

03-016

A NOVEL HYBRID EVOLUTIONARY APPROACH FOR CAPTURING THE EXPERT KNOWLEDGE INTO THE UNEQUAL AREA FACILITY LAYOUT PROBLEM.

García Hernández, Laura ¹; Palomo Romero, Juan María ²; Salas Morera, Lorenzo ²; Arauzo Azofra, Antonio ²; García Hernández, Jose A. ³

¹ Universidad de Córdoba, ² Universidad de Córdoba, ³ Junta de Andalucía

This article presents a new hybrid evolutionary system for considering human expert knowledge into the Unequal Area Facility Layout Problem (UA-FLP). The novel hybrid system proposed consists of an Interactive Genetic Algorithm combined with two different niching methods with the aim of allowing interaction between the hybrid algorithm and the human expert designer, normally called Decision Maker (DM) in the field of UA-FLP. The participation of the DM knowledge into the approach guides the complex search process, adjusting it to the DM's preferences. The whole population associated to facility layout designs is evaluated by the DM, who only gives a subjective evaluation to a set of representative individuals of the population in each iteration (in order to avoid fatiguing him/her). The inclusion of niching techniques into the approach allow the preservation of the population diversity avoiding that similar solutions will be presented to the DM in the same algorithm iteration. The novel proposed approach was tested using a interesting case study of facility layout designs. The results of the experiments, which successfully validate the novel proposed approach, are presented, compared and discussed in this study.

Keywords: *Unequal Area Facility Layout Problem; Interactive Evolutionary Computation; Expert Knowledge; Niching method*

UNA NUEVA PROPUESTA HÍBRIDA EVOLUTIVA PARA CAPTURAR EL CONOCIMIENTO EXPERTO EN EL PROBLEMA DE DISTRIBUCIÓN EN PLANTA DE ÁREAS DESIGUALES

En este artículo se presenta un nuevo sistema evolutivo híbrido para considerar el conocimiento experto humano en el problema de distribución de instalaciones de áreas desiguales. El nuevo sistema híbrido propuesto, consta de un algoritmo interactivo genético combinado con dos métodos niching diferentes con el objetivo de permitir la interacción entre el algoritmo híbrido y el experto humano. La participación del conocimiento del experto en la guía del complejo proceso de búsqueda, lo ajusta a las preferencias del diseñador. La población asociada a los esquemas de distribución en planta es evaluada por el diseñador, que sólo da una evaluación subjetiva a un conjunto de individuos representativos de la población en cada iteración (para evitar fatigarlo). La inclusión de técnicas de niching en la propuesta permite la preservación de la diversidad de la población evitando que soluciones similares se presenten al diseñador en la misma iteración del algoritmo. La nueva propuesta ha sido probada mediante un caso de estudio de distribución en planta. Los resultados de los experimentos, que se validan satisfactoriamente el nuevo enfoque propuesto, se presentan, comparan y analizan en este estudio.

Palabras clave: *Problema de distribución en planta de áreas desiguales; Computación interactiva evolutiva; Conocimiento experto; Métodos de niching*

Correspondencia: Laura García Hernández ir1gahel@uco.es

1. Introducción

La distribución de las instalaciones de una planta, que a menudo se conoce como el problema de distribución en planta, se sabe que tiene un impacto significativo en los costes de fabricación, productos en proceso, los tiempos y la productividad (Drira, Pierreval, & Hajri-Gabbouj, 2007). Dónde ubicar las instalaciones y el diseño eficiente de las instalaciones son cuestiones estratégicas importantes y fundamentales en cualquier industria manufacturera (Singh & Sharma, 2006). Una buena distribución de las instalaciones contribuyen a la eficiencia global de las operaciones y pueden reducir los costes operativos totales entre un 20% y un 50% (Tompkins et al., 2010).

Dependiendo de las características que se consideren, varios problemas se incluyen en el campo del diseño de distribución en planta de las instalaciones (Kusiak & Heragu, 1987). En particular, el trabajo propuesto en este artículo se centra en el problema de distribución en planta de áreas desiguales (comúnmente llamado, UA-FLP) tal y como se describe por Armour y Buffa (Armour & Buffa, 1963), quien lo formuló como un diseño de planta rectangular que es compuesta de instalaciones rectangulares desiguales que deben ser dispuestas de una manera eficaz.

Los problemas de distribución en planta pueden modelarse usando diferentes técnicas como: la programación lineal entera, programación entera mixta (Izadinia, Eshghi, & Salmani, 2014) y los métodos gráficos (Heragu & Kusiak, 1991). Varias técnicas se han aplicado, como branch and bound (Solimanpur & Jafri, 2008) y la teoría de grafos (Kim & Kim, 1995). Recientemente, los investigadores se han centrado en los métodos meta-heurísticos, como la búsqueda tabú (McKendall et al, 2006;. Scholz, Petrick, & Domschke, 2009), el recocido simulado (Sahin, Ertogral, & Türkbey, 2010), el sistema de hormigas (Komarudin & Wong, 2010, 2012a) y los algoritmos genéticos (AG) (Aiello, Scalia, & Enea, 2013; Goldberg, 1989). Los últimos métodos se han utilizado comúnmente en el problema de distribución en planta de áreas desiguales (García-Hernández et al., 2009; Michalewicz et al., 1996).

En general, el problema de diseñar una disposición física de instalaciones consiste en la optimización y la consideración de determinados objetivos (principalmente, el flujo de materiales) (Aiello, Scalia, & Enea, 2012) y criterios. Sin embargo, Babbar-Sebens y Minsker (2012), declaró que estos enfoques de optimización no describen adecuadamente toda la información cualitativa importante que es considerada como esencial por un experto humano que está involucrado en la fase de diseño (por ejemplo, ingenieros). Sin embargo, es difícil tener en cuenta estas características cualitativas utilizando una heurística clásica o sistema de optimización de meta-heurística (Brintup, Ramsden, & Tiwari, 2007). Supuestamente, la manera más razonable para considerar todas estas características en el proceso de selección consiste en incluir personalmente el tomador de decisiones en el proceso (Brintup et al., 2007), esto proporciona ventajas adicionales y elimina el requisito de especificar toda la información conveniente sobre el diseño de la instalación de antemano, ofreciendo al diseñador la capacidad de aprender de sus propias preferencias, y estimular la creatividad de los usuarios (García-Hernández et al., 2013).

Brintup et al. (2006) destacaron el hecho de que la computación evolutiva interactiva puede contribuir en gran medida a mejorar el diseño optimizado mediante la participación de los usuarios en la búsqueda de una solución satisfactoria. Sin embargo, en los problemas de optimización de la vida real, a veces no estamos satisfechos con sólo una solución óptima. En este sentido, los algoritmos genéticos a menudo carecen de la capacidad de encontrar

múltiples soluciones óptimas. Muchos autores han utilizado métodos de niching para resolver este problema ya que pueden mantener una población diversa y no son tan propensos a converger prematuramente como los algoritmos genéticos (Yu & Suganthan, 2010). Niching es un término genérico que se refiere a una técnica para encontrar y preservar múltiples nichos estables o partes favorables en la población de las soluciones, posiblemente alrededor de óptimos locales, para evitar la convergencia a una solución única (Gren y Pierreval, 2013).

En este artículo se presenta un nuevo algoritmo evolutivo híbrido para incorporar el conocimiento experto humano en el problema de distribución en planta de áreas desiguales. Este enfoque consiste en un algoritmo genético interactivo que se combina con dos métodos niching para permitir la interacción entre el proceso de algoritmo y el diseñador, preservando al mismo tiempo la diversidad de la población de soluciones.

2. Definición del problema.

El problema de distribución en planta de áreas desiguales fue propuesto por Armour y Buffa (1963). En él se considera una planta rectangular con dimensiones fijas, y un conjunto de instalaciones que, con un área requerida, donde la suma de las áreas de las instalaciones debe ser inferior o igual al área total de la planta. El objetivo es la asignación de las instalaciones de la planta en base a un criterio de optimización dado con la restricción de no superposición de las instalaciones.

Muchos autores han tenido en cuenta criterios cuantitativos para resolver el problema de distribución en planta de áreas desiguales, incluidos los costos de manipulación de materiales, requerimientos de adyacencia, requisitos de distancia entre instalaciones y una relación de aspecto deseada.

Sin embargo, como se discutió en la sección anterior, existen características cualitativas que deberían ser consideradas por el diseñador en el diseño de la distribución de las instalaciones. Estas pueden ser:

- La distribución del espacio sobrante en el diseño de la planta.
- La colocación instalación.
- La orientación que el diseñador prefiere para una instalación.
- Lugares que no se desean para una cierta instalación para evitar ciertos factores (por ejemplo, ruido, malos olores, humedad)
- Cualquier otra preferencia subjetiva que el diseñador considera necesaria en el diseño final de la distribución de las instalaciones.

3. Enfoque propuesto

3.1. Representación de la planta

La estructura usada es la estructura flexible de bahías propuesta por Tong (1991) se utiliza para representar la distribución de la planta. La planta rectangular se define por sus dimensiones de altura y anchura. Este área se divide en una dirección en bahías de diferentes anchos, que se subdividen para asignar las instalaciones que componen el diseño de la planta.

3.2. Estructura de codificación

La estructura de codificación que se utiliza en nuestro enfoque para representar cada

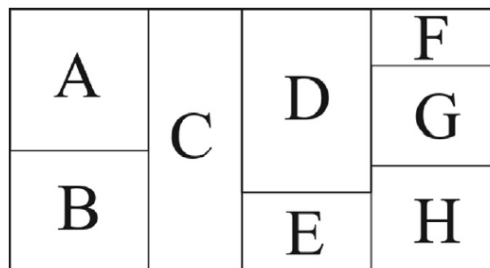
individuo de la población es similar al propuesto por García-Hernández et al. (2013). Esta estructura contiene un cromosoma que se compone de 2 segmentos. El primer segmento encierra la secuencia de instalaciones organizadas bahía por bahía, de izquierda a derecha y de arriba a abajo. Contiene n etiquetas que representan cada una de las instalaciones en el diseño de la planta. El segundo segmento incluye las divisiones de bahías en la distribución de la planta. Contiene $n-1$ elementos binarios.

Un valor de 0 indica que la instalación correspondiente se coloca en la misma bahía que la anterior, y un valor de 1 indica que este elemento es la última instalación en la presente bahía. Un ejemplo de una estructura de codificación de individuos y su representación asociada se muestran en las siguientes figuras:

Figura 1: 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 2: Representación de la planta asociada al cromosoma anterior



3.3. Algoritmo evolutivo híbrido

En esta sección se explica el algoritmo evolutivo híbrido en detalle. El algoritmo se ilustra en la Fig. 3.

En la primera etapa del algoritmo, una población inicial de n instalaciones se genera aleatoriamente. A continuación, se clasifica la población en grupos utilizando el algoritmo fuzzy c-means que fue descrita y usada en el trabajo de García-Hernández et al. (2013). Así, un subconjunto de individuos que se compone de los elementos representativos de cada categoría se muestra al diseñador.

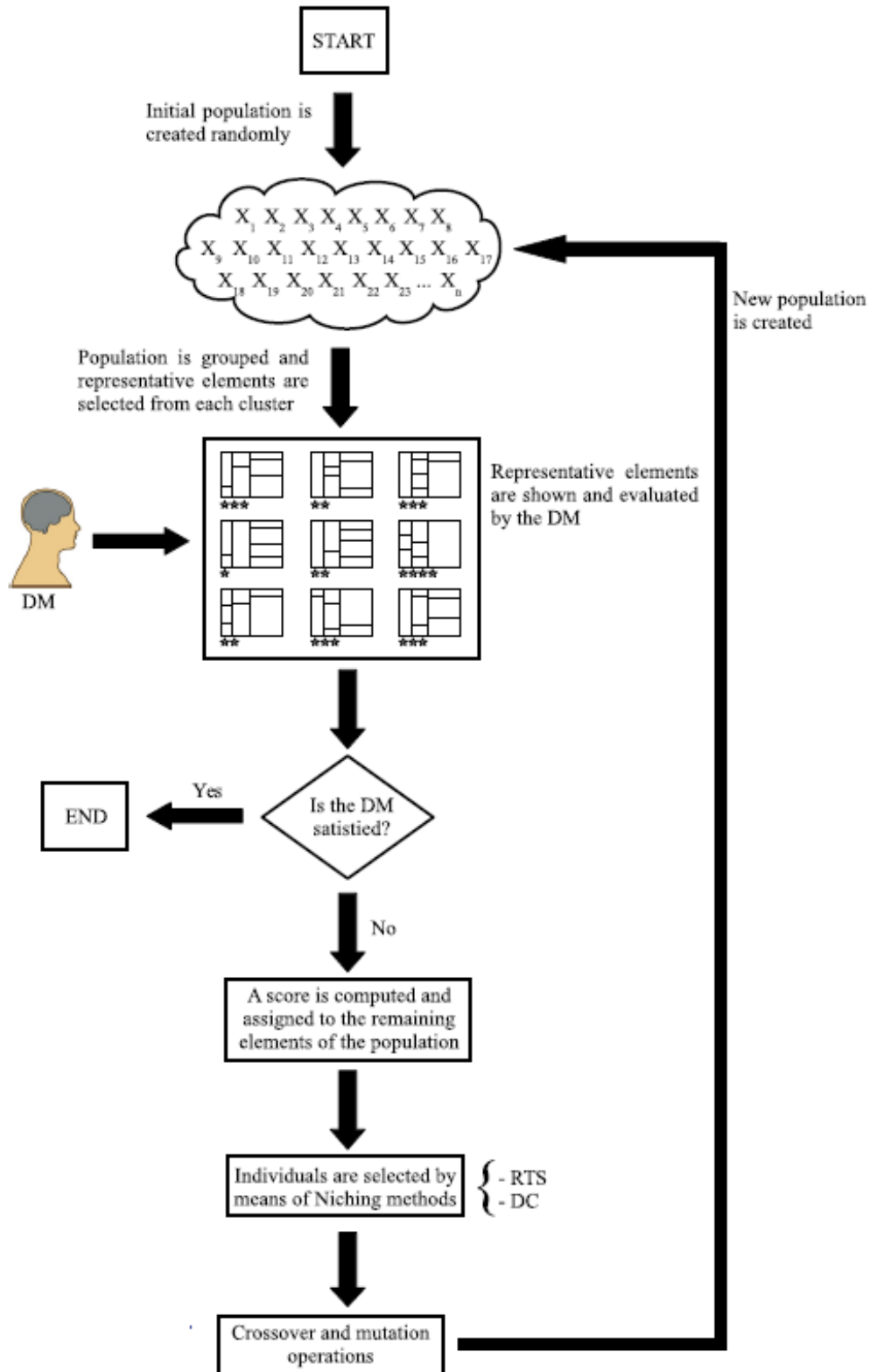
Nueve soluciones se muestran al diseñador en cada generación como era propuesto por García-Hernández et al. (2013). Éste evalúa las soluciones de acuerdo con sus preferencias para cada diseño. En cada iteración, el diseñador da una calificación a cada una de las soluciones, la puntuación puede ir desde 1 (solución poco satisfactoria) a 5 (solución muy satisfactoria). La aptitud de los demás individuos ($eval$) se calcula utilizando la ecuación 1.

$$eval = \sum_{i=1}^c m_{ij} e_i \quad (1)$$

donde c es el número de grupos, m_{ij} es el grado de pertenencia de j individuo en el grupo i , y e_i es la puntuación dada por el diseñador al individuo representativo del grupo i . En esta etapa, un método de selección se aplica a la población. En este estudio, se utilizan dos técnicas niching: la técnica DC descrita en el trabajo de De Jong (1975) y la técnica RTS descrita en el artículo de Harik (1995). Finalmente, operadores de cruce y mutación (que se describirán en las secciones 3.3.1 y 3.3.2, respectivamente) se aplican, y la nueva población se crea, siendo más probable que se adapte mejor a las preferencias del diseñador que la

población anterior. El proceso se repite hasta que el diseñador encuentra un diseño de distribución en planta que satisface sus preferencias.

Figura 3: Diagrama del algoritmo evolutivo híbrido propuesto



3.3.1. Operadores de cruce

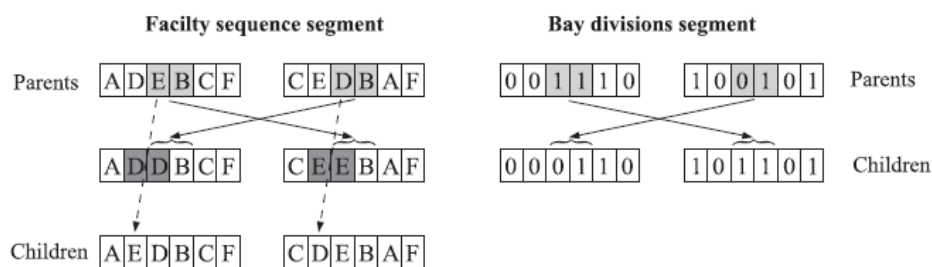
En el método propuesto, dos operadores de cruce diferentes son aplicados a los dos segmentos del cromosoma.

El segmento de secuencia de instalaciones se recombina mediante el método de cruce Partial Mapped Crossover tal y como se describe en el trabajo de Sadrzadeh (2012), lo que garantiza que no haya instalaciones que se repitan en un diseño de distribución en planta. Los pasos de este método son los siguientes:

1. Selecciona al azar una subcadena.
2. Cambio de la subcadena seleccionada entre los padres.
3. Regular la descendencia con la relación de proyección.

El segundo segmento (segmento división de bahías) es atravesado por medio de los n-puntos de cruce (Holand, 1992; Starkweather et al., 1991). En este caso, un solo punto de cruce (que se selecciona al azar) se utiliza para el intercambio de las subseries entre los padres para crear la descendencia. Ejemplos de ambos operadores de cruce se muestra en la Fig. 4.

Figura 4: Ejemplos de operadores de cruce que son aplicados a cada segmento



3.3.2. Operador de mutación

En nuestro enfoque, el operador de mutación se aplica con una cierta probabilidad (normalmente baja) para cada segmento del cromosoma. En el segmento de secuencia de instalaciones, dos instalaciones son elegidas al azar y conmutadas. En el segmento de división de bahías, un índice es escogido al azar, y su valor se invierte.

4. Experimentación.

En esta sección se describe la metodología experimental, esta es realizada sobre la base de las preferencias que se consideran interesantes para el diseñador. Un caso de prueba ha sido seleccionado para probar nuestro algoritmo híbrido y los resultados se validan comparándolas con los obtenidos por García-Hernández et al. (2013). Este caso fue definido por Aiello, Enea, y Galante (2006). En este problema, el diseño de la planta se compone de 20 instalaciones que debe ser destinada dentro de un área cuadrada de 70 m de lado. En la definición del problema, cada una de las instalaciones posee un tamaño de área (m^2) a partir de la cual, y dependiendo del cromosoma generado, un algoritmo calcula el alto y ancho de cada una de las instalaciones, así como genera la representación de la planta en una imagen, la cual, posteriormente, será mostrada al DM (García-Hernández et al., 2013).

Los grupos de preferencias determinan escenarios de prueba que contienen preferencias que se consideran importantes para el diseñador en cada caso de prueba. Los grupos de preferencias considerados se enumeran a continuación:

1. Una disposición de la planta donde la instalación 'G' toca cualquier lado de la planta, la

instalación 'J' está situado en una esquina, la instalación 'T' es adyacente a la instalación 'G' y la instalación 'T' es una instalación interior.

2. Una disposición de la planta cuando la instalación 'G' es una instalación exterior, la instalación 'T' no es adyacente a la instalación 'G', la instalación 'J' está situado en una esquina, y la instalación 'G' es adyacente a la instalación 'P' (que es una instalación de interior).

3. Una disposición de la planta que se divide cinco bahías, donde: la instalación 'C' es una instalación exterior, y la instalación 'G' es una instalación interior que no es adyacente a la instalación 'T'.

4.1. Metodología

El objetivo principal de la metodología propuesta es definir una planta distribución de las instalaciones que satisface las preferencias del diseñador, es decir, un diseño con una puntuación de 5 (el valor máximo posible). Adicionalmente, para evitar la fatiga del diseñador, el número de generaciones que se requiere para llegar a la meta debe ser lo más bajo posible.

El algoritmo evolutivo híbrido propuesto contiene parámetros que han sido afinados previamente de forma empírica. Debido a que el diseñador evalúa nueve diseños durante cada iteración el algoritmo, no es viable realizar una prueba masiva. Para evitar los efectos aleatorios, se repite la prueba tres veces con el algoritmo propuesto para cada uno de los grupos de preferencias que han sido definidos como importantes para el diseñador. Los resultados de estos experimentos son comparados con los encontrados usando el enfoque propuesto por García-Hernández et al. (2013).

La Tabla 1 muestra los resultados que se obtuvieron utilizando nuestro enfoque con las dos técnicas de niching que han sido implementadas (llamados como Niching (DC) y Niching (RTS), respectivamente) y los obtenidos utilizando la metodología de García-Hernández et al. (2013) (denominada en la tabla como IGA). Se ha considerado el número de generaciones (llamada Gen. en la tabla) que se necesitan para satisfacer cada uno de los grupos de preferencias definidos en la sección anterior. Estos tres grupos de preferencias se encuentran representados en la tabla mediante la columna G.I. Para cada uno de los grupos de preferencias se ha lanzado el algoritmo propuesto un total de tres repeticiones. Además, se ha considerado la media (Av.) y la desviación estándar (Std. Dev), con el objeto de analizar los resultados arrojados por las pruebas. La tabla muestran el número de iteraciones de los algoritmos que se necesitan para llegar a una solución satisfactoria, que se define como la solución que tiene la puntuación máxima. Los mejores valores alcanzados en cada conjunto de preferencias se han resaltado en negrita.

Tabla 1: Comparativa del nº de generaciones que son necesarias para satisfacer al diseñador

G.I.	IGA			Niching (DC)			Niching (RTS)		
	Gen.	Av.	Std. Dev.	Gen.	Av.	Std. Dev.	Gen.	Av.	Std. Dev.
1	5 9 7	7	2	3 4 5	4	1	4 4 5	4.33	0.57
2	9 11 8	9.33	1.52	7 6 7	6.66	0.57	7 8 5	6.66	1.52
3	10 13 9	10.66	2.08	7 9 6	7.33	1.52	8 10 8	8.66	1.15

Como se muestra en la Tabla 1, el número de generaciones que es necesario para llegar a una solución satisfactoria por el diseñador varía entre repeticiones con el mismo grupo de

preferencias. El mínimo número de generaciones que se requiere para llegar a una solución que satisface todas las preferencias del diseñador varía entre 3 y 6 para el caso 1. Nuestro enfoque ha llegado a una solución satisfactoria en menos generaciones que mediante García-Hernández et al. (2013) en las tres repeticiones.

En la Tabla 2, se representa la diversidad de las soluciones obtenidas por los diferentes algoritmos. En este caso, el valor de los resultados mostrado en esta tabla corresponde con el número de individuos representativos idénticos mostrados al DM en una misma iteración, cuantificando la diversidad de las soluciones que se están generando.

Como se muestra en la Tabla 2, nuestro enfoque generó una mayor diversidad de individuos representativos que los obtenidos por García-Hernández et al. (2013) en todas las repeticiones. Ya que este último, genera un máximo de 5 elementos repetidos por generación durante todas las ejecuciones, y nuestro enfoque ha generado un máximo de 2 elementos repetidos.

Tabla 2: Comparativa de la diversidad de las soluciones de las plantas generadas

G.I.	IGA			Niching (DC)			Niching (RTS)		
	Gen.	Av.	Std. Dev.	Gen.	Av.	Std. Dev.	Gen.	Av.	Std. Dev.
1	3 4 4	3.66	0.57	1 1 0	<u>0.66</u>	0.57	2 1 0	1	1
2	4 5 4	4.33	0.57	0 1 0	<u>0.33</u>	0.57	2 0 0	0.66	1.15
3	5 3 4	4	1	0 0 0	<u>0</u>	0	0 0 1	0.33	0.57

5. Conclusiones.

Este estudio presenta un nuevo algoritmo evolutivo híbrido para el problema de distribución en planta de instalaciones de área desigual. Utilizando el enfoque sugerido, es posible incluir las preferencias del diseñador en el proceso de diseño por medio de un algoritmo genético interactivo. El conocimiento del experto guía el proceso de búsqueda hacia sus preferencias sin necesidad de especificarla al comienzo del proceso.

Debido a que la intervención de los diseñadores es esencial para la evolución adecuada del algoritmo, la selección de soluciones que se le mostrará durante cada iteración es crucial para el éxito del algoritmo. El conjunto de soluciones debe ser representativa de la población, y sus componentes debe ser lo suficientemente diferente de uno a otro para evitar que el diseñador de tener que evaluar soluciones similares o idénticos en el mismo paso del algoritmo. Nuestro enfoque híbrido incluye una combinación de un algoritmo genético interactivo con dos métodos diferentes de niching para preservar la diversidad de soluciones.

El análisis empírico muestra que el enfoque propuesto es capaz de ofrecer diseños de instalaciones que satisfacen las preferencias del diseñador y preservan la diversidad de soluciones durante todo el proceso de diseño. Los resultados experimentales muestran que el método proporciona soluciones satisfactorias que no se repiten y es más rápido que el enfoque con el que ha sido comparado. Además, el nuevo enfoque produce menos fatiga y reduce la sobrecarga del diseñador. Así, el nuevo enfoque híbrido mejora los resultados que se obtienen por el método anterior.

Incorporar el conocimiento experto aumenta la calidad y aplicabilidad de los diseños de las soluciones que se obtienen sólo mediante la optimización matemática, ya que permite tener en cuenta las preferencias cualitativas del diseñador, lo cual es imposible en otros enfoques. Sin embargo, el método adolece de requerir la participación continua del diseñador en la

evaluación de soluciones durante el proceso, por lo que encontrar una manera de acortar el proceso sería beneficioso. Además, acortando el proceso computacional permitiría al diseñador para incorporar requisitos cualitativos más complejos, lo que aumentaría las posibles aplicaciones del método.

En el futuro, podría ser interesante utilizar más de un diseñador en el enfoque. Esto podría enriquecer los diseños de plantas mediante la incorporación de los conocimientos y la experiencia de varios expertos humanos. Además, para ayudar al diseñador en el proceso de evaluación, un enfoque que incluya un sistema de aprendizaje en el algoritmo también se podría investigar.

6. Bibliografía.

- Aiello, G., Enea, M., & Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and electre method. *Robotics and Computer-Integrated Manufacturing*, 22, 447–455.
- Aiello, G., Scalia, G. L., & Enea, M. (2012). A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications*, 39(12), 10352–10358.
- 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.
- Armour, G. C., & Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9, 294–309.
- Babbar-Sebens, M., & Minsker, B. S. (2012). Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design. *Applied Soft Computing*, 12(1), 182–195.
- Bezdek, J. C., Ehrlich, R., & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10, 192–203.
- Brintup, A. M., Ramsden, J., & Tiwari, A. (2007). An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Computers in Industry*, 58, 279–291.
- Brintup, A. M., Takagi, H., Tiwari, A., & Ramsden, J. (2006). Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems. *Journal of Biological Physics and Chemistry*, 6, 137–146.
- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems (Ph.D. thesis), University of Michigan, Ann Arbor: University Microfilms No. 76-9381.
- Dira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2), 255–267.
- García-Hernández, L., Arauzo-Azofra, A., Salas-Morera, L., Pierreval, H., & Corchado, E. (2013). Recycling plants layout design by means of an interactive genetic algorithm. *Intelligent Automation and Soft Computing*, 19(3), 457–468.
- García-Hernández, L., Arauzo-Azofra, A., Pierreval, H., & Salas-Morera, L. (2009). Encoding structures and operators used in facility layout problems with genetic algorithms. In: *Intelligent Systems Design and Applications*, 2009. ISDA '09. Ninth International Conference on, (pp. 43–48).
- 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.
- Goldberg, D. E. (1989). *Genetic algorithms in Search, optimization, and machine learning*. Addison-Wesley Professional.
- Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1), 1–13.
- Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. In L. J. Eshelman (Ed.), *ICGA* (pp. 24–31). Morgan Kaufmann.
- Izadinia, N., Eshghi, K., & Salmani, M. H. (2014). A robust model for multi-floor layout problem. *Computers & Industrial Engineering*, 78(0), 127–134.
- Kim, J. Y., & Kim, Y. D. (1995). Graph theoretic heuristics for unequal-sized facility layout problems. *Omega*, 23(4), 391–401.
- Komarudin & Wong, K. Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research*, 202(3), 730–746.
- Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operational Research*, 29(3), 229–251.
- Mahfoud, S. W. (1992). Crowding and preselection revisited. *Parallel Problem Solving from Nature*, 27–36.
- Mahfoud, S.W. (1995). *Niching Methods for Genetic Algorithms* (Ph.D. thesis), Urbana, IL, USA: University of Illinois at Urbana-Champaign.
- McKendall, A. R., Jr., & Jaramillo, J. R. (2006). A tabu search heuristic for the dynamic space allocation problem. *Computers and Operations Research*, 33(3), 768–789.
- Michalewicz, Z., Dasgupta, D., Riche, R. G. L., & Schoenauer, M. (1996). Evolutionary algorithms for constrained engineering problems. *Evolutionary Computation*, 4, 1–32.
- 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., Ertogral, K., & Türkbey, O. (2010). A simulated annealing heuristic for the dynamic layout problem with budget constraint. *Computers and Industrial Engineering*, 59(2), 308–313.
- 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.
- Solimanpur, M., & Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers and 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. In *Proceedings of the fourth international conference on genetic algorithms* (pp. 69–76). Morgan Kaufmann.
- Tompkins, J., White, J., Bozer, Y., & Tanchoco, J. (2010). *Facilities planning* (4rd ed. edition). New York: Wiley.
- Tong, X. (1991). *SECOT: A Sequential Construction Technique For Facility Design*. Doctoral

Dissertation, University of Pittsburg.

Yu, E., & Suganthan, P. (2010). Ensemble of niching algorithms. *Information Sciences*, 180(15), 2815–2833.

