

UN ALGORITMO METAHEURÍSTICO BASADO EN RECOCIDO SIMULADO CON ESPACIO DE BÚSQUEDA GRANULAR PARA EL PROBLEMA DE LOCALIZACIÓN Y RUTEO CON RESTRICCIONES DE CAPACIDAD*

John Willmer Escobar**; Rodrigo Linfati***

Recibido: 09/08/2012

Aceptado: 05/11/2012

RESUMEN

Consideramos 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 construidas para satisfacer las demandas de los clientes. El objetivo es minimizar la suma de los costos de abrir depósitos, de los costos de los vehículos usados, y de los costos variables asociados con la distancia recorrida por las rutas. En este paper, proponemos una metaheurística basada en simulado y recocido con espacio de búsqueda granular para resolver el problema CLRP. Experimentos computacionales en instancias de benchmarking tomadas de la literatura muestran que el algoritmo propuesto es capaz de obtener, dentro de tiempos computacionales razonables, soluciones de alta calidad mostrando su eficacia.

Palabras clave: problema de localización y ruteo; simulado y recocido; búsqueda granular: algoritmos metaheurísticos.

* Artículo resultado de una investigación parcialmente financiada por el MIUR (Ministero Istruzione, Università e Ricerca) Italia, y la Pontificia Universidad Javeriana, Cali, Colombia

** Corresponding Author. Researcher, University of Bologna. Profesor Tiempo Completo, Pontificia Universidad Javeriana Cali, Colombia. PhD(C) in Operations Research, University of Bologna, Italy. Magister en Ingeniería Industrial, Universidad del Valle, Cali, Colombia. Especialista en Finanzas, Universidad del Valle, Cali, Colombia. Ingeniero Industrial, Universidad del Valle, Cali, Colombia. Departamento de Ingeniería Civil e Industrial, Pontificia Universidad Javeriana, Calle 18 No. 118-250, 26239 Cali, Colombia. Department of Electronics, Computer Sciences and Systems (DEIS), University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy. Email: johnwillmer.escobar2@unibo.it. Fax: +572 3218266

*** Profesor Tiempo Completo, Universidad del Bio-Bío, Chile. PhD in Operations Research, University of Bologna, Italy. Magister en Ingeniería Industrial, Universidad de Concepcion, Concepcion, Chile. Ingeniero Civil Industrial, Universidad de Concepcion, Concepcion, Chile. Department of Electronics, Computer Sciences and Systems (DEIS), University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy. Email: rlinfati@ubiobio.cl

A META-HEURISTIC ALGORITHM BASED ON THE SIMULATED ANNEALING WITH GRANULAR SEARCH SPACE FOR THE CAPACITATED LOCATION ROUTING PROBLEM

Abstract

The article deals with the Capacitated Location Routing Problem (CLRP) where the goal is to determine the depots to be opened, the customers to be assigned to each deposit opened, and the routes to be constructed for fulfilling customers' demands. The objective is to minimize the sum of costs resulting from opening depots, costs resulting from used vehicles, and variable costs associated to the distance across the routes. In this paper, a metaheuristic based on simulated annealing with granular search space for solving the CLRP is proposed. Computational experiments on benchmarking instances taken from literature show that the proposed algorithm is able to obtain high-quality solutions within reasonable computational times, thus showing its efficiency.

Key words: location routing problem; simulated annealing; granular search; metaheuristic algorithms.

INTRODUCCIÓN

El estudio de los problemas de localización y ruteo es un área relativamente nueva de investigación en el campo de la logística dentro de la ingeniería industrial e ingeniería de sistemas (específicamente en el campo de ciencias de la computación). Este tipo de problemas incluye dos decisiones fundamentales de la cadena de abastecimiento: la localización de instalaciones y el ruteo de vehículos. Un gran número de trabajos se ha dedicado a estudiar cada una de estas decisiones de manera independiente. Este hecho se puede explicar debido a que la localización es una decisión estratégica, la cual es tomada por un horizonte largo de tiempo. Por su parte, el ruteo es un aspecto operacional modificado varias veces en un tiempo corto. Sin embargo, según Rand [1], las decisiones de localizar depósitos frecuentemente son influenciadas por los costos de transporte, y viceversa. De igual forma, Salhi y Nagy [2] han encontrado que el costo logístico puede ser reducido a lo largo del tiempo, cuando se consideran decisiones de localización y ruteo simultáneamente.

Este artículo considera el problema de localización y ruteo con restricciones de capacidad en depósitos y vehículos (CLRP). El CLRP puede ser descrito como un problema de grafos de la siguiente manera. Sea $G = (V, A)$ un grafo completo no dirigido, en el cual $V = \{1, \dots, m + n\}$ es el conjunto de vértices y A es el conjunto de arcos. El conjunto de vértices es dividido en dos subconjuntos. Vértices $i = 1, \dots, m$ corresponden a los depósitos potenciales, cada uno con capacidad W_i y costo fijo de apertura O_i . Vértices $j = m + 1, \dots, m + n$ corresponden a los clientes, cada uno con valor de demanda D_j . Una flota homogénea de vehículos k , cada uno con capacidad Q , es disponible en cada depósito $i \in V$. Cada vehículo cuando es utilizado para realizar una ruta genera un costo fijo F . Un costo no negativo c_{ij} es asociado con cada arco $(i, j) \in A$.

El objetivo del CLRP es determinar los depósitos a abrir, los clientes a asignar a cada depósito

abierto y las rutas a construir para satisfacer la demanda de los clientes con el mínimo costo global. Este costo incluye los costos fijos de apertura de los depósitos, los costos fijos asociados a los vehículos usados por las rutas y los costos de los arcos visitados por cada ruta. En el CLRP las siguientes restricciones son consideradas: i) cada ruta debe comenzar y terminar en el mismo depósito; ii) cada cliente debe ser visitado por una ruta exactamente una vez; iii) la suma de las demandas de los clientes visitados en una ruta no debe exceder la capacidad del vehículo Q ; iv) la suma de las demandas de los clientes asignados a un depósito no debe exceder su capacidad W_i ; v) los flujos entre depósitos no son permitidos.

Prins et al. [3] han propuesto una formulación de tres índices para el CLRP. Dicha formulación utiliza las variables binarias $y_i = 1$ si el depósito i es abierto, $f_{ij} = 1$ si el cliente j es asignado al depósito i y $x_{ijk} = 1$ si el arco (j, l) es visitado desde j a l en la ruta desarrollada por el vehículo k . El problema entonces es formulado mediante el siguiente modelo:

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

Sujeto a

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

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

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

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

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

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

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

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

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

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

La función objetivo (1) suma los costos de los depósitos abiertos, los costos de los arcos visitados por las rutas y los costos fijos asociados con el uso de los vehículos. El grupo de restricciones (2) garantiza que cada cliente sea visitado exactamente por una ruta y que solamente tenga un predecesor en la secuencia de la ruta. El grupo de restricciones (3) y (8) son asociadas con las capacidades del depósito y vehículo, respectivamente. Restricciones (4) y (5) aseguran la continuidad de cada ruta y determina que cada una de ellas comience y termine en el mismo depósito. La eliminación de los subtours es satisfecha a través de las restricciones (6). Restricciones (7) especifica que un cliente puede ser asignado a un depósito solamente si existe una ruta que lo une. Finalmente restricciones (9), (10) y (11) representan las variables binarias usadas en el modelo.

El CLRP es considerado un problema NP-hard debido a que es la 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). En este artículo se propone un algoritmo metaheurístico basado en simulado y reconocido con espacio de búsqueda granular para resolver el CLRP. La literatura existente para el problema de CLRP es descrita en la Sección 1. La Sección 2 detalla el algoritmo propuesto. Los resultados computacionales son presentados en la Sección 3. Finalmente, conclusiones e investigación futura se detallan en la Sección 4.

1. REVISIÓN DE LA LITERATURA

La literatura relacionada con problemas de localización y ruteo ha sido condensada en Min et al. [4] y Nagy y Salhi [5]. En [4] se propone una

clasificación para los problemas de localización y ruteo basada en el método de solución y las perspectivas del problema. Por su parte, en [5] se propone una clasificación más amplia basada en diferentes aspectos tales como el método de solución, la estructura jerárquica, el tipo de información, el período de planeación, etc.

De acuerdo con el método de solución, el problema CLRP ha sido resuelto mediante algoritmos exactos y heurísticos. Métodos exactos están basados en formulaciones de dos y tres índices. Formulaciones de tres índices han sido propuestas en [3]. Por otro lado, formulaciones de dos índices han sido propuestas por Laporte et al. [6] y Belenguer et al. [7]. Elementos comunes de estos trabajos son técnicas de relajación, desigualdades válidas y algoritmos de separación. En términos generales, estos trabajos han sido capaces de probar optimalidad solo en instancias que contienen menos de 100 clientes. Debido a lo anterior, algoritmos heurísticos han sido propuestos como alternativa para la solución de problemas de localización y ruteo de mediana y gran escala (problemas con más de 100 clientes).

En la literatura existente, varios trabajos han considerado métodos heurísticos para resolver el CLRP. Usualmente, métodos heurísticos determinan buenas soluciones con razonables tiempos de ejecución. Estas técnicas emplean diferentes métodos de solución para modelar la relación entre los dos subproblemas (problema de localización y problema de ruteo).

En [5] se ha realizado una clasificación de los algoritmos heurísticos en cuatro categorías: métodos secuenciales, métodos basados en clústeres, métodos iterativos, métodos jerárquicos. Métodos secuenciales resuelven el problema de localización y luego el problema de ruteo. Estos algoritmos, según Salhi y Rand [8], no permiten una retroalimentación entre los dos subproblemas. Métodos iterativos resuelven ambos subproblemas en un modo iterativo, retroalimentando la solución entre

dos subproblemas. Estos métodos pueden resolver el CLRP de dos maneras: resolviendo el problema de ruteo de vehículos y asignando un depósito a cada ruta o resolviendo el problema de localización de depósitos y construyendo al menos una ruta para cada depósito. Métodos iterativos han sido estudiados recientemente por [3] y Tuzun y Burke [9].

En [5] se propone un algoritmo de dos fases que intercambia información entre ellas. En la primera fase, los clientes son agregados en “super clientes”, y el correspondiente problema de localización de instalaciones con restricciones de capacidad es resuelto por medio de la técnica de relajación Langragiana. En la segunda fase, un procedimiento metaheurístico basado en una búsqueda granular tabú (ver Toth y Vigo [10]) con un solo vecindario fue usado para resolver el resultante problema de ruteo con múltiples depósitos. Al final de cada iteración, información sobre los arcos predominantes es transmitida durante la siguiente fase. En [9] se ha propuesto un algoritmo tabú de dos fases: ruteo y localización. Este algoritmo itera entre la localización y ruteo para encontrar mejores soluciones para instancias medianas y grandes. En este trabajo, resultados para instancias de hasta 200 clientes han sido reportados.

Métodos basados en clústeres han sido propuestos por Barreto et al. [11]. En esta metodología, los clientes son agrupados en clústeres de acuerdo con la capacidad del vehículo. Luego, el problema de viajero de negocios (TSP) es resuelto para cada grupo de clientes. Finalmente, los grupos de clientes son asignados a cada depósito abierto. Métodos jerárquicos resuelven el CLRP usando una estructura jerárquica. El problema de localización es solucionado como problema principal, y luego el subsecuente problema de ruteo es resuelto como problema subordinado. En particular, estos métodos resuelven el problema de localización en un modo aproximado aplicando una subrutina que resuelve el correspondiente problema de ruteo.

Otras metaheurísticas para el CLRP han sido propuestas por Prins et al. [12]. En este trabajo, un algoritmo basado en un procedimiento aleatorio de búsqueda adaptativa goloso (GRASP) con estrategia de path-relinking ha sido propuesto. Los mismos autores han propuesto un algoritmo memético con administración de la población (MA|PM). Para mayores detalles, los lectores son referidos a Prins et al. [13].

Recientes metaheurísticas han sido propuestas por Duhamel et al. [14], Yu et al. [15] y Escobar et al. [16]. En [14] se propone un método basado en un híbrido GRASP con una búsqueda local evolucionada (ELS). En [15] se propone un procedimiento aleatorio basado en la heurística de recocido simulado con tres vecindades. Finalmente, en [16] se propone un algoritmo heurístico de dos fases para la solución del problema CLRP.

2. DESCRIPCIÓN DEL ALGORITMO PROPUESTO

En esta sección, se explica el algoritmo propuesto para resolver el CLRP. El algoritmo propuesto está basado en la metaheurística de recocido simulado con búsqueda de espacio granular propuesta en [10]. Recocido simulado es una metaheurística popular usada para resolver problemas discretos y continuos. La metaheurística recocido simulado adopta su nombre gracias a la analogía con el proceso físico de recocido con sólidos. En Eglese [17] se puede observar una buena visión general sobre el desarrollo teórico y la aplicación de la metaheurística de recocido simulado.

La principal característica de la metaheurística de recocido simulado es que proporciona una alternativa eficiente para escapar de óptimos locales al permitir movimientos “peores” respecto a la función objetivo (hill-climbing moves) en aras de encontrar un óptimo global. A medida que la temperatura se reduce a cero, los movimientos “peores” se producen con menos frecuencia convergiendo a óptimos globales. El procedimiento de

recocido simulado comienza desde una solución aleatoria inicial. En cada iteración, una nueva solución es tomada del vecindario predefinido de la actual solución. El valor de la función objetivo de la nueva solución es comparado con la actual para determinar si un mejoramiento ha sido alcanzado.

En el algoritmo propuesto, se construye una solución inicial factible mediante un procedimiento híbrido introducido en [16]. Para mejorar dicha solución, un procedimiento basado en recocido simulado con espacio de búsqueda granular es aplicado. El procedimiento híbrido inicial permite obtener una solución factible con un tiempo de cómputo reducido. Una de las ventajas de usar este procedimiento es que permite localizar correctamente los depósitos, lo cual reduce el espacio de búsqueda del procedimiento de mejoramiento de un CLRP a un MDVRP. En otras palabras, el procedimiento de recocido simulado propuesto no considera decisiones de cierre y apertura de centros de distribución. Los procedimientos descritos con anterioridad son detallados en las siguientes Secciones.

2.1 Solución inicial

El procedimiento híbrido inicial introducido por [16] es capaz de encontrar soluciones de alta calidad con tiempos computacionales reducidos. Este procedimiento combina métodos exactos y heurísticos. A continuación se menciona cada uno de los pasos para la construcción de la solución inicial:

Paso 1: Un TSP tour gigante es construido considerando todos los clientes, mediante el conocido procedimiento heurístico Lin-Kernighan (LKH). Para mayores detalles ver Helsgaun [18].

Paso 2: El TSP tour gigante es dividido en varios grupos de clientes llamados clústeres. El criterio para obtención de los grupos de clientes es la capacidad de los vehículos.

Paso 3: Para cada depósito i y cada clúster g , el procedimiento LKH es aplicado para encontrar el correspondiente TSP tour. De esta manera, es

posible obtener la longitud de la ruta l_{ig} asignando el depósito i al clúster g .

Paso 4: Los depósitos son asignados a los clústeres resolviendo el correspondiente problema de localización de planta con restricciones de una sola fuente de abastecimiento (“*Single Source Capacitated Plant Location Problem*”). De esta manera se determina qué depósitos serán abiertos y cuáles TSP tours serán asignados a cada depósito.

Paso 5: Finalmente un procedimiento de división es aplicado a la solución actual. Este procedimiento busca reducir la distancia recorrida, adicionando nuevas rutas y asignándolas a depósitos diferentes. Este procedimiento es efectivo solamente cuando el costo por el uso de un vehículo F es reducido. Para las tres combinaciones de dos arcos, arcos (r,s) y (t,u) , los siguientes pasos son repetidos:

- Arcos (r,s) y (t,u) son removidos de la ruta actual;
- El arco (t,u) es insertado en la nueva ruta;
- El conjunto de clientes que pertenecen a la cadena que conecta vértice s con vértice t en la ruta considerada es seleccionado como un nuevo clúster;
- Para cada depósito para el cual la asignación de los clientes satisface las restricciones de capacidad, el procedimiento LKH es aplicado para encontrar el correspondiente TSP tour de cada depósito abierto i al nuevo clúster;
- El clúster es asignado al depósito para el cual el costo de la ruta es mínimo;
- Finalmente un procedimiento de mejoramiento para el problema de ruteo de vehículos (VRP) propuesto en [16] es aplicado para el depósito asignado.

2.2 Fase de mejora de la solución inicial

En la segunda fase del procedimiento, el algoritmo trata de mejorar la solución inicial (x_0) obtenida por el procedimiento heurístico anteriormente descrito, mediante la aplicación de la

metaheurística de recocido simulado con un espacio de búsqueda granular. Este procedimiento trata de mejorar la solución mediante movimientos que involucran mover clientes dentro y entre rutas.

2.2.1 Espacio de búsqueda granular

En un procedimiento de búsqueda local, el número de vecindades que se puede alcanzar con un movimiento simple crece directamente con las dimensiones del problema. Para reducir considerablemente el tiempo de cálculo requerido en la evaluación de vecindades, en [10] se ha propuesto el uso de espacios de “búsqueda granular” para el problema de ruteo de vehículos con restricciones de capacidad. La metodología de búsqueda granular usa una lista de “buenos vecindarios”, llamada Lista de Candidatos a ser evaluados para obtener una nueva solución.

El espacio de búsqueda granular, determinado por la Lista de Candidatos, es obtenido por la unión de los arcos “cortos”, los arcos incidentes a los depósitos y los arcos que pertenecen a las mejores soluciones encontrados durante la búsqueda. Los arcos “cortos” son los que su distancia es menor que el valor de granularidad (ϑ), definido este como el producto entre un factor de esparcificación (β) y el costo promedio de los arcos que componen una “buena” solución. El costo promedio de estos arcos está definido como el valor objetivo de una “buena” solución (z) dividido por el número de clientes (n) más el número de depósitos (K), como se observa en la ecuación (12). El factor de esparcificación β sirve para controlar el número de arcos que pertenecen a la Lista de Candidatos. En [10] se aconseja el uso de un valor de β entre 1.0 y 2.5, lo cual permite seleccionar entre el 10% y el 20% de los arcos de un grafo completo. En este caso, se evalúan algunas vecindades reduciendo notablemente el tiempo de cálculo.

$$\vartheta = \beta * \frac{z}{n + K} \quad (12)$$

El cambio en el valor del factor de esparcificación β permite considerar simultáneamente estrategias de diversificación e intensificación. El valor de β es alterado a medida que va evolucionando la búsqueda. Al inicio de la búsqueda el valor de β es ajustado a un valor pequeño β_0 . Si la solución actual no mejora después de N_i iteraciones, el valor de β es incrementado hasta β_d , permitiendo al algoritmo explorar nuevas áreas de búsqueda (estrategia de diversificación). Cuando se desea intensificar la búsqueda, el valor de β después de N_r iteraciones es reducido nuevamente al valor β_0 .

2.2.2 Vecindades

El algoritmo propuesto utiliza cinco vecindades, las cuales son evaluadas hasta que se cumpla la condición de término (número de iteraciones N_{stop}).

- **Shift:** un cliente es transferido desde su ubicación actual a otra posición en la misma ruta o en una ruta diferente (perteneciente al mismo depósito o a otro diferente).
- **Swap:** dos clientes intercambian su posición, es decir, el cliente u toma el lugar del cliente v o viceversa. Los clientes pueden pertenecer a la misma ruta o a diferentes rutas (perteneciente al mismo depósito o a otro diferente).
- **Two-Opt:** Dos arcos no consecutivos son eliminados de la solución actual y las rutas son reconectadas de forma diferente para obtener una nueva solución. Si los arcos pertenecen a la misma ruta, el movimiento es considerado como el método tradicional Two-Opt para TSP. Si ambos arcos pertenecen a rutas diferentes que están abastecidas por el mismo depósito, se obtiene el método tradicional Two-Opt para VRP. Finalmente, si las dos rutas pertenecen a diferentes depósitos, para reconectar las rutas es necesario realizar movimientos adicionales con los arcos incidentes al depósito.
- **Exchange:** dos clientes consecutivos son transferidos de su ubicación actual, a una nueva ubicación en la misma ruta o en una ruta

diferente (perteneciendo al mismo depósito o a otro depósito diferente).

- **Inter-tour Exchange:** es una extensión del movimiento de Swap considerando dos pares de clientes consecutivos que pertenecen a diferentes rutas. El arco que conecta a los clientes consecutivos se conserva en la solución actual.

2.2.3 Algoritmo de recocido simulado

El algoritmo de recocido simulado es un procedimiento iterativo que permite movimientos de “no mejora” de forma probabilística. Este método fue introducido por Kirkpatrick et al. [19], quienes fueron los primeros en usarlo para resolver problemas de optimización combinatoria. El algoritmo de recocido simulado imita la evolución de un sistema físico inestable hacia un equilibrio termodinámico a una temperatura fija. En cada iteración de dicho procedimiento, una nueva solución (x') es seleccionada de forma aleatoria de las vecindades de la solución actual (x).

El algoritmo de recocido simulado acepta nuevas soluciones de acuerdo con dos criterios: i) si el valor de la función objetivo de la nueva solución es mejor, ii) si el valor de la función objetivo de la nueva solución es peor, y un valor aleatorio generado entre cero y uno es menor que la diferencia entre

la solución actual y la nueva solución dividido por la temperatura del sistema T . Al inicio del algoritmo, la temperatura del sistema es ajustada en un valor T_0 . Este valor disminuye cada N_{cool} iteraciones de manera proporcional a un factor de enfriamiento α , donde $\alpha < 1$. El Algoritmo 1 resume el procedimiento de Recocido y Simulado propuesto.

3. EXPERIENCIA COMPUTACIONAL

El algoritmo propuesto (GSA) ha sido implementado en C++ y los experimentos fueron ejecutados en una CPU Intel Core Duo (2.00 Ghz) bajo Linux Ubuntu 11.04 con 2 GB de memoria. El desarrollo del algoritmo propuesto ha sido probado en 2 sets de instancias de benchmarking de la literatura propuestas en [9] y Barreto [20]. En ambos set de datos, los clientes y depósitos potenciales son representados por puntos en el plano cartesiano. De esta forma, el costo de transporte de un arco es la distancia euclidiana calculada como un número real de doble precisión. El primer set de datos (propuesto por Tuzun y Burke [9]) considera 36 instancias con depósitos sin restricciones de capacidad. El número de clientes es determinado en el intervalo [100, 200], y el número de depósitos potenciales es 10 o 20. La capacidad del vehículo es determinada como 150. El set de datos propuesto

Algoritmo 1: Pseudocódigo de Recocido Simulado

Entrada: Solución inicial (x_0).

Salida: Solución final (x').

$x = x' = x_0$ $T = T_0$ Iter = 0

Mientras iter < MaxIter

 Generar solución x' en el vecindario $N(x)$ de forma aleatoria

Si $f(x') < f(x)$ $x = x'$ **Sino**

$r = \text{Aleatorio}(0,1)$

Si $r < e^{-\frac{f(x')-f(x)}{T}}$

$x = x'$ **Si** $f(x') < f(x')$

$x' = x'$

$T = \alpha * T$

 iter = iter+1

Fin

en [20] considera 13 instancias obtenidas de problemas clásicos de ruteo de vehículos con restricciones de capacidad adicionando nuevos depósitos con sus correspondientes capacidades y costos fijos. El número de clientes varía entre 21 y 150, y el número de depósitos potenciales de 5 a 10.

3.1 Ajuste de parámetros

En particular, para cada instancia, cinco corridas independientes del algoritmo han sido ejecutadas con cinco semillas diferentes. Los resultados reportados corresponden a los mejores valores

encontrados durante las ejecuciones y el tiempo de cómputo corresponde al tiempo total de las cinco ejecuciones.

Los mejores resultados de la heurística son reportados en las tablas 1 y 2. El siguiente set de parámetros ha sido obtenido después de varias corridas en los dos sets de instancias: $\beta_0 = 1.50$, $\beta_d = 2.50$, $N_s = 2000$, $N_r = 1000$, $T_0 = 1000$, $\alpha = 0.97$, $N_{cool} = 1200$, $MaxIter = 6000$. Estos valores han sido utilizados para la solución de todas las instancias consideradas.

Tabla 1. Resultados para el grupo de instancias propuestas por Tuzun-Burke (1999)

Instancia	BR	GRASP [2006]			MA PM [2006]			LRGTS [2007]			GRASP + ELS [2010]			SALRP [2010]			2-Phase HGTS [2012]			GSA [2012]		
		Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time
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	1490,82	1,19	151
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	1486,27	2,56	244
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	1407,26	0,76	123
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	1474,01	2,91	159
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	1167,16	0,00	246
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	220
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	791,66	0,00	181
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	254
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	157
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	1251,22	0,42	242
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	137
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	159
131112	1944,57	2006,70	3,20	113	1959,39	0,76	129	1946,01	0,07	13	1944,57	0,00	518	1953,85	0,48	743	1961,75	0,88	485	1944,57	0,00	353
131122	1856,51	1888,90	1,74	161	1881,67	1,36	144	1875,79	1,04	19	1864,24	0,42	705	1899,05	2,29	835	1856,51	0,00	298	1871,13	0,79	273
131212	1984,25	2033,93	2,50	100	1984,25	0,00	111	2010,53	1,32	11	1992,41	0,41	727	2057,53	3,69	456	2012,69	1,43	406	2012,69	1,43	411
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	263
132112	1445,25	1508,33	4,36	118	1448,27	0,21	168	1448,65	0,24	23	1453,78	0,59	103	1453,30	0,56	750	1445,25	0,00	449	1453,78	0,59	446
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	491
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	269
132222	931,44	940,80	1,00	143	934,79	0,36	196	936,93	0,59	14	945,81	1,54	224	934,62	0,34	842	931,49	0,01	335	931,44	0,00	320
133112	1705,36	1736,90	1,85	93	1720,30	0,88	144	1729,31	1,40	18	1712,11	0,40	271	1720,81	0,91	742	1705,36	0,00	444	1745,23	2,34	425
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	333
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	1234,83	2,61	497
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	1156,27	0,03	371

Instancia	BR	GRASP [2006]			MA PM [2006]			LRGTS [2007]			GRASP + ELS [2010]			SALRP [2010]			2-Phase HGTS [2012]			GSA [2012]		
		Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time
121112	2265,59	2384,01	5,23	385	2293,99	1,25	523	2296,52	1,37	41	2295,90	1,34	655	2324,10	2,58	1328	2265,59	0,00	522	2265,59	0,00	503
121122	2166,43	2288,09	5,62	410	2277,39	5,12	458	2207,50	1,90	40	2203,57	1,71	432	2258,16	4,23	1455	2166,43	0,00	603	2187,86	0,99	526
121212	2246,39	2273,19	1,19	311	2274,57	1,25	378	2260,87	0,64	33	2246,39	0,00	1566	2260,30	0,62	1319	2249,40	0,13	527	2256,32	0,44	486
121222	2237,81	2345,10	4,79	419	2376,25	6,19	436	2259,52	0,97	40	2265,53	1,24	2192	2326,53	3,96	1428	2237,81	0,00	558	2253,32	0,69	506
122112	2106,26	2137,08	1,46	338	2106,26	0,00	351	2120,76	0,69	48	2106,47	0,01	1521	2112,65	0,30	1320	2121,93	0,74	522	2121,93	0,74	474
122122	1718,65	1807,29	5,16	370	1771,53	3,08	378	1737,81	1,11	59	1779,05	3,51	618	1722,99	0,25	1400	1749,10	1,77	691	1718,65	0,00	605
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	1473,27	0,39	747
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	1082,99	0,04	578
123112	1973,28	2044,66	3,62	283	1973,28	0,00	413	1984,06	0,55	43	2004,33	1,57	1451	1994,16	1,06	1318	1984,77	0,58	542	1990,87	0,89	475
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	1970,91	2,01	586
123212	1778,41	1788,70	0,58	199	1782,23	0,21	353	1786,79	0,47	34	1778,80	0,02	1398	1779,10	0,04	1314	1778,41	0,00	697	1779,10	0,04	533
123222	1390,74	1408,63	1,29	296	1396,24	0,40	530	1401,16	0,75	43	1453,82	4,54	2202	1396,42	0,41	1427	1390,87	0,01	518	1390,74	0,00	489
Average		1556,51	2,86	163	1532,19	1,23	207	1528,75	1,21	22	1522,01	0,67	607	1526,41	0,86	826	1519,05	0,52	392	1521,55	0,65	368
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 Duo (2.00 Ghz)			Core 2 Duo (2.00 Ghz)		
CPU index		314			314			314			4373			4046			1398			1398		

Fuente: elaboración propia

Tabla 2. Resultados para el grupo de Instancias propuestas por Barreto (2004)

Instance	BR	GRASP [2006]			MA PM [2006]			LRGTS [2007]			GRASP + ELS [2010]			SALRP [2010]			2-Phase HGTS [2012]			GSA [2012]		
		Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time	Costo	Gap BR	CPU time
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	580,4	2,62	46
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	868,9	2,36	91
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	841,7	1,00	220
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	362,0	1,74	152
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	44881,8	2,09	399
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	7
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	11
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	13
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	43
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	24
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	42
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	13
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	226
Average		4569,07			1,58			20			4523,24			2,01			36			4539,41		
CPU		Pentium 4 (2.4 Ghz)			Pentium 4 (2.4 Ghz)			Pentium 4 (2.4 Ghz)			Core2Quad (2.83 Ghz)			Core2Quad (2.66 Ghz)			Core2Duo (2.00 Ghz)			Core2Duo (2.00 Ghz)		
CPUindex		314			314			314			4373			4046			1398			1398		

Fuente: elaboración propia

3.2 Estudio comparativo

El algoritmo propuesto ha sido comparado con las siguientes heurísticas publicadas para el CLRP: GRASP en [12], MA|PM en [13], LRGTS en [3], GRASP+ELS en [14], SALRP en [15] y 2-Phase HGTS en [16]. Los resultados reportados para GRASP, MA|PM, LRGTS, SALRP, y 2-Phase HGTS corresponden a una sola corrida del algoritmo asociado. Finalmente, GRASP+ELS ha sido ejecutado cinco veces considerando cinco semillas randómicas, y el costo reportado es el mejor resultado sobre las corridas con el tiempo computacional requerido para alcanzar la mejor solución dentro de la correspondiente ejecución.

En las tablas 1-2, la siguiente notación es usada:

Instancia	nombre de la instancia;
Costo	costo de solución obtenida por cada método (en una sola corrida o la mejor corrida);
BR	costo del mejor resultado encontrado por los algoritmos considerados;
Gap BR	variación porcentual del costo encontrado por cada algoritmo con respecto al valor de BR;
CPU time	tiempo de ejecución en segundos en la CPU usada por cada algoritmo;
Average	promedio de los valores de la columna;
CPU	CPU usado por cada algoritmo;
CPU index	Passmark performance test para cada CPU.

En las tablas 1 y 2, para cada instancia, los valores para los cuales el costo de cada algoritmo es igual al valor de BR son resaltados en negrilla. Cuando el algoritmo propuesto (GSA) mejora el valor del BR, su resultado es subrayado. Finalmente, el CPU index es obtenido por el Passmark Performance Test. Este bien conocido test está relacionado con el CPU y el desarrollo de memoria. Una CPU es rápida cuando el valor del CPU index es alto.

En las tablas 1 y 2, nosotros reportamos resultados para el set de instancias propuestos por Tuzun y Burke (1999) y Barreto (2004), respectivamente. En particular, valores promedios de Gap BR, CPU time y CPU index son reportados. La tabla 1 muestra que el algoritmo propuesto provee un valor promedio global de Gap BR menor que GRASP, MA|PM, LRGTS, GRASP+ELS, SALRP. Solamente 2-Phase HGTS obtiene, con mayores tiempos de computación, valores ligeramente superiores para el valor promedio de Gap BR. En cuanto al tiempo computacional, el algoritmo GSA es más veloz que GRASP+ELS, SALRP y 2-Phase HGTS, los cuales obtuvieron los mejores resultados previos en término de Gap BR. Sin embargo, el CPU time usado por GASP+ELS no representa el tiempo global requerido para encontrar las mejores soluciones, desde que este corresponde al CPU time para cada instancia para obtener las mejores soluciones en su respectiva ejecución. Algoritmos GRASP, MA|PM, LRGTS toman menos tiempo computacional, pero ellos parecen ser menos robustos en términos de la calidad de la solución.

Finalmente, en la tabla 2 se muestran los resultados para el set de datos propuestos por Barreto (2004). Los resultados remarcan que nuestro algoritmo es competitivo en términos de calidad de la solución. El algoritmo GSA es capaz de obtener mejores resultados que GRASP, MA|PM y LRGTS pero superado por GRASP+ELS, SALRP y 2-Phase HGTS.

4. CONCLUSIONES Y FUTURAS INVESTIGACIONES

Nosotros proponemos un algoritmo efectivo basado en la metaheurística recocido con un espacio de búsqueda granular para el problema de localización y ruteo con restricciones de capacidad (CLRP). El algoritmo propuesto ha sido evaluado considerando instancias de benchmarking tomadas de la literatura. Los experimentos computacionales muestran que el algoritmo propuesto es capaz de obtener, dentro de razonables tiempo de computa-

ción, varias soluciones obtenidas por los métodos anteriormente publicados y nuevas mejores soluciones. Los resultados obtenidos sugieren que el algoritmo propuesto es competitivo con los algoritmos anteriormente publicados. El algoritmo GSA podría ser aplicado a otros problemas de logística similares como el problema de ruteo con múltiples depósitos (MDVRP), el problema de localización y ruteo con restricciones de periodicidad (PLRP), etc.

5. AGRADECIMIENTOS

El trabajo del primer autor ha sido parcialmente financiado por el MIUR (Ministero Istruzione, Università e Ricerca) Italia, y la Pontificia Universidad Javeriana, Cali, Colombia. Este soporte es altamente agradecido.

REFERENCIAS

- [1] G. Rand, "Methodological choices in depot location studies", *Operational Research Quarterly*, vol. 27, n.º 1, pp. 241-249, 2003.
- [2] S. Salhi, y G. Nagy, "Consistency and robustness in location - routing", *Studies in Locational Analysis*, vol. 13, pp. 3-19, 1999.
- [3] C. Prins et al., "Solving the capacitated location routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic", *Transportation Science*, vol. 41, pp. 470-483, 2007.
- [4] H. Min et al., "Combined location-routing problems: A synthesis and future research directions", *European Journal of Operational Research*, vol. 108, n.º 1, pp. 1-15, 1998.
- [5] G. Nagy, y S. Salhi, "Location - routing: issues, models and methods", *European Journal of Operational Research*, vol. 177, n.º 2, pp. 649-672, 2007.
- [6] G. Laporte et al., "An exact algorithm for solving a capacitated location-routing problem", *Annals of Operations Research*, vol. 6, n.º 9, pp. 291-310, 1986.
- [7] J. Belenguer et al., "A Branch and Cut method for the Capacitated Location-Routing Problem", *Computers and Operations Research*, vol. 38, n.º 6, pp. 931-941, 2011.
- [8] S. Salhi, y G.K. Rand, "The effect of ignoring routes when location depots", *European Journal of Operational Research*, vol. 39, pp. 150-156, 1989.
- [9] D. Tuzun, y LI. Burke, "A two-phase tabu search approach to the location routing problem", *European Journal of Operational Research*, vol. 116, pp. 87-99, 1999.
- [10] P. Toth, y D. Vigo, "The granular tabu search and its application to the vehicle routing problem", *INFORMS Journal on Computing*, vol. 15, n.º 4, pp. 333-346, 2003.
- [11] S. Barreto et al., "Using cluster analysis in capacitated location routing problem", *European Journal of Operational Research*, vol. 179, n.º 3, pp. 968-977, 2007.
- [12] C. Prins et al., "Solving the capacitated location routing problem by a GRASP complemented by a learning process and path relinking", *4OR - A Quarterly Journal of Operations Research*, vol. 4, n.º 3, pp. 221-238, 2006.
- [13] C. Prins et al., "A memetic algorithm with population management (MA|PM) for the capacitated location - routing problem", *Lecture notes in computer science*, vol. 3906, pp. 183-194, 2006.
- [14] C. Duhamel et al., "A GRASP x ELS approach for the capacitated location-routing problem", *Computer and Operations Research*, vol. 37, n.º 1, pp. 1912-1923, 2010.
- [15] V.F. Yu et al., "A simulated annealing heuristic for the capacitated location routing problem". *Computer and Industrial Engineering*, vol. 58, pp. 288-299, 2010.
- [16] J.W. Escobar et al., "A two-phase hybrid heuristic algorithm for the capacitated location-routing problem", *Computers and Operations Research*, vol. 40, n.º 1, pp. 70-79, 2013.
- [17] R.W. Eglese, "Simulated annealing: a tool for operational research", *European Journal of Operational Research*, vol. 46, pp. 271-281, 1990.
- [18] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic", *European Journal of Operational Research*, vol. 126, n.º 1, pp. 106-130, 2000.
- [19] S. Kirkpatrick et al., "Optimization by Simulated Annealing", *Science*, vol. 220, pp. 671-680, 1983.
- [20] S. Barreto, Analise e modalização de problemas de localização - distribuição, PhD thesis, University of Aveiro, Portugal, October 2004.