

Revista Electrónica Nova Scientia

Ubicación óptima de unidades de servicio bajo condiciones de capacidad limitada mediante un método metaheurístico

Optimal location of capacitated service units through a metaheuristic optimization approach

Roger Z. Ríos Mercado¹, Dagoberto R. Quevedo Orozco²

¹Programa de Posgrado en Ing. de Sistemas, Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León (UANL)

²SINTEC, San Pedro, Nuevo León

México

Roger Z. Ríos Mercado. UANL, AP 111-F, Cd. Universitaria, San Nicolás de los Garza NL 66455, México.
E-mail: roger.rios@uanl.edu.mx

© Universidad De La Salle Bajío (México)

Resumen

Introducción: El problema de localización de p -centro capacitado consiste en ubicar p instalaciones y asignar usuarios a cada una de ellas, de tal manera que se minimice la distancia máxima entre cualquier usuario y su instalación asignada, sujeto a la capacidad en la demanda restringida por cada instalación. Este trabajo propone una metodología heurística para la solución del problema, los resultados de la experimentación demuestran la calidad de la heurística propuesta en relación a los métodos existentes en la literatura.

Método: Se propone una metodología heurística para la solución de este problema, la cual integra varios componentes, tales como un método voraz-adaptativo con una selección probabilística, búsqueda local voraz iterada y una búsqueda descendente por entornos variables.

Resultados: La evidencia empírica sobre un conjunto de instancias de localización usualmente utilizadas en la literatura, revela el impacto positivo de cada uno de los componentes desarrollados y de la calidad de la heurística propuesta en relación a los métodos existentes. Por ejemplo, la heurística propuesta pudo encontrar soluciones factibles a todas las instancias probadas excepto dos, mientras que el mejor de los otros tres métodos probados falló en 18 de las instancias.

Conclusión: Se encontró empíricamente que la heurística propuesta supera a la mejor heurística existente para este problema en términos de calidad de la solución, tiempo de ejecución y confiabilidad en la búsqueda de soluciones factibles en instancias difíciles.

Palabras Clave: Investigación de operaciones; Optimización combinatoria; Localización discreta; Problema de p -centro capacitado; Metaheurísticas.

Abstract

Introduction: The capacitated vertex p -center problem is a location problem that consists of placing p facilities and assigning customers to each of these facilities so as to minimize the maximum distance between any customer and its assigned facility, subject to demand capacity constraints for each facility.

Method: In this work, a metaheuristic for this location problem is presented. It integrates several components such as a greedy randomized construction with an adaptive probabilistic sampling scheme and an iterated greedy local search with variable neighborhood descent.

Results: Empirical evidence over a widely used set of benchmark instances on location literature reveals the positive impact of each of the developed components and the quality of the solutions delivered by the heuristic when compared with existing methods. For instance, the proposed heuristic was able to find feasible solutions to all but two instances, while the best of the existing methods failed in 18 of these instances.

Conclusion: It is found empirically that the proposed heuristic outperforms the best existing heuristic for this problem in terms of solution quality, running time, and reliability on finding feasible solutions for hard instances.

Keywords: Operations research; Combinatorial optimization; Discrete location; Capacitated p -center problem; Metaheuristics.

1 Introducción

La Investigación de Operaciones es la rama de las matemáticas que estudia y aborda problemas de toma de decisiones. Dentro de este campo, la optimización discreta se enfoca en problemas donde la naturaleza de las variables de decisión es de carácter discreto. En particular, en este trabajo se aborda el problema de *p*-centro capacitado (CpCP, por sus siglas en inglés: *Capacitated p-Center Problem*), que consiste en ubicar *p* instalaciones y asignar usuarios a cada una de ellas, de tal manera que se minimice la distancia máxima entre cualquier usuario y su instalación asignada, sujeto a la capacidad en la demanda restringida por cada instalación.

El CpCP es un problema de marcada importancia y relevancia en el campo de ciencias de localización ya que existen diversas aplicaciones prácticas que pueden ser modeladas como un CpCP. El criterio *minmax* que compone al problema es usualmente aplicado para minimizar los efectos adversos en el peor de los casos durante un servicio proporcionado. Usos comunes pueden ser encontrados en la planificación de distritos escolares o en el diseño del sistema de cobertura en salud, particularmente problemas que surgen en situaciones de emergencia tales como proporcionar servicios médicos mediante la localización de instalaciones de emergencia o ambulancias, o la ubicación de las estaciones de bomberos, donde es claro que el preservar la vida humana es lo más importante. Las aplicaciones anteriores, naturalmente muestran la presencia de un límite en la capacidad de servicio que cada instalación puede ofrecer.

Éste problema se define como sigue. Se tiene un conjunto de nodos V que representan a los usuarios o clientes (nodos que generan demanda) que necesitan algún tipo de servicio y un conjunto de nodos U que contiene a puntos donde pueden ubicarse las instalaciones; una demanda w_i para cada usuario $i \in V$; una capacidad de instalación s_j para cada nodo $j \in U$; una distancia d_{ij} entre cada par de nodos $i, j \in V \cup U$ y número dado p de ubicaciones deseadas. La capacidad s_j se refiere a que cada instalación, dependiendo del lugar j donde puede ser instalada, sólo puede atender un número limitado de la demanda. Se debe decidir dónde ubicar p instalaciones, y a cuál instalación debe asignarse cada usuario con el fin de minimizar la mayor distancia entre cualquier usuario y su instalación, sujeto a la restricción de capacidad de cada instalación. El CpCP es un problema técnicamente clasificado como NP-duro (Kariv y Hakimi, 1979), es decir, sumamente difícil de resolver en el sentido que cualquier algoritmo de solución que pretenda encontrar soluciones óptimas al problema emplea un

tiempo de cómputo que crece exponencialmente con el tamaño del problema en el peor de los casos, a menos que $NP = P$. Aquí P es la clase de problemas resolubles en tiempo polinomial. La consecuencia directa de este resultado es que para intentar resolver instancias del problema de mediana a gran escala es necesario recurrir a métodos heurísticos los cuales deben ser capaces de encontrar soluciones de buena calidad en tiempos de cómputo razonables.

El problema de p -centro en su versión sin capacidad ha sido ampliamente estudiado. Farahani y Hekmatfar (2009), y Elloumi et al. (2004) ofrecen una extensa revisión bibliográfica. Sin embargo la investigación en el $CpCP$ ha sido más limitada. El trabajo más significativo desde la perspectiva de optimización exacta se debe a Özsoy y Pinar (2006) y Albareda-Sambola et al. (2010). En el primer trabajo, los autores presentan un método exacto basado en la solución de una serie problemas de cobertura (SCP por sus siglas en inglés: *set covering problem*), usando un solucionador de programas lineales entero-mixtos (MILP por sus siglas en inglés: *mixed-integer linear programming*) durante una búsqueda iterativa de las distancias de cobertura. En el segundo trabajo, los autores proponen un método exacto basado en una relajación lagrangiana con una reformulación de cobertura. Evidentemente, dada la complejidad del problema, estos métodos exactos exhiben ciertas limitaciones, particularmente cuando se intentan aplicar a la resolución de instancias del problema de tamaño relativamente mayor.

Hasta donde se conoce, el método heurístico más significativo para el $CpCP$ se debe a Scaparra et al. (2004). En su trabajo, los autores desarrollaron una heurística basada en una búsqueda local a gran escala con un vecindario de múltiples intercambios representado por un grafo de mejora, explotando los principios teóricos de la optimización en redes, aquí se abordan instancias con 402 nodos y 40 instalaciones. En los últimos 13 años no ha habido, hasta donde tenemos conocimiento otro método heurístico para el $CpCP$ más exitoso que el de Scaparra et al. (2004).

El objetivo principal de nuestro trabajo es la integración de varios componentes heurísticos avanzados que han tenido éxito en otros problemas de optimización combinatoria dentro de una metaheurística para el $CpCP$. El término *metaheurística* se refiere a un método de solución a problemas de optimización que guía y orquesta una interacción entre procedimientos heurísticos constructivos, de búsqueda local y estrategias algorítmicas para crear un proceso sistemático capaz de escapar de óptimos locales y llevar a cabo una búsqueda robusta sobre un espacio de soluciones más amplio. Si bien, una metaheurística utiliza procedimientos y estrategias más sofisticadas que una heurística, esto no garantiza encontrar un óptimo global al problema. Para este fin, se desarrolla un método voraz-adaptativo con una selección pro-

babilística durante la construcción de la solución. Además, el método se ve reforzado por una fase de búsqueda local voraz iterada y una búsqueda descendente por entornos variables.

El valor y contribución del trabajo consiste precisamente en saber explotar adecuadamente la estructura particular del problema para poder desarrollar componentes que impacten significativamente en la calidad de las soluciones obtenidas. Los resultados experimentales del trabajo empírico donde se evalúan ampliamente cada uno de los componentes, así como la metaheurística propuesta en forma global, indican el exitoso impacto de cada uno de los componentes integrados en la metaheurística. Cuando se compara con el trabajo existente, el alcance de la heurística propuesta ofrece en la mayoría de los casos soluciones de mejor calidad que los encontrados por la heurística existente, así como un esfuerzo computacional significativamente menor. Por otra parte, si se compara con todo en el trabajo existente, incluyendo los mejores métodos exactos desarrollados hasta ahora, nuestra heurística resulta ser más robusta en la búsqueda de soluciones factibles para los conjuntos de instancias más difíciles.

2 Formulación del Problema

Sea $U = \{u_1, \dots, u_m\}$ el conjunto de ubicaciones potenciales para las p instalaciones y $V = \{v_1, \dots, v_n\}$ el conjunto de usuarios. Típicamente m es mucho mayor que p . Este trabajo asume el caso especial cuando $U = V$. Sea d_{ij} la distancia entera entre los nodos i y j . Cada $j \in V$ tiene una demanda entera w_j y una capacidad entera s_j . Una p -partición de V se define como una colección de subconjuntos $X = \{X_1, \dots, X_p\}$, con $K = \{1, \dots, p\}$, donde $X_k \subset V$ es un subconjunto de V formado por un subconjunto de nodos tal que $\bigcup_{k \in K} X_k = V$ y $X_k \cap X_q = \emptyset$ para todo $k, q \in K, k \neq q$. El conjunto de centros está dado por $P = \{c(1), \dots, c(p)\}$, donde $c(k)$ es el centro para el subconjunto X_k , es decir, el nodo que aloja la instalación que brinda servicio a los usuarios en X_k . El problema puede ser representado como

$$\min \left\{ \max_{k \in K} f(X_k) : X \in \Pi \right\}, \quad (1)$$

donde Π es el conjunto de soluciones factibles del problema, es decir, la colección de todas las p -particiones de V , y la función objetivo (a ser minimizada) está dada por la distancia entre la unidad más alejada del centro y el centro de cada territorio, es decir,

$f(X_k) = \max \{d_{j,c(k)} : j \in X_k\}$. Para un territorio dado X_k , su centro (o p -centro) $c(k)$ está dado por:

$$c(k) = \operatorname{argmin}_{i \in X_k} \left\{ \max_{j \in X_k} \left\{ d_{ij} : \sum_{q \in X_k} w_q \leq s_i \right\} \right\}. \quad (2)$$

Para un X_k dado, en caso que ningún $i \in X_k$ cumpla con la restricción de capacidad $\sum_{q \in X_k} w_q \leq s_i$, entonces se define su correspondiente $f(X_k) = \infty$. Para un X_k dado, el cálculo de $c(k)$ se obtiene fácil y rápidamente al resolver un problema de 1-centro, lo que puede hacerse con bastante eficiencia con una complejidad de $O(n)$. Definimos como $l(j)$ el índice o territorio $k \in K$ al cual pertenece el usuario j , es decir, $l(j) = k$ si y solo si $j \in X_k$.

3 Descripción de la Heurística

En los últimos años, una importante tendencia en el campo de las metaheurísticas es el de integrar diferentes componentes, dando como resultado, métodos híbridos exitosos que tratan de explotar la estructura de un problema específico. Como consecuencia de esto, a veces no está claro cómo nombrar una metaheurística, ya que utiliza los conceptos de diversos métodos. En este sentido, el método de solución propuesto utiliza un procedimiento de búsqueda local voraz iterada (IGLS por sus siglas en inglés: *iterated greedy local search*) como marco principal. En su fase de construcción, se emplea un método voraz-adaptativo con una selección probabilística. A este mecanismo de construcción se le conoce como GRASP (Feo y Resende, 1995) y ha sido muy utilizado con éxito en problemas de optimización combinatoria similares, tal como la segmentación de distritos comerciales (Ríos-Mercado y Fernández, 2009; Ríos-Mercado y Escalante, 2016). Posteriormente en la fase de mejora se aplica un método IGLS con una búsqueda descendente por entornos variables (VND por sus siglas en inglés: *variable neighborhood descent*). IGLS es una técnica originalmente propuesta por Ruiz y Stützle (2007) que resulta de una extensión de una heurística llamada búsqueda local iterada (ILS por sus siglas en inglés: *iterated local search*), Lourenço *et al.* (2002), proporciona una extensa explicación y revisión bibliográfica de este método. IGLS aplica iterativamente fases de destrucción, reconstrucción y búsqueda local a una solución inicial dada. Con el fin de escapar de un óptimo local y explorar otras regiones en el espacio de soluciones, IGLS aplica un procedimiento de perturbación para generar nuevos puntos de partida para la búsqueda local,

cambiando la solución actual. La destrucción/reconstrucción o perturbación consiste en la eliminación de algunos elementos de la solución actual con un criterio específico, seguido por un algoritmo voraz para obtener una nueva solución. Después de que una solución ha pasado a través de una fase de perturbación, se aplica un procedimiento de búsqueda local. El último paso es decidir, en base a un criterio de mejora, si la solución obtenida después de la búsqueda local debe sustituir a la solución incumbente en la siguiente iteración. IGLS sigue iterando hasta que se cumple algún criterio de parada. Una ventaja de IGLS es que permite diversificar y mejorar a lo largo de la búsqueda sin la necesidad de emplear estructuras de memoria complejas. Su simplicidad hace que sea aplicable a varios problemas de optimización combinatoria. Por ejemplo, Ruiz y Stützle (2007, 2008), Fanjul-Peyro y Ruiz (2010) y Urlings et al. (2010) proveen resultados significativos en el estado del arte para diferentes problemas de secuenciación de maquinas. Huerta-Muñoz et al. (2017) aplican exitosamente IGLS a un problema de segmentación de mercados.

El método VND (Hansen y Mladenović, 1999) es un algoritmo de mejora iterativo donde son usados t vecindarios que, por lo general, se ordenan según su tamaño. El algoritmo comienza con un primer vecindario realizando procesos de mejora iterativa hasta que se alcanza el óptimo local. Cuando no se encuentra una mejora para el h -ésimo vecindario entonces VND continúa la búsqueda en el $(h+1)$ -ésimo vecindario. Si se encuentra una mejora, el proceso de búsqueda comienza de nuevo en el primer vecindario. Cuando se termina de examinar el último vecindario, entonces el proceso de búsqueda termina y el procedimiento devuelve la solución final, que es un óptimo local con respecto a todos los t vecindarios. Se ha demostrado que VND puede mejorar considerablemente el rendimiento de los algoritmos de mejora iterativos respecto a la calidad de la solución final y el tiempo requerido en comparación con otros algoritmos iterativos. La idea de hibridizar ILS con VND ha sido utilizado con éxito anteriormente en otros problemas de optimización combinatoria (Subramanian et al., 2010; Martins et al., 2011).

El enfoque IGLS/VND propuesto se representa en el Algoritmo 1. Una solución inicial se obtiene en los (pasos 2-3). La búsqueda local (pasos 6-7) se realiza mientras la solución sigue mejorando. Sea $B(X)$ el subconjunto de particiones cuello de botella en X , es decir las particiones que contienen a la máxima de las distancias entre un usuario y un centroide actual, definido por $B(X) = \{k \in K : f(X_k) = f(X)\}$ y sea $J(X)$ el conjunto que contiene los usuarios con la máxima distancia desde una instalación en cada subconjunto, es decir,

$J(X) = \{j \in X_k : d_{j,c(k)} = f(X) \text{ para } k \in B(X)\}$. Para comparar dos soluciones X y X' , se utiliza un criterio efectivo de mejora propuesto en Scaparra et al. (2004) que incluye la reducción de los elementos de cuello de botella en la solución, definido como

$$X' \text{ es mejor que } X \Leftrightarrow \begin{cases} f(X') < f(X), \text{ ó} \\ f(X') = f(X), B(X') \subseteq B(X), J(X') \subset J(X) \end{cases} \quad (3)$$

Este criterio se cumple si X' disminuye el valor de la función objetivo o si X' reduce el número de particiones cuello de botella mientras no se incremente el valor de la función objetivo, sin crear nuevas particiones y nodos con algún cuello de botella. La solución X se actualiza si se encuentra una mejor solución de acuerdo con el criterio (3); en caso contrario, se aplica una perturbación de la solución X' (paso 11). El algoritmo se detiene cuando se alcanza el número máximo de iteraciones (indicado por r_{\max}).

Algoritmo 1 Procedimiento IGLS/VND

Require: $V = \{1, \dots, n\}$, $p \in \mathbb{N}$, $\alpha \in [0.0, 1.0]$ y $r_{\max} \in \mathbb{N}$

```

1:  $X' \leftarrow \text{Construccion}(V, p)$ 
2:  $X' \leftarrow \text{VND}(X')$ 
3:  $X \leftarrow X'$ 
4: while  $r_{\max} > 0$  do
5:    $X' \leftarrow \text{Perturbacion}(X', \alpha)$ 
6:    $X' \leftarrow \text{VND}(X')$ 
7:   if  $X'$  es mejor que  $X$  then
8:      $X \leftarrow X'$ 
9:   else
10:     $X' \leftarrow \text{Agitar}(X')$ 
11:   end if
12:    $r_{\max} \leftarrow r_{\max} - 1$ 
13: end while
14: return  $X$ 

```

Algoritmo 1. Pseudo-código de IGLS que toma como entrada: un conjunto de nodos V , un número p de instalaciones deseadas, un valor α , de destrucción de la solución durante la fase de perturación, y el máximo número de iteraciones r_{\max} .

El método contiene cuatro componentes, descritos brevemente a continuación:

CONSTRUCCIÓN: Consta de dos etapas: (a) localización de instalaciones y (b) asignación de usuarios. La primera etapa realiza una estrategia basada en método voraz aleatorizado y adaptativo con selección probabilística que diversifica el conjunto de centros potenciales. En la segunda etapa, un método voraz determinista con un criterio de distancia y capacidad realiza la asignación de los usuarios.

PERTURBACION: Aplica una destrucción y reconstrucción de la solución. El objetivo es

reducir los elementos cuello de botella. La fase de destrucción consiste en desconectar el $\alpha\%$ (definido como parámetro de destrucción α) de nodos localizados en X_k (excluyendo al centro localizado en $c(k)$) que tengan una probabilidad mayor de empeorar la solución, definida por un criterio de distancia y capacidad.

VND: Emplea dos diferentes estructuras de vecindario denotadas por N_1 y N_2 , que realizan movimientos de *reinserción* (cambio de un nodo $j \in X_k$ a la partición X_q) e *intercambio* (intercambio de los nodos $j \in X_k$ y $i \in X_q$ a la partición X_q y X_k respectivamente). Para cada uno de los vecindarios, el movimiento potencial toma en cuenta los factores de distancia y capacidad.

AGITACIÓN: Este método realiza una eliminación y reconstrucción total de varias particiones X_k candidatas, lo cual diversifica parcialmente la estructura de la solución actual, guiando el método a otras regiones de la búsqueda más prometedoras durante las siguientes fases de mejora.

Un mayor detalle de la composición formal de los métodos se discute en Quevedo-Orozco y Ríos-Mercado (2015).

4 Resultados Computacionales

La heurística propuesta se evalúa en un conjunto de instancias de prueba obtenidos de la literatura especializada, a continuación, se detalla cada uno de los componentes de la experimentación.

Descripción de las instancias de prueba: Para cada experimento, se utilizaron cinco conjuntos de instancias distintas, los cuales han sido utilizados en trabajos anteriores sobre este problema.

(Grupo A) Beasley OR-Library: Este conjunto de instancias, propuesto en Beasley (1990) para el problema de la p -mediana capacitada (*CpMP*, por sus siglas en inglés: *capacitated p-median problem*), contiene dos subgrupos de 10 casos con capacidad igual en cada instalación. El primer subgrupo contiene 50 nodos de demanda y 5 instalaciones, el otro subconjunto tiene 100 nodos de demanda y 10 instalaciones que deben ser ubicadas.

(Grupo B) Galvão y ReVelle: Este conjunto de instancias se ha generado por Scaparra et al. (2004) específicamente para el *CpCP* basado en el conjunto de instancias de Galvão y Re-

Velle (1996) para el SCP. Este conjunto contiene instancias con 100 y 150 usuarios, y de 5 a 15 instalaciones, con capacidad de la instalación variable. El conjunto original se compone de dos redes de nodos que se han generado aleatoriamente.

(Grupo C) Lorena y Senne: Este conjunto, propuesto en Lorena y Senne (2004) para el *CpMP* incluye 6 instancias a gran escala cuyo tamaño varía entre 100 y 402 usuario y de 10 a 40 instalaciones, con igual capacidad en cada instalación.

(Grupo D) Ceselli y Righini: existe un conjunto de instancias agregado recientemente a la OR-Library y propuesto en Ceselli y Righini (2005), el cual se considera difícil de resolver por los problemas de localización capacitados, tales el *CpMP* y *CpCP*, ya que la complejidad de su solución se ve significativamente influenciada por la relación entre el número de nodos (n) y el número de instalaciones a ubicar (p). Cuanto menor sea la relación n/p la instancia tiende a ser más difícil de resolver debido a la complejidad combinatorial del problema (Kariv, O., y Hakimi, S.L., 1979). Se observa un comportamiento similar cuando se utiliza este conjunto para el *CpCP*. En los conjuntos anteriores A, B, y C, esta relación n/p es 10 o más. El conjunto D está compuesto por 4 subconjuntos α , β , γ y δ , cada subconjunto consta de 40 instancias creadas a partir de la siguiente serie: 20 de ellas tienen un grafo con 50 y 100 nodos; los otros 20 casos fueron generados aleatoriamente con cardinalidad entre 150 y 200. Los mismos 40 casos fueron resueltos con diferente número de centros para cada subconjunto de instancias: α con $p = \lfloor n/10 \rfloor$, β con $p = \lfloor n/4 \rfloor$, γ con $p = \lfloor n/3 \rfloor$ y δ con $p = \lfloor 2n/5 \rfloor$. La capacidad total se mantiene en todos los subconjuntos como $s_i = \lceil 12n/p \rceil$. Nos referimos a estos cuatro conjuntos de instancias diferentes como D- α , D- β , D- γ , y D- δ , cuyo literal expresa la complejidad de cada subconjunto, donde D- α se refiere al subconjunto fácil y D- δ al subconjunto difícil.

(Grupo E) Reinelt: Este conjunto fue generado por Lorena y Senne (2004) específicamente para el *CpMP* basado en las instancias del TSPLIB compilado por Reinelt (1994). El conjunto contiene 5 instancias con 3,038 usuarios y entre 600 a 1,000 instalaciones, con una capacidad de la instalación por instancia, calculada por $s_i = \left\lceil 0.8 \sum_{j \in V} w_j / p \right\rceil$. Este conjunto se considera de gran escala y con proporciones de n/p entre 3 y 5.

Condiciones computacionales: La heurística se implementó en C++ y fue compilada con la versión GNU g++ 4.4.5. Para obtener soluciones exactas o cotas inferiores, se utilizaron los métodos exactos de Özsoy y Pinar (2006), denominado como OP, y Albareda-Sambola et al. (2010), denominado como ADF. En estos métodos se utiliza, ILOG CPLEX 12.5 como solucionador del subproblema de programación lineal (LP, por sus siglas en inglés). Se estableció el límite de tiempo de 1 hora y un uso máximo de memoria de 1 Gb para los conjuntos de instancias de prueba A, B, C, D- α , D- β , D- γ y D- δ ; para el conjunto E se fijo en 6 horas y un uso de memoria de 4 Gb. Cada uno de los experimentos se llevó a cabo en un equipo con AMD Opteron 2.0 GHz (x16), 32 GiB de RAM en Debian 6.0.8 GNU/Linux Kernel 2.6.32-5 en una arquitectura de 64 bits.

Ajuste de parámetros algorítmicos: El propósito de este experimento es primeramente afinar la heurística con respecto al parámetro α para cada conjunto de instancias. Esto se consigue mediante un detallado análisis estadístico tal como se describe a continuación. La variable de respuesta estudiada es el promedio del porcentaje de la desviación relativa o DR, definida de la manera siguiente:

$$DR = 100 \times \frac{f(X) - f^{LB}}{f^{LB}}, \quad (4)$$

donde $f(X)$ es el valor de la función objetivo encontrada por el método para cada caso en particular y f^{LB} su mejor cota inferior conocida. Esta cota se obtuvo mediante la aplicación de cualquiera de los métodos exactos OP o ADF. En algunos casos, esta cota inferior resultó ser la solución óptima, pero cuando un método exacto cumple la condición de parada por límite de tiempo o uso de la memoria, se reporta una cota inferior. En otras palabras, DR indica una sobreestimación de la calidad de solución en términos de que tan lejos está del óptimo global del problema. Por ejemplo, un DR=5% indica que la solución reportada por el método está a un 5% o menos de distancia del valor óptimo global. El límite de iteración heurística se fijó en 100 para los conjuntos de instancias A, B, C, D- α , D- β , D- γ y D- δ , y 50 para el conjunto E; y se ha ejecutado para todos los valores del parámetro α entre 0 y 1 en intervalos de 0.1, en todos los conjuntos de instancias, realizando 5 repeticiones en este experimento. Se separaron los resultados en 8 bloques, de acuerdo con el número de conjuntos de instancias con sus respectivos subconjuntos y estudiamos un solo factor α de cada bloque compuesto por 5 repeticiones, donde DR es la variable de respuesta.

Basado en una prueba Tukey con diferencia significativa honesta (HDS, por sus siglas

en inglés: *honest significant difference*) del 95%, que permitirá identificar diferencias significativas entre los diversos bloques de prueba. Se observó que el conjunto $\{0.7, 0.6, 0.7, 0.6, 0.4, 0.5, 0.4, 0.4\}$ de valores del parámetro α dio los mejores resultados para los conjuntos de instancias A, B, C, D- α , D- β , D- γ , D- δ y E, respectivamente. Para los experimentos siguientes, se han utilizado estos valores de α y ejecutado la heurística con 1,000 iteraciones como límite para los conjuntos de instancias A, B, C, D- α , D- β , D- γ y D- δ , y 200 iteraciones como límite para el conjunto E, con 30 repeticiones.

Comparación con el estado del arte: El principal propósito de estos experimentos es proporcionar una comparación detallada entre la heurística propuesta (denominada ahora como QR y descrita en el Algoritmo 1) y la de Scaparra et al. (2004) (denominado SPS) que es la mejor existente para este problema. Además, se ha decidido incluir los resultados obtenidos por los métodos exactos OP y ADF. La Tabla 1 resume la comparación entre los métodos para todos los conjuntos de instancias en términos de su promedio de DR, tiempo de ejecución, y el número de casos infactibles para cada método. La sección de "Tiempo promedio (s)" da el tiempo de ejecución en segundos y "Promedio de desviación (%)" expresa el porcentaje de DR con respecto del valor óptimo o la mejor cota inferior conocida para cada método. En el caso de los métodos exactos, la desviación representa la diferencia entre la cota inferior y superior. Para el método propuesto QR, se muestra el rendimiento promedio de tiempo durante las 30 repeticiones independientes, donde "QR²" indica el promedio del mejor DR encontrado para la instancia, sobre todas las repeticiones. La sección de "Número de fallos" representa el total de instancias infactibles por cada uno de los métodos. Se dice que una solución es infactible, cuando esta es reportada después de que el criterio de paro es satisfecho, pero no cumple alguna de las restricciones del problema.

Tabla 1: Resumen de la comparación entre los métodos para todos los conjuntos de instancias.

Conjunto	Promedio de desviación (%)				Tiempo promedio (s)				Número de fallos			
	ADF	OP	SPS	QR ²	ADF	OP	SPS	QR	ADF	OP	SPS	QR
A	0.00	0.00	5.41	0.23	27.53	33.46	5.34	0.39	0	0	0	0
B	0.00	0.00	6.70	3.41	178.53	154.52	27.30	0.76	0	0	0	0
C	0.00	0.00	31.79	3.58	233.63	414.76	285.43	2.39	0	0	0	0
D- α	0.18	0.00	12.62	2.72	255.36	139.72	25.22	0.84	1	0	0	0
D- β	4.74	1.62	18.03	18.17	640.68	498.19	26.47	2.73	9	6	5	0
D- γ	4.27	0.86	16.99	29.68	941.87	496.28	27.05	4.49	12	3	8	0
D- δ	5.75	1.93	17.56	58.85	1320.19	453.30	44.66	5.55	14	4	21	2
E	167.34	-	445.91	190.58	21600.00	1167.06	1877.61	350.42	5	5	0	0

Analizando esta tabla, se observa que QR es considerablemente más rápido para todos los conjuntos de instancias en todos los métodos. Respecto a la calidad de la solución, el método propuesto proporciona soluciones aceptables para los conjuntos de instancias A, B, C y para los subconjuntos D- α y D- β . En los subgrupos D- γ y D- δ , la comparación entre QR y SPS en términos de calidad de la solución no tiene demasiado sentido porque SPS falla en la búsqueda de soluciones factibles para un gran número de casos. Como podemos ver SPS falla en la búsqueda de una solución factible en 34 casos, mientras QR falla sólo en 2. En el análisis del conjunto de instancias E, QR claramente supera a todos métodos, cuyo tiempo de cómputo y recursos crecen considerablemente; en el aspecto de factibilidad, QR es aún mejor, ya que fue capaz de encontrar soluciones factibles para todos los casos de este conjunto de instancias. Cuando se compara con los métodos exactos sobre todos los conjuntos de instancias, QR es aún más confiable en términos de número de soluciones factibles encontradas. Los métodos exactos ADF y OP fallaron en la búsqueda de soluciones factibles en 41 y 18 instancias, respectivamente, mientras QR falló en sólo 2 instancias.

Análisis de los componentes: En este último experimento, evaluamos el valor que cada componente da a la heurística. Consideramos los tres componentes esenciales del método: Perturbación, VND y Agitación. El experimento consiste en deshabilitar un componente a la vez y ejecutar la heurística utilizando 1,000 iteraciones como límite para los conjuntos de instancias A, B, C, D- α , D- β , D- γ y D- δ , y 200 como límite iteraciones para el conjunto E, con 30 repeticiones. Se utiliza el mismo conjunto de valores α del experimento anterior.

Tabla 2: Análisis de los componentes dentro de la heurística QR.

Conjunto	Todos	Componentes			Contribución (%)		
		Perturbación	VND	Agitación	Perturbación	VND	Agitación
A	0.23	17.25	2.77	2.70	77.25	11.52	11.23
B	3.41	12.53	4.66	5.33	74.19	10.16	15.65
C	3.58	36.63	8.32	15.59	66.38	9.51	24.12
D- α	2.72	31.87	5.61	5.39	83.98	8.32	7.70
D- β	18.17	106.97	33.19	29.04	77.42	13.09	9.48
D- γ	29.68	195.23	46.81	45.03	83.60	8.65	7.75
D- δ	58.85	261.94	105.76	92.62	71.57	16.53	11.90
E	190.58	496.62	245.65	192.91	84.20	15.15	0.64

La Tabla 2 muestra la comparación de los componentes de cada conjunto de instancias. En esta tabla, la columna "Todos" representa la diferencia promedio cuando todos los componentes están habilitados, que coincide con el valor que se muestra en la columna QR² de la Tabla 1. Cada columna en la sección "Componentes" representan el componente desha-

bilitado durante el experimento y muestra el valor promedio de desviación obtenido. La sección "Contribución %" muestra el porcentaje que el componente ofrece con respecto al valor total que se muestra en "Todos". Por ejemplo, en la fila asociada al conjunto A, QR obtiene soluciones que tienen una desviación promedio de 0.23%. Cuando QR se ejecuta con el componente de Perturbación deshabilitado, la desviación media empeora al 17.25%. Esto representa una contribución 77.25%, como se indica en la Contribución (%), calculada de la manera siguiente:

$$\text{Contribución} = 100 \times \frac{\text{perturbación} - \text{todos}}{\sum_{i=1}^c (D_i - \text{Todos})} \quad (5)$$

donde c es el número total de los componentes, y D_i es la calidad promedio de la solución obtenida cuando el i -ésimo componente es desactivado. Es notable que el componente más influyente dentro de la heurística es la perturbación, seguido de VND y Agitación. Esto es consistente con el análisis estadístico, el cual demostró que el parámetro α , utilizado en la perturbación, influye en la variable de respuesta. Sin embargo, los otros dos componentes agregan valor al rendimiento general. La aportación de VND varía de 8.32% a 16.53%, y la aportación del método de Agitación varía de 0.64% a 24.12%.

Análisis Asintótico: La Figura 1 muestra una comparación de los métodos en términos de su tiempo de ejecución asintótica y recursos de memoria utilizadas con respecto al tamaño de la instancia. La estadística de la memoria indica el tamaño máximo de bits utilizado (Loosemore et al, 2012), es decir, el número máximo de bits de memoria física que cada método utilizó simultáneamente. Como puede verse, los recursos utilizados por el método propuesto son más bajos que los utilizados por los otros tres métodos. En particular, los requisitos de uso de memoria de los dos métodos exactos, como se esperaba, son considerablemente más grandes.

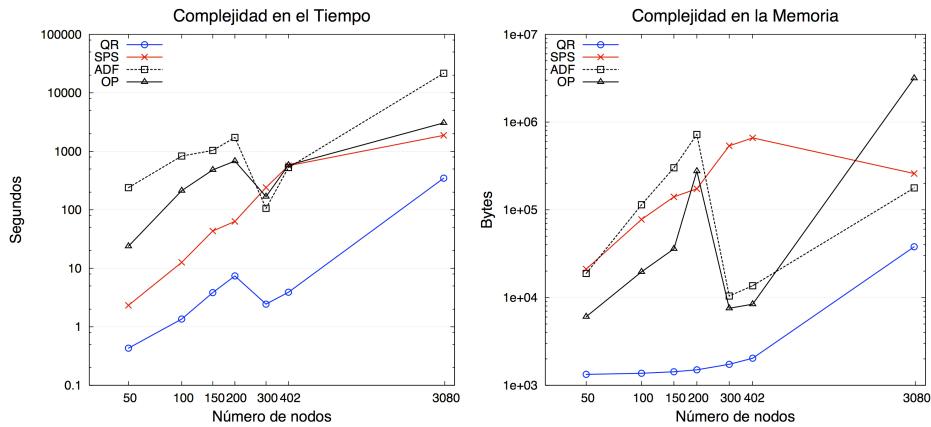


Figura 1: Comparación asintótica en función de n entre los métodos en términos de tiempo de ejecución y uso de memoria. Los ejes se encuentran en escala logarítmica.

5 Conclusiones

En este trabajo de investigación se ha ilustrado la aplicación de un método heurístico que integra varios componentes avanzados de la optimización metaheurística, tales como un método voraz-adaptativo con una selección probabilística y búsqueda local voraz iterada con una búsqueda descendente por entornos variables para uno de los problemas de mayor importancia en el campo de ciencias de localización.

Los resultados revelan contundentemente que la heurística propuesta supera a la mejor heurística conocida en términos de calidad de la solución, el tiempo de ejecución y confiabilidad en la búsqueda de soluciones factibles en los casos más difíciles, convirtiéndose en un avance significativo en el estado del arte en esta rama del conocimiento. El rendimiento del método propuesto es más robusto que el de los métodos exactos existentes, requiriendo menos esfuerzo computacional y memoria para la obtención de soluciones de calidad razonablemente buena para los conjuntos de instancias A, B, y C. Para los casos difíciles en los conjuntos de instancias D- γ y D- δ , las desviaciones respecto a optimalidad de las soluciones heurísticas resultaron ser de calidad aceptable, sin embargo, son obtenidas más rápidamente que el método existente. Para este conjunto difícil, nuestra heurística encontró soluciones factibles para casi todos los casos probados, que es claramente superior a la heurística de SPS, ya que ésta última falló en varias ocasiones. Para el conjunto a gran escala E, ambos métodos exactos fracasaron en la búsqueda de soluciones óptimas en todas las instancias. Nuestra heurística encontró mejores soluciones que la mejor solución reportada por los métodos exactos y además superó claramente a SPS. En un análisis detallado de los componentes, hemos visto que

el éxito de la heurística se debe principalmente al método de perturbación y VND. Sin embargo, para los conjuntos de instancias A, B, y C, el método de agitación resultó muy útil también.

El método propuesto ofrece soluciones robustas en un corto tiempo para los problemas discutidos previamente en la literatura. Para los conjuntos D y E analizados en este trabajo, el método garantiza un mayor éxito en la búsqueda de soluciones factibles que la de la heurística existente. Esto se logró gracias a una explotación adecuada de la estructura matemática del problema, lo cual permitió una eficaz e inteligente explotación de la misma en el desarrollo de los componentes, superando notablemente a los mejores métodos existentes. Por ejemplo, fue necesario definir los conjuntos cuellos de botella que dependen a su vez de la estructura min-max de la función objetivo.

Agradecimientos

Agradecemos los comentarios de tres revisores anónimos los cuales ayudaron a mejorar la exposición del artículo. Este trabajo fue apoyado por el Consejo Mexicano Nacional de Ciencia y Tecnología (Apoyo CONACYT CB-2011-01-166397) y la Universidad Autónoma de Nuevo León (Apoyo UANL-PAICYT CE728-11). También agradecemos a María Scaparra y Juan A. Díaz por proporcionarnos el código fuente de sus métodos.

Referencias

- Albareda-Sambola, M., Díaz, J.A., y Fernández, E. (2010). Lagrangean duals and exact solution to the capacitated p -center problem. *European Journal of Operational Research*, 201(1):71–81.
- Beasley, J.E. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–72.
- Ceselli, A., y Righini, G. (2005). A branch-and-price algorithm for the capacitated p -median problem. *Networks*, 45(3):125–42.
- Elloumi, S., Labbé, M., y Pochet, Y. (2004). A new formulation and resolution method for the p -center problem. *INFORMS Journal on Computing*, 16(1):84–94.
- Farahani, R.Z. y Hekmatfar, M. (2009). Editores. *Facility Location: Concepts, Models, Algorithms and Case Studies*. Physica-Verlag, Heidelberg, Alemania.
- Fanjul-Peyro, L. y Ruiz, R. (2010). Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research*, 207(1):55–69.
- Feo, T.A., y Resende, M.G.C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133.
- Hansen, P., y Mladenović, N. (1999). An introduction to variable neighborhood search. En: S.

- Voß, S. Martello, I.H. Osman y C. Roucairol (editores), *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pp. 433–58, Kluwer, Boston, EUA.
- Huerta-Muñoz, D.L., Ríos-Mercado, R.Z., y Ruiz, R. (2017). An iterated greedy heuristic for a market segmentation problem with multiple attributes. *European Journal of Operational Research*, 261(1):75–87.
- Kariv, O., y Hakimi, S.L. (1979). An algorithmic approach to network location problems, I: The p -centers. *SIAM Journal of Applied Mathematics*, 37(3):513–38.
- Loosemore, S., Stallman, R.M., McGrath, R., Oram, A., y Drepper, U. (2012). *The GNU C Library Reference Manual: For version 2.17*. Free Software Foundation.
- Lorena, L.A.N., y Senne, E.L.F. (2004). A column generation approach to capacitated p -median problems. *Computers & Operations Research*, 31(6):863–76.
- Lourenço, H.R., Martin, O., y Stützle, T. (2002). Iterated local search. En: F. Glover F y G.A. Kochenberger (editores), *Handbook of Metaheuristics*, pp. 321–353, Kluwer, Boston, EUA.
- Martins, A.X., Duhamel, C., Souza, M.C., Saldanha, R.R., y Mahey, P. (2011). A VND-ILS heuristic to solve the RWA problem. En: J. Pahl, T. Reiners y S. Voß (editores), *Network Optimization*. Lecture Notes in Computer Science, vol. 6701, pp. 577–582, Springer, Berlin, Alemania, 2011.
- Özsoy, F.A., y Pınar, M.Ç. (2006). An exact algorithm for the capacitated vertex p -center problem. *Computers & Operations Research*, 33(5):1420–36.
- Quevedo-Orozco, D.R., y Ríos-Mercado, R.Z. (2015). Improving the quality of heuristic solutions for the capacitated vertex p -center problem through iterated greedy local search with variable neighborhood descent. *Computers & Operations Research*, 62:133–144.
- Reinelt, G. (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer, Heidelberg, Alemania.
- Galvão, R. D. y ReVelle, C. (1996). A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88(1):114–123.
- Ríos-Mercado, R.Z., y Escalante, H.J. (2016). GRASP with path relinking for commercial districting. *Expert Systems with Applications*, 44:102–113.
- Ríos-Mercado, R.Z. y Fernández, E. (2009). A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776.
- Ruiz, R., y Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–49.
- Ruiz, R., y Stützle, T. (2008). An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143–59.
- Scaparra, M.P., Pallottino, S., y Scutellà, M.G. (2004). Large-scale local search heuristics for the capacitated vertex p -center problem. *Networks*, 43(4):241–55.
- Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S., y Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–911.

- Urlings, T., Ruiz, R., y Stützle, T. (2010). Shifting representation search for hybrid flexible flowline problems. *European Journal of Operational Research*, 207(2):1086–95.