

A NOVEL PARALLEL EVOLUTIONARY APPROACH FOR THE FACILITY LAYOUT PROBLEM

Palomo-Romero, Juan M.; Salas-Morera, Lorenzo; García-Hernández, Laura
Universidad de Córdoba

The placement of facilities in a plant is known to have a significant impact on work in process, lead times, productivity and manufacturing costs. Where to locate facilities and the efficient design of those facilities are important and fundamental strategic issues in any manufacturing industry. A novel evolutionary approach based on island model to solve the unequal facility layout problems (UA-FLP) using a flexible bay structure representation is presented in this paper. This one consists on a genetic algorithm based on a multiple-population coarse-grained model, whose aim is to maintain the diversity of the population and get a better sampling of the search space, providing a better quality of solutions. Our proposal has been tested with a collection of well-known problems. Moreover, the results obtained have been compared with the best proposals published in the literature. Our approach have obtained, in many cases, better solutions than previous ones.

Keywords: facility layout problem; evolutionary algorithms; genetic algorithms; parallel model

UN NUEVO ENFOQUE EVOLUTIVO PARALELO EN EL PROBLEMA DE DISTRIBUCIÓN EN PLANTA

La distribución de las instalaciones de una planta tiene un impacto significativo en los costes de fabricación, productos en proceso, los tiempos y la productividad. Dónde ubicar las instalaciones y el diseño eficiente de las instalaciones son cuestiones estratégicas importantes y fundamentales en cualquier industria manufacturera. En este artículo se presenta un nuevo enfoque evolutivo basado en un modelo de islas para la resolución del problema de distribución en planta de áreas desiguales (UA-FLP) usando como representación la estructura de bahías flexibles (FBS). Este enfoque consiste en un algoritmo genético basado en un modelo de grano grueso de múltiples poblaciones, cuyo objetivo es mantener la diversidad de la población y obtener un mejor muestreo del espacio de búsqueda, ofreciendo una mejor calidad de las soluciones. Nuestra propuesta ha sido probada con una serie de problemas conocidos. Además, los resultados obtenidos han sido comparados con las mejores propuestas publicadas en la literatura, obteniendo nuestra nueva propuesta, en un gran número de casos, mejores soluciones que las anteriores.

Palabras clave: distribución en planta; algoritmos evolutivos; algoritmos genéticos; modelo paralelo

1. Introducción

El problema de la colocación de instalaciones en una planta, que se conoce como el problema de distribución en planta, o *facility layout problem* (FLP), es bien conocido que tiene un impacto significativo en los costes de fabricación, la productividad y los plazos de entrega (Drira, Pierreval & Hajri-Gabouj, 2007). Dónde ubicar las instalaciones y el diseño eficiente de las instalaciones son cuestiones estratégicas importantes y fundamentales para la producción de bienes o prestación de servicios (Singh & Sharma, 2006). Una instalación es una entidad que facilita la realización de cualquier trabajo. Puede ser una máquina, un centro de trabajo, una celda de fabricación, un departamento, un almacén, etc (Heragu, 2008). En la mayoría de los casos, el objetivo principal de los problemas de distribución en planta es reducir al mínimo los costos de manejo de materiales.

En cualquier industria manufacturera, la ubicación de las instalaciones y un eficiente diseño de la planta son aspectos a tener muy en cuenta (Singh & Sharma, 2006). Un excelente diseño de la distribución de las instalaciones contribuye a la eficiencia global de las operaciones, y puede reducir los costes operativos totales en una horquilla del 20% al 50% (Tompkins et al., 2010).

Desde la década de 1990, la investigación en este campo se ha centrado, principalmente, en los problemas de distribución en planta de área desigual (UA-FLP) (Scholz, Petrick & Domschke, 2009), el cual es considerado uno de los más importantes problemas de diseño de plantas industriales (Meller & Gau, 1996). En particular, el trabajo propuesto en este artículo se centra en el problema tal y como se describe por Armour y Buffa (1963), quienes lo formularon como un diseño de planta rectangular que está compuesto por instalaciones rectangulares desiguales que deben ser dispuestas de una manera eficaz.

Varias técnicas se han aplicado en la resolución de los FLP, tales como el método ramificación y poda (*branch and bound*) (Solimanpur & Jafari, 2008), programación entera mixta (*mixed integer programming*) (Bozer & Wang, 2012) y la teoría de grafos (Kim & Kim, 1995). Recientemente, muchos investigadores se han centrado en los métodos meta-heurísticos, tales como el enfriamiento simulado (*simulated annealing*) (Sahin, 2011), la búsqueda tabú (Scholz, Petrick & Domschke, 2009), el sistema de hormigas (Ulutas & Kulturel-Konak, 2012) y los algoritmos genéticos (AG) (Ou-Yang & Utamina, 2013).

Los AG (Holland, 1992) son métodos computacionales que se basan en los principios de la selección natural (Darwin, 1859). De acuerdo con Darwin, los individuos en una población compiten entre sí para obtener recursos naturales y encontrar una pareja para procrear. Los individuos más exitosos de la población tendrán una mayor probabilidad de reproducirse y crear más individuos de una nueva generación que los otros individuos más débiles.

Los AG son algoritmos de búsqueda globales que se han utilizado comúnmente para resolver problemas de optimización y búsqueda. Además, estos algoritmos computacionales se han utilizado en diversos campos de estudio, tales como los sistemas de control en ingeniería, la robótica, el reconocimiento de patrones, el reconocimiento de voz, la industria del agua, y, naturalmente, en los problemas de distribución de las instalaciones.

Muchos estudios han abordado el problema UA-FLP mediante el uso de AG. En primer lugar, Tate y Smith (1995) implementan un algoritmo genético simple para la resolución del UA-FLP. Este enfoque considera una función adaptativa de penalización para encontrar buenas soluciones factibles. Esta función penaliza a los diseños de plantas que no cumplen con las restricciones correspondientes. Además, en este trabajo se proporcionaron los

resultados de varios problemas de prueba que demostraron la solidez de su algoritmo. Kochhar, Foster y Heragu (1998) proponen un algoritmo basado en AG para resolver el FLP de planta única, teniendo en cuenta los departamentos de tamaños iguales y desiguales. Evaluaron su algoritmo utilizando varios problemas de prueba y concluyeron que el AG puede ofrecer una mejor alternativa cuando el objetivo es encontrar una serie de diseños "razonablemente buenos". Wang, Hu y Ku (2005) presentaron un estudio que combina un AG con análisis de varianza (ANOVA). Su estudio pone de manifiesto que los resultados experimentales muestran que el enfoque es más viable para hacer frente a los problemas de diseño de instalaciones en el mundo real. El enfoque de Konak et al. (2006) consta de una formulación de programación entera mixta para encontrar soluciones óptimas para el UA-FLP. Los autores han probado su algoritmo con un conjunto de problemas de la literatura y, para algunos de ellos, obtuvieron mejores resultados que los publicados hasta la fecha. Komarudin y Wong (2010) presentan un algoritmo de optimización de colonia de hormigas usando un método basado en árbol (en concreto, *slicing tree*) para representar fácilmente los problemas sin restringir excesivamente el espacio de soluciones. Más tarde, Wong y Komarudin (2010) publicaron un enfoque de sistemas de hormigas a la solución del UA-FLP por medio de una estructura de bahía flexible (FBS). En el mismo documento, diseñaron una mejora a la representación de FBS, orientada a la resolución de problemas con los espacios vacíos. El algoritmo se detiene cuando se alcanza uno de los siguientes criterios: (a) un número máximo de iteraciones; o (b) un número máximo de iteraciones sin mejorar la mejor solución. En la misma dirección, un algoritmo de optimización basado en colonia de hormigas fue propuesto por Kulturel-Konak y Konak (2011). En este trabajo se propone la optimización de colonia de hormigas y lo probó en 21 problemas del tipo UA-FLP bien conocidos de la literatura, desde problemas con pocos departamentos hasta grandes problemas de hasta 62 departamentos. Los resultados obtenidos se compararon con las soluciones previamente más conocidos. Este algoritmo encuentra las mejores soluciones conocidas previamente en un muy corto periodo de tiempo de CPU. Recientemente, el trabajo de Kulturel-Konak (2012) propuso un enfoque de búsqueda tabú probabilístico que utiliza la programación lineal para determinar las características de los departamentos y sus ubicaciones dentro de las bahías. El enfoque de programación lineal propuesto se utiliza para resolver 13 casos de FLP de diferentes tamaños de la literatura. Esta propuesta obtuvo los mismos mejores resultados hasta la fecha para muchos problemas, y proporcionó mejores resultados para algunos de ellos. En la obra de Aiello, Scalia y Enea (2013), un algoritmo genético multiobjetivo (MOGA) se utilizó para evolucionar la población. Este enfoque implica la clasificación de acuerdo con un conjunto de criterios propuestos por el tomador de decisiones (*Decision Maker*), por medio de un procedimiento de múltiples criterios de toma de decisiones. En contraste, los enfoques para la resolución del UA-FLP mediante algoritmos genéticos interactivos mediante el uso de criterios subjetivos fueron propuestos por García-Hernández et al. (2013). En estas propuestas, el usuario guía la evolución de la población mediante la asignación de puntuaciones subjetivas a un conjunto de instalaciones representativas de la población.

Hasta la fecha, se han aplicado muy pocos enfoques genéticos basados en el modelo de islas para resolver los problemas de distribución de la instalación. El modelo de basado en islas puede ser considerado como una herramienta para mantener la diversidad de la población. El modelo de islas, basado en la evolución de una población que se divide en múltiples subpoblaciones independientes, ayuda a preservar la diversidad genética y explorar los múltiples espacios de búsqueda (Whitley, Rana & Heckendorn, 1998). Por otra parte, este modelo parece mejorar el rendimiento con respecto a los modelos secuenciales con una única población (Cantú-Paz, 1998).

Por desgracia, en base a nuestro conocimiento, ningún trabajo ha considerado resolver el problema del UA-FLP utilizando el modelo de islas. Debido a las características principales de los AGP, tales como el mantenimiento de la diversidad de la población, búsqueda

paralela de múltiples puntos en el espacio, y superior eficiencia en comparación con el AG secuencial, esta línea de investigación podría ser útil para obtener resultados prometedores para el problema en cuestión.

Por lo tanto, en este trabajo, se propone un nuevo algoritmo genético paralelo basado en un modelo de isla (IMGGA) para resolver el UA-FLP usando la representación FBS.

2. Algoritmos genéticos paralelos

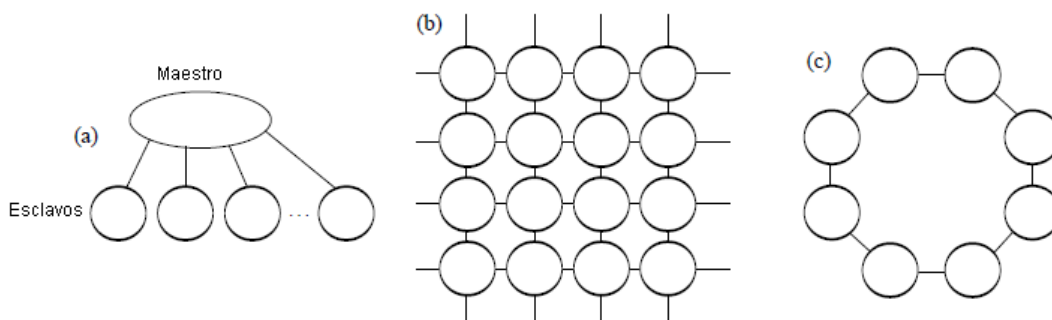
Los AGP surgen de la necesidad de cálculo requerido por problemas de extrema complejidad, cuyo tiempo de ejecución usando AG tradicionales, de tipo secuencial, es, a menudo, prohibitivo. Un AGP se compone, generalmente, de una serie de AG cada uno actuando en una parte de la población, o en poblaciones independientes, con o sin comunicación entre ellos (Solar, Parada & Urrutia, 2002).

Cantú-Paz (1998) propuso una clasificación de AGP en tres categorías (ver figura 1):

1. Población global con un modelo maestro-esclavo. Como en un AG tradicional, existe una sola población, pero en este caso, esta población se evalúa mediante varios procesos independientes. Sin embargo, la selección y el cruce se aplican a toda la población.
2. Población única de grano fino. Este modelo de AGP se basa en una población estructurada espacialmente. El caso ideal es tener un solo individuo en cada nodo. La selección y el cruce se limitan a los nodos vecinos. Cada individuo en cada generación podría ser cruzado con otro vecino, el cual es seleccionado entre todos los posibles vecinos candidatos.
3. Múltiples poblaciones de grano grueso. Este modelo consta de varias subpoblaciones distribuidas que intercambian individuos de vez en cuando (mediante un procedimiento denominado *migración*). Por esta razón, este tipo de AGP es más difícil de entender que el resto de los modelos, puesto que este enfoque cambia el comportamiento de un AG tradicional.

En nuestro trabajo, utilizamos un modelo del último tipo descrito (3). En este modelo, debido al grado de independencia que mantienen las islas y el intercambio de individuos a través de la migración, es posible explorar diferentes regiones del espacio de búsqueda, lo cual mejora la calidad de búsqueda, ofreciendo soluciones de mejor calidad y que requieren menos evaluaciones para encontrarlas. Este enfoque también puede ser considerado como un medio para mantener la diversidad de la población (Starkweather et al., 1991), una de las principales cualidades de las que haremos uso en nuestro trabajo.

Figura 1: Clasificación de los algoritmos genéticos paralelos



3. Formulación del problema

Nuestro trabajo se centra en la solución del problema UA-FLP, el cual, como se ha indicado anteriormente, es uno de los problemas de diseño de distribución en planta más estudiados en la literatura (Bozer & Wang, 2012; García-Hernández et. al, 2013; Kim & Kim, 1995; Kulturel-Konak, 2012; Scholz, Petrick & Domschke, 2009; Wong & Komarudin, 2010)

El UA-FLP fue descrito por Armour y Buffa (1963) como una región rectangular fija de dimensiones H (alto) y W (ancho), y una colección de n instalaciones de área especificada (A_i), donde la suma de las áreas de las instalaciones es menor o igual que el área de la planta (HW). Véase la ecuación 1.

$$\sum_i^n A_i \leq H \times W \quad (1)$$

El objetivo es asignar las instalaciones de la planta en base a unos criterios de optimización dado, teniendo en cuenta que las instalaciones no se pueden superponer.

3.1 Función objetivo

Cada individuo de la población se compone de un cromosoma que representa una solución codificada genéticamente para el problema, es decir, un diseño de distribución en planta. Cada uno de estos diseños posee una puntuación (*fitness*), que representa la calidad de la solución. Esta puntuación se calcula mediante una función objetivo.

Los diferentes objetivos y criterios que han sido considerados en la problemática del UA-FLP son: requisitos de adyacencia, distancia, coste del flujo de materiales, o la satisfacción de una relación de aspecto. La optimización puede realizarse con un criterio o una combinación de ellos (Aiello, Scalia & Enea, 2013).

La mayor parte de las investigaciones consideran sólo el flujo de materiales, minimizando directamente el coste total de manejo de materiales entre instalaciones (a veces llamados departamentos) (Wang, Hu & Ku, 2005).

La función para evaluar las soluciones de nuestra propuesta se basa en el coste de manejo de materiales, que incorpora una función de penalización adaptativa (Tate & Smith, 1995), para dirigir el proceso de búsqueda a las regiones con solución factible. La función objetivo que minimiza el coste total de manejo de materiales se presenta en la ecuación 2.

$$\min OF = \sum_{i=1}^n \sum_{j=1, i \neq j}^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \quad (2)$$

$$i, j = 1, 2, \dots, n$$

donde n es el número de departamentos o instalaciones; f_{ij} es el coste de mover el material desde el departamento i al departamento j ; d_{ij} es la distancia entre los departamentos i y j ; D_{inf} es el número de departamentos inviables; V_{feas} es el mejor valor factible encontrado de la función objetivo; V_{all} es el mejor valor de la función objetivo encontrado hasta el momento; k es el parámetro que ajusta la dureza de la función de penalización adaptativa, establecido en '3' (Tate & Smith, 1995).

El procedimiento de cálculo de la distancia es presentado en la ecuación 3, donde el centro del departamento es el punto definido por x e y .

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \quad (3)$$

3.2. Representación de la planta

En nuestro trabajo, la representación de la distribución en planta es realizada mediante la estructura de bahía flexible (FBS) propuesta por Tong (1991). De entre los diferentes mecanismos expuestos en la literatura para representar los diseños de plantas, FBS ha sido ampliamente utilizado. Una muestra del gran número de trabajos de investigación que usaron este esquema de representación demuestra esta afirmación: Tate y Smith (1995); Konak et al. (2006); Wong y Komarudin (2010); Kulturel-Konak y Konak (2011); Ulutas y Kulturel-Konak (2012); Kulturel-Konak (2012); y, García-Hernández et al. (2013).

FBS posee las siguientes características: (a) la planta es un área rectangular definida por el alto y ancho de la misma; (b) la planta es dividida, en una dirección, en bahías paralelas; y (c) cada bahía es subdividida para la colocación de cada instalación.

3.3. Codificación

La estructura de codificación que se utiliza en nuestra propuesta para representar cada individuo de la población es similar al propuesto por García-Hernández et al. (2013). Esta estructura contiene un cromosoma que se descompone en dos segmentos:

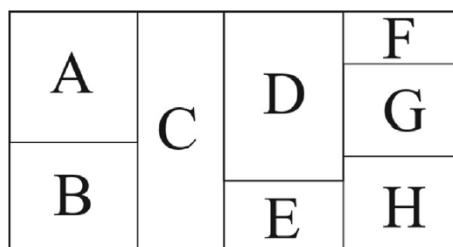
4. El primer segmento encierra la secuencia de instalaciones organizadas bahía por bahía, de izquierda a derecha y de arriba a abajo. Contiene n etiquetas que representan cada una de las instalaciones del diseño de la planta.
5. El segundo segmento incluye las divisiones de bahías en la distribución de la planta. Contiene $n-1$ elementos binarios. Un valor '0' indica que la instalación correspondiente se coloca en la misma bahía que la anterior, y un valor '1' indica que este elemento es la última instalación en la bahía actual.

Un ejemplo de una estructura de codificación de individuos y su representación asociada se muestran en las figuras 2 y 3.

Figura 2: Ejemplo de codificación del cromosoma

| Facility sequence | | | | | | | | Bay divisions | | | | | | |
|-------------------|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

Figura 3: Representación de la planta asociada al cromosoma anterior



3.4. Algoritmo genético paralelo basado en un modelo de isla (IMGA)

En esta sección se explica la propuesta del presente trabajo, un algoritmo genético paralelo basado en un modelo de isla, el cual hemos llamado IMGA.

En primer lugar, se describe nuestro enfoque por medio de pseudocódigo (figura 4). En las siguientes subsecciones, se explican los detalles de los pasos más importantes del algoritmo.

Figura 4: Pseudocódigo del algoritmo propuesto (IMGA)

```
Generar una población inicial de  $p$  individuos;  
Evaluar el fitness de todos los individuos;  
Distribuir la población en  $i$  islas;  
repetir  $c$  veces:  
  para cada isla hacer:  
    repetir  $g$  veces:  
      Seleccionar el mejor individuo para la reproducción;  
      Aplicar la operación de cruce;  
      Aplicar la operación de mutación;  
      Actualizar el fitness de los individuos modificados;  
      Generar la población de la nueva generación;  
    end;  
    Migrar los  $m$  mejores individuos a la isla vecina;  
  end;  
end;
```

3.4.1. Operador de selección

Existen muchos operadores de selección para los AG, cada uno de ellos con diferentes características. Un operador de selección ideal debe ser simple de codificar y eficiente, tanto en arquitecturas no-paralelas como paralelas (Miller & Goldberg, 1996).

En nuestra propuesta, nosotros hemos utilizado el operador de selección por torneo. La selección por torneo es una variante de los métodos de selección basados en *ranking*.

La operativa es la siguiente: un número de individuos (dos en nuestra propuesta) son elegidos aleatoriamente de la población, siendo el individuo con mejor *fitness* de dicha pareja, el seleccionado como padre para las futuras operaciones (cruce y mutación). Este proceso se repite tanto como individuos deban ser elegidos.

3.4.2. Operadores de cruce

En nuestra propuesta, se aplican dos operadores de cruce diferentes, dependiendo del segmento del cromosoma.

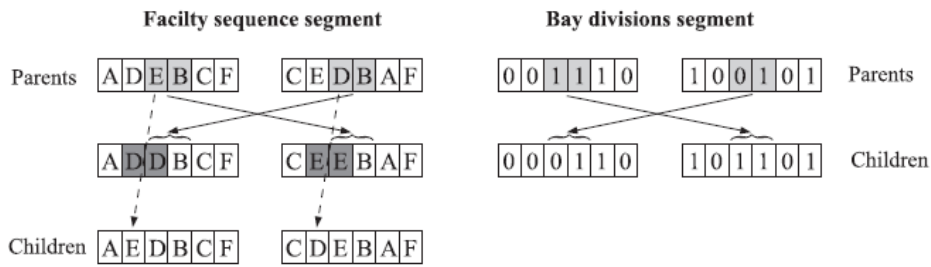
El segmento 1 (*facility sequence*) se recombina mediante el método de cruce *PMX*, tal y como se describe en el trabajo de Sadrzadeh (2012), lo que garantiza que no existan instalaciones que se repitan en un mismo diseño. Los pasos de este método son los siguientes:

1. Seleccionar aleatoriamente una subcadena.
2. Intercambiar la subcadena seleccionada entre los padres.
3. Regular la descendencia con la relación de proyección.

El segmento 2 (*bay divisions*) es cruzado por medio del método de n -puntos de cruce (Holand, 1992; Starkweather et al., 1991). En este caso, un solo punto de cruce (seleccionado al azar) se utiliza para el intercambio de las subseries entre los padres para crear la descendencia.

En la figura 5 se muestran ejemplos de ambos operadores de cruce.

Figura 5: Ejemplos de operadores de cruce



3.4.3. Operadores de mutación

En nuestro trabajo, el operador de mutación es aplicado con una cierta probabilidad (generalmente baja) para cada segmento del cromosoma. En el segmento 1, se eligen dos instalaciones al azar para ser intercambiadas. En el segmento 2, se escoge un índice, el cual será invertido.

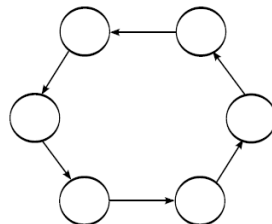
3.4.4. Proceso de migración

En nuestro modelo basado en islas, utilizamos una topología en forma de anillo. Este enfoque es fácil de implementar, además de ser utilizado ampliamente (Alba & Troya, 1999). En nuestra propuesta, los individuos elegidos de cada isla migran a la *isla vecina* (ver figura 6).

La migración es un operador que guía el intercambio de individuos entre las islas. Cuando la migración ocurre, la información de las diferentes regiones del espacio de búsqueda es intercambiada entre las islas, lo cual provee una mayor diversidad en la población (Pereira & Sacco, 2008).

Este operador está compuesta por dos parámetros: (a) la cantidad de individuos que migran (m en nuestra propuesta), y (b) la frecuencia en la que se produce la migración (g en nuestro algoritmo).

Figura 6: Topología de anillo



4. Experimentos

Nuestra propuesta ha sido probada en un total de doce configuraciones de un problema bien conocido de la literatura, en concreto el conocido como AB20, de Armour y Buffa (1963), en el cual se consideran diferentes restricciones, en concreto el valor máximo de la relación de aspecto (α).

La tabla 1 presenta las propiedades de las diferentes configuraciones del problema utilizadas para la experimentación, tales como el tamaño de la planta, el número de departamentos y el valor máximo de α .

La propuesta IMGA contiene parámetros que han sido configurados o tuneados (*parameters tuning*). En este caso, las pruebas para determinar estos parámetros han sido realizadas durante el desarrollo del algoritmo, mediante los problemas bien conocidos de la literatura.

Tabla 1: Características de los problemas de pruebas

| Nombre del problema | Número de departamentos | Tamaño de la planta (W x H) | Relación de aspecto |
|---------------------|-------------------------|-----------------------------|---------------------------|
| AB20-ar170667 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 1.70667$ |
| AB20-ar175 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 1.75$ |
| AB20-ar2 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 2$ |
| AB20-ar3 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 3$ |
| AB20-ar4 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 4$ |
| AB20-ar5 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 5$ |
| AB20-ar7 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 7$ |
| AB20-ar10 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 10$ |
| AB20-ar15 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 15$ |
| AB20-ar25 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 25$ |
| AB20-ar50 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 50$ |
| AB20-ar1000 | 20 | 2.00 x 3.00 | $\alpha_{\max} = 1000$ |

Tabla 2: Parámetros del algoritmo IMGA

| Parámetro | Valor |
|--|-------|
| Tamaño total de la población (p) | 500 |
| Número de islas (i) | 10 |
| Tamaño de la subpoblación de cada isla (p/i) | 50 |
| Número de ciclos de migraciones (c) | 70 |
| Número de generaciones entre cada proceso de migración (g) | 15 |
| Número de individuos que migran (m) | 5 |
| Probabilidad de cruce (p_c) | 0.6 |
| Probabilidad de mutación (p_m) | 0.03 |

En la tabla 2 se muestran los parámetros del algoritmo IMGA utilizados en la experimentación, cuyo resultados serán comparados con las mejores propuestas de la literatura en la siguiente sección.

5. Resultados y comparativa con otras propuestas

Las soluciones encontradas por nuestra propuesta IMGA (para los problemas bien conocidos descritos anteriormente) han sido comparadas con las mejores soluciones reportadas por otras propuestas basadas en la representación FBS encontradas en la

literatura, las cuales se describen a continuación: un algoritmo de optimización basado en colonia de hormigas propuesto por Kulturel-Konak y Konak (2011); y un enfoque mediante búsqueda tabú probabilística presentado por Kulturel-Konak (2012).

El test comparativo entre la propuesta IMGGA y las mejores de la literatura (presentadas en el párrafo anterior) se resume en la tabla 3. Los mejores valores alcanzados por cada problema se encuentran resaltados en negrita.

En la comparativa con la optimización basada en colonia de hormigas de Kulturel-Konak y Konak (2011), nuestra propuesta mejoran las soluciones del trabajo citado en 8 de los 12 casos.

Igualmente, con respecto al sistema de hormigas de Kulturel-Konak (2012), nuestro enfoque alcanza, de nuevo, mejores soluciones en 10 de los 12 problemas.

Tabla 3: Test comparativo entre IMGGA y las mejores propuestas de la literatura

| Nombre del problema | Algoritmo propuesto (IMGGA) | Kulturel-Konak (2012) | Kulturel-Konak & Konak (2011) |
|---------------------|-----------------------------|-----------------------|-------------------------------|
| AB20-ar170667 | 5885.68 | 5895.33 | 5845.3 |
| AB20-ar175 | 6291.87 | 5895.33 | 5845.3 |
| AB20-ar2 | 5858.41 | 5895.33 | 5845.3 |
| AB20-ar3 | 5419.49 | 5594.29 | 5674.3 |
| AB20-ar4 | 5286.65 | 5360.8 | 5450.4 |
| AB20-ar5 | 5256.1 | 5297.6 | 5297.6 |
| AB20-ar7 | 4844.49 | 4793.47 | 4793.5 |
| AB20-ar10 | 4367.57 | 4440.68 | 4440.7 |
| AB20-ar15 | 4100.17 | 4119.8 | 4116.8 |
| AB20-ar25 | 3324.49 | 3399.16 | 3526.5 |
| AB20-ar50 | 2382.74 | 2457.79 | 2706.5 |
| AB20-ar1000 | 1588.49 | 1638.53 | 1638.5 |

En la tabla 4 se presenta un resumen de la comparativa entre la propuesta de este trabajo y la mejor solución encontrada en la literatura para cada uno de los problemas. En esta tabla, nosotros presentamos la siguiente información: la mejor solución alcanzada por IMGGA, la diferencia porcentual entre la mejor solución encontrada por nuestro algoritmo y la mejor solución de la literatura, y el diseño de la distribución en planta de las mejores soluciones encontradas por IMGGA (expresado bahía por bahía, de izquierda a derecha, de arriba abajo).

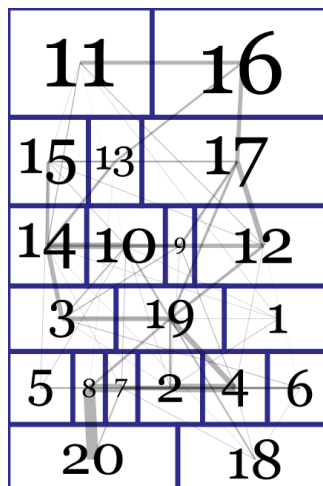
Tabla 4: Resumen de los resultados

| Nombre del problema | Mejor solución obtenida por IMGGA | Diferencia (%) ^a | Diseño de la distribución en planta |
|---------------------|-----------------------------------|-----------------------------|--|
| AB20-ar170667 | 5885.68 | -0.69 | 16-17 11-15-12 13-14-9-10 1-3-19 5-6-8-7-4 18-20-2 |
| AB20-ar175 | 6291.87 | -7.08 | 20-18-1 2-4-19-6-3 7-8-9-10-14 5-13-12-15 16-17-11 |
| AB20-ar2 | 5858.41 | -0.22 | 20-6-18 5-8-7-2-4 13-9-3-19 15-14-10 17-1-12 16-11 |
| AB20-ar3 | 5419.49 | 3.23 | 16-11 17-13-15 12-9-10-14 1-19-3 6-4-2-7-8-5 18-20 |
| AB20-ar4 | 5286.65 | 1.40 | 20-18 5-8-7-2-4-6 3-19 15-14-10-12 1-13-9-17 |

| | | | |
|--|----------------|-------|---|
| | | | 11-16 |
| AB20-ar5 | 5256.1 | 0.79 | 1-16-11 17-13 12-9-15 3-14 19-10 6-4-2-7-8-20 18-5 |
| AB20-ar7 | 4844.49 | -1.05 | 11 16 17-15 12-13 1-10-9-14 5-19-3 20-8-7-2- 4-6-18 |
| AB20-ar10 | 4367.57 | 1.67 | 11 16 17 12 15-13-14-10-9-19-3-1 5-20-8-7-2-4- 6-18 |
| AB20-ar15 | 4100.17 | 0.41 | 1 18 20-8-7-2-4-19-10-9-14-3-6 12 15 17 5-13 16 11 |
| AB20-ar25 | 3324.49 | 2.25 | 11 16 13 17 12 15 10 14 3 19 4 2 20-8- 7-9-5-1 6 18 |
| AB20-ar50 | 2382.74 | 3.15 | 11 16 13 17 12 15 9 14 10 3 19 4 2 6 7 8 20 5 18 1 |
| AB20- ar1000 | 1588.49 | 3.42 | 1 18 5 20 8 7 6 2 4 19 3 10 14 9 15 12 17 13 16 -11 |
| ^a 100 * (mejor solución literatura – mejor solución IMGA) / (mejor solución IMGA) | | | |

Por último, en la figura 7 se muestra la mejor solución obtenida para el problema AB20-ar3 mediante la propuesta presentada en este trabajo.

Figura 7: Mejor solución del problema AB20-ar3 obtenida por IMGA



6. Conclusiones

Este trabajo presenta un nuevo algoritmo genético paralelo basado en un modelo de isla utilizando la estructura de bahía flexible para representar las soluciones a problemas de distribución en planta de área desigual.

Debido al grado de independencia que existe entre las islas, y al intercambio de individuos mediante el proceso de migración, el algoritmo propuesto puede explorar las diferentes regiones del espacio de búsqueda, mejorando la calidad de la búsqueda y ofreciendo mejores soluciones. En este sentido, nuestro enfoque mejora muchas soluciones en comparación con las mejores propuestas de la literatura hasta la fecha.

Este enfoque es la primera contribución en cuanto al uso de AGP para resolver el UA-FLP. Por lo tanto, basándonos en los resultados obtenidos en este estudio, parece que los AGP son una prometedora línea de investigación para continuar trabajando en mejorar la potencia y eficiencia de estos enfoques, haciendo uso de la paralelización de CPUs.

7. Bibliografía

- Aiello, G., Scalia, G. L., & Enea, M. (2013). A non dominated ranking multi objective genetic algorithm and electre method for unequal area facility layout problems. *Expert Systems with Applications*, 40 (12), 4812 – 4819.
- Alba, E., & Troya, J. M. (1999). A survey of parallel distributed genetic algorithms. *Complexity*, 4 (4), 31–52.
- Armour, G. C., & Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9, 294–309.
- Bozer, Y. A., & Wang, C. T. (2012). A graph-pair representation and mip-model-based heuristic for the unequal-area facility layout problem, *European Journal of Operational Research*, 218 (2), 382 – 391.
- Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs Paralleles*, 10.
- Darwin, C. (1859). *On the Origin of the Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life*, John Murray.
- Dira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31 (2), 255–267.
- García-Hernández, L., Pierreval, H., Salas-Morera, L., & Arauzo-Azofra, A. (2013). Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. *Applied Soft Computing*, 13(4), 1718–1727.
- Heragu, S. (2008). *Facilities Design, Third Edition, 500 Tips*. Taylor & Francis.
- Holland, J.H. (1992). *Adaptation in natural and artificial systems*, MIT Press, Cambridge, MA, USA.
- Kim, J. Y., & Kim, Y. D. (1995). Graph theoretic heuristics for unequal-sized facility layout problems, *Omega*, 23 (4), 391–401.
- Kochhar, J. S., Foster, B. T., & Heragu, S. S. (1998). Hope: A genetic algorithm for the unequal area facility layout problem, *Computers & Operations Research*, 25 (78), 583 – 594.
- Komarudin, K., & Wong, Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research*, 202 (3), 730 – 746.
- Konak, A., Kulturel-Konak, S, Norman, B. & Smith, A. (2006) A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, 34, 660–672.
- Kulturel-Konak, S., & Konak, A. (2011). Unequal area flexible bay facility layout using ant colony optimization. *International Journal of Production Research*, 49 (7), 1877–1902.
- Kulturel-Konak, S. (2012). A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. *European Journal of Operational Research*, 223 (3), 614 – 625.
- Meller, R., & Gau, K. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15, 351–366.
- Miller, B. L., Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4 (2), 113–131.
- Ou-Yang, C. & Utamima, A. (2013). Hybrid estimation of distribution algorithm for solving single row facility layout problem. *Computers & Industrial Engineering*, 66 (1), 95 – 103.

- Pereira, C. M., & Sacco, W. F. (2008). A parallel genetic algorithm with niching technique applied to a nuclear reactor core design optimization problem. *Progress in Nuclear Energy*, 50 (7), 740 – 746.
- Sadrzadeh, A. (2012). A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Computers and Industrial Engineering*, 62(4), 1055–1064.
- Sahin, R. (2011). A simulated annealing algorithm for solving the bi-objective facility layout problem, *Expert Systems with Applications*, 38 (4), 4460 – 4465.
- Scholz, D., Petrick, A., & Domschke, W. (2009). Stats: A slicing tree and tabu search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research*, 197 (1), 166 – 178.
- Singh, S., & Sharma, R. (2006). A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*, 30, 425–433.
- Solar, M., Parada, V., Urrutia, R. (2002). A parallel genetic algorithm to solve the set covering problem. *Computers & Operations Research*, 29 (9), 1221 – 1235.
- Solimanpur, M., & Jafari A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm, *Computers & Industrial Engineering*, 55 (3), 606–619.
- Starkweather, T., Mcdaniel, S., Whitley, D., Mathias, K., Whitley, D., & Dept, M. E. (1991). A comparison of genetic sequencing operators. *Proceedings of the fourth international conference on genetic algorithms*, 69–76.
- Tate, D. M., & Smith, A. E. (1995). Unequal area facility layout using genetic search. *IIE Transactions*, 27, 465–472.
- Tompkins, J., White, J., Bozer, Y., & Tanchoco, J. (2010). *Facilities Planning*, Wiley, New York, 4rd edition.
- Tong, X. (1991). SECOT: A Sequential Construction Technique For Facility Design. Doctoral Dissertation, University of Pittsburg.
- Ulutas, B.H., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve UA-FLP. *Expert Systems with Applications*, 39 (5), 5384 – 5395.
- Wang, M. J., Hu, M. H., & Ku, M. Y. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *Computers & Industrial Engineering*, 56 (2), 207–220.
- Whitley, D., Rana, S., & Heckendorn, R. B. (1998). The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7, 33–47.
- Wong, Y., & Komarudin, K. (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Expert Systems with Applications*, 37 (7), 5523–5527.