} \in \boldsymbol{J}$ tiene una demanda $\boldsymbol{d_j}$ la cual debe ser satisfecha por un depósito abierto. Un conjunto de vehículos idénticos, cada uno con capacidad $\boldsymbol{q}$ y costo fijo $\boldsymbol{f}$, está disponible en cada depósito $\boldsymbol{i}$. Cada arco $(\boldsymbol{i}, \boldsymbol{j}) \in \boldsymbol{E}$ es asociado con un costo de viaje $\boldsymbol{c_{ij}}$. La meta del CLRP es determinar los depósitos a ser abiertos, los clientes a ser asignados a cada depósito abierto y las rutas a ser desarrolladas para satisfacer las demandas de los clientes. Las siguientes restricciones son impuestas:

DYNA 82 (189), pp. 243-251. February, 2015 Medellín. ISSN 0012-7353 Printed, ISSN 2346-2183 Online
DOI: http://dx.doi.org/10.15446/dyna.v82n189.48552

- Cada ruta debe comenzar y finalizar en el mismo depósito;
- Cada cliente debe ser visitado exactamente una vez por una sola ruta;
- La carga total de cada ruta debe no exceder la capacidad del vehículo $\boldsymbol{q}$;
- La carga total de las rutas asignadas a un depósito abierto i no debe exceder su capacidad $\boldsymbol{w_i}$.

La función objetivo del CLRP está determinada por la suma de los costos de apertura, de los costos de viaje y de los costos fijos asociados con los vehículos usados. El CLRP es conocido por ser un problema NP-hard, desde que este es una generalización de dos problemas NP-hard: el problema de localización de instalaciones con restricciones de capacidad (CFLP) y el problema de ruteo de vehículos con múltiples depósitos (MDVRP).

Algoritmos exactos han sido propuestos para resolver el CLRP en [1-3]. Estas aproximaciones han sido capaces de probar optimalidad en instancias de tamaño mediano (menores o iguales a 150 clientes). Debido a esto, heurísticas y metaheurísticas han sido propuestas para resolver exitosamente instancias de CLRP de gran tamaño.

En [4] se propone una búsqueda tabú de dos fases para resolver el CLRP. La búsqueda tabú de dos fases itera entre la etapa de localización y la etapa de ruteo, en aras de obtener mejores soluciones para instancias de gran tamaño. En este trabajo, resultados para instancias de hasta 200 clientes son reportados. Otro algoritmo basado en una aproximación de dos fases ha sido propuesto por [5]. En la primera fase (fase de localización), los clientes son agregados en nodos llamados "súper-clientes" y el problema de localización con restricciones de capacidad es resuelto a través de una relajación Langragiana. En la segunda fase (fase de ruteo), una búsqueda tabú granular (GTS) con un vecindario es usado para resolver el correspondiente problema de ruteo de vehículos (VRP) para cada depósito.

Métodos basados en agrupación han sido propuesto por [6]. Estos métodos consideran tres etapas. En la primera etapa, el conjunto de clientes son divididos en grupos de acuerdo a la capacidad de vehículos. En la segunda etapa, un problema de viajero de negocios (TSP) es resuelto para cada grupo. Finalmente, en la tercer etapa, cada tour TSP es agrupado en "súper-clientes" para resolver el correspondiente problema de localización de instalaciones con restricciones de capacidad (CFLP).

En [7] se propone un procedimiento aleatorio goloso (GRASP) con una estrategia de conectividad de soluciones de alta calidad para el CLRP. Los mismos autores (ver [8]) han propuesto una algoritmo memético con administración de la población.

Recientemente, varios algoritmos metaheurísticos han sido propuestos por [9-13]. En [9], una búsqueda de vecindario variable con siete vecindarios ha sido propuesto. En el segundo trabajo ([10]), se propone una metaheurística basada en un GRASP híbrido con una búsqueda local evolutiva (ELS). En [11] se propone un algoritmo basado en recocido simulado con tres vecindarios aleatorios para el CLRP. En [12] se han introducido una búsqueda de vecindario variable para el problema de localización y ruteo periódico con restricciones de capacidad (PLRP) y para el CLRP. En este trabajo, una combinación de una búsqueda de vecindario variable (VNS) pura con la solución de varios ILPs es considerada para la solución del CLRP. Finalmente, en [13] se ha propuesto un algoritmo exitoso de dos fases, el cual es capaz de encontrar soluciones de alta calidad dentro de tiempos computacionales cortos.

En este artículo, se introduce una metaheurística basada en una búsqueda tabú granular dentro de una búsqueda iterativa local para el CLRP. El algoritmo propuesto explota la potencialidad de la búsqueda local iterativa que ha sido estudiada en [14] y la búsqueda tabú eficiente propuesta en [15]. En la literatura revisada hasta el momento no se ha hallado la combinación del GTS dentro de un esquema de ILS. El artículo está organizado de la siguiente forma. En la sección 2 el marco general del algoritmo es detallado. Sección 3 presenta el algoritmo propuesto, denominado GITS. Experimentos computacionales en las instancias de benchmarking son mostradas en Sección 4. Finalmente, sección 5 detalla las conclusiones e investigación futura.

2. Marco general del algoritmo

2.1. Esquema de la Búsqueda Local Iterativa (ILS)

Esta técnica ha sido propuesta a principios de los años 1980 ([16]). La búsqueda local iterativa es principalmente una versión mejorada del algoritmo Ascenso de Colina (Hill Climbing) con reinicios aleatorios. Este método realiza una búsqueda de óptimos locales, ya que realiza un tour desde un óptimo local a otro.

La ILS se basa en dos principios. Primero, el ILS no escoge un reinicio aleatorio al azar, sino que busca un punto de reinicio lo suficientemente lejano para obtener un óptimo local diferente al actual, pero a su vez, suficientemente cerca para que no ser una búsqueda aleatoria de óptimos locales. Segundo, cuando encuentra un nuevo óptimo local debe decidir si utilizarlo como nuevo punto de partida para el reinicio o no, en caso de escoger siempre el nuevo óptimo local se estaría en presencia de una caminata aleatoria, si sólo se seleccionan óptimos locales nuevos mejores que el anterior, se estaría haciendo un proceso de Ascenso de Colina (Hill Climbing). La ILS escoge un punto intermedio entre ambos extremos.

2.2. Espacio de búsqueda granular

El algoritmo propuesto utiliza la idea de un espacio de búsqueda granular introducida por [15], el cual está basado en la utilización de un grafo incompleto que contiene los arcos incidentes a los depósitos, los arcos que pertenecen a las mejores soluciones encontradas durante la búsqueda y los arcos cuyo costo es menor que un valor de $\vartheta = \beta\bar{z}$, donde $\bar{z}$, es el costo promedio de los arcos de la mejor solución factible encontrada durante la búsqueda y β es un factor de esparsificación, actualizado dinámicamente durante la búsqueda. La modificación dinámica del grafo incompleto permite al algoritmo alternar entre etapas extensas de intensificación y etapas breves de diversificación. Un valor pequeño de β es asociado con etapas de intensificación, mientras que un valor alto de β es asociado con etapas de diversificación. La idea general de la búsqueda tabú granular es encontrar soluciones de gran calidad, conservando las características de la búsqueda tabú original, dentro de tiempos de computación cortos.

2.3. Procedimiento de solución inicial

La solución inicial S_o es construida usando un algoritmo híbrido propuesto por [13]. Este algoritmo es usado para construir la solución inicial, debido a la capacidad de obtener buenas soluciones factibles dentro de tiempos de computación razonables. Primero, un tour TSP gigante es construido conteniendo todos los clientes mediante la conocida heurística Lin-Kernighan (LKH) ([17]). Luego el tour gigante es partido en varios grupos para cada uno de los cuales, la capacidad del vehículo es satisfecha.

Para cada depósito i y cada grupo g, un TSP tour es ejecutado para obtener la distancia real de asignar el depósito i al cluster g. Luego, los depósitos son asignados a los grupos resolviendo un modelo de programación entera mixta correspondiente al bien conocido problema de localización con restricciones de capacidad y fuente única (ver, e.g. [18] y [19]).

Note que el procedimiento de solución inicial es repetido n veces (donde n es el número de clientes) seleccionando cada vértice como vértice inicial para dividir el tour gigante. Finalmente, un procedimiento de división es aplicado para reducir el costo de la distancia recorrida por las rutas adicionando nuevas rutas y asignándolas a diferentes depósitos [13]. El procedimiento de división es aplicado dos veces manteniendo la solución mejor solución factible encontrada.

2.4. Vecindarios

El algoritmo propuesto permite soluciones infactibles con respecto a las capacidades de los vehículos y de los depósitos. Dada una solución factible S durante la búsqueda tabú granular, se asigna un valor de la función objetivo $F_1(S)$. El valor de $F_1(S)$ es igual a la suma de los costos fijos de los depósitos abiertos, del costo de la distancia recorrida por los arcos que pertenecen a las rutas de la solución S, y del costo fijo de los vehículos usados en S. En adición, para cada solución infactible S con respecto a la capacidad del depósito, se adiciona a $F_1(S)$ un término de penalización obtenido por la multiplicación de la capacidad extra del depósito por un parámetro de penalización dinámicamente actualizado durante la búsqueda $\propto_d$. De esta manera, el valor de la función objetivo $F_2(S)$ es obtenido. Un procedimiento similar es aplicado al valor de la función objetivo para cualquier solución S infactible con respecto a la capacidad de los vehículos usando un factor de penalización $\propto_r$. Note que si la solución S es factible $F_2(S) = F_1(S)$.

En particular, si se han encontrado soluciones infactibles respecto a la capacidad del depósito durante N_{fact} iteraciones, el valor del factor de penalización $\propto_d$ es calculado como el $\min\{\alpha_{max}, \propto_d \times \delta_{inc}\}$, donde $\delta_{inc} > 1$. De lo contrario, si soluciones factibles se han encontrado con respecto a la capacidad del depósito durante N_{fact} iteraciones, el valor del factor de penalización $\propto_d$ es calculado como $\max\{\alpha_{min}, \propto_d \times \delta_{red}\}$, donde $\delta_{red} < 1$. N_{fact}, $\propto_d$, α_{max}, α_{min}, δ_{inc} y δ_{red} son parámetros ajustados. Una aproximación similar es aplicada para calcular el valor del factor de penalización $\propto_r$. El algoritmo metaheurístico aplica movimientos entre rutas y al interior de las rutas correspondiente a los siguientes vecindarios:

- **Inserción:** Un cliente es transferido de su posición actual a otra posición, dentro de la misma ruta o en una ruta diferente (que pertenece al mismo depósito o a otro diferente).
- **Intercambio:** Un par de clientes son intercambiados al mismo tiempo en la misma ruta o en una ruta diferente (que pertenece al mismo depósito o a otro diferente).
- **Two-opt:** Esta es una extensión del movimiento tradicional Two opt. Si los dos clientes pertenecen a la misma ruta, el movimiento es considerado como el bien conocido movimiento intra ruta Two-opt ([17]). Si los clientes pertenecen a una ruta diferente que pertenece a un mismo depósito, el movimiento es similar al tradicional Two-opt en el problema de ruteo de vehículos (VRP). De lo contrario, si cada par de clientes pertenecen a depósitos diferentes, es necesario desarrollar un movimiento adicional con los arcos incidentes a los depósitos para reconectar las rutas de forma correcta.
- **Doble inserción:** Dos clientes son transferidos de su posición actual a otra posición manteniendo el arco que los conectan. Los clientes pueden ser insertados en la ruta actual o en una ruta diferente (perteneciendo al mismo depósito o a uno diferente).
- **Doble intercambio:** Un par de arcos son intercambiados al mismo tiempo. Los arcos puedes estar en la misma ruta o en una ruta diferente (que pertenece al mismo depósito o a otro diferente).

Un movimiento es realizado sólo si los nuevos arcos insertados en la solución están en el espacio de búsqueda granular.

3. Descripción del algoritmo propuesto

Esta sección presenta la discusión del algoritmo desarrollado (GITS) para la solución del CLRP. El algoritmo propuesto utiliza una búsqueda tabú granular (GTS) internamente dentro de una búsqueda iterativa local (ILS) para las estructuras de vecindario descritas en la sección anterior. En particular, ILS controla la búsqueda global; mientras que el GTS guía el proceso de búsqueda local usando los diferentes vecindarios y el eficiente espacio de búsqueda descrito en la sección 2.2.

En particular, después de construir la solución inicial S_0 (mediante el procedimiento descrito en la sección 2.3), GITS itera a través de diferentes vecindarios para mejorar la mejor solución factible encontrada durante la búsqueda. El algoritmo propuesto selecciona un vecindario de las estructuras descritas en la sección 2.4, y aplica una búsqueda tabú granular (GTS) en el vecindario seleccionado hasta que un mínimo local es encontrado. Luego el vecindario es reiniciado aleatoriamente y de nuevo el procedimiento GTS es aplicado. El algoritmo es ejecutado por N_{stop} iteraciones (donde N_{stop} es un parámetro dado).

El procedimiento GTS explora el espacio de solución moviéndose en cada iteración de una solución S a la mejor solución (solución con el menor valor en el vecindario $N(S)$, de $F_2(S)$, inclusive si esta es infactible). El último movimiento es declarado como tabú. El tamaño de la lista tabú es aleatoriamente ajustado en el intervalo $[t_{min}, t_{max}]$, donde t_{min} y t_{max} son paramétricos.

El GTS considera tres estrategias de diversificación. La primera estrategia está basada en la diversificación granular propuesta por [15]. Inicialmente el factor de esparsificación es ajustado a un valor pequeño β_0. Cuando la mejor solución factible no es mejorada después de N_{beta} iteraciones, el factor de esparsificación β es incrementado a β_d. Luego un nuevo grafo es calculado, y N_{change} iteraciones son desarrollados comenzando de la mejor solución factible encontrada. Finalmente, el factor de esparsificación es reseteado a su valor original β_0 y la búsqueda continua. β_0, β_d, N_{beta} y N_{change} son paramétricos.

La segunda estrategia de diversificación está basada en un esquema de penalización: en la selección del mejor movimiento a ser desarrollado para cualquier solución infactible S, un término extra de penalidad (φ) es adicionado al valor de la función objetivo $F_2(S)$. El valor de φ es igual al producto de la diferencia absoluta Δ_{obj} entre dos valores sucesivos de la función objetivo, de la raíz cuadrada del número de rutas r, y del factor de escala g ([20]). Finalmente, la última estrategia de diversificación determina cada N_{third} iteraciones (donde N_{third} es un parámetro dado) una solución factible usando el procedimiento VRPH propuesto por [13] para el problema del CLRP.

Finalmente, un procedimiento de perturbación es aplicado cuando el GTS se mantiene en óptimo local por un N_{shak} número de iteraciones (donde N_{shak} es un parámetro dado). En particular, este procedimiento aplica el movimiento de inserción considerando tres rutas aleatorias al mismo tiempo (por mayores detalles ver [21]). En particular, nosotros consideramos un procedimiento aleatorio de selección de las rutas a ser perturbadas. El algoritmo selecciona la primera ruta (k1) aleatoriamente. La segunda ruta (k2) es el vecino más cercano de k1, y la tercera ruta (k3) es el vecino más cercano de k2, diferente a k1. La evaluación de la distancia entre rutas se calcula teniendo en cuenta su "centro de gravedad".

Para cada cliente i1 de la ruta k1, cada cliente i2 de la ruta k2, cada arco (h2,j2) de la ruta k2 (con h2≠i2 y j2≠i2), y cada arco (h3,j3) de la ruta k3, se obtiene una nueva solución S considerando los siguientes movimientos: 1) remover cliente i1 de ruta k1 e insertarlo entre vértices h2 y j2 en ruta k2; 2) eliminar cliente i2 de ruta k2 e insertarlo entre vértices h3 y j3 en ruta k3. El movimiento asociado con el valor mínimo de la función objetivo es implementado. El procedimiento de perturbación permite al algoritmo explorar nuevas regiones del espacio de solución.

3.1. Pseudocódigo del algoritmo propuesto

```
S <- Solución Inicial
N <- Vecindario Inicial

WHILE (Condición de Término)
        S' = GTS(S, N)
        SI (S' es mejor que S_best)
                S_best = S'
        SI (S' cumple criterio de aceptación)
                S = S'
        S = Reinicio(S)
    N = Reinicio(N)
        S = Perturbación(S)
```

4. Descripción del algoritmo propuesto

El algoritmo propuesto (GITS) fue implementado en C++ y los experimentos han sido desarrollados en una Intel Core Duo (un solo core ha sido usado) CPU (2.00 Ghz) bajo Linux Ubuntu 11.04 con 2 GB de memoria. El desarrollo del algoritmo propuesto ha sido testeado en tres conjuntos de instancias propuestas en [4,22,23]. En todos los conjuntos, los clientes y los depósitos son representados por puntos en el plano. De esta manera, el costo de un arco es calculado como la distancia euclidiana, multiplicada por 100 y calculado como un entero (ver [22]), o calculado como número real de doble precisión (ver [4,23]).

El conjunto de datos propuesto por [23] considera 36 instancias con depósitos sin restricciones de capacidad. El número de clientes es ajustado en el intervalo (100, 200), y el número de depósitos potenciales está entre 10 o 20. La capacidad del vehículo es ajustado a 150. El conjunto de datos propuesto en [22] considera 30 instancias. El número de clientes es ajustado en el intervalo [20, 200], y el número de depósitos potenciales es 5 o 10. La capacidad del vehículo es 70 o 150. Finalmente, el conjunto de datos propuesto en [4] contiene 13 instancias obtenidas desde problemas clásicos del VRP adicionando nuevos depósitos con sus correspondientes capacidades y costos fijos. El número de clientes está entre 21 a 150, y el número de depósitos potenciales de 5 a 10.

4.1. Ajuste de parámetros

En particular, para cada instancia, una corrida del algoritmo ha sido ejecutada. Los mejores resultados del algoritmo son reportados en Tablas 1-4. El siguiente set de parámetros ha sido obtenido después de varias corridas en los tres set de instancias: $N_{fact} = 10$, $\propto_d = 0.0075F_1(S_0)$, $\propto_r = 0.0050F_1(S_0)$, $\alpha_{max} = 0.04F_1(S_0)$, $\alpha_{min} = 1.00$, $\delta_{inc} = 2.00$, $\delta_{red} = 0.30$, $t_{min} = 3$, $t_{max} = 6$, $\beta_0 = 1.80$, $\beta_d = 2.40$, $N_{beta} = 2.00n$, $N_{change} = 1.00n$, $g = 0.01$, $N_{third} = 1.50n$, $N_{shak} = 0.20n$, y $N_{stop} = 7n$. Estos valores han sido usados para las soluciones de las instancias consideradas.

4.2. Estudio comparativo

El algoritmo propuesto ha sido comparado con los siguientes algoritmos publicados para el CLRP: GRASP ([7]), MA|PM ([8]), LRGTS ([5]), GRASP+ELS ([10]), SALRP ([11]), VNS+VLNS ([12]) y 2-Phase HGTS ([13]). Los resultados se presentan según aparecen en la literatura. Note que los resultados promedio no están disponibles para

Tabla 1.Resultados resumidos en Gap BR y Tiempo CPU para el conjunto completo de instancias

Set	Size	GRASP		MA|PM		LRGTS		GRASP+ELS		SALRP		VNS+VLNS		2-Phase HGTS		GITS	
		Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU	Gap BR	Tiempo CPU
Tuzun-Burke	36	2,91	163	1,29	207	1,27	22	0,72	607	0,92	826	--	--	0,57	392	0,49	201
Prodhon	30	3,49	97	1,27	96	0,63	18	0,96	258	0,31	422	0,76	7	0,42	176	0,47	91
Barreto	13	1,58	20	2,01	36	1,61	18	0,03	188	0,25	161	--	--	0,74	105	0,93	53
Promedio Global		**2,91**	**114**	**1,40**	**137**	**1,08**	**20**	**0,70**	**405**	**0,58**	**564**	--	--	**0,54**	**263**	**0,56**	**135**
Total NBR		**8**		**18**		**8**		**35**		**33**		**2**		**26**		**35**	
CPU		**Pentium 4 (2.4 Ghz)**		**Pentium 4 (2.4 Ghz)**		**Pentium 4 (2.4 Ghz)**		**Core 2 Quad (2.83 Ghz)**		**Core 2 Quad (2.66 Ghz)**		**Core 2 Quad (2.83 Ghz)**		**Core 2 Duo (2.00 Ghz)**		**Core 2 Duo (2.00 Ghz)**	
Índice CPU		**314**		**314**		**314**		**4373**		**4046**		**4373**		**1398**		**1398**	

Fuente: Elaboración Propia

todos los algoritmos. Los resultados reportados para GRASP, MA|PM, LRGTS, SALRP, y 2-Phase HGTS corresponde a una sola corrida del algoritmo asociado. VNS+VLNS ha sido ejecutado 30 veces para cada instancia, y el costo reportado y tiempo computacional corresponde al costo promedio y tiempo promedio sobre las corridas. Finalmente, GRASP+ELS ha sido ejecutado cinco veces considerando cinco diferentes semillas, y los costos reportados corresponden a las mejores soluciones sobre las corridas y el tiempo computacional requerido para alcanzar dichas soluciones. En Tablas 1-4, la siguiente notación es usada:

- Instancia: nombre de la instancia;
- *n*: número de clientes;
- *m*: número de depósitos potenciales;
- Costo: costo de la solución obtenida por cada método (en una sola corrida o la mejor corrida);
- BR: costo del mejor resultado entre los algoritmos considerados;
- CPU: CPU usado por cada algoritmo;
- Índice CPU: Passmark performance test para cada CPU;
- Tiempo CPU: tiempo de ejecución en segundos en la CPU usada por cada algoritmo;
- Gap BR: porcentaje gap del costo de la solución encontrada por cada algoritmo con respecto al BR;
- NBR: número de BR obtenidos por cada algoritmo.

En Tablas 2-4, los valores para los cuales cada algoritmo es capaz de encontrar el valor de BR, son escritos en negrilla. Cuando el algoritmo propuesto (GITS) mejora el valor de BR, este resultado es subrayado. Finalmente, el índice CPU es obtenido por el Passmark Performance Test[1]. Este es un bien conocido benchmark test. Una CPU es más rápida cuando el índice CPU índex es alto. En Tabla 1, se reporta un resumen de resultados obtenidos por los algoritmos considerados para el set completo de instancias. En particular, valores promedio del Gap BR, CPU time, Total NBR y CPU índex son reportados. Tabla 1 muestra que el algoritmo propuesto provee el valor promedio más pequeño de Gap BR que el reportado por GRASP, MA|PM, LRGTS, GRASP+ELS, SALRP y VNS+VLNS. Solamente algoritmo 2-Phase HGTS provee, con mayores CPU times, ligeramente mejores valores promedio de Gap BR.

En cuanto al tiempo de CPU, GITS es más rápido que GRASP+ELS, SALRP y 2-Phase HGTS, los cuales han sido los algoritmos previamente publicados que han sido capaces de obtener los mejores resultados. Es de notar que el CPU time usado por algoritmo GRASP+ELS no representa el tiempo global requerido para encontrar la mejor solución, desde que este corresponde al tiempo de CPU utilizado para encontrar la mejor solución en su respectiva corrida. Por otro lado, algoritmos GRASP, MA|PM, LRGTS and VNS+VLNS toman menos CPU time, pero ellos parecen ser menos robustos que el algoritmo propuesto en términos de la calidad de solución.

Además, el algoritmo propuesto junto con el GRASP+ELS son capaces de obtener el mayor número de BKS. El algoritmo propuesto ha sido capaz de alcanzar 35 BR de 79 instancias consideradas en este artículo.

Los resultados para el conjunto de datos propuesto en [4] son reportados en Tabla 2. Los resultados muestran que el algoritmo propuesto es capaz de producir 18 BR sobre 36 instancias; de las cuales 12 corresponden a nuevos valores de BR. En este conjunto de datos, se puede observar que el algoritmo supera todas las otras heurísticas en términos del promedio Gap BR y número global de soluciones de BR encontradas.

Los resultados para el conjunto de datos propuestos en [22] son detallados en Tabla 3. En promedio, el algoritmo propuesto es capaz de obtener mejores resultados que GRASP, MA|PM, LRGTS, GRASP+ELS y VNS+VLNS. Sólo, SALRP y 2-Phase HGTS proveen también con tiempos computacionales mayores, ligeramente mejores resultados de Gap BR.

Finalmente, Tabla 4 reporta los resultados obtenidos para el set de instancias propuestas en Barreto [23]. Los resultados muestran que el algoritmo propuesto es competitivo en términos de la calidad de la solución. Algoritmo GITS es capaz de obtener mejores resultados promedios que GRASP, MA|PM y LRGTS, pero es superado por algoritmos GRASP+ELS, SALRP y 2-Phase HGTS.

[1] Passmark® Software Pty Ltd, http://www.passmark.com

Tabla 2. Resultados para las instancias propuesta por [4]

Instance	BR	GRASP			MA|PM			LRGTS			GRASP + ELS			SALRP			2-Phase HGTS			GITS		
		Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU
111112	**1473,36**	1525,25	3,52	33	1493,92	1,40	33	1490,82	1,19	3	**1473,36**	0,00	233	1477,24	0,26	369	1479,21	0,40	152	1479,21	0,40	84
111122	**1449,20**	1526,90	5,36	41	1471,36	1,53	36	1471,76	1,56	8	**1449,20**	0,00	9	1470,96	1,50	274	1486,27	2,56	239	1491,09	2,89	126
111212	**1396,59**	1423,54	1,93	28	1418,83	1,59	36	1412,04	1,11	4	**1396,59**	0,00	112	1408,65	0,86	231	1407,26	0,76	120	1406,55	0,71	74
111222	**1432,29**	1482,29	3,49	36	1492,46	4,20	36	1443,06	0,75	8	**1432,29**	0,00	114	**1432,29**	0,00	420	1474,01	2,91	146	1460,03	1,94	99
112112	**1167,16**	1200,24	2,83	28	1173,22	0,52	33	1187,63	1,75	8	**1167,16**	0,00	27	1177,14	0,86	348	**1167,16**	0,00	232	1188,66	1,84	83
112122	**1102,24**	1123,64	1,94	34	1115,37	1,19	43	1115,95	1,24	8	**1102,24**	0,00	259	1110,36	0,74	342	**1102,24**	0,00	224	**1102,24**	0,00	105
112212	**791,66**	814,00	2,82	23	793,97	0,29	38	813,28	2,73	5	792,03	0,05	5	**791,66**	0,00	360	**791,66**	0,00	201	803,27	1,47	96
112222	**728,30**	747,84	2,68	38	730,51	0,30	49	742,96	2,01	6	**728,30**	0,00	48	731,95	0,50	418	**728,30**	0,00	254	**728,30**	0,00	126
113112	**1238,49**	1273,10	2,79	23	1262,32	1,92	38	1267,93	2,38	4	1240,39	0,15	55	**1238,49**	0,00	300	**1238,49**	0,00	160	**1238,49**	0,00	82
113122	**1246,00**	1272,94	2,16	36	1251,32	0,43	48	1256,12	0,81	6	**1246,00**	0,00	233	1247,28	0,10	428	1251,22	0,42	237	1247,27	0,10	127
113212	**902,26**	912,19	1,10	20	903,82	0,17	35	913,06	1,20	4	902,30	0,00	249	**902,26**	0,00	291	**902,26**	0,00	135	**902,26**	0,00	71
113222	**1018,29**	1022,51	0,41	38	1022,93	0,46	63	1025,51	0,71	5	**1018,29**	0,00	196	1024,02	0,56	316	**1018,29**	0,00	157	**1018,29**	0,00	85
131112	**1933,67**	2006,70	3,78	113	1959,39	1,33	129	1946,01	0,64	13	1944,57	0,56	518	1953,85	1,04	743	1961,75	1,45	485	**1933,67**	0,00	179
131122	**1852,35**	1888,90	1,97	161	1881,67	1,58	144	1875,79	1,27	19	1864,24	0,64	705	1899,05	2,52	835	1856,51	0,22	298	**1852,35**	0,00	173
131212	**1983,09**	2033,93	2,56	100	1984,25	0,06	111	2010,53	1,38	11	1992,41	0,47	727	2057,53	3,75	456	2012,69	1,49	406	**1983,09**	0,00	184
131222	**1801,39**	1856,07	3,04	133	1855,25	2,99	144	1819,89	1,03	16	1835,25	1,88	415	**1801,39**	0,00	833	1803,01	0,09	302	1803,01	0,09	175
132112	**1443,80**	1508,33	4,47	118	1448,27	0,31	168	1448,65	0,34	23	1453,78	0,69	103	1453,30	0,66	750	1445,25	0,10	449	**1443,80**	0,00	186
132122	**1444,17**	1456,82	0,88	166	1459,83	1,08	155	1492,86	3,37	28	**1444,17**	0,00	662	1455,50	0,78	828	1452,07	0,55	493	1452,07	0,55	210
132212	**1204,42**	1240,40	2,99	134	1207,41	0,25	201	1211,07	0,55	19	1219,86	1,28	459	1206,24	0,15	752	**1204,42**	0,00	270	**1204,42**	0,00	128
132222	**931,49**	940,80	1,00	143	934,79	0,35	196	936,93	0,58	14	945,81	1,54	224	934,62	0,34	842	**931,49**	0,00	335	931,90	0,04	177
133112	**1701,34**	1736,90	2,09	93	1720,30	1,11	144	1729,31	1,64	18	1712,11	0,63	271	1720,81	1,14	742	1705,36	0,24	444	**1701,34**	0,00	182
133122	**1402,94**	1425,74	1,63	128	1429,34	1,88	156	1424,59	1,54	19	**1402,94**	0,00	524	1415,85	0,92	833	1416,74	0,98	342	1416,74	0,98	175
133212	**1203,44**	1223,70	1,68	89	**1203,44**	0,00	154	1216,32	1,07	15	1214,82	0,95	251	1216,84	1,11	756	1234,83	2,61	526	1235,05	2,63	207
133222	**1155,96**	1231,33	6,52	135	1158,54	0,22	223	1162,16	0,54	14	**1155,96**	0,00	375	1159,12	0,27	837	1156,05	0,01	380	1156,05	0,01	208
121112	**2258,02**	2384,01	5,58	385	2293,99	1,59	523	2296,52	1,71	41	2295,90	1,68	655	2324,10	2,93	1328	2265,59	0,34	522	**2258,02**	0,00	315
121122	**2166,20**	2288,09	5,63	410	2277,39	5,13	458	2207,50	1,91	40	2203,57	1,73	432	2258,16	4,25	1455	2166,43	0,01	603	**2166,20**	0,00	300
121212	**2239,65**	2273,19	1,50	311	2274,57	1,56	378	2260,87	0,95	33	2246,39	0,30	1566	2260,30	0,92	1319	2249,40	0,44	527	**2239,65**	0,00	287
121222	**2236,73**	2345,10	4,85	419	2376,25	6,24	436	2259,52	1,02	40	2265,53	1,29	2192	2326,53	4,01	1428	2237,81	0,05	558	**2236,73**	0,00	351
122112	**2103,82**	2137,08	1,58	338	2106,26	0,12	351	2120,76	0,81	48	2106,47	0,13	1521	2112,65	0,42	1320	2121,93	0,86	522	**2103,82**	0,00	278
122122	**1722,99**	1807,29	4,89	370	1771,53	2,82	378	1737,81	0,86	59	1779,05	3,25	618	**1722,99**	0,00	1400	1749,10	1,52	691	1750,66	1,61	433
122212	**1467,54**	1496,75	1,99	243	**1467,54**	0,00	323	1488,55	1,43	38	1474,25	0,46	514	1469,10	0,11	1299	1473,27	0,39	724	1469,45	0,13	318
122222	**1082,59**	1095,92	1,23	309	1088,00	0,50	505	1090,59	0,74	39	1085,69	0,29	1243	1088,64	0,56	1429	**1082,59**	0,00	616	1084,28	0,16	349
123112	**1969,38**	2044,66	3,82	283	1973,28	0,20	413	1984,06	0,75	43	2004,33	1,77	1451	1994,16	1,26	1318	1984,77	0,78	542	**1969,38**	0,00	261
123122	**1932,05**	2090,95	8,22	399	1979,05	2,43	406	1986,49	2,82	53	1964,40	1,67	1273	**1932,05**	0,00	1412	1958,98	1,39	617	1972,75	2,11	344
123212	**1778,28**	1788,70	0,59	199	1782,23	0,22	353	1786,79	0,48	34	1778,80	0,03	1398	1779,10	0,05	1314	1778,41	0,01	697	**1778,28**	0,00	349
123222	**1390,87**	1408,63	1,28	296	1396,24	0,39	530	1401,16	0,74	43	1453,82	4,53	2202	1396,42	0,40	1427	**1390,87**	0,00	518	1392,26	0,10	317
Promedio			**2,91**	**163**		**1,29**	**207**		**1,27**	**22**		**0,72**	**607**		**0,92**	**826**		**0,57**	**392**		**0,49**	**201**
NBR			**0**			**2**			**0**			**11**			**7**			**11**			**18**	

Fuente: Elaboración Propia

Tabla 3. Resultados para las instancias propuesta por [22]

Instance	BR	GRASP			MA|PM			LRGTS			GRASP + ELS			SALRP			VNS+VLNS			2-Phase HGTS			GITS		
		Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU
20-5-1a	54793	55021	0,42	0	54793	0,00	0	55131	0,62	1	54793	0,00	0	54793	0,00	20	54890	0,18	1	54793	0,00	3	54793	0,00	2
20-5-1b	39104	39104	0,00	0	39104	0,00	0	39104	0,00	0	39104	0,00	0	39104	0,00	15	39104	0,00	1	39104	0,00	4	39104	0,00	3
20-5-2a	48908	48908	0,00	0	48908	0,00	1	48908	0,00	1	48908	0,00	0	48908	0,00	19	48933	0,05	1	48945	0,08	3	48945	0,08	2
20-5-2b	37542	37542	0,00	0	37542	0,00	0	37542	0,00	0	37542	0,00	0	37542	0,00	15	37542	0,00	1	37542	0,00	4	37611	0,18	3
50-5-1a	90111	90632	0,58	2	90160	0,05	4	90160	0,05	0	90111	0,00	3	90111	0,00	75	90389	0,31	2	90402	0,32	27	90785	0,75	13
50-5-1b	63242	64761	2,40	2	63242	0,00	5	63256	0,02	1	63242	0,00	0	63242	0,00	58	63472	0,36	3	64073	1,31	27	63936	1,10	9
50-5-2a	88298	88786	0,55	3	88298	0,00	5	88715	0,47	2	88643	0,39	11	88298	0,00	95	89859	1,77	2	89342	1,18	23	89342	1,18	12
50-5-2b	67308	68042	1,09	3	67893	0,87	5	67698	0,58	2	67308	0,00	16	67340	0,05	59	68013	1,05	4	68479	1,74	21	68438	1,68	10
50-5-2bis	84055	84055	0,00	2	84055	0,00	5	84181	0,15	3	84055	0,00	0	84055	0,00	75	84208	0,18	3	84055	0,00	23	84126	0,08	8
50-5-2bbis	51822	52059	0,46	3	51822	0,00	6	51992	0,33	1	51822	0,00	11	51822	0,00	66	52131	0,60	5	52087	0,51	29	52270	0,86	9
50-5-3a	86203	87380	1,37	3	86203	0,00	5	86203	0,00	0	86203	0,00	0	86456	0,29	74	86727	0,61	2	86203	0,00	66	86203	0,00	18
50-5-3b	61830	61890	0,10	3	61830	0,00	8	61830	0,00	1	61830	0,00	0	62700	1,41	58	62095	0,43	3	61830	0,00	38	61885	0,09	20
100-5-1a	276137	279437	1,20	28	281944	2,10	33	277935	0,65	9	276960	0,30	148	277035	0,33	349	278292	0,78	5	276186	0,02	157	276137	0,00	75
100-5-1b	214885	216159	0,59	24	216656	0,82	44	214885	0,00	9	215854	0,45	68	216002	0,52	269	216286	0,65	8	214892	0,00	136	216878	0,93	59
100-5-2a	194124	199520	2,78	18	195568	0,74	45	196545	1,25	3	194267	0,07	212	194124	0,00	349	195022	0,46	4	194625	0,26	145	194920	0,41	76
100-5-2b	157150	159550	1,53	23	157325	0,11	45	157792	0,41	4	157375	0,14	125	157150	0,00	212	158217	0,68	5	157319	0,11	193	157364	0,14	82
100-5-3a	200242	203999	1,88	21	201749	0,75	36	201952	0,85	3	200345	0,05	141	200242	0,00	250	201748	0,75	4	201086	0,42	163	200788	0,27	69
100-5-3b	152467	154596	1,40	20	153322	0,56	43	154709	1,47	3	152528	0,04	221	152467	0,00	197	154917	1,61	4	153663	0,78	168	153617	0,75	68
100-10-1a	287864	323171	12,27	38	316575	9,97	31	291887	1,40	14	301418	4,71	48	291043	1,10	270	291775	1,36	9	289755	0,66	277	287864	0,00	203
100-10-1b	234210	271477	15,91	30	270251	15,39	45	235532	0,56	14	269594	15,11	186	234210	0,00	203	238058	1,64	10	238002	1,62	152	237632	1,46	117
100-10-2a	243778	254087	4,23	39	245123	0,55	31	246708	1,20	15	243778	0,00	260	245813	0,83	261	245614	0,75	6	245768	0,82	92	245768	0,82	52
100-10-2b	203988	206555	1,26	30	205052	0,52	39	204435	0,22	10	203988	0,00	139	205312	0,65	199	205718	0,85	7	204252	0,13	99	204252	0,13	42
100-10-3a	250882	270826	7,95	35	253669	1,11	36	258656	3,10	14	253511	1,05	164	250882	0,00	338	255140	1,70	7	254716	1,53	125	255070	1,67	82
100-10-3b	204815	216173	5,55	40	204815	0,00	45	205883	0,52	11	205087	0,13	203	205009	0,09	240	207410	1,27	7	205837	0,50	144	207171	1,15	78
200-10-1a	476778	490820	2,95	518	483497	1,41	431	481676	1,03	63	486467	2,03	1521	481002	0,89	1428	481141	0,92	18	476778	0,00	671	478154	0,29	320
200-10-1b	377716	416753	10,34	379	380044	0,62	579	380613	0,77	60	382329	1,22	359	383586	1,55	1336	381516	1,01	16	378289	0,15	476	377716	0,00	239
200-10-2a	449006	512679	14,18	554	451840	0,63	351	453353	0,97	60	452276	0,73	112	450848	0,41	1796	452374	0,75	17	449951	0,21	483	449006	0,00	231
200-10-2b	374717	379980	1,40	368	375019	0,08	401	377351	0,70	78	376027	0,35	1610	376674	0,52	1245	376836	0,57	16	374961	0,07	530	374717	0,00	290
200-10-3a	471978	496694	5,24	425	478132	1,30	266	476684	1,00	78	478380	1,36	1596	473875	0,40	1776	475344	0,71	18	472321	0,07	624	471978	0,00	330
200-10-3b	362827	389016	7,22	290	364834	0,55	341	365250	0,67	74	365166	0,64	591	363701	0,24	1326	365705	0,79	15	363252	0,12	389	362827	0,00	214
Promedio			3,49	97		1,27	96		0,63	18		0,96	258		0,31	422		0,76	7		0,42	176		0,47	91
NBR			4			11			6			13			15			2			7			10	

Fuente: Elaboración Propia

Tabla 4. Resultados para las instancias propuesta por [23]

Instance	BR	GRASP			MA|PM			LRGTS			GRASP + ELS			SALRP			2-Phase HGTS			GITS		
		Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU	Cost	Gap BR	Tiempo CPU
Christofides69-50x5	**565,6**	599,1	5,92	3	**565,6**	0,00	4	586,4	3,68	3	**565,6**	0,00	8	**565,6**	0,00	53	580,4	2,62	45	582,7	3,02	22
Christofides69-75x10	**848,9**	861,6	1,50	10	866,1	2,03	9	863,5	1,72	10	850,8	0,22	86	**848,9**	0,00	127	**848,9**	0,00	94	865,2	1,92	45
Christofides69-100x10	**833,4**	861,6	3,38	26	850,1	2,00	45	842,9	1,14	28	**833,4**	0,00	127	838,3	0,59	331	838,6	0,62	234	837,1	0,44	111
Daskin95-88x8	**355,8**	356,9	0,31	18	**355,8**	0,00	34	368,7	3,63	18	**355,8**	0,00	130	**355,8**	0,00	577	362,0	1,74	148	361,6	1,63	97
Daskin95-150x10	**43963,6**	44625,2	1,50	156	44011,7	0,11	255	44386,3	0,96	119	**43963,6**	0,00	1697	45109,4	2,61	323	44578,9	1,40	456	44578,9	1,40	199
Gaskell67-21x5	**424,9**	429,6	1,11	0	**424,9**	0,00	0	**424,9**	0,00	0	**424,9**	0,00	0	**424,9**	0,00	18	**424,9**	0,00	6	**424,9**	0,00	4
Gaskell67-22x5	**585,1**	**585,1**	0,00	0	611,8	4,56	0	587,4	0,39	0	**585,1**	0,00	15	**585,1**	0,00	17	**585,1**	0,00	9	**585,1**	0,00	6
Gaskell67-29x5	**512,1**	515,1	0,59	0	**512,1**	0,00	1	**512,1**	0,00	0	**512,1**	0,00	9	**512,1**	0,00	24	**512,1**	0,00	11	**512,1**	0,00	7
Gaskell67-32x5	**562,2**	571,9	1,73	1	571,9	1,73	1	584,6	3,98	1	**562,2**	0,00	18	**562,2**	0,00	27	**562,2**	0,00	40	**562,2**	0,00	20
Gaskell67-32x5	**504,3**	**504,3**	0,00	1	534,7	6,03	1	504,8	0,10	1	**504,3**	0,00	34	**504,3**	0,00	25	**504,3**	0,00	22	**504,3**	0,00	15
Gaskell67-36x5	**460,4**	**460,4**	0,00	1	485,4	5,43	1	476,5	3,50	1	**460,4**	0,00	0	**460,4**	0,00	32	**460,4**	0,00	39	**460,4**	0,00	22
Min92-27x5	**3062,0**	**3062,0**	0,00	0	**3062,0**	0,00	1	3065,2	0,10	0	**3062,0**	0,00	35	**3062,0**	0,00	23	**3062,0**	0,00	11	**3062,0**	0,00	7
Min92-134x8	**5709,0**	5965,1	4,49	50	5950,0	4,22	111	5809,0	1,75	48	5719,3	0,18	280	**5709,0**	0,00	522	5890,6	3,18	252	5920,8	3,71	134
Promedio			**1,58**	**20**		**2,01**	**36**		**1,61**	**18**		**0,03**	**188**		**0,25**	**161**		**0,74**	**105**		**0,93**	**53**
NBR			**4**			**5**			**2**			**11**			**11**			**8**			**7**	

Fuente: Elaboración Propia

5. Conclusiones y futuras investigaciones

En este artículo se propone un algoritmo efectivo para el problema de localización y ruteo con restricciones de capacidad (CLRP). El desarrollo del algoritmo propuesto ha sido evaluado considerando instancias de *benchmarking* propuestas en la literatura. Los resultados computacionales muestran el algoritmo propuesto es capaz de producir, dentro de tiempos computacionales cortos, varias soluciones obtenidas por los métodos previamente publicados y algunas nuevas mejores soluciones. Los resultados obtenidos sugieren que el algoritmo propuesto es competitivo con los algoritmos previamente publicados. El algoritmo propuesto podría ser aplicado a otros problemas de ruteo con múltiples depósitos como el problema de ruteo de vehículos con múltiples depósitos (MDVRP), el problema de localización y ruteo con restricciones de periodicidad (PLRP), etc.

Agradecimientos

Este trabajo ha sido parcialmente soportado por la Pontificia Universidad Javeriana, Cali, la Universidad del Bío-Bío, Chile y la Universidad Nacional de Colombia. Este soporte es gratamente agradecido.

Bibliografía

[1] Laporte, G., Nobert, Y. and Arpin, D., An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research, 6 (9), pp. 291-310, 1986. http://dx.doi.org/10.1007/BF02023807

[2] Contardo, C., Gendron, B. and Cordeau, J.F., A branch-and-cut-and-price algorithm for the capacitated location-routing problem. CIRRELT, 2011.

[3] Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C. and Calvo, R.W., A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research, 38 (6), pp. 931-941, 2011. http://dx.doi.org/10.1016/j.cor.2010.09.019

[4] Tuzun, D. and Burke, L.I., A two-phase tabu search approach to the location routing problem. European Journal of Operational Research, 116 (1), pp. 87-99, 1999. http://dx.doi.org/10.1016/S0377-2217(98)00107-6

[5] Prins, C., Prodhon, C., Ruiz, A., Soriano, P. and Calvo, R.W., Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. Transportation Science, 41 (4), pp. 470-483, 2007. http://dx.doi.org/10.1287/trsc.1060.0187

[6] Barreto, S., Ferreira, C., Paixao, J. and Santos, B.S., Using clustering analysis in a capacitated location-routing problem. European Journal of Operational Research, 179 (3), pp. 968-977, 2007. http://dx.doi.org/10.1016/j.ejor.2005.06.074

[7] Prins, C., Prodhon, C. and Calvo, R.W., Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR: A Quarterly Journal of Operations Research, 4 (3), pp. 221-238, 2006a.

[8] Prins, C., Prodhon, C. and Calvo, R.W., A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. Lecture Notes on Computer Science, 3906, pp. 183-194, 2006b. http://dx.doi.org/10.1007/11730095_16

[9] Jabal-Ameli, M.S., Aryanezhad, M.B. and Ghaffari-Nasab, N., A variable neighborhood descent based heuristic to solve the capacitated location-routing problem. International Journal of Industrial Engineering Computations, 2 (1), pp. 141-154, 2011. http://dx.doi.org/10.5267/j.ijiec.2010.06.003

[10] Duhamel, C., Lacomme, P., Prins, C. and Prodhon, C., A GRASP× ELS approach for the capacitated location-routing problem. Computers & Operations Research, 37 (11), pp. 1912-1923, 2010. http://dx.doi.org/10.1016/j.cor.2009.07.004

[11] Yu, V.F., Lin, S.W., Lee, W. and Ting, C.J., A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering, 58 (2), pp. 288-299, 2010. http://dx.doi.org/10.1016/j.cie.2009.10.007

[12] Pirkwieser, S. and Raidl, G.R., Variable neighborhood search coupled with ILP-based very large neighborhood searches for the (periodic) location-routing problem. In Hybrid Metaheuristics, Springer Berlin Heidelberg, pp. 174-189, 2010. http://dx.doi.org/10.1007/978-3-642-16054-7_13

[13] Escobar, J.W., Linfati, R. and Toth, P., A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. Computers & Operations Research, 40 (1), pp. 70-79, 2012. http://dx.doi.org/10.1016/j.cor.2012.05.008

[14] Gendreau, M. and Potvin, J.Y., Handbook of metaheuristics. Vol. 2. New York, Springer, 2010. http://dx.doi.org/10.1007/978-1-4419-1665-5

[15] Toth, P. and Vigo, D., The granular tabu search and its application to the vehicle-routing problem, INFORMS Journal on Computing, 15 (4), pp.333-346, 2003. http://dx.doi.org/10.1287/ijoc.15.4.333.24890

[16] Baxter, J., Local optima avoidance in depot location. Journal of the Operational Research Society, 32 (9), pp. 815-819, 1981. http://dx.doi.org/10.1057/jors.1981.159 http://dx.doi.org/10.2307/2581397

[17] Lin, S. and Kernighan, B.W.m An effective heuristic algorithm for the traveling-salesman problem. Operations Research, 21 (2), pp. 498-516, 1973. http://dx.doi.org/10.1287/opre.21.2.498

[18] Barcelo, J. and Casanovas, J., A heuristic Lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research, 15 (2), pp. 212-226, 1984. http://dx.doi.org/10.1016/0377-2217(84)90211-X

[19] Klincewicz, J. and Luss, H., A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints. Journal of the Operational Research Society, 37 (5), pp. 495-500, 1986. http://dx.doi.org/10.1057/jors.1986.84 http://dx.doi.org/10.2307/2582672

[20] Taillard, E., Parallel iterative search methods for vehicle routing problems. Networks, 3 (8), pp. 661-673, 1993. http://dx.doi.org/10.1002/net.3230230804

[21] Renaud, J., Laporte, G. and Boctor, F., A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), pp. 229-235, 1996. http://dx.doi.org/10.1016/0305-0548(95)O0026-P

[22] Prins, C., Prodhon, C. and Wolfler-Calvo, R., Nouveaux algorithmes pour le probleme de localisation et routage sous contraintes de capacite. MOSIM., 4e´me Conference Francophone de Modelisation et Simulation, Nantes, France, 4, pp. 1115-122. 2004.

[23] Barreto, S., Analise e modelizacao de problemas de localizacao distribuicao, PhD. Thesis, University of Aveiro, Aveiro, Portugal, 2004.

J.W. Escobar, es Dr. en Investigación de Operaciones de la Universidad de Bologna, Italia. Docente de programas de Pregrado y Posgrado de la Pontificia Universidad Javeriana y Universidad del Valle, Colombia. Sus intereses de investigación incluyen el diseño e implementación de efectivos algoritmos exactos y heurísticos para problemas de optimización combinatoria.

R. Linfati, es Dr en Investigación de Operaciones de la Universidad de Bologna, Italia. Profesor en el Departamento de Ingeniería Industrial de la Universidad del Bío-Bío, Chile. Sus intereses de investigación incluyen el diseño e implementación de efectivos algoritmos exactos y heurísticos para problemas de optimización combinatoria.

W. Adarme-Jaimes, recibió el grado de Ing. Industrial en 1993 de la Universidad Industrial de Santander, Colombia; es Esp. en Ingeniería de la Producción y Mejoramiento Continuo en 1997 de la Universidad Pedagógica y Tecnológica de Colombia, Colombia, es MSc. en Ingeniería con énfasis Logística en 2007 de la Universidad del Valle, Colombia, Dr. en Ingeniería, Industria y Organizaciones en 2011 de la Universidad Nacional de Colombia, Bogotá, Colombia. Profesor en la Facultad de Ingeniería de la Universidad Nacional de Colombia, Sede Bogotá. Director del grupo de investigación en Logística Sociedad, Economía y Productividad (SEPRO). Ha dirigido en los últimos diez años seis investigaciones sobre logística para diferentes sectores de la economía colombiana, con publicación de resultados en revistas indexadas. Es consultor en política pública sobre logística y sistemas de abastecimiento para los Ministerios de Comercio, Transporte y Salud de Colombia y Alcaldía mayor de Bogotá. Director de 14 tesis de maestría y tres tesis de Doctorado en Logística. Ponente en congresos internacionales sobre logística en Alemania, México, Panamá, Venezuela y Colombia.