

Soluciones al problema p-hub usando algoritmos genéticos

Pedro Jiménez Latorre¹, Ángel Barbancho Sánchez, César Hervás Martínez¹

Resumen- El problema P-hub mediano es un problema, considerado NP-complejo, de máxima actualidad dentro de los problemas de localización. Las aproximaciones que encontramos en la literatura se basan, en general, en técnicas de computación evolutiva y en heurísticas desarrolladas ad hoc para resolver este problema. Lo novedoso de este trabajo consiste en proponer un algoritmo genético, totalmente parametrizado, que proporciona soluciones aceptables al problema P-hub mediano sin restricciones de capacidad y con asignación simple.

Nuestro algoritmo utiliza individuos con representación entera, operadores de cruce y mutación especialmente diseñados para esta representación y calcula el tamaño de la población inicial en función de la complejidad del problema.

Palabras clave: Localización de hub, algoritmo genético.

I. INTRODUCCIÓN

El problema p-hub mediano con asignación simple, parte de n nodos interconectados entre sí, entre los que se produce un determinado flujo que origina unos costes. Para su solución se localizan p hubs ($p < n$). Esto es, se encuentran p nodos que servirán de enlace entre los demás. Los nodos hub estarán conectados entre sí, pero un nodo no hub solamente estará conectado con uno y solo un hub. Este es un problema de localización, conexión y asignación, que encuentra los hubs idóneos y asigna a cada hub, un conjunto de nodos. Los conjuntos de nodos asignados a cada hub deben ser disjuntos.

En la literatura encontramos diversos acercamientos al problema, o a problemas similares, que utilizan técnicas de computación evolutiva, aunque es raro encontrar un algoritmo genético propuesto para resolver el problema. Las técnicas más usadas para afrontar este tipo de problemas son Branch-and-Bound como se puede ver en [9], [10], [16] y [25], Greedy como en [3] y búsqueda tabú (ver [1], [15] y [28]). Otras técnicas, dentro de la computación evolutiva, menos utilizadas son GRASP [15],

Simulated annealing [9] y redes neuronales de Hopfield [26]. Los pocos algoritmos genéticos que aparecen en la literatura para resolver este problema, casi siempre son algoritmos híbridos como en [22], que utiliza un algoritmo genético con búsqueda multiarranque y en [1] que es un algoritmo híbrido utilizando técnicas de búsqueda tabú. Más recientemente en [6] encontramos algoritmos genéticos para resolver diversos problemas de localización, aunque no se afronta el problema del p-hub mediano con asignación simple.

Todo esto, nos motivó para desarrollar un algoritmo genético puro, que intentase dar soluciones aceptables a este problema que tiene múltiples aplicaciones en campos de tanta utilidad y actualidad como son las telecomunicaciones [7] y [29], telefonía móvil [23], el transporte aéreo o de cualquier tipo [5], [14], [21] y [30], la mensajería [9], [10] y [13], los servicios de emergencia [7], etc.

Pretendíamos conseguir un algoritmo genético totalmente parametrizado, que abordase con garantías, problemas de gran envergadura. Para ello, partimos de la librería JCLEC [31]², e integramos dentro de ella a nuestro algoritmo.

En la sección II se describe el problema y se presenta la formulación matemática utilizada para resolverlo. La sección III muestra la estructura del individuo y las fases del algoritmo utilizado. En la sección IV se muestran las pruebas realizadas. Las dos últimas secciones están dedicadas a las conclusiones obtenidas.

II. DESCRIPCIÓN DEL PROBLEMA

Los problemas de localización aparecen cuando los encargados de tomar decisiones deben seleccionar donde se ubicarán una o varias instalaciones o servicios [31]. Las instalaciones³, pueden ser de muy diverso tipo, como hospitales, industrias,

¹ Departamento de Informática y Análisis Numérico.
Universidad de Córdoba.
Campus Universitario de Rabanales, Edificio C-2.
e-mail: pjimenez@uco.es

² JCLEC (Java Class Library for Evolution Computation) es una biblioteca de clases Java para Computación Evolutiva desarrollada por el grupo de investigación Aprendizaje y Redes Neuronales (AYRNA) de la Universidad de Córdoba.

³ Recientemente está siendo de amplio interés el estudio de las instalaciones de telefonía móvil [23] y las generadas en las redes de telecomunicaciones en general [29].

aeropuertos, plantas generadoras de electricidad, mercados, gasolineras etc. Las decisiones, en estos problemas, se toman bajo una serie de criterios preestablecidos, cumpliendo unas determinadas restricciones de forma que se satisfagan de forma óptima las necesidades demandadas por los usuarios. Uno de los criterios determinantes de estos problemas es decidir que norma se utilizará para estimar las distancias.

Cuando el problema trata de una localización dentro de una ciudad, se suele usar una norma rectilínea⁴. Que se adapta mejor a la estructura de la calles que hay que recorrer, para unir dos puntos. Cuando se trata de problemas en zonas rurales o en el espacio aéreo, se suele utilizar la norma euclídea, que dice que la distancia entre dos puntos viene dada por la longitud del segmento que los une.

Los criterios usados, relacionados de forma directa con el tipo de instalaciones o servicios a localizar, van desde el que precisa que los servicios estén tan próximos a los puntos de demanda como sea posible, hasta el que pretende lo opuesto. Las restricciones pueden imponer que el número de instalaciones a localizar esté prefijado, que los servicios dispongan de ciertas capacidades que no puedan ser sobrepasadas, etc. En función de las combinaciones de criterios, restricciones, entorno sobre el que se desarrolla el problema y cualquier otro elemento que intervenga en la determinación óptima de las instalaciones, aparecen diferentes modelos que son resueltos de manera diferente [5].

El problema de localización del p-hub [20] y [21], es un problema de localización, conexión y asignación, en el que el conjunto de localizaciones coincide con el de puntos de demanda. La asignación se consigue eligiendo los nodos hub de manera que cada nodo no hub esté conectado con uno o con varios nodos hub. Cada hub se conecta con cada uno de los puntos de demanda a los que da servicio, mediante un arco. Todos los hub están conectados mediante un subgrafo completo.

Cuando cada nodo se conecta con un único hub, al problema se le conoce como de asignación simple. El caso en el que los nodos puedan conectarse con mas de un hub, se conoce como problema de asignación múltiple.

Las soluciones de estos problemas se pueden representar mediante un grafo $G=(V,E)$, en el que los vértices de G se corresponden con los puntos de

demanda y localización. Los arcos de este grafo, representan a las asignaciones o conexiones.

Supongamos que V_D es el conjunto de vértices que se corresponde con los puntos de demanda, V_L el conjunto de vértices que representan a las localizaciones, E_L los arcos que sólo tienen vértices de localización, y E_D el conjunto de arcos en los que al menos un vértice es punto de demanda. Llamaremos subgrafo de localización-conexión a (V_L, E_L) y grafo parcial de asignación a (V, E_D) . El subgrafo (V_L, E_L) es el que conecta a todos los hub.

O'Kelly [21], describe el problema considerando los siguientes valores

W_{ij} = número de unidades de tráfico que se envían desde el punto i al j .

C_{ij} = coste unitario por unidad de tráfico enviada sobre el arco (i,j) .

Se supone que tanto W_{ii} como C_{ii} son igual a 0, para todo i . Si tanto i como j son hubs, el coste por unidad de tráfico sobre el arco (i,j) es igual a aC_{ij} . Siendo a el coeficiente de coste para transferencia entre dos hubs. Se consideran también otros dos coeficientes β y γ que sirven para calcular el costo de "recogida" desde un punto origen hasta un hub, y de "distribución" desde un hub hasta un punto de destino.

Las variables de decisión vienen definidas por

$$X_{ij} = \begin{cases} 1, & \text{si el punto } i \text{ es asignado al hub } j \\ 0, & \text{en otro caso} \end{cases}$$

$$Y_j = \begin{cases} 1, & \text{si el punto } j \text{ es un hub} \\ 0, & \text{en otro caso} \end{cases}$$

La formulación matemática del problema de asignación simple es

$$\min \sum_{i=1}^n \sum_{j=1}^n W_{ij} \left(\beta \sum_{k=1}^n X_{ik} C_{ik} + \gamma \sum_{m=1}^n X_{mj} C_{mj} + \alpha \sum_{k=1}^n \sum_{m=1}^n X_{ik} X_{jm} C_{km} \right) \quad (1)$$

sujeta a las siguientes restricciones

$$\sum_{j=1}^n X_{ij} = 1, \quad i = 1, \dots, n$$

$$X_{ij} \leq Y_j \quad i = 1, \dots, n; \quad j = 1, \dots, n$$

$$\sum_{j=1}^n Y_j = p$$

$$X_{ij}, Y_j \in \{0,1\} \quad i = 1, \dots, n \quad j = 1, \dots, n$$

Campbell propuso en [7], una formulación para el problema p-hub mediano con asignación simple,

⁴ La norma rectilínea, mide la distancia entre dos puntos en función del camino más corto que hay entre ellos. Cada camino está formado por un conjunto encadenado de segmentos que parten de uno de ellos y llega hasta el otro.

cuyo número de variables era del orden $O(n^4)$. Para ello, definió la fracción de flujo desde un nodo origen i hasta un nodo destino j , a través de los hubs k y m , como X_{ijkm} . El coste por unidad de flujo entre i y j , vía los hubs k y m es $C_{ijkm} = C_{ik} + \alpha C_{km} + C_{mj}$.

La formulación matemática resulta entonces:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{m=1}^n W_{ij} C_{ijkm} X_{ijkm} \quad (2)$$

sujeto a

$$\sum_{k=1}^n Y_k = p$$

$$\sum_{k=1}^n \sum_{m=1}^n X_{ijkm} = 1, \quad i = 1, \dots, n, \quad j = 1, \dots, n$$

$$Y_k \in \{0, 1\}, \quad k = 1, \dots, n$$

$$X_{ijk} \in \{0, 1\}, \quad i = 1, \dots, n; \quad k = 1, \dots, n$$

$$X_{ik} \leq Y_k, \quad i = 1, \dots, n; \quad k = 1, \dots, n$$

$$\sum_{j=1}^n \sum_{m=1}^n (W_{ij} X_{ijkm} + W_{ji} X_{jimk}) = \sum_{j=1}^n (W_{ij} + W_{ji}) X_{ik},$$

$$i = 1, \dots, n; \quad k = 1, \dots, n$$

$$X_{ijkm} \geq 0, \quad i = 1, \dots, n; \quad j = 1, \dots, n; \quad k = 1, \dots, n; \quad m = 1, \dots, n$$

Otras formulaciones de este problema se pueden ver en [19], [30] y en [8]. En [19] O'Kelly y otros, presentan una linearización de las formulaciones utilizadas en [30]. Jaime Ebery [8], desarrolla nuevas formulaciones para el problema que son estudiadas para problemas con dos o tres hub solamente.

En nuestro algoritmo utilizamos la formulación de O'Kelly, a pesar de ser la más compleja.

Utilizando la formulación de Campbell, concretamente el modelo desagregado que propone en [7], Mayer y Wagner han desarrollado un algoritmo Branch-and-Bound, llamado HubLocator [18] que resuelve problemas de asignación múltiple con hasta 30 nodos. Este algoritmo, se basa en las técnicas de ajuste dual y ascendente dual desarrolladas por Klineciewicz en [16].

III. ESTRUCTURA DEL INDIVIDUO Y DESCRIPCIÓN DE LAS FASES DEL ALGORITMO

El problema que queremos abordar con nuestro algoritmo, se plantea de la siguiente forma: tenemos un conjunto de n nodos interconectados entre si, entre los que se produce un determinado flujo que origina unos costes. Para su solución se localizan p hubs ($p < n$). Esto es, se encuentran p nodos que servirán de enlace entre los demás. Los nodos hub

estarán conectados entre si, pero un nodo no hub solamente estará conectado con uno y solo un hub.

Para representar a cada individuo se ha utilizado un esquema de representación entera. Su estructura está definida en un vector de enteros de dimensión n , en el que los índices representan a cada uno de los nodos de la red y los valores de los mismos definen o distinguen al nodo en cuestión, como hub o como no hub. Los índices comienzan en 0 y acaban en $n-1$. Los valores permitidos para cada nodo, según que sea hub o no lo sea, son los siguientes:

- Los elementos del vector cuyo índice corresponde a un nodo no hub, sólo podrán tener valores comprendidos entre $[0, p-1]$.
- Los elementos del vector cuyo índice corresponde a un nodo hub, tendrán valores comprendidos entre $[p, 2p-1]$.

Cada valor asignado a un nodo no hub, indica a que hub está conectado, teniendo en cuenta que los índices empiezan en 0. Más claramente, esto quiere decir que un valor 2, en uno de estos elementos, quiere decir, que ese elemento está conectado con el tercer hub.

Esta representación tiene la ventaja de que los nodos asignados a un hub, son los que tienen un valor igual al del hub menos p , siendo p el número de hubs.

A. Tamaño de la Población

El tamaño de la población influye en la diversidad de la población y en el efecto que sobre ésta tiene la presión selectiva. Varios investigadores han estudiado este problema desde diferentes perspectivas. Grefenstette [12] aplicó un meta algoritmo genético para controlar los parámetros de otro, incluyendo tamaño de población y método de selección. Golberg en [11] hace un análisis teórico del tamaño de población óptimo. En [25] se hace un estudio de la influencia que tienen los parámetros en el proceso de búsqueda. En [27] se propone un algoritmo que ajusta el tamaño de la población con respecto a la probabilidad de error de la selección. Otros métodos propuestos consisten en variar dinámicamente el tamaño de la población [2]. En el problema que se nos plantea, este parámetro está íntimamente relacionado con la complejidad del problema, por lo que su valor depende del conjunto de datos estudiado.

En este sentido, hemos trabajado en una formulación que determina el valor de la población inicial en función del número de nodos de la red, del número de hubs presentes en la misma y del porcentaje de error que se pretende obtener:

$$P_{\text{inicial}} = \begin{cases} np^{\frac{\log 30}{\log n/3} - 1 + u} & \text{para } p \leq n/3 \\ n \left(\frac{n-p+1}{2} \right)^{\frac{\log 30}{\log n/3} - 1 + u} & \text{para } p > n/3 \end{cases} \quad (3)$$

$$u \in (0, 1] \\ p \in (0, n]$$

donde n es el número de nodos.

p es el número de hubs.

u es el umbral de acierto expresado en tanto por uno.

Un valor de u igual a 1 indica que queremos llegar al óptimo; un valor de 0.98, indicará al algoritmo que puede parar cuando haya llegado a un valor cercano al 98% del óptimo. Indudablemente, el algoritmo no puede garantizar esos valores, pero los parámetros han sido elegidos a partir de problemas reales cuyo óptimo era conocido. Cuando se ejecuta el algoritmo, el criterio de parada viene establecido por el número de generaciones que se quieren evaluar, que se introduce como parámetro.

Cuando $p=1$ el problema es trivial, ya que sólo hay un hub al que todos los nodos están conectados. El problema alcanza su máxima complejidad matemática cuando el número de hubs está en el intervalo

$$p \in \left[\frac{n}{2} - 1, \frac{n}{2} + 1 \right]$$

Esto se debe a que el número de Stirling de clase II $S(n, p)^5$, toma sus mayores valores en ese intervalo.

La fórmula anterior se pensó para que el tamaño de población fuese creciendo en función de la complejidad del problema, pero que a la vez, guardara un equilibrio para que no hiciera caer al algoritmo en una rápida convergencia o en un desperdicio de recursos computacionales y de tiempo.

En la literatura no hemos encontrado problemas reales en los que el número de hubs fuese mayor a n

⁵ Sin tener en cuenta las distintas posibilidades de asignación, hay

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}$$

formas diferentes de escoger p hubs sobre un conjunto de n nodos. Pero cuando le añadimos la asignación entonces el número de soluciones es el número de Stirling de clase II

$$S(n, p) = \sum_{k=0}^n \frac{(-1)^k (n-k)^p}{k!(n-k)!}$$

dividido por 3, pero hemos querido dar un valor a la población inicial que permita resolver problemas de este estilo.

Cuando u es igual a 1, i.e. cuando más forzamos al algoritmo, los valores de la población inicial están comprendidos entre n y $30n$, lo que nos da un valor razonable de individuos en función de la complejidad del problema.

El número de individuos de la población inicial, para un número fijo de nodos, crece con el número de hubs, hasta que este llega a ser un tercio del número de nodos, para después ir descendiendo nuevamente.

B. Fase de Inicialización

La inicialización de los valores de los genes sigue una distribución aleatoria uniforme, que toma valores dentro del rango de cada uno de los genes. Este rango es el intervalo $[0, p-1]$ para los nodos no hubs y $[p, 2p-1]$ para los hubs. Esta inicialización aleatoria se lleva a cabo teniendo en cuenta que los individuos de la población inicial sean totalmente diferentes. Es decir los individuos se escogen al azar eliminando las repeticiones.

C. Fase de Cruce

En la fase de cruce se aplica el *Cruce Uniforme*. Una cualidad de este tipo de cruce es que es un operador con alta capacidad de *ruptura de esquemas*, es decir, que el cruce no propaga los esquemas de unos individuos a otros, evitando así que los esquemas más importantes puedan guiar la búsqueda.

Así el *cruce uniforme* se caracteriza porque el valor de cada uno de los genes de los hijos viene determinado por la elección aleatoria de los valores de estos genes en sus padres. Para ello se utiliza una probabilidad P_u que indica la frecuencia con la que el valor de un gen del primer hijo se toma del primer padre. Cuando un hijo recibe el gen de uno de sus padres, el otro hijo recibirá el gen del otro padre.

Hay que decir que el *Cruce Uniforme* implementado ha sido adaptado a las restricciones del problema, no siendo exactamente igual que el original, aunque siguiendo fielmente su funcionamiento básico.

De manera detallada el cruce implementado hace lo siguiente: recorre el cromosoma de izquierda a derecha. Si ambos genes (tanto el de un padre como el del otro en una determinada posición), son nodos *no hubs*, se genera un 0 o un 1 de forma aleatoria siguiendo una distribución uniforme. Si el número generado es 0 se produce una herencia sin cruce, es decir el hijo 1 hereda el gen del padre 1 y el hijo 2 hereda el gen del padre 2. Pero si el número generado es 1 entonces sí se produce cruce. Por lo tanto el hijo 1 hereda del padre 2 y el hijo 2 hereda

del padre 1. Si cualquiera de los genes a cruzar fuera un nodo *hub*, se omite cualquier tipo de cambio dejando ambos genes tal y como están, pasando a la siguiente posición. Como se puede observar cada hijo tiene una probabilidad 0.5 de heredar un determinado gen.

La elección de los individuos a cruzar se realiza utilizando una selección por torneo de tamaño 2. Una vez elegidos los individuos, el operador de cruce se aplica a todos sus genes. En todos los cruces aplicados, los hijos sustituirán a los padres independientemente de que la aptitud de los hijos sea peor que la de los padres.

D. Fase de Mutación

En esta fase, y debido a las restricciones del problema, se ha implementado dos casos particulares de mutación, *Mutación de hub por nodo Asociado* y *Mutación de hub por nodo No Asociado*. Nos referimos por nodo asociado a un hub, a un nodo que está conectado con ese hub.

En estos dos tipos especiales de mutación se genera un cambio entre los hubs, lo que implica una importante variación en la fisonomía de la red.

1) Mutación de hub por nodo asociado

Este tipo de mutación intercambia de forma aleatoria un nodo hub por otro nodo asociado a dicho hub. Esto se traduce en el grafo en un cambio en el conjunto de hubs y en una reasignación de algunos nodos.

2) Mutación de hub por nodo no asociado

Es similar a la anterior, intercambia de forma aleatoria un nodo hub por otro nodo, en este caso, no asociado a dicho hub. La elección de dicho nodo no se hace de forma totalmente aleatoria sino que tiene un factor heurístico. El algoritmo escoge del conjunto de nodos hub aquel que se encuentra más cercano (según la distancia euclídea) a nuestro hub y después escoge de forma aleatoria uno de los nodos asociados al mismo.

Un aspecto importante de ambas mutaciones es que el número de hub mutados en cada realización de las mismas es parametrizable.

E. Fase de Selección

Existen dos factores muy importante que influyen en el proceso de evolución de un algoritmo genético: la diversidad de la población y la presión selectiva. Estos factores están estrechamente relacionados, de forma que un incremento en la presión selectiva

supone un decremento de la diversidad de la población y viceversa. Por tanto, es importante establecer un equilibrio entre estos dos factores. La política de selección es la encargada de fijar este compromiso. Ante la disyuntiva entre la selección por ranking y por torneo, nos hemos decantado, en el cruce, por la *selección por torneo de 2* utilizada en [32] por las siguientes razones:

- La complejidad de la selección por torneo es menor que la de la selección por ranking [4].
- La presión selectiva es mayor, lo que nos va a permitir medir hasta que punto la mutación, y especialmente el cruce es capaz de mantener la diversidad de la población.

En cambio, en la mutación hemos utilizamos *selección aleatoria* por razones derivadas de las características intrínsecas del problema. Para asegurarnos que el mejor individuo siempre pasa a la siguiente generación se utilizará elitismo. Se ha demostrado, tanto teórica, como empíricamente [17] y [32] la conveniencia de esta filosofía. Así pues la dinámica del proceso de selección será: seleccionar el mejor individuo, a continuación, aplicar una selección por torneo de 2, sobre el 100% de la población, para seleccionar los individuos a cruzar, y por último aplicar una selección aleatoria, también sobre el total de la población, para seleccionar los individuos a mutar por cada tipo de mutación según cada caso.

F. Fase de Decisión

Para poder estudiar la velocidad de convergencia, así como la calidad de las soluciones, se han establecido 2 criterios de parada:

- Número máximo de generaciones.
- Alcance de una aptitud tolerable.

En el primero el número máximo de generaciones ha sido establecido en 1000. Evidentemente esta restricción tan fuerte se puede modificar según la envergadura del problema. De todos modos, en generaciones anteriores se suelen encontrar óptimos locales difíciles de superar, por lo que no se consiguen grandes avances, en general, aumentando el número de generaciones.

El segundo criterio es dependiente del factor u que sirve para crear el tamaño de la población inicial, y que en la fase de desarrollo y pruebas del algoritmo, ha servido además, como factor de parada cuando la diferencia entre el óptimo conocido de un problema y el mejor individuo encontrado en una generación era menor del u por uno.

Realmente este umbral no garantiza que nos acerquemos hasta ese nivel del óptimo, puesto que

éste es desconocido, evidentemente, en los futuros problemas en los que se podrá utilizar el algoritmo.

G. Fase de Reemplazo

Normalmente el reemplazo consiste en que la descendencia o hijos derivados en una generación pasarán a ser los padres de la siguiente generación. En nuestro algoritmo, inicialmente, se implantó este tipo de reemplazo, pero a consecuencia de un problema de *convergencia prematura* nos vimos obligados a cambiarlo.

La *Convergencia Prematura* es un problema derivado de la pérdida del equilibrio exploración-explotación y consiste en desviarse hacia zonas del espacio de búsqueda que no contienen el óptimo global. Para evitar este problema se han propuesto diferentes mecanismos que afectan a las distintas características de los AGs. Uno de estos mecanismos consiste en cambiar la población inicial. En [11], Goldberg sugiere *reinicializar* la ejecución de un AG usando una nueva población cuando su nivel de convergencia es sustancial. La reinicialización se hace con individuos generados aleatoriamente junto con los mejores individuos de la población convergida anterior. Tras la convergencia, el mejor elemento encontrado se utiliza como plantilla para reinicializar la población al completo. Para formar cada individuo en la nueva población se muta aleatoriamente una porción del 35% de este elemento.

En nuestro caso hemos adaptado lo anteriormente expuesto a las características de nuestro problema. Cada cierto número de generaciones⁶, nuestro algoritmo reinicializa la población en la fase de reemplazo.

Inicialmente se ordenan los individuos de la población en función de su aptitud. Una vez ordenados se recorre la población mutando los individuos repetidos de manera que al terminar obtengamos una población totalmente heterogénea. La mutación consiste en la permuta de dos alelos del individuo. Se escoge aleatoriamente un alelo. Si el alelo a mutar corresponde a un nodo no hub se intercambia, al azar, por otro nodo no hub y si el alelo corresponde a un nodo hub se permuta por otro nodo no hub, el cual puede estar asociado al mismo o no.

IV. EXPERIMENTOS Y RESULTADOS

Nuestro algoritmo ha sido probado para los problemas (20, 3)⁷, (20, 4), (20, 5), (25, 3), (25, 4),

(25, 5), (50, 3), (50, 4) y (50, 5) y todos los casos se estudiaron con cinco variantes en las probabilidades de cruce y mutación ((0.4, 0.3, 0.3), (0.6, 0.1, 0.3), (0.6, 0.2, 0.2), (0.6, 0.3, 0.1) y (0.8, 0.1, 0.1))⁸. A su vez, sobre cada tipo de experimento se realizaron pruebas con umbrales del 1, del 0.98 y del 0.95 del óptimo.

Los datos utilizados para probar el algoritmo, se pueden encontrar en la dirección

<http://www.ms.ic.ac.uk/jeb/orlib/phubinfo.html>.

Éstos corresponden a datos reales correspondientes al *Servicio Postal Australiano* y constan de una matriz de 200 nodos y 8 hubs. En el mismo fichero podemos encontrar también un pequeño programa en C que nos permite hacer subproblemas, con diferentes tamaños y número de hubs, partiendo del conjunto matriz. También nos proporcionan soluciones óptimas para varias combinaciones de ambos parámetros, en las cuales nos hemos basado para probar la eficiencia de nuestro algoritmo.

La Tabla 1 muestra los resultados obtenidos para problemas de diversa complejidad. En todos los casos se realizaron 30 ensayos. La probabilidad de cruce de los ejemplos es de 0.6 y se realizaron 10 de las pruebas para mutaciones (0.6, 0.1, 0.3), 10 para (0.6, 0.2, 0.2) y 10 para (0.6, 0.3, 0.1). La primera columna de la tabla muestra la complejidad del problema. La segunda el óptimo conocido. La tercera columna nos da el número de aciertos de los 30 ensayos. La cuarta nos indica los tiempos medios en segundos que tarda cada experimento. La quinta y sexta muestran la media y la desviación típica de los experimentos. La séptima el error medio de cada ensayo con respecto al óptimo. La octava el número medio de generaciones que realizó el algoritmo. La última columna, indica en que generación, de media, se alcanzó el mejor valor en cada ensayo.

Las pruebas se realizaron en un ordenador con un procesador Pentium II a 400 mhz, dotado de una memoria RAM de 128 MB.

Para determinar si la acción, conjunta o por separado, de los diferentes factores produce efectos significativos sobre el resultado del algoritmo realizamos análisis de varianza incluyendo como factores la probabilidad de cruce (P_{cruce}), el número de nodos (N_{nodos}) y el número de hubs (N_{hubs}).

El estudio se realizó para un nivel de confianza del 95%, es decir, con un 0.05% de margen de error, por lo tanto. Basándonos en los resultados obtenidos en el análisis se realizó para cada uno de los factores

⁶ Este es un parámetro que se introduce al algoritmo y que en las pruebas realizadas ha variado entre las 10 y las 20 generaciones.

⁷ En cada pareja de números el primero indica números de nodos del problema y el segundo el número de hubs.

⁸ El primer número de la terna indica probabilidad de cruce, el segundo probabilidad de mutación por nodo asociado y el tercero probabilidad por nodo no asociado.

Tabla 1: Resultados con umbral $U=1$ sobre 30 evaluaciones

Problema	Optimo	Aciertos	Tiempos Medios	Aptitud Media	Desv. Típica	Error Medio	Parada Media	Media Mejor
(20,3)	151533.08	13	46 seg	153058.11	1982.85	0.0100	601.47	178.23
(20,4)	135624.88	2	62 seg	139548.43	1960.06	0.0289	948.83	217.87
(20,5)	123130.09	2	67 seg	125709.93	2493.63	0.0209	940.27	178.13
(25,3)	155256.32	10	86 seg	158219.91	2179.82	0.0190	701.13	138.23
(25,4)	139197.17	2	98 seg	141623.58	3399.93	0.0174	993.37	276.97
(25,5)	123574.29	7	110 seg	127653.91	4540.36	0.0330	832.17	220.90
(50,3)	158569.93	6	320 seg	162086.96	2047.33	0.0221	843.00	318.17
(50,4)	143378.05	17	352 seg	146453.54	4396.03	0.0214	580.17	307.57
(50,5)	132366.95	2	367 seg	137937.94	1799.70	0.0420	958.63	424.77

influyentes (número de nodos, número de hubs y probabilidad de cruce), un test de comparaciones múltiples de medias Tamhane para cuantificar esas diferencias y calcular en cada caso cual es el mejor valor medio.

En el test de Tamhane realizado para la probabilidad de cruce, encontramos que existían grandes diferencias significativas tanto entre P_c de 0.6 y P_c de 0.4, como entre $P_c=0.6$ y $P_c=0.8$. Además dicho test era concluyente en señalar que el valor medio más bajo corresponde a $P_c=0.6$. Eso nos hizo considerar a la probabilidad de cruce de 0.6 como la mejor para nuestro algoritmo.

V. CONCLUSIONES DEL ANÁLISIS

Una vez realizado el análisis estadístico sobre los principales factores que intervienen en el AG podemos sacar las siguientes conclusiones:

- Los errores relativos son menores cuando el número de nodos es pequeño y mayores cuando el número de nodos aumenta. Esta parece una conclusión lógica ya que al aumentar el número de nodos estamos aumentando el espacio de búsqueda. Se ha visto que no había demasiadas diferencias entre $N=20$ y $N=25$, pero sí entre $N=20$ y $N=50$, lo cual nos indica que los errores relativos aumentan de forma gradual con respecto al número de nodos.
- Los errores son menores cuanto menor es el número de hubs
- La probabilidad de cruce, según nuestros experimentos, debe estar en torno a un porcentaje cercano al 60%.
- La variación de la probabilidad de mutación, no influye de manera significativa en el resultado del algoritmo, aunque en función de nuestros

experimentos el porcentaje aconsejado debe estar en torno al 20% para cada tipo de mutación, es decir, un porcentaje repartido equitativamente entre la *mutación hub por nodo asociado* y la *mutación hub por nodo no asociado*.

VI. CONCLUSIONES

Según nuestros conocimientos, existen pocos trabajos sobre la aplicación de los algoritmos genéticos al problema P-hub. Con el algoritmo implementado no se puede garantizar encontrar siempre el óptimo. Sin embargo en los experimentos realizados, siempre hemos llegado como mínimo una vez al óptimo en los problemas en los que disponíamos del dato, ver Tabla 1.

Por otra parte la potencia del algoritmo radica en que en pocas generaciones nos acercamos a un valor bastante cercano al óptimo (ver Tabla 1), algo que es muy relevante dada la complejidad de este tipo de problemas.

Con el desarrollo de nuestro algoritmo hemos conseguido:

- Desarrollar una sólida representación genética del problema, que podrá servir de base en futuras investigaciones sobre este tipo de problemas.
- Desarrollar nuevos operadores genéticos de cruce y mutación que cumplen las restricciones del problema.
- Parametrizar el funcionamiento del algoritmo lo que facilita la manipulación de las características del mismo a la hora de realizar diferentes tipos de experimentos.

A partir de los Análisis de Varianza realizados sobre los resultados, hemos afinado los parámetros de nuestro algoritmo, en función de los principales factores que intervienen en el problema.

VII. REFERENCIAS

- [1] Sue Abdinour-Helm. A hybrid heuristic for the uncapacited hub location problem. *European Journal of Operational Research*, 106:489-499, 1998.
- [2] J. Arabas, Z. Michalewicz, and J. Mulawka. Gavaps - a genetic algorithm with varying population size. In Z. Michalewicz, J. Krawczyk, M. Kazemi, and C. Janikow, editors, *First IEEE International Conference on Evolutionary Computation*, volume 1, pages 73-78, Orlando, 27-29 June 1994. IEEE Service Center, Piscataway, NJ.
- [3] Turgut Aykin. The hub location and routing problem. *European Journal of Operational Research*, 83 pp. 200-219, 1995.
- [5] M. L. Brandeau, Samuel S. Chiu. An Overview of Representative Problems in Location Research. *Management Science*. Vol. 35, pp. 645-674, 1989.
- [6] Rajan Batta, Jorge H. Jaramillo, Joy Bhadury. On the use of genetic algorithms to solve location problems. *Computers and Operations Research*, 29:761-779, 2002.
- [7] J. F. Campbell. Integer Programming formulations of Discrete Hub Location Problems. *European Journal of Operational Research* Vol. 72 pp. 387-405, 1994.
- [8] Jamie Ebery. Solving large single allocation p-hub problems with two or three hubs. *European Journal of Operational Research* 128, pp 447-458, 2001.
- [9] A. T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, Vol. 4 No. 3, pp 139-154, 1996.
- [10] J. Ebery, M. Krishnamoorthy, A.T. Ernst and N. Bolland. The capacited multiple allocation hub location problem: Formulations and algorithms. *European Journal of Operational Research* 120, pp 614-631, 2000.
- [11] D. E. Goldberg. Sizing populations for serial and parallel genetic algorithms. In J. Schaffer, editor, *3rd International Conference on Genetic Algorithms*, pages 70-79, San Mateo, CA, 1989. Morgan Kaufmann.
- [12] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Mans, and Cybernetics*, 16(1):122-128, 1986.
- [13] M.J. Kuby and R.G. Gray. The hub network design problem with stopover and feeders: The case of Federal Express. *Transportation Research*. vol 27A, No. 1, pp. 1-12, 1993.
- [14] J.G. Klincewicz. Heuristics for the p-hub location problem. *European Journal of Operational Research*, 53, pp. 25-37, 1991.
- [15] J.G. Klincewicz. A voiding local optima in the p-hub location problem using tabu search and grasp. *Annals of Operations Research* 40, pp 283-302, 1992.
- [16] John G. Klincewicz. A dual algorithm for the uncapacitated hub location problem. *Location Science*, 4(3):173-184, 1996.
- [17] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 1994.
- [18] HubLocator: an exact solution metod for the multiple allocation hub location problem. *Computers & Operations Research* 29, pp. 715-739, 2002.
- [19] M.E. O'Kelly and D.L. Bryan, J. Skorin-Kapov and D. Skorin-kapov. Hub network design with single and multiple allocation: A computational study. *Location Science*, Vol 4, No 3, pp. 125-138, 1996.
- [20] M.E. O'Kelly. The location of interacting hub facilities. *Transportation Science*, 20(2):92-106, 1986.
- [21] M.E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393-404, 1987.
- [22] M. Pérez, F. Almeida and J. M. Moreno-Vega. Genetic algorithm with multistart search for the p-hub median problem. *Proceedings of the Euromicro Conference*, pp. 702-707, 1998.
- [23] S. Pierre and F. Houéto. A tabu search approach for assigning cells switches in cellular mobile networks. *Computer Communications* 25, pp. 464-477, 2002.
- [24] Juan Prawda. Métodos y modelos de investigación de operaciones, vol II. Editorial LIMUSA, 1981.
- [25] J. Schaffer, R. Caruana, L. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In J. Schaffer, editor, *3rd International Conference on Genetic Algorithms*, pages 51-60, San Mateo, CA, 1989. Morgan Kaufmann.
- [26] K Smith, M. Krishnammoorthy and M. alaniswami. Neural Versus Traditional approaches to the location of interacting hub facilities. *Location science*, Vol 4. No. 3, pp. 155-171, 1996.
- [27] R. E. Smith. Adaptively resizing populations: An algorithm and analysis. In S. Forrest, editor, *5th International Conference on Genetic Algorithms*, page 653, San Mateo, CA, 1993. Morgan Kaufmann.
- [28] J. Skorin-Kapov and D. Skorin-kapov. On tabu search for the location of interacting hub facilities. *European Journal of Operational Research*, 73:502-509, 1994.
- [29] Joseph Sarkis, R.P. Sundarraj. Hub location at Digital Equipment Corporation: A comprehensive analysis of qualitative and quantitative factors. *European Journal of Operational Research*, 137, pp. 502-509, 2002.
- [30] D. Skorin-kapov, J. Skorin-Kapov and M. O'Kelly. Tight linear programing relaxations of uncapacited p-hub median problem. *European Journal of Operational Research*, 94, pp. 582-593, 1996.
- [31] S. Ventura, C. Hervas y D. Ortiz. Jclec: Una biblioteca de clases en java. *Primer congreso iberoamericano de algoritmos evolutivos y bioinspirados. AEB'02*, 2002.
- [32] B. T. Zhang and J. J. Kim. Comparison of selection methods for evolutionary optimization. *Evolutionary Optimization*, 2(1):55-70, 2000.