

El problema de asignación de frecuencias generalizado con pesos y su solución

The generalized frequency assignment problem and its solution

David F. Muñoz

Departamento de Ingeniería Industrial y Operaciones, Instituto Tecnológico Autónomo de México, Río Hondo 1, 01080 Ciudad de México, México

E-mail: davidm@itam.mx

Recibido 17 de octubre del 2016; aceptado el 10 de diciembre del 2016

Resumen

En este artículo se reporta el desempeño de 15 métodos heurísticos para encontrar soluciones iniciales y 4 meta-heurísticas para resolver un problema de asignación de frecuencias en el que el valor de las frecuencias asignadas depende de pesos correspondientes a los sitios donde se asigna la frecuencia. Los diferentes algoritmos fueron probados en un conjunto de problemas que se generaron utilizando un generador que representa situaciones similares a la asignación de frecuencias FM en México. Los resultados experimentales mostraron que las heurísticas que consideran los pesos de los sitios tienen un mejor desempeño y, de entre las 4 meta-heurísticas probadas, el mejor desempeño lo obtuvo el algoritmo basado en templado simulado.

Descriptor: *asignación de frecuencias, asignación de canales, problema de T-coloreo, meta-heurísticas*

Abstract

We report performance of 15 heuristics and 4 metaheuristics to solve a frequency assignment problem where the value of an assigned frequency depends on the weights of the corresponding site where the frequency is assigned. Different algorithms were tested on two sets of problems, the first one corresponds to the well-known Philadelphia problems and the second one to situations similar to the assignment of FM frequencies in Mexico. Experimental results showed that the heuristics that take into account the site weights performed the best and, among the four metaheuristics tested, the algorithm based on simulated annealing performed the best.

Keywords: *frequency assignment, channel assignment, T-coloring problem, metaheuristics*

1. Introducción

Desde su introducción a la comunidad de Investigación de Operaciones [1], los problemas de asignación de frecuencias (FAP por Frequency Assignment Problem) han sido ampliamente tratados. Entre 1997 y 2004 tuvo lugar una serie de conferencias auspiciadas por el Center for Discrete Mathematics and Theoretical Computer Science, desarrollándose algoritmos, competencias e instancias de problemas que se han resumido en [2] y en

<http://mat.gsia.cmu.edu/COLOR04/>. Los FAP también han sido ampliamente estudiados en la Comunidad Europea bajo el auspicio de la European Cooperation for Science & Technology, Action 259 (wireless flexible personalised communications), cuyos resultados, junto con una amplia bibliografía se pueden encontrar en <http://fap.zib.de/index.php/>. Por otro lado, existen varios artículos que hacen una revisión de los problemas y métodos de solución relacionados con los FAP [3, 4].

El FAP que es motivo de la presente investigación es una variante del problema conocido en la literatura inglesa como el graph set T-coloring problem (GSTCP), ver [5], el cual es una generalización del graph T-coloring problem (GTCP) en el que se debe asignar un conjunto de canales (en vez de un solo canal) a un conjunto de sitios seleccionados de un conjunto finito de canales en cada sitio. En [6] se presenta un estudio sistemático, tanto de heurísticas para obtener soluciones iniciales del GSTCP como de métodos de búsqueda, cuando el objetivo es el de maximizar el número de canales asignados a los sitios.

En el problema que abordamos en esta investigación, el objetivo es el de maximizar una suma ponderada de los canales asignados a cada sitio, al que llamaremos problema de asignación de frecuencias generalizado con pesos (GWFAP, por sus siglas en inglés), y que es particularmente importante cuando existe un beneficio diferenciado por la asignación de canales en los distintos sitios.

La principal contribución de este artículo es un estudio bastante amplio de esta variante del GSTPC, ampliando los resultados inicialmente presentados en [7], luego de su posterior aplicación para la asignación de canales para televisión digital y estaciones de radio FM por la Secretaría de Comunicaciones y Transporte de México. Los resultados aquí reportados pueden ser de utilidad no sólo para los FAP, sino también para problemas que tienen una formulación similar, como son la programación de horarios y el mantenimiento de flotillas de transporte [8].

Luego de esta introducción, en la siguiente sección se define con mayor precisión el problema que es motivo de esta investigación, así como la notación a utilizar. En la sección 3 se describen las heurísticas que fueron probadas, mientras que en la sección 4 se presentan los algoritmos de búsqueda desarrollados. A continuación en la sección 5 se discuten resultados obtenidos al aplicar las heurísticas y los algoritmos de búsqueda a un conjunto de problemas. Finalmente, en la sección 6 se presentan las conclusiones.

2. Formulación en programación lineal

Con el objetivo de presentar una formulación en programación lineal para el problema de asignación de frecuencias que es de nuestro interés, para cada sitio $v \in V$ denotemos por $F_v \subseteq T$ al conjunto de frecuencias disponibles para el sitio v , es decir, el

conjunto S_v de frecuencias asignadas al sitio v debe cumplir $S_v \subseteq F_v$. Esta condición es necesaria porque en nuestra aplicación, ya existen algunas frecuencias asignadas en determinados sitios, mismas que prohíben la asignación de ciertas frecuencias en cada sitio. Bajo esta notación, las variables de decisión para nuestra formulación en programación lineal son:

$$X_{vf} = \begin{cases} 1, & \text{si la frecuencia } f \text{ se asigna al sitio } v, \\ 0, & \text{de otra forma,} \end{cases} \quad (1)$$

para $v \in V, f \in F_v$, y la función objetivo a maximizar adopta la forma:

$$Z = \sum_{v \in V} w_v \sum_{f \in F_v} X_{vf}, \quad (2)$$

donde w_v es el peso (valor) asignado al sitio $v \in V$. Las restricciones son de la forma

$$X_{vf} + X_{ug} \leq 1, \quad (3)$$

para $v, u \in V$, y $f \in F_v, g \in F_u$, y se establecen para todas las frecuencias que interfieren si se asignan a los sitios correspondientes. En la práctica se establecen utilizando reglas de separación por distancia o modelos de propagación para el cálculo de la magnitud de la interferencia; en ambos casos se considera la potencia y la altura de las correspondientes antenas de transmisión. El modelo más utilizado para este fin es el modelo de Longley-Rice, ver e.g. [9].

3. Heurísticas para obtener soluciones iniciales

Como es muy conocido, el problema de coloreo y sus variantes, como el GWFAP, es un problema NP-completo [10], lo que significa que para un número grande de sitios, los métodos exactos de solución (e.g., ramificación y acotamiento) podrían no ser capaces de proporcionar una solución del problema en un tiempo razonable de cómputo. Es por esta razón que la construcción de soluciones iniciales, utilizando procedimientos rápidos (llamados heurísticas), puede ser de utilidad para disponer de una solución al problema, siempre y cuando no esté muy alejada del óptimo. Por otro lado, una buena solución inicial puede ser también utilizada como punto de partida de algún algoritmo de búsqueda (meta-heurística), como los que se discuten en la sección 4.

```

1. Hacer  $j \leftarrow 1$  e ir al paso 2.
2. Hacer  $v \leftarrow \pi_j$  e ir al paso 3.
3. While  $|F_v| > 0$ 
    • Asignar la menor frecuencia  $f_v \in F_v$  al sitio  $v$ .
    • Para cada sitio  $u \in V$ , excluir de  $F_u$  todas las
      frecuencias  $f_u \in F_u$  que interfieren con la
      frecuencia  $f_v$  en el sitio  $v$ .
End while
4. Hacer  $j \leftarrow j + 1$ , si  $j > p$  detenerse, de otra forma ir al
paso 2.

```

Figura 1: Pseudo-código del algoritmo ST-GREEDY.

Las heurísticas que consideramos en esta investigación pueden agruparse en 3 categorías a las que identificaremos por los nombres: ST-GREEDY, G-DSATUR y G-RLF [6], cuyos significados explicamos a continuación.

Las heurísticas correspondientes a la categoría ST-GREEDY parten de una permutación $\pi = (\pi_1, \dots, \pi_p)$ de los sitios (donde p es el número de sitios), y en cada paso se asignan canales en forma voraz, de acuerdo con el algoritmo que se presenta en la figura 1. Al inicio del algoritmo de la figura 1, cada conjunto F_v se inicializa con el conjunto de canales factibles para el sitio, y cada vez que se asigna una frecuencia a un sitio, estos conjuntos deben actualizarse excluyendo todas las frecuencias que interfieren con la frecuencia asignada en el sitio correspondiente.

Se han probado 6 heurísticas correspondientes a la categoría ST-GREEDY, cada una correspondiente a un tipo de permutación de los sitios, como se indica a continuación:

RO: La permutación se genera en forma aleatoria, asignando en cada paso la misma probabilidad a cada sitio no seleccionado.

LWF: La permutación corresponde al ordenamiento de los sitios de mayor a menor por peso, rompiendo empates de acuerdo con el mayor grado ajustado del sitio.

LDF: La permutación corresponde al ordenamiento de los sitios de mayor a menor por grado del sitio.

LADF: La permutación corresponde al ordenamiento de los sitios de mayor a menor por grado ajustado del sitio.

SDL: La permutación corresponde al ordenamiento menor al final por grado del sitio.

SADL: La permutación corresponde al ordenamiento menor al final por grado ajustado del sitio.

Para calcular el grado de un sitio, decimos que existe un arco entre dos sitios $v, u \in V$ (se permite $v = u$) si existen dos frecuencias $f_v \in F_v, f_u \in F_u$ tales que la asignación simultánea de la frecuencia f_v al sitio v y f_u al sitio u está prohibida, de esta manera, el grado de un sitio es el número total de arcos que salen del sitio. Bajo la misma notación, decimos que la diferencia de frecuencias del arco es $|f_v - f_u|$, y el grado ajustado de un sitio es la suma de las diferencias de frecuencia de los arcos que salen del sitio. Por otro lado, el “ordenamiento menor al final” que se utiliza para obtener las permutaciones SLG y SLAG se obtiene al secuenciar, empezando por el final, al sitio que tenga el menor grado (o grado ajustado) sin considerar a los arcos de los sitios ya secuenciados. Las heurísticas correspondientes a la categoría G-DSATUR se basan en una generalización de la heurística DSATUR, propuesta por Costa [11] para el GTCP. Bajo estas heurísticas se asignan frecuencias en forma voraz, de acuerdo con un ordenamiento de los sitios, como en el algoritmo ST-GREEDY de la figura 1, pero el orden de los sitios se determina de acuerdo con el grado de saturación del sitio (GSS), el cual es el número de frecuencias que fueron prohibidas por las frecuencias de los sitios que se secuenciaron previamente. Se han probado las siguientes 6 heurísticas correspondientes a esta categoría:

LSWF: El orden de los sitios se obtiene de acuerdo con el criterio de mayor GSS primero y se rompen empates por peso del sitio.

SSWF: El orden de los sitios se obtiene de acuerdo con el criterio de menor GSS primero y se rompen empates por peso del sitio.

LSDF: El orden de los sitios se obtiene de acuerdo con el criterio de mayor GSS primero y se rompen empates por grado del sitio.

SSDF: El orden de los sitios se obtiene de acuerdo con el criterio de menor GSS primero y se rompen empates por grado del sitio.

LSADF: El orden de los sitios se obtiene de acuerdo con el criterio de mayor GSS primero y se rompen empates por grado ajustado del sitio.

SSADF: El orden de los sitios se obtiene de acuerdo con el criterio de menor GSS primero y se rompen empates por grado ajustado del sitio.

Las heurísticas correspondientes a la categoría G-RLF ya no utilizan la asignación voraz por sitio, sino que, empezando por la frecuencia más pequeña, se asigna la misma frecuencia a todos los sitios que así lo permitan, considerando que cada vez que se asigna una frecuencia, se prohíben las frecuencias que causan interferencia en cada uno de los sitios. Se han probado las siguientes 3 heurísticas en esta categoría:

RLWF: El orden de asignación de la misma frecuencia en los sitios es por peso del sitio.

RLDF: El orden de asignación de la misma frecuencia en los sitios es por grado del sitio.

RLADF: El orden de asignación de la misma frecuencia en los sitios es por grado ajustado del sitio.

4. Algoritmos de búsqueda

Los algoritmos de búsqueda más explorados para el GSTCP, y para el GTCP en general, corresponden a implementaciones de 3 métodos generales de búsqueda: templado simulado (SA por Simulated Annealing), búsqueda tabú (TS por Tabu Search) y algoritmo genético (GA por Genetic Algorithm), y es por esta razón que en esta sección presentaremos algoritmos para resolver el GWFAP basados en cada uno de estos 3 principios. Además de estos algoritmos, presentaremos también otro algoritmo basado en otro método de búsqueda denominado entropía cruzada (CE por Cross Entropy).

4.1. Vecindad de una solución

La implementación de algoritmos basados en SA o en TS consideran que cada solución de un problema tiene un conjunto de soluciones llamadas vecinas, y la búsqueda de soluciones óptimas o cercanas a la óptima, ocurre explorando soluciones vecinas, a partir de una solución inicial. Es conveniente mencionar que, aunque algunas implementaciones de algoritmos de búsqueda consideran movimientos a soluciones no factibles, es decir, que violan alguna restricción por interferencia [12], todos los algoritmos

considerados en esta investigación, consideran sólo soluciones factibles del GWFAP, por lo que el concepto de vecindad que hemos utilizado para las implementaciones de los algoritmos SA y TS, considera sólo vecinos factibles.

Para definir la vecindad de una solución utilizaremos la notación introducida en la sección 3 para definir el GSTCP, denotando al conjunto de sitios por $V = \{v_1, \dots, v_p\}$, cualquier solución del GWFAP puede verse como una colección $x = \{S_1^x, \dots, S_p^x\}$, donde $S_i^x \subseteq F_{v_i}$ es el conjunto de frecuencias asignadas al sitio $v_i, i = 1, 2, \dots, p$. Decimos que otra solución $y = \{S_1^y, \dots, S_p^y\}$ es una solución vecina de x , si ésta puede obtenerse utilizando el siguiente procedimiento:

1. Seleccionar un sitio $v_i \in V$, y una frecuencia no asignada $f \in F_{v_i} - S_i^x$.
2. Para cada sitio $v_j \in V$, hacer $S_j^y = S_j^x - U_j^y(f)$, donde $U_j^y(f) \subseteq S_j^x$ es el conjunto de frecuencias que, al ser usadas en el sitio v_j , causan interferencia con la frecuencia f en el sitio v_i .

Se entenderá por vecindad de una solución x , al conjunto de todas las soluciones que son vecinas de x .

Nótese que el paso 2 del procedimiento que acabamos de mencionar es, en esencia, el mismo que está dentro del While de la figura 1, por lo que, para generar los vecinos de una solución, se puede re-usar parte del código del algoritmo ST-GREEDY.

4.2. Algoritmo de templado simulado

El procedimiento de búsqueda SA es un procedimiento de búsqueda estocástica que se inspira en la dinámica de ciertos fenómenos termodinámicos como el congelamiento y cristalización de los líquidos, o el calentamiento y enfriamiento lento de ciertos metales para lograr una estructura estable (de menor energía). Es por esta razón que el procedimiento SA utiliza una variable llamada temperatura (variable Tem en la figura 2), cuyo valor decrece durante la ejecución del algoritmo, esta función es la analogía de la energía en el fenómeno termodinámico.

```

Hacer  $Tem = T_0$  y generar la solución inicial  $x_0$ 
Calcular  $Z(x_0)$  e inicializar  $N_c$ 
While  $Tem > T_{min}$ 
  For  $i = 1$  to  $N_c$ 
    Generar una solución vecina  $x_1$  y calcular  $Z(x_1)$ 
    Generar  $r$  uniforme en  $(0,1)$ 
    If  $r < P[Z(x_0), Z(x_1)]$ 
       $x_0 \leftarrow x_1, Z(x_0) \leftarrow Z(x_1)$ 
    End if
  End for
  Hacer  $Tem \leftarrow \alpha Tem$ 
End While

```

Figura 2: Pseudo-código del algoritmo SA.

En la figura 2 se presenta el pseudo-código del procedimiento SA (para el caso de maximización) que hemos utilizado en las dos implementaciones que hemos probado para resolver el WGFAP. Como podemos apreciar de la figura, partiendo de una temperatura inicial T_0 y de una solución inicial x_0 , se generan ciclos en los que se exploran N_c soluciones vecinas (cada una a partir de la anterior).

Cada solución vecina generada puede ser aceptada, en cuyo caso actualiza a la solución anterior, o rechazada. De acuerdo con la figura 2, una solución vecina es aceptada si $P[Z(x_0), Z(x_1)] > r$, donde r se distribuye uniformemente en $(0,1)$, y:

$$P[Z(x_0), Z(x_1)] = \begin{cases} e^{-(Z(x_0) - Z(x_1))/BT}, & Z(x_1) \leq Z(x_0), \\ 0, & \text{de otra forma,} \end{cases} \quad (4)$$

Esta condición permite que se acepten soluciones con menor valor de la función objetivo, para evitar la absorción por un óptimo local, y la aceptación de valores menores tiene menor probabilidad a medida que baja la temperatura.

La constante B en (4) recibe el nombre de constante de Boltzmann, es una constante de escala, y debe ajustarse empíricamente. Nótese que para aplicar el algoritmo de la figura 2 se requiere de la especificación de otros parámetros, como son la temperatura inicial T_0 , la temperatura final T_{min} , el número de iteraciones N_c del ciclo principal, y α (número entre 0 y 1 que determina la velocidad de enfriamiento). Los detalles sobre estos parámetros

pueden consultarse en [7].

4.3. Algoritmo de búsqueda tabú

La definición de vecindad introducida en la sección anterior, aunque en teoría permite alcanzar una solución óptima a través de movimientos entre vecinos, tiene el inconveniente de que, al prohibirse las frecuencias que interfieren con la frecuencia entrante, podría haber espacio para incorporar una frecuencia adicional que no cause interferencia con las otras frecuencias asignadas en la solución. Debido a esta razón se decidió diseñar un procedimiento que trate de mejorar una solución factible dada, tratando de incorporar frecuencias en forma voraz. A continuación definimos dicho procedimiento, al cual llamaremos mutación.

El procedimiento de mutación parte de una solución $x = \{S_1^x, \dots, S_p^x\}$, donde $S_i^x \subseteq F_{V_i}$, y de una permutación $\pi = (\pi_1, \dots, \pi_p)$ de los sitios, y a continuación:

1. Se exploran todos los sitios v_i (en el orden indicado por π), y se recorren las soluciones vecinas de x (en el orden de las frecuencias más pequeñas de $F_{V_i} - S_i^x$), para determinar una solución óptima $y = \{S_1^y, \dots, S_p^y\}$, y el correspondiente valor de la función objetivo $Z(y)$.
2. Si $Z(y) \leq Z(x)$ el procedimiento termina, y se devuelve a x como mutación de x , de otra forma se hace $x \leftarrow y$ y se regresa al paso 1.

Se dice que una solución x del WGFAP es l -optimal si ningún vecino tiene $Z(y) > Z(x)$, por lo que el procedimiento anterior regresa una mutación de x que es l -optimal.

Los algoritmos de búsqueda basados en el principio TS [13] almacenan soluciones exploradas (llamadas memorias de corto y largo plazo), con el objetivo de no repetir la búsqueda alrededor de soluciones que ya fueron exploradas. Como se describe en la figura 3, en cada ciclo del algoritmo, se exploran todas las soluciones vecinas de una solución x_0 y a cada una se le aplica el procedimiento de mutación descrito en la sección anterior. De estas soluciones, se almacenan la mejor solución tabú y la mejor solución que no es tabú. Si la solución tabú es mejor, se puede permitir que ésta sea la solución a explorar en el próximo ciclo, si alcanza el llamado criterio de

aspiración (que a continuación explicamos). De otra forma, la próxima solución a explorar será la solución que no es tabú.

```

Inicializar memorias de corto y largo plazo
Generar solución inicial  $x_0$ 
While not finished
  Inicializar  $Z_B = -\infty, Z_T = -\infty$ 
  For  $i=1$  to  $P$  and each  $f \in F_{v_i} - S_{v_i}^{x_0}$ 
    Generar la solución vecina  $x_1$  con  $f \in S_{v_i}^{x_1}$  y aplicar mutación a  $x_1$ 
    If  $x_1$  es tabú
      If  $Z(x_1) > Z_T$  then  $x_T \leftarrow x_1, Z_T \leftarrow Z(x_1)$ 
    Else
      If  $Z(x_1) > Z_B$  then  $x_B \leftarrow x_1, Z_B \leftarrow Z(x_1)$ 
    End if
  End for
  If  $Z_T > Z_B$  and se cumple criterio de aspiración
     $x_0 \leftarrow x_T, Z(x_0) \leftarrow Z(x_T)$ 
  Else
     $x_0 \leftarrow x_B, Z(x_0) \leftarrow Z(x_B)$ 
  End if
  Actualizar memorias de corto y largo plazo
End While
  
```

Figura 3: Pseudo-código del algoritmo TS.

Para explicar el mecanismo que determina si una solución es tabú, denotemos por T_v^k al conjunto de las frecuencias que se asignaron al sitio v al inicio de la k -ésima repetición del ciclo principal; si en esta repetición la solución vecina x_1 (ver figura 3) trata de asignar una frecuencia $f \in F_{v_i} - S_{v_i}^{x_0}$, tal que $f \in T_{v_i}^j$ para algún $j \in \{k-1, \dots, k-M-1\}$, se dice que la solución x_1 es tabú debido a la memoria de corto plazo (el parámetro M es la longitud de la memoria de corto plazo). El movimiento se considera también tabú (debido a la memoria de largo plazo) si $\sum_{j=1}^k I(f \in T_{v_i}^j) > k\beta$, donde β es otro parámetro, y $I(f \in T_{v_i}^j)$ es 1 si $f \in T_{v_i}^j$, y 0 de otra forma.

El criterio de aspiración de una solución tabú es que el valor de su función objetivo sea mayor que el de las soluciones de los últimos M ciclos, donde M es la longitud de la memoria de corto plazo. Es conveniente mencionar que el algoritmo TS, que probaremos en la sección 5, termina cuando se excede un determinado tiempo de corrida.

4.4. Algoritmo genético

Los procedimientos de búsqueda basados en GA [14] tratan de imitar los procesos naturales de

evolución y selección natural, por lo que (como se ilustra en la figura 4), parten de una población inicial de soluciones $\{x_1, \dots, x_P\}$. En cada ciclo principal del algoritmo, se genera una nueva generación. Bajo el GA que hemos implementado para el WGFAP, cada elemento de la población anterior (x_i) genera un descendiente (x) al combinarse con otro elemento (x_k) elegido al azar, utilizando un procedimiento al que llamaremos cruzamiento (del inglés crossover). Luego de aplicar mutación al descendiente, éste puede reemplazar al padre con menor valor de la función objetivo, si su valor no es peor que el de ambos, y de esta manera, algunos elementos de la siguiente generación pueden provenir de la anterior, o ser descendientes. En nuestra implementación, la terminación ocurre cuando se excede un tiempo de corrida determinado.

```

Generar población inicial  $\{x_1, \dots, x_Q\}$ 
While not finished
  For  $i=1$  to  $Q$ 
    Generar descendiente  $x$  a partir de padres  $x_i$  y  $x_k$ 
    Aplicar mutación a  $x$ 
    If  $Z(x) \geq Z(x_i) = \min\{Z(x_i), Z(x_k)\}$  then  $x_i \leftarrow x$ 
  End for
End While
  
```

Figura 4: Pseudo-código del algoritmo GA.

El procedimiento de cruzamiento de dos soluciones $z = \{S_1^z, \dots, S_p^z\}$, $y = \{S_1^y, \dots, S_p^y\}$ que hemos implementado, asume que el conjunto de frecuencias factibles para el descendiente en el sitio v_i es $F_{v_i} = S_i^z \cup S_i^y$, y determina las frecuencias asignadas aplicando el algoritmo ST-GREEDY de la figura 1, con base en una permutación aleatoria de los sitios. Para generar la población inicial también se ha aplicado el algoritmo ST-GREEDY, partiendo de la permutación por mayor peso de los sitios $\pi^1 = (\pi_1^1, \pi_2^1, \dots, \pi_p^1)$ para el primer elemento de la población, se generaron $Q = p$ permutaciones “dando vuelta a la ruleta”, es decir,

$$\pi^2 = (\pi_p^1, \pi_1^1, \dots, \pi_{p-1}^1), \dots, \pi^p = (\pi_2^1, \pi_3^1, \dots, \pi_p^1, \pi_1^1).$$

4.5. Algoritmo de entropía cruzada

Los algoritmos basados en CE fueron propuestos por Rubinstein y Kroese [15], y son algoritmos de búsqueda estocástica que alteran las probabilidades de selección en el espacio de búsqueda, utilizando el concepto de CE para asignar mayor probabilidad a los elementos con mejor valor para la función objetivo.

```

Inicializar distribución  $q$  y hacer  $i \leftarrow 0, done = false$ 
While not done
  Hacer  $i \leftarrow i+1$  y generar muestra  $X_1, \dots, X_m$  usando  $q$ 
  Ordenar los valores  $Z(X_{j_1}) \leq \dots \leq Z(X_{j_m})$ , hacer  $B_i \leftarrow Z(X_{j_m})$ 
  y usar los valores  $X_{j_k}, \dots \leq X_{j_m}, k = \lfloor \rho_0 m \rfloor$ 
  para actualizar  $q$ .
  Si para  $i \geq d, B_{i-d} = B_{i-d+1} = \dots = B_i$  hacer  $done = true$ 
End While
    
```

Figura 5: Pseudo-código del algoritmo CE.

En nuestro algoritmo CE usamos el hecho de que la solución se puede obtener aplicando ST-GREEDY a la “permutación correcta de sitios” [16], por lo que se propone muestrear el sitio i de la distribución q_i , donde la distribución inicial se obtuvo asignando probabilidades proporcionales a los pesos. La distribución $q = (q_1, \dots, q_p)$ se actualiza en cada iteración usando la distribución empírica de las mejores soluciones, considerando la fracción ρ_0 , donde ρ_0 es menor (aunque cercano) a 1.

5. Resultados experimentales

Para correr nuestros experimentos se desarrolló un generador de problemas que toma en cuenta las principales características de la asignación de frecuencias en México. Dado un entero positivo n , se generan n^2 sitios con coordenadas (LAT_i, LON_j) , donde $LAT_i = LAT_1 + (i-1)ALAT, LON_j = LON_1 + (j-1)ALON$, para $i, j = 1, \dots, n$. Posteriormente, se determina aleatoriamente el tipo de antena (de entre 6 diferentes tipos) y el peso que posee cada sitio, así como las frecuencias que ya están asignadas, de entre 100 posibles canales FM. A continuación se calculan las distancias entre los sitios y, considerando el tipo de antena, las frecuencias asignadas y las tablas de distancias mínimas de separación que se aplican en México, se determina cuáles de las 100 frecuencias siguen siendo factibles de asignarse a cada uno de los sitios.

Tabla 1: Promedio de frecuencias factibles y tiempos de corrida (seg.) de CPLEX variando n .

n	4	9	16	25	36	49	64
Sitios	4	9	16	25	36	49	64
Factibles	48	45	40	40	39	38	38
Tiempo	0.1	0.3	1.2	5.1	81.9	532	5213

El primer experimento consistió en correr 10 repeticiones para cada valor de n , tratando de resolver el problema utilizando el solver CPLEX y registrando los tiempos de corrida. Los resultados hasta $n = 8$ se muestran en la tabla 1. Para $n = 9$ CPLEX ya no fue capaz de proporcionar una solución, nuestra estación de trabajo reportó demasiada memoria requerida por CPLEX, a la vez que el proceso se tornó excesivamente lento.

Posteriormente se corrieron 10 repeticiones con cada una de las 15 heurísticas propuestas y el tamaño más grande ($n = 8$) de problemas para los que CPLEX proporcionó la solución exacta. En la figura 6 se reporta el porcentaje promedio de la solución óptima que se obtuvo con cada heurística, donde se puede apreciar que el mejor desempeño lo obtuvo la heurística RLWF (86%), seguida de cerca por LWF (82%).

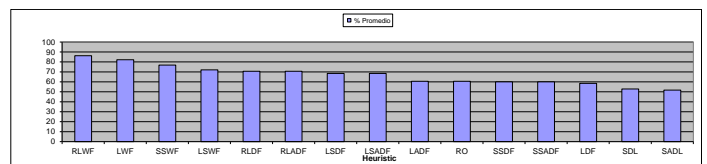


Figura 6: Porcentajes promedio de la solución óptima para 15 heurísticas.

Debido a que SA y CE tardaban entre 420 y 750 segundos en proporcionar una solución para problemas de tamaño $n = 8$, finalmente se reporta, en la figura 7, el porcentaje promedio de la solución óptima para 10 repeticiones del problema de tamaño $n = 8$ corriendo un máximo de 750 segundos, donde se puede apreciar que el mejor desempeño lo obtuvo la meta-heurística SA (99.2%), seguida de CE (95.7%).

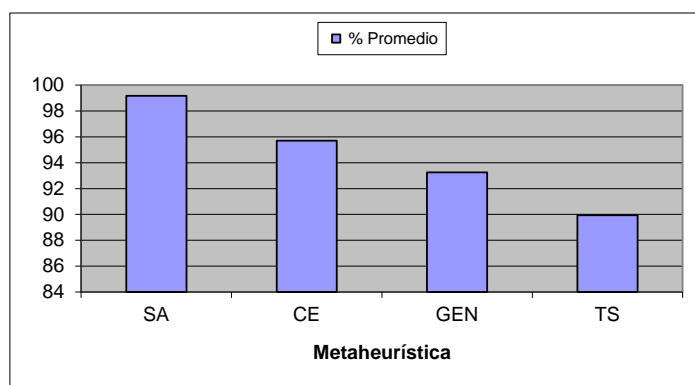


Figura 7: Porcentajes promedio de la solución óptima para 4 meta-heurísticas.

5. Conclusiones

Los resultados obtenidos al probar 15 métodos heurísticos para soluciones iniciales del GWFAP muestran que es conveniente considerar los pesos de las frecuencias, habiéndose desempeñado mejor las heurísticas LWF, RLWF y LSWF.

De los 4 métodos de búsqueda probados en esta investigación, el método SA es el que obtuvo el mejor desempeño promedio, como se reporta en la literatura para otros FAP [12]. El algoritmo CE tuvo un desempeño razonable, aunque se pudo observar que la necesidad de producir una muestra aleatoria grande en cada ciclo del algoritmo hace que éste consuma un tiempo significativo de cómputo.

Agradecimientos

El autor expresa su gratitud a la Asociación Mexicana de Cultura A. C. y al Instituto Federal de Telecomunicaciones de México por el apoyo recibido para realizar esta investigación.

Referencias

- [1] B.H. Metzger. *Spectrum management technique*. Presentación en el 38th National ORSA Meeting, Detroit, MI, 1970.
- [2] D. S. Johnson, A. Mehrotra y M. S. Trick (eds.). Special Issue on Computational Methods for Graph Coloring and its generalizations. *Discrete Applied Mathematics* **156** (2008).
- [3] K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino y A. Sassano. Models and solution techniques for frequency assignment problems. *Annals of Operations Research* **153** (2007) 79-129.
- [4] A. Eisenblätter, M. Grötschel y A. M. C. A. Koster. Frequency assignments and ramifications of coloring. *Discussiones Mathematicae Graph Theory* **22** (2002) 51-88.
- [5] W. K. Hale. Frequency assignment: theory and applications. *Proceedings of the IEEE* **68** (1980) 1497-1514.
- [6] M. Chiarandini M. y T. Stützle. Stochastic local search algorithms for graph set T-colouring and frequency assignment. *Constraints* **12** (2007) 371-403.
- [7] D. F. Muñoz y D. F. Muñoz. Algorithms for the generalized weighted frequency assignment problem. *Computers and Operations Research* **39** (2012), 3256-3266.
- [8] B. A. Tesman. Set T-colorings. *Congressus Numerantium* **77** (1990) 229-242.
- [9] J. S. Seybold. *Introduction to RF propagation* (John Wiley, NY, 2005).
- [10] M. R. Garey y D. R. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).
- [11] D. Costa. On the use of some known methods for T-colorings of graphs. *Annals of Operations Research* **41** (1993) 343-358.
- [12] S. Hurley y D. Smith. Meta-heuristics and channel assignment. En R. Lesse y S. Hurley (eds.), *Methods and Algorithms for Radio Channel Assignment*, Oxford University Press (2002), 22-44.
- [13] F. Glover, E. Taillard y D. de Werra. A user's guide to tabu search. *Annals of Operations Research* **41** (1993) 3-28.
- [14] D. Goldberg. *Genetic algorithms in search, optimization and machine learning* (Addison Wesley, NY, 1989).
- [15] R. Y. Rubinstein y D. P. Kroese (2004). *The Cross-Entropy Method. A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning* (Springer, NY, 2004).
- [16] C. L. Valenzuela. A study of permutation operators for minimum span frequency assignment using an order based representation. *Journal of Heuristics* **7** (2001) 5-21.