

# Universidad de Córdoba

– Programa de Doctorado de Computación  
Avanzada, Energía y Plasmas –



– TESIS DOCTORAL –

---

## **Evolutionary Computation Strategies applied to the UA-FLP**

---

Estrategias de computación evolutiva aplicadas al UA-FLP

---

Autor:

**Juan María Palomo Romero**

Directores:

Prof. Dr. Lorenzo Salas Morera

Prof. Dra. Laura García Hernández

Córdoba, noviembre de 2019

TITULO: *Estrategias de computación evolutiva aplicadas al UA-FLP*

AUTOR: *Juan María Palomo Romero*

---

© Edita: UCOPress. 2020  
Campus de Rabanales  
Ctra. Nacional IV, Km. 396 A  
14071 Córdoba

[https://www.uco.es/ucopress/index.php/es/  
ucopress@uco.es](https://www.uco.es/ucopress/index.php/es/ucopress@uco.es)

---





**TÍTULO DE LA TESIS: Evolutionary Computation Strategies applied to the UA-FLP**

**DOCTORANDO/A: Juan María Palomo Romero**

### **INFORME RAZONADO DE LOS DIRECTORES DE LA TESIS**

Juan María Palomo Romero ha realizado su tesis doctoral dentro de la línea *Técnicas Avanzadas de Control, Modelado, Simulación y Optimización en La Ingeniería* del programa de doctorado de *Computación Avanzada, Energía y Plasma* de la UCO, bajo la dirección de los doctores Lorenzo Salas Morera y Laura García Hernández.

El plan de investigación inicial del doctorando ha sido adaptado a cuatro objetivos principales. Los tres primeros se corresponden con las labores de estudio profundo de la literatura, análisis y diseño de diferentes técnicas para la resolución de los problemas UA-FLP. El cuarto objetivo consiste en el desarrollo, implementación, pruebas y aplicación práctica de la investigación realizada en los tres primeros objetivos.

Las labores de investigación realizadas se han visto respaldadas por resultados muy satisfactorios:

- 1 Publicación en una revista JCR del primer cuartil (publicado en junio de 2015): L. García-Hernández, J. M. Palomo-Romero, L. Salas-Morera, A. Arauzo-Azofra, and H. Pierreval. "A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem". *Expert Syst. Appl.*, vol. 42, no. 10, pp. 4697–4708, Jun. 2015.
- 2 Comunicación en un congreso internacional (CIDIP 2015): García Hernández, Laura; Palomo Romero, Juan María; Salas Morera,

Lorenzo; Arauzo Azofra, Antonio; García Hernández, Jose A. "A Novel Hybrid Evolutionary Approach for Capturing the Expert Knowledge into the Unequal Area Facility Layout Problem". *19th International Congress on Project Management and Engineering*. 2015.

- 3 "Premio Joven Investigador" en el anterior congreso internacional.
- 4 Comunicación en un congreso internacional (CIDIP 2016): Palomo Romero, Juan María; Salas Morera, Lorenzo; García Hernández, Laura. "A novel parallel evolutionary approach for the facility layout problem". *20th International Congress on Project Management and Engineering*. 2016.
- 5 Publicación en una revista JCR del primer cuartil (publicado en febrero de 2017): J. M. Palomo-Romero, L. Salas-Morera, and L. García-Hernández. "An island model genetic algorithm for unequal area facility layout problems". *Expert Syst. Appl.*, vol. 68, pp. 151–162, 2017.

Además, cabe resaltar que, a fecha de noviembre de 2019, los dos artículos han sido citados un total de 51 ocasiones (datos de Scopus).

Por otro lado, se ha diseñado un novedoso algoritmo híbrido genético paralelo que, mediante una estrategia que promueve la diversificación y mantiene la calidad de las soluciones, explora un mayor espacio de búsqueda con un alto grado de diversidad. Este algoritmo ha sido implementado y se han realizado las experimentaciones oportunas, obteniendo de ellas resultados muy interesantes. En la actualidad se está finalizando la redacción del artículo (tercero relacionado con la tesis).

Además, el doctorando ha participado en otras líneas de investigación, como dirección de proyectos mediante software libre, *m-Health*, y educación universitaria en el ámbito de la ingeniería, en los cuales se han obtenido, como resultado dos papers publicados en revistas JCR:

- L. Salas-Morera, A. Arauzo-Azofra, L. García-Hernández, J. M. Palomo-Romero, and J. L. Ayuso-Muñoz, "New Approach to the Distribution of Project Completion Time in PERT Networks". *J. Constr. Eng. Manag.*, vol. 144, no. 10, 2018.

- L. Salas-Morera, M. a A. Cejas-Molina, J. M. Palomo-Romero, and L. García-Hernández, "Factors Affecting Engineering Students Dropout: A Case Study". *Int. J. Eng. Educ.*, vol. 35, no. 1, pp. 156–167, 2019.

Igualmente, cabe resaltar que, en este último año, el doctorando, en colaboración con el grupo de investigación, ha publicado un capítulo para la Colección Editorial "Dirección e Ingeniería de Proyectos" en la editorial Springer, ISBN: 978-84-9828-667-0. El capítulo del libro se titula "Aplicación del software libre PPC-Project a la docencia práctica de gestión de proyectos en Ingeniería Informática".

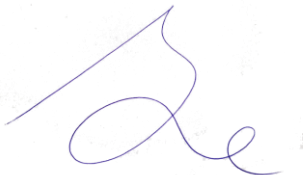
El cumplimiento del plan de investigación ha sido muy completo, incluso arrojando mayores resultados en forma de publicaciones y comunicaciones, de los que, en un principio, cabía esperar.

Por último, con respecto al plan de formación, el doctorando ha completado todas las actividades formativas obligatorias. Además, está participando en diferentes actividades formativas optativas, como la asistencia a congresos internacionales, cursos dirigidos a doctorandos, revisión de artículos en revistas internacionales de impacto (JCR) y, miembro del Comité de Programa de un congreso internacional.

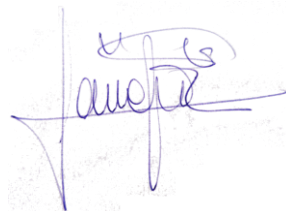
Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 4 de noviembre de 2019

Firma de los directores



Fdo.: Lorenzo Salas Morera



Fdo.: Laura García Hernández



---

## RESUMEN

---

En la presente tesis doctoral se desarrollan dos aproximaciones distintas al problema de distribución en planta de áreas desiguales (*UA-FLP*).

En primer lugar, se trata de incorporar el conocimiento del diseñador experto a los algoritmos clásicos de optimización, de forma que, además de buscar buenas soluciones desde el punto de vista cuantitativo, por ejemplo minimizando el flujo de materiales, se introduzca la posibilidad de que el diseñador aporte su experiencia y preferencias personales. Para facilitar la intervención humana en el proceso de búsqueda de soluciones, se ha utilizado un procedimiento de *clustering*, el cual permite clasificar las soluciones subyacentes en el conjunto de búsqueda, de forma que se presente al diseñador un número suficientemente representativo y, a la vez, evitándole una fatiga innecesaria. Además, en esta primera propuesta se han implementado dos técnicas de *niching*, denominadas *Deterministic Crowding* y *Restricted Tournament Selection*. Estas técnicas tienen la capacidad de mantener ciertas propiedades dentro de la población de soluciones, preservar múltiples nichos con soluciones cercanas a los óptimos locales, y reducir la probabilidad de quedar atrapado



en ellos. De esta manera el algoritmo se enfoca simultáneamente en más de una región (nicho) en el espacio de búsqueda, lo cual es esencial para descubrir varios óptimos en una sola ejecución.

Por otro lado, en la segunda aproximación al problema, se ha implementado una estrategia evolutiva paralela, muy útil para los problemas de alta complejidad en los que el tiempo de ejecución con un enfoque evolutivo secuencial es prohibitivo. La propuesta desarrollada, denominada *IMGGA*, está basada en un algoritmo genético paralelo de grano grueso con múltiples poblaciones o *islas*. Este enfoque se caracteriza por evolucionar varias subpoblaciones independientemente, entre las que se intercambian individuos, haciendo posible explorar diferentes regiones del espacio de búsqueda, al mismo tiempo que se mantiene la diversidad de la población, permitiendo la obtención de buenas y diversas soluciones.

Con ambas propuestas se han realizado experimentos que han arrojado resultados muy satisfactorios, encontrando buenas soluciones para un conjunto de problemas bien conocidos en la bibliografía. Estos buenos resultados han permitido la publicación de dos artículos indexados en el primer decil del ranking *JCR (Journal Citation Reports)*.

---

## ABSTRACT

---

The present doctoral thesis develops two different approaches to the *Unequal Area Facility Layout Problem (UA-FLP)*.

The first approach encompasses the designer's knowledge on classic optimization of algorithms in pursuance of good quantitative solutions (e.g. minimizing the materials flow) and also opens the possibility to include the contribution of the designer by means of his expertise and personal preferences. A *clustering* procedure has been used to facilitate human intervention in the process of finding solutions. This allows the underlying solutions to be classified in the search in order to present the designer with sufficiently representative solutions and, at the same time, avoiding unnecessary fatigue. In addition, two *niching* techniques have been implemented, called *Deterministic Crowding* and *Restricted Tournament Selection*. These techniques have the ability to maintain certain properties within the solutions space, preserve multiple niches with solutions close by local optimums, and reduce the probability of being trapped in them. In this way, the algorithm focuses simultaneously on more than one region (niche) in the search space, which is essential to discover several optimums in a single execution.

The second approach to the problem comprises the implementation of a parallel evolutionary strategy. This method is useful for problems of high complexity in which the execution time using a sequential evolutionary approach is prohibitive. The proposal developed, called *IMGA (Island Model Genetic Algorithm)*, is based on a parallel genetic algorithm of multiple-population coarse-grained. This is characterized by evolving several subpopulations independently among which individuals are exchanged. Different regions of the search space can be explored while the diversity of the population is maintained. Satisfactory and diverse solutions have been obtained as a result of this method.

Experiments with both proposals have been carried out with satisfactory results, providing good solutions for a set of problems well known in the literature. These results were already published in two papers indexed in the first decile of the *JCR (Journal Citation Reports)* ranking.

---

## AGRADECIMIENTOS

---

Con la presente tesis doctoral culmino una de las etapas más intensas y gratificantes de mi vida académica. Han sido años de esfuerzo, sacrificio y superación que, sin duda, han merecido la pena. No solo por el aprendizaje, tanto personal como académico, sino por la cantidad de experiencias vividas en un lugar tan fascinante como es la Universidad.

Quisiera comenzar dando las gracias a la Universidad de Córdoba, la cual ha sido un importante pilar en mi formación, tanto para el mundo profesional como investigador. A todos los profesores que hacen grande a esta institución. En especial, plasmar mi agradecimiento a tres de ellos. A Cristina María Gámez, por la confianza que depositó en mí desde el primer momento, y por sus innumerables y sabios consejos. A Nicolás Luis Fernández, por transmitir sus conocimientos con esa gran pasión que le caracteriza. Y por último, a Antonio Araúzo, por su inestimable ayuda como compañero en el grupo de investigación.

A mis queridos directores Lorenzo Salas y Laura García. Agradeceles enormemente la dedicación y el apoyo incondicional durante este complejo, pero enriquecedor, camino. Por la comprensión en

todo momento y las facilidades que me han brindado para poder compaginar el trabajo investigador con mi vida profesional y personal. Agradecerles su confianza, y en haberme dado la oportunidad de participar en la vida docente universitaria, labores que he realizado con orgullo y vocación. A Laura, por sus consejos, apoyo y orientación en las labores de investigación y redacción de artículos. A Lorenzo, por su paciencia infinita, su trabajo incansable y por contar siempre conmigo. *Continuamos para bingo.*

A mi familia, el apoyo más importante cuando más lo he necesitado. A mis padres, Juan y Encarnación, que con su esfuerzo lo han puesto todo a mi disposición para poder formarme intelectual y humanamente. Agradecido porque me inculcaran sus valores, la importancia de la educación y la constancia para conseguir mis objetivos y metas. Y a la mejor hermana del mundo, Alicia, siempre dispuesta a escucharme, ayudarme y alegrarme el día. La culminación de esta tesis doctoral es tanto mérito mío como vuestro. Gracias por vuestro amor. Gracias de todo corazón.

A todos ellos, mil gracias.

*Caminante, no hay camino,  
se hace camino al andar.*

— **Antonio Machado**

*El logro de un objetivo  
debe ser el punto de partida de otro.*

— **Alexander Graham Bell**

*El mayor enemigo del conocimiento  
no es la ignorancia,  
sino la ilusión del conocimiento.*

— **Stephen Hawking**



*A mi familia*





---

## ÍNDICE DE CONTENIDOS

---

Índice de figuras	III
Índice de tablas	V
Acrónimos	VII
1 Introducción	1
1.1 Definición del problema de distribución en planta . .	3
1.2 Enfoques para la resolución del Facility Layout Problem	7
1.3 El diseñador experto . . . . .	14
1.4 La diversidad de las soluciones . . . . .	17
2 Objetivos	19
3 Metodología	21
3.1 Algoritmos genéticos . . . . .	22
3.2 Problema de distribución en planta de áreas desiguales	26
3.2.1 Formulación del problema . . . . .	29
3.2.2 Codificación de las plantas . . . . .	33

3.3	Enfoques . . . . .	34
3.3.1	Técnicas de <i>niching</i> . . . . .	35
3.3.2	Algoritmos genéticos paralelos . . . . .	40
3.4	Conjunto de problemas ( <i>Datasets</i> ) . . . . .	42
4	Resultados	53
5	Conclusiones y futuras líneas de investigación	57
	Bibliografía	65
	ANEXOS	107
A	A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem	109
B	An island model genetic algorithm for unequal area facility layout problems	125

---

## ÍNDICE DE FIGURAS

---

Figura 1.1	Ejemplo de solución de problema <i>FLP</i> ( <i>Carton packs</i> ) con 11 instalaciones . . . . .	6
Figura 1.2	Ejemplo de solución de problema <i>DFLP</i> con 6 instalaciones y 3 periodos . . . . .	7
Figura 1.3	Representación discreta de una planta . . . . .	8
Figura 1.4	Representación de un problema <i>SRFLP</i> con 6 instalaciones . . . . .	9
Figura 3.1	Diagrama de flujo del algoritmo genético . . . . .	23
Figura 3.2	Ejemplo de codificación de un individuo . . . . .	24
Figura 3.3	Ejemplo de operador de cruce en un algoritmo genético . . . . .	25
Figura 3.4	Ejemplo de operador de mutación en un algoritmo genético . . . . .	26
Figura 3.5	Formas de las instalaciones . . . . .	27
Figura 3.6	Instalación con la misma área y diferentes <i>aspect ratio</i> . . . . .	27
Figura 3.7	Representación de una planta con <i>STS</i> . . . . .	29
Figura 3.8	Representación de una planta con <i>FBS</i> . . . . .	29
Figura 3.9	Tipos de distancias . . . . .	33

Figura 3.10	Ejemplo de codificación de una planta . . . .	34
Figura 3.11	Diagrama de flujo del algoritmo <i>Deterministic Crowding</i> . . . . .	38
Figura 3.12	Diagrama de flujo del algoritmo <i>Restricted Tournament Selection</i> . . . . .	39

---

## ÍNDICE DE TABLAS

---

Tabla 1.1	Flujo de materiales entre instalaciones (problema <i>Carton packs</i> ) . . . . .	7
Tabla 3.1	Características de los problemas para la experimentación . . . . .	44
Tabla 3.2	Problema <i>O7</i> . Flujo de materiales entre departamentos. . . . .	45
Tabla 3.3	Problema <i>O8</i> . Flujo de materiales entre departamentos. . . . .	45
Tabla 3.4	Problema <i>O9</i> . Flujo de materiales entre departamentos. . . . .	45
Tabla 3.5	Problema <i>vC10</i> . Flujo de materiales entre departamentos. . . . .	46
Tabla 3.6	Problema <i>Ba12</i> . Flujo de materiales entre departamentos. . . . .	46
Tabla 3.7	Problema <i>MB12</i> . Flujo de materiales entre departamentos. . . . .	47
Tabla 3.8	Problema <i>Slaughterhouse</i> . Flujo de materiales entre departamentos. . . . .	47

Tabla 3.9	Problema <i>Ba14</i> . Flujo de materiales entre departamentos. . . . .	48
Tabla 3.10	Problema <i>AB20</i> . Flujo de materiales entre departamentos. . . . .	49
Tabla 3.11	Problema <i>Aiello20</i> . Flujo de materiales entre departamentos. . . . .	50
Tabla 3.12	Problema <i>SC30</i> . Flujo de materiales entre departamentos. . . . .	51
Tabla 3.13	Problema <i>SC35</i> . Flujo de materiales entre departamentos. . . . .	52

---

## ACRÓNIMOS

---

**ACO** *Ant Colony Optimization.*

**AR** *Aspect Ratio.*

**BB** *Branch and Bound.*

**BRKGA** *Biased Random-Key Genetic Algorithm.*

**CX** *Cycle crossover operator.*

**DC** *Deterministic Crowding.*

**DFLP** *Dynamic Facility Layout Problem.*

**DRLP** *Double Row Layout Problem.*

**FBS** *Flexible Bay Structure.*

**FLP** *Facility Layout Problem.*

**GA** *Genetic Algorithm.*

**IMGA** *Island Model Genetic Algorithm.*

**LP** *Linear Programming.*

**MFLP** *Multi-Floor Layout Problem.*

**MIP** *Mixed-Integer Programming.*

**OX** *Order crossover.*



**PBX** *Position based crossover operator.*

**PGA** *Parallel Genetic Algorithm.*

**PMX** *Partially-mapped crossover.*

**PSO** *Particle Swarm Optimization.*

**QAP** *Quadratic Assignment Problem.*

**RTS** *Restricted Tournament Selection.*

**SA** *Simulated Annealing.*

**SFLP** *Static Facility Layout Problem.*

**SRFLP** *Single Row Facility Layout Problem.*

**STS** *Slicing Tree Structure.*

**UA-FLP** *Unequal Area Facility Layout Problem.*

---

## INTRODUCCIÓN

---

En un mercado dominado por la competencia y la demanda, que exige gran variedad de productos con ciclos de vida cada vez más cortos, las empresas manufactureras tratan de adaptar sus procesos de fabricación implementando sistemas flexibles que permitan hacer frente a la incertidumbre que esta situación genera. Lograr la eficiencia y la flexibilidad demandada pasa necesariamente por una correcta ordenación de los medios productivos que permita, no sólo hacer frente con éxito a las situaciones actuales, sino también, a posibles escenarios futuros [1].

Debido a la competencia existente en la industria, se debe analizar todos los posibles caminos hacia la reducción del coste. En muchas industrias, es casi imposible asegurar una ventaja frente a la competencia, ya que en cualquiera de los factores principales, como son los materiales, maquinarias, métodos de distribución y salarios, se ha alcanzado una estandarización que impide una

ventaja significativa. Por lo tanto, la dirección debe asegurar, a través de los detalles que afecten al precio de coste, sus márgenes de beneficio. Uno de estos importantes detalles es la distribución en planta.

El problema de distribución en planta, comúnmente conocido como Facility Layout Problem (*FLP*), trata de determinar la mejor ubicación para los departamentos o instalaciones que componen una planta industrial [2]. Para ello, ha de tenerse en consideración ciertos criterios y restricciones. Así, según Tompkins, White, Bozer y otros [3], una buena distribución de las instalaciones que componen la planta industrial puede reducir entre un 20% y un 50% los costes totales de producción. Una instalación es una entidad que facilita el desempeño de cualquier trabajo. Puede ser una máquina, un departamento, un almacén, un matadero, un centro de trabajo, una fábrica agroalimentaria, un hospital, un taller mecánico, etc.

Según plantea Hicks [4], la responsabilidad del ingeniero industrial es la de diseñar una planta de producción que elabore el producto especificado a la tasa estipulada de producción a un coste mínimo. La forma en la que se distribuyen las instalaciones en una planta industrial afecta directamente a los costes de producción, eficiencia de la planta y productividad [5]. Es por ello por lo que en la década de 1950 comienzan los intentos por establecer una metodología que permitiera afrontar uno de los principales problemas que aún hoy en día sigue siendo estudiado ampliamente en el ámbito industrial: el problema de la distribución en planta [6].

## 1.1 DEFINICIÓN DEL PROBLEMA DE DISTRIBUCIÓN EN PLANTA

A lo largo de los años, diferentes autores han formulado definiciones del problema que nos ocupa. Debido a la variedad de consideraciones encontradas en la literatura, los investigadores no están de acuerdo con una definición común del *FLP*.

Koopmans y Beckmann [7], unos de los primeros en considerar esta clase de problemas, definen *FLP* como un problema industrial en el que el objetivo es organizar las instalaciones para minimizar el coste de transporte de materiales entre ellas.

Muther [6], otro de los primeros estudiosos del *FLP*, plantea que la distribución en planta implica la coordinación física de los elementos industriales. *Esta ordenación, ya practicada o en proyecto, incluye, tanto los espacios necesarios para el movimiento del material, almacenamiento, trabajadores indirectos y todas las otras actividades o servicios, como el equipo de trabajo y el personal de taller.* El objetivo es, según Muther [6], obtener una ordenación de las áreas de trabajo y del equipo, que sea la más económica para el trabajo, al mismo tiempo que la más segura y satisfactoria para los empleados, de modo que sea posible fabricar el producto a un coste suficientemente reducido como para poder venderlo con un buen margen de beneficio en un mercado de competencia. Según esta filosofía de diseño, se pretende conseguir algo más que la mera disposición de la maquinaria y los equipos sobre una superficie res-

tringida, siendo el fin principal la obtención de una organización óptima del trabajo, tanto desde el punto de vista de los costes de producción como de la ergonomía y la seguridad [8].

Por su parte, Moore [9] define la distribución en planta óptima, como aquella que proporciona la máxima satisfacción a todas las partes que se ven involucradas en el proceso de implantación. En este punto cabe hacer referencia a la complejidad del problema que nos atañe, el cual generalmente no dispone de una única solución (sub)óptima, y que el número de variables implicadas puede ser tan grande que una aproximación intuitiva al problema es desaconsejable.

Tate y Smith [10] y Shayan y Chittilappilly [11] definen el *FLP* como un grupo de problemas de optimización consistentes en la partición de una región plana de dimensiones conocidas en departamentos de área conocida, de tal manera que se minimice el coste asociado con las interacciones previstas entre dichos departamentos. Los trabajos de Meller, Narayanan y Vance [12] y Azadivar y Wang [13] apuntan en la misma dirección, considerando que el objetivo es encontrar una disposición no superpuesta de  $n$  instalaciones rectangulares en un plano rectangular, minimizando la distancia entre ellas.

Meller y Gau [14] añaden restricciones al problema, expresando que el objetivo del *FLP* es determinar la distribución más eficiente de un número de departamentos indivisibles minimizando los costes

del transporte de materiales dentro de la planta, considerando dos grupos de restricciones: a) los requerimientos de área, y b) las restricciones de localización de los departamentos.

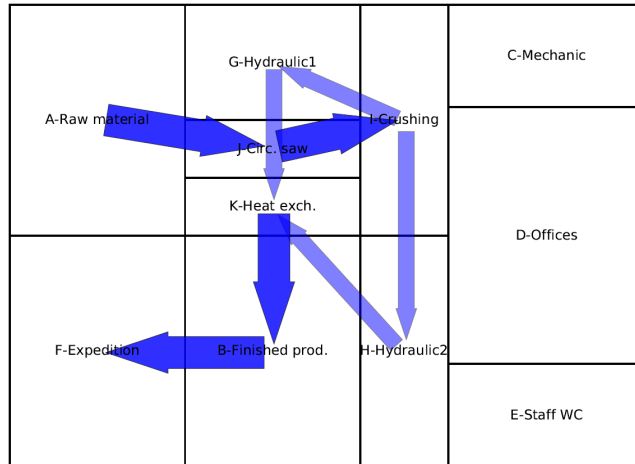
Cabe añadir que el *FLP* puede ser enfocado para la optimización de un objetivo o bien de varios objetivos, lo que se conoce como *optimización multiobjetivo* [15]-[20].

Aunque, como se ha indicado anteriormente, no existe una definición formalmente consensuada, el *FLP* puede definirse como un problema que consiste en ordenar instalaciones (*facilities*) en una región plana de dimensiones conocidas, respetando una serie de restricciones, cuya meta es obtener el diseño de planta óptimo en función de uno o varios objetivos.

Por último, es necesario añadir que existen dos categorías de *FLP* en función de la evolución de su diseño. El problema de distribución en planta estático (Static Facility Layout Problem (*SFLP*)), y el Dynamic Facility Layout Problem (*DFLP*).

El primero de ellos es el que identificamos generalmente como *FLP*, sin necesidad de especificar que es *estático*, ya que es el tipo de problema estudiado de una manera más amplia a lo largo de la bibliografía, desde los inicios [21]-[25] hasta la actualidad [26]-[30].

Generalmente, este problema consiste en el diseño de una planta con el objetivo de minimizar el coste de manejo de materiales,



**Figura 1.1:** Ejemplo de solución de problema *FLP* (*Carton packs*) con 11 instalaciones

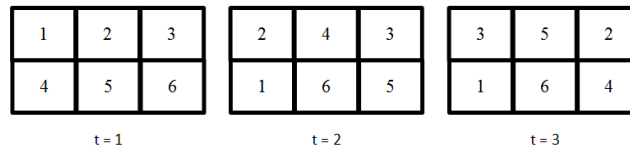
en base al flujo de materiales entre las instalaciones. En la figura 1.1 se muestra un ejemplo de solución de diseño de *FLP* (*estático* o *SFLP*) de una planta de reciclaje de envases de cartón (*carton packs*). Este caso real, descrito por García-Hernández, Arauzo-Azofra, Salas-Morera y otros [31], acompaña una tabla del flujo de materiales entre instalaciones (ver tabla 1.1). Como puede observarse en la solución del diseño, las instalaciones con un flujo mayor de materiales han sido situadas a poca distancia, ya que cuanto mayor sea la distancia entre ellas, mayor será el coste de manejo de materiales, y por ende, de toda la planta.

Y en segundo lugar, el problema de diseño dinámico de instalaciones (*DFLP*), que se caracteriza en que el diseño de la planta va cambiando por periodos (ver figura 1.2), debido, generalmente, a que la ruta que efectúa el flujo de material cambia con el tiempo. Este tipo de problemas es utilizado para entornos volátiles, en los

**Tabla 1.1:** Flujo de materiales entre instalaciones (problema *Carton packs*)

Desde	Hasta	Flujo (%)
A-Raw material	J-Circular saw	100
J-Circular saw	I-Crushing machine	100
I-Crushing machine	G-Hydraulic press 1	50
I-Crushing machine	H-Hydraulic press 2	50
G-Hydraulic press 1	K-Heat exchanger	50
H-Hydraulic press 2	K-Heat exchanger	50
K-Heat exchanger	B-Finished product w.	100
B-Finished product w.	F-Expedition	100

que la demanda no es estable, cambiando la producción en diferentes periodos. Aunque no ha sido un problema muy estudiado en el pasado, recientemente está aumentando su aceptación en la comunidad investigadora [32]-[40].

**Figura 1.2:** Ejemplo de solución de problema *DFLP* con 6 instalaciones y 3 periodos

## 1.2 ENFOQUES PARA LA RESOLUCIÓN DEL FACILITY LAYOUT PROBLEM

Dependiendo de cómo se represente el diseño de la planta, discreta o continua, las formulaciones clásicas encontradas en la literatura conducen en su mayoría, respectivamente, a problemas de asignación cuadrática (Quadratic Assignment Problem (*QAP*)) o programación entera mixta (Mixed-Integer Programming (*MIP*)). En



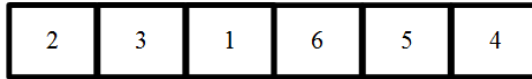
el caso de la representación discreta, la planta se divide en bloques rectangulares con la misma área y forma, y cada bloque se asigna a una instalación Wang, Hu y Ku [41]. Si las instalaciones tienen áreas desiguales, cada una de ellas puede abarcar varios bloques, pero siempre ocupando bloques completos, ya que son indivisibles (ver figura 1.3). En la representación continua todas las instalaciones están ubicadas en cualquier lugar dentro del plano, sin solaparse entre ellas. En este tipo de representación no se divide la planta por bloques, ya que las instalaciones pueden tener tamaños y formas diversas (ver figura 1.1).

1	1	4
2	4	4
3	3	5
3	3	5

Figura 1.3: Representación discreta de una planta

La primera formulación de instalaciones de igual tamaño fue realizada por Koopmans y Beckmann [7] para modelar la ubicación geográfica de plantas interdependientes como *QAP*. La representación discreta es utilizada, generalmente, para problemas de planta de fila única (Single Row Facility Layout Problem (*SRFLP*)), como bien puede observarse en los trabajos de Diponegoro y Sarker [42], Palubeckis [43], Solimanpur, Vrat y Shankar [44] y Wang

y Sarker [45], y en el ejemplo de la figura 1.4; entre otros. Esta representación también es muy habitual encontrarla en propuestas para resolver el *DFLP* [46], [47]. Sin embargo, la suposición de un tamaño igual de para todas las instalaciones limita en gran medida su aplicación [48], conduciendo a que muchos autores consideren la representación continua más relevante [49].



**Figura 1.4:** Representación de un problema *SRFLP* con 6 instalaciones

Montreuil [50] fue uno de los primeros en presentar una formulación *MIP* para el *FLP* utilizando una representación continua. Cuando el número de instalaciones es pequeño, como por ejemplo, en las propuestas de Liu y Meller [51] y de Meller, Narayanan y Vance [12], el problema puede ser resuelto de manera óptima. Otros autores han empleado *MIP* para resolver el *FLP* como Bozer y Wang [52]; Castillo y Westerlund [53]; Montreuil [50]; Sherali, Fraticelli y Meller [54]; Meller, Chen y Sherali [55]; entre otros. Además, en numerosos trabajos de problemas con múltiples plantas (Multi-Floor Layout Problem (*MFLP*)) ha sido aplicada la formulación *MIP* [56]-[59]. Desafortunadamente, el modelo *MIP* generalmente necesita una gran cantidad de variables binarias para satisfacer ciertas restricciones, lo que resulta de poca eficiencia para resolver problemas a gran escala [60].

Uno de los métodos exactos más utilizados es el conocido como ramificación y poda (Branch and Bound (*BB*)). Bazaraa [61]

diseña un método *BB* para optimizar el problema resultante. Como resultado, el algoritmo proporciona diseños óptimos y subóptimos al ingeniero de diseño para su consideración y modificación. Por su parte, Kim y Kim [62] consideran en su trabajo el problema de ubicar los puntos de entrada y salida (E/S) de cada departamento para un diseño dado. El objetivo del problema es minimizar la distancia total de los flujos de material entre los puntos de E/S. En su enfoque, utiliza un modelo de programación lineal (Linear Programming (*LP*)) para minimizar la distancia total de flujos para obtener un límite inferior. A continuación, aplican un algoritmo *BB* para encontrar una ubicación óptima de los puntos de E/S. La propuesta de Kohara, Yamamoto y Suzuki [63] introduce una condición que elimina automáticamente los diseños que son simétricos, ya que tienen el mismo valor objetivo que el diseño original, y aplica esta técnica a un algoritmo basado en métodos *BB*. La propuesta de Karatas y Yakıcı [64] trata de resolver el *FLP* multiobjetivo en una aplicación real como es la localización de estaciones de servicios de emergencia. Esta propuesta se basa principalmente en una combinación de *BB* con técnicas de programación de objetivos iterativos. Por su parte, Rosenblatt [65], optimiza un *DFLP* de pequeño tamaño (6 instalaciones y 5 períodos) utilizando un algoritmo *BB*.

Sin embargo, cuando el número de instalaciones es mayor, es necesario hacer uso de métodos subóptimos para abordar el problema. En este contexto, los métodos metaheurísticos han sido muy empleados. Por un lado, los métodos de búsqueda global como

la búsqueda tabú [66], [67], un método presentado por Glover [68]; y el enfriamiento simulado (Simulated Annealing (*SA*)) [69], [70]. Por otro lado, los enfoques evolutivos como los sistemas de hormigas (Ant Colony Optimization (*ACO*)) [71], [72], y los algoritmos genéticos (Genetic Algorithm (*GA*)) [73], entre otros.

Chiang y Kouvelis [74] desarrollaron un algoritmo de búsqueda tabú para resolver el *FLP*. Utilizaron un vecindario basado en el intercambio de dos localizaciones de instalaciones e incluyeron una lista tabú de tamaño dinámico y memoria a largo plazo. Por su parte, Samarghandi y Eshghi [75] proponen un nuevo algoritmo basado en la búsqueda tabú para el *SRFLP*, en el que utilizan una técnica llamada memoria adaptativa para la diversificación e intensificación. Kothari y Ghosh [76] presentan dos implementaciones de búsqueda tabú: una basada en la propuesta anteriormente mencionada [75], y otra que usa un vecindario de inserción [77]. También presentan técnicas para acelerar significativamente la búsqueda de dos vecindarios, arrojando mejores resultados para un gran número de problemas *SRFLP* estudiados previamente en la bibliografía. Este tipo de problema, muy estudiado en la literatura [78], tiene aplicaciones prácticas como la localización de instalaciones a lo largo de un corredor [79], la disposición de libros en un estante [80], o la disposición de máquinas en línea recta recorrida por un vehículo guiado automatizado [81], [82], entre otras.

El *SA* es un método basado en la simulación de *MonteCarlo*, que permite resolver problemas difíciles de optimización combinatoria.

Se inspira en el proceso del recocido del acero, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. Este proceso físico fue simulado por primera vez por Metropolis, Rosenbluth, Rosenbluth y otros [83]. La primera aplicación de este método a problemas de optimización fue llevada a cabo por Kirkpatrick, Gelatt y Vecchi [84]. Se han publicado numerosos enfoques para la resolución del *FLP* mediante el uso de algoritmos *SA* [85]-[90], como por ejemplo el trabajo de Chwif, Barretto y Moscato [91], en el que aborda algunos aspectos prácticos que hasta entonces se habían explorado con poca frecuencia en la literatura, como las instalaciones con diferentes áreas, formas y orientaciones, así como las instalaciones fijas; o la novedosa versión de *SA* propuesta por Grobelny y Michalski [92], basada en patrones lingüísticos [93]-[95] y un enfoque de teoría difusa. Por otro lado, se publican enfoques de este algoritmo para la resolución del *FLP* de tipo multiobjetivo [96]-[98]. Por su parte, McKendall y Shang [99] presentan dos algoritmos basados en *SA*: una propuesta simple de *SA* aplicada al *DFLP*, y una segunda heurística basada en la primera que añade estrategias de *look-ahead/look-back* [99].

*ACO* es otra de las técnicas metaheurísticas aplicadas en la actualidad, propuesta por Dorigo y Gambardella [100], y basada en el comportamiento de las hormigas cuando están buscando un camino entre la colonia y una fuente de alimentos. Este método se ha aplicado ampliamente para resolver diferentes problemas de

optimización combinatoria [101]-[104]. En el caso que nos ocupa, los algoritmos *ACO* se han implementado con éxito para resolver problemas *FLP* formulados como *QAP* [82], [105]. *ACO* puede ser considerada como una técnica de optimización prometedora para resolver problemas complejos de optimización combinatoria como el *DFLP*, tal y como se aborda en los trabajos de Baykasoglu, Dereli y Sabuncu [106] y de Chen [107].

Por otro lado, en diferentes trabajos se han realizado propuestas basadas en la descomposición [108], [109], como en la investigación de Guan, Lin, Feng y otros [110], en el que se divide el problema del diseño de planta de doble fila *Double Row Layout Problem (DRLP)* en dos subproblemas. En el primer subproblema, se aplica una primera búsqueda local para optimizar las secuencias de instalaciones en doble fila. En el segundo subproblema, se aplica la técnica *Particle Swarm Optimization (PSO)* para optimizar los espacios libres ajustables entre instalaciones adyacentes bajo la condición de que las secuencias de las instalaciones sean fijas.

Otro de los enfoques prometedores con un sinnúmero de aplicaciones son los algoritmos genéricos (*GA*). Algoritmos de búsqueda global que generalmente se han utilizado para resolver problemas de optimización y búsqueda [111]-[113], y que se ha aplicado con éxito a otros campos tan diversos como los procesos industriales del acero [114], [115]; la industria energética [116], [117]; el sector nuclear [118], [119]; en el ámbito de la biomedicina tanto en el diagnóstico [120], [121] como en la predicción [122], [123]; y en

optimización del uso de combustible [124], [125]; entre muchos otros.

Con el objeto de aunar las bondades de diferentes técnicas, las propuestas híbridas para la resolución del *FLP* están proliferando en los últimos años, sobre todo las enfocadas a la resolución del *DFLP* [33], [34], [99], [107], [126], [127]. Por su lado, Guan y Lin [128] proponen un algoritmo híbrido basado en la búsqueda de vecindario variable y *ACO* para resolver el problema *SRFLP*. Para el Unequal Area Facility Layout Problem (*UA-FLP*) se han publicado diversas propuestas híbridas, como el *GA* combinado con una estrategia de descomposición a través de deconstrucciones y reconstrucciones de soluciones parciales presentado por Paes, Pessoa y Vidal [129]; o el enfoque de Gonçalves y Resende [130], que combina un algoritmo genético con claves aleatorias (Biased Random-Key Genetic Algorithm (*BRKGA*)), para determinar el orden de colocación y las dimensiones de cada instalación, y un modelo de programación lineal, para ajustar las soluciones. Por su parte, Ou-Yang y Utamima [131] proponen un algoritmo para la resolución del *SRFLP*, consistente en una hibridación de algoritmos de estimación de distribución, *PSO* y búsqueda tabú.

### 1.3 EL DISEÑADOR EXPERTO

Aunque el flujo de material entre las diferentes instalaciones que conforman la planta es el criterio más considerado a la hora de resolver el problema que nos atañe, existen otras características que

---

pueden tenerse en cuenta. Estas pueden ser de tipo cuantitativo o cualitativo. En el primer caso, podemos hablar de requerimientos de cercanía o proximidad entre dos instalaciones, satisfacer alguna restricción de forma, etc. En el segundo caso, hablamos de otro tipo de preferencias que pueden ser útiles o necesarias cuando realizamos el diseño de la planta industrial [132], como por ejemplo: deseo de una concreta ubicación de una instalación en particular, necesidad de evitar la localización para cierto departamento, la seguridad, el ruido, la flexibilidad, etc.

De acuerdo con Brintup, Ramsden y Tiwari [133], las características cualitativas son difíciles de considerar, ya que no pueden ser modeladas como una función clásica de optimización, pueden variar durante el proceso de diseño, son específicas para cada problema en particular, y pueden ser desconocidas a priori. Para solucionar esta problemática, se requiere el conocimiento del diseñador experto, incluyendo sus preferencias a la hora de buscar la mejor solución de diseño. Para abordar esta problemática se utiliza un método interactivo que se retroalimenta de la información suministrada del conocimiento del experto. Brintup, Takagi, Tiwari y otros [134] consideran que un método de resolución interactivo favorece la obtención de soluciones satisfactorias en el diseño de la planta. De acuerdo con García-Hernández, Pierreval, Salas-Morera y otros [135], la inclusión del conocimiento del diseñador experto en la propuesta ofrece una serie de ventajas adicionales como la no necesidad de especificar por adelantado toda la información



deseable en el diseño de la planta, ofrecer al diseñador experto la capacidad de aprender sobre sus propias preferencias, estimular la creatividad de los diseñadores [136], o seleccionar la mejor solución de compromiso cuando existe un conflicto entre objetivos [137].

Existen algunos estudios en los que se ha incorporado el conocimiento del diseñador experto en el proceso de búsqueda de soluciones en el *FLP*. Así, Quiroz, Louis, Banerjee y otros [138] sugieren un enfoque colaborativo aplicado al diseño de planta. Por su parte, Zhao, Li, Yang y otros [139] propone un enfoque que consiste en un algoritmo inmune basado en la optimización cooperativa humano-computadora. Los algoritmos basados en inmunidad [140], originados en la década de 1990, permiten incorporar la memoria inmune, la extracción y la inoculación de anticuerpos eficientes, así como el mecanismo de inhibición y promoción de anticuerpos en los sistemas inmunes biológicos. Por todo ello, los enfoques basados en inmunidad son útiles para prevenir la convergencia prematura y, al mismo tiempo, acelerar la tasa de convergencia [141]. Sin embargo, este enfoque no permite al diseñador cambiar sus preferencias iniciales por otras nuevas que podrían aparecer durante el proceso.

La primera propuesta interactiva que considera características cualitativas específicamente para el *UA-FLP*, fue propuesta por García-Hernández, Pierreval, Salas-Morera y otros [135]. Aunque esta propuesta mejoró con respecto a los métodos anteriores, el

---

algoritmo se centró principalmente en introducir el conocimiento del diseñador mediante la evaluación de soluciones representativas que se seleccionaban de toda la población utilizando un método de agrupamiento. Desafortunadamente, el método de selección mostraba un punto débil con respecto a la falta de diversidad en la población, por lo que con frecuencia se presentaban soluciones similares al diseñador experto en la misma iteración del algoritmo, así como durante todo el proceso. Este hecho causaba fatiga al diseñador experto, lo que lo llevaba a evaluar soluciones repetidas (que habían sido evaluadas previamente) y prolongaba innecesariamente la duración de la ejecución del algoritmo.

Por lo tanto, se considera necesario investigar nuevos enfoques que permitan evitar sobrecargar al diseñador experto durante la ejecución del algoritmo interactivo, siendo esencial preservar la diversidad de la población al mismo tiempo que se evite evaluar diseños repetidos.

#### 1.4 LA DIVERSIDAD DE LAS SOLUCIONES

Uno de los principales problemas con el que nos encontramos a la hora de afrontar el *FLP* es la búsqueda de soluciones diversas, ya que cuanto mayor diversidad podamos mantener, mayor probabilidad tendremos de encontrar buenas soluciones.

Los *GA* han ganado popularidad porque se pueden aplicar en una amplia gama de problemas, incluido el problema que nos

ocupa, el *FLP*, siendo uno de los algoritmos más utilizados para abordarlo. Sin embargo los comportamientos de los *GA* no han demostrado ser tan adaptativos como se esperaba [142]. Un problema común es el de la convergencia prematura [143], que consiste en que las soluciones convergen hacia una zona en la que no existe una solución óptima. El grado de diversidad de la población es una de las principales causas de convergencia prematura [144], [145]. Por lo general, la pérdida de diversidad se hace cada vez más notable a medida que se ejecutan más iteraciones [131] Por lo tanto, para llegar a los óptimos globales y explorar adecuadamente el espacio de búsqueda, es muy importante mantener la diversidad de la población.

Una de las principales líneas de investigación para resolver los problemas de quedar atrapado en los óptimos locales, consiste en el estudio de los algoritmos genéticos paralelos (Parallel Genetic Algorithm (*PGA*)). La adopción de un enfoque paralelo puede considerarse como una forma de mejorar la diversificación de la búsqueda y retrasar la convergencia prematura [146], características clave para explorar un espacio de búsqueda más amplio y obtener mejores soluciones.

Es por ello que, en vista de los primeros estudios publicados, consideramos que esta línea de investigación de los *PGA* puede ser prometedora para el mantenimiento de la diversidad de la población y, por consecuencia, para la obtención de mejores y diversas soluciones para el *FLP*.

# 2

---

## OBJETIVOS

---

El objetivo principal de esta tesis doctoral es mejorar los enfoques existentes para la resolución del *UA-FLP*, mediante el desarrollo de propuestas basadas en algoritmos genéticos.

Estas nuevas propuestas exploran dos líneas de investigación prometedoras: *a)* la resolución del problema teniendo en cuenta los aspectos cualitativos para ayudar al diseñador experto, y *b)* la aplicación de nuevas técnicas para el mantenimiento de diversidad de la población que permita explorar un amplio espacio de búsqueda con mayores probabilidades de obtener mejores soluciones.

A continuación, se presentan los objetivos secundarios para lograr los principales objetivos anteriormente mencionados:

- Análisis del estado del arte del *FLP* para identificar los enfoques en los que se está trabajando y los nuevos desafíos que se presentan.
- Análisis de los distintos conjuntos de datos con los que realizar las experimentaciones de los nuevos enfoques a desarrollar.
- Estudio de las nuevas técnicas de búsqueda automática para la resolución de problemas *FLP*.
- Análisis de los distintos algoritmos evolutivos paralelos existentes, así como su aplicación a problemas de optimización, con especial atención al *FLP*.
- Diseño e implementación de una nueva propuesta para la resolución del *UA-FLP* teniendo en cuenta los aspectos cualitativos, permitiendo la interacción entre el algoritmo y el diseñador humano experto.
- Diseño e implementación de un nuevo algoritmo híbrido genético paralelo que, mediante una estrategia que promueva la diversificación y mantenga la calidad de las soluciones, explore un mayor espacio de búsqueda con un alto grado de diversidad.

# 3

---

## METODOLOGÍA

---

Este capítulo resume los métodos y herramientas utilizados para el desarrollo de algoritmos propuestos en la presente tesis doctoral.

En primer lugar se describen los *GA*, enmarcados en los métodos evolutivos que están basados en la evolución natural.

Seguidamente, se presenta el problema del *UA-FLP*, en cuya resolución se centra las líneas de investigación de este trabajo.

En tercer lugar, se introducen las dos principales líneas de investigación: la mejora en la búsqueda de soluciones con aspectos cualitativos guiando al diseñador experto, y la persecución de la diversidad con el objeto de ampliar el espacio de búsqueda.

Por último, se describen los *datasets*, es decir las colecciones de datos que definen los problemas sobre los cuales se realiza la experimentación de los algoritmos desarrollados en esta tesis.

### 3.1 ALGORITMOS GENÉTICOS

Presentados en la década de 1970 por Holland [147], estos métodos computacionales se basan en la evolución natural, descrita en el trabajo de Darwin [148], en la que los individuos de una población compiten para obtener recursos naturales y encontrar parejas. Los individuos más exitosos en una población tienen una mayor probabilidad de reproducción, por lo tanto, crearán más descendencia que los individuos menos exitosos.

De acuerdo con García-Hernández, Araúzo-Azofra, Pierreval y otros [149], los GA han sido empleados ampliamente para resolver el problema de distribución en planta de áreas desiguales (*UA-FLP*) [41], [129], [130], [150]-[152]. Este tipo de enfoque está arrojando resultados muy prometedores, por lo que es un claro candidato para seguir investigando en esta línea.

En la figura 3.1 se muestra el diagrama de flujo de un algoritmo genético simple. El objetivo de este algoritmo es manipular una población de individuos que representan soluciones candidatas a un problema.

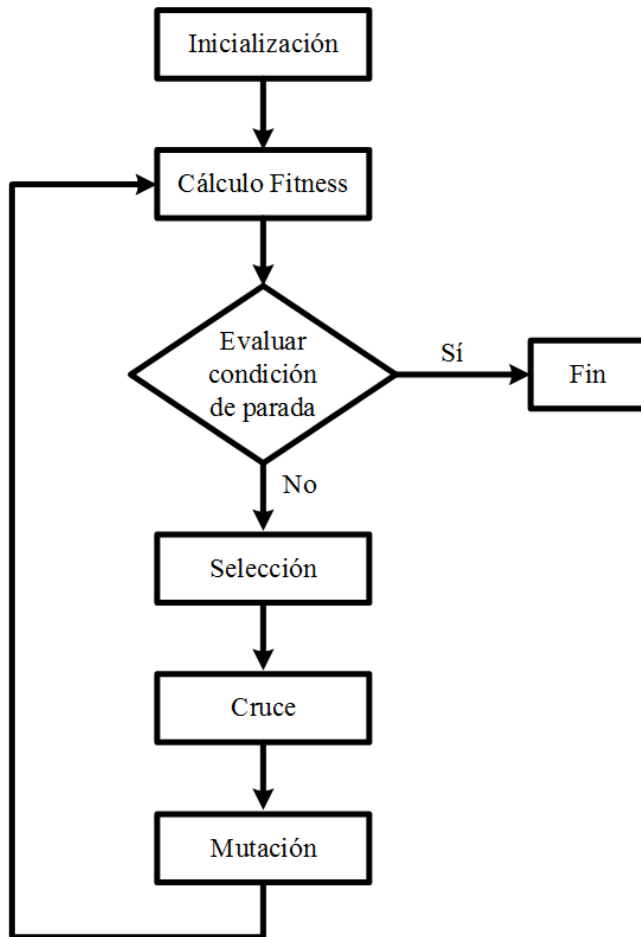
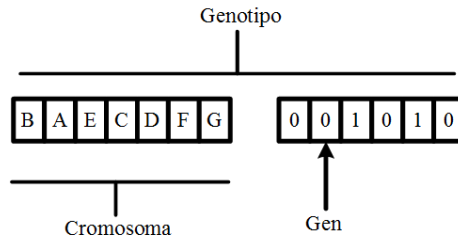


Figura 3.1: Diagrama de flujo del algoritmo genético

En primer lugar, se genera aleatoriamente la población inicial (inicialización), que normalmente puede contener varios cientos o miles de posibles soluciones. Cada individuo (solución) se codifica como un *genotipo*, el cual, a su vez, está compuesto por uno o varios *cromosomas*. Un *cromosoma* está formado por una serie de *genes*. Los diferentes valores que puede tomar un gen se denominan *alelos*. En la figura 3.2 se muestra un ejemplo de codificación de un individuo.





**Figura 3.2:** Ejemplo de codificación de un individuo

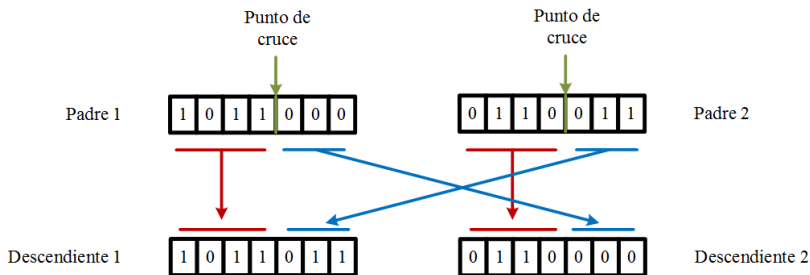
Una vez generada la población inicial, se realizarán distintas iteraciones que finalizarán cuando se cumpla la condición de parada.

Se calcula la puntuación (*fitness*) de cada individuo de la población, con una función predefinida, la cual, en caso de un problema de maximización, cuanto mayor sea el valor de la función, mayor será su *fitness*; en caso de minimización, cuanto menor sea el valor de la función, mayor será su *fitness*.

El siguiente paso consiste en la selección. Los individuos que presentan un *fitness* elevado aumentan las posibilidades de ser seleccionados para reproducirse con otro individuo y pasar su material genético a otra iteración del algoritmo. Aunque el objetivo de la selección es garantizar que los mejores individuos tienen una mayor probabilidad de ser padres, ya que genéticamente sus hijos pueden heredar las bondades de los padres, también es necesario dar oportunidad de reproducirse a los individuos menos buenos, ya que su material genético puede resultar útil en el proceso de reproducción, aportando diversidad. Entre las diferentes estrategias de selección empleadas a lo largo de la literatura, destacan las si-

guientes: torneo [135], *ranking* [153], ruleta [154], elitismo [155] y selección de *Boltzmann* [156], entre otros.

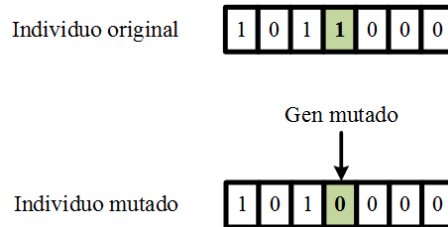
La operación de cruce producirá nuevos individuos que presentarán las características de sus padres. Por lo tanto, se genera una nueva población de individuos generalmente mejorados, que reemplazará a la población anterior. Entre los diferentes operadores de cruce cabe destacar los más habituales [157]: cruce de  $n$ -puntos, Position based crossover operator (*PBX*) [158], Partially-mapped crossover (*PMX*) [159], Cycle crossover operator (*CX*) [160], Order crossover (*OX*) [161], entre otros. En la figura 3.3 se muestra un ejemplo de cruce de tipo  $n$ -puntos entre dos individuos (padres). En concreto, se utiliza una técnica de corte en un único punto, en el que el cromosoma de los nuevos individuos (descendientes) se forma con una parte del cromosoma de cada padre.



**Figura 3.3:** Ejemplo de operador de cruce en un algoritmo genético

Finalmente, y por lo general, con una baja probabilidad, se seleccionan aleatoriamente individuos a los que se aplica un proceso de mutación. El objetivo de este proceso es alterar el cromosoma del individuo para mantener la diversidad de la población [162]. Aunque existen diferentes operadores de mutación, el más común

es la mutación de cadena de bits. Este sencillo proceso consiste en cambiar aleatoriamente los genes del cromosoma que conforma el individuo (ver figura 3.4).



**Figura 3.4:** Ejemplo de operador de mutación en un algoritmo genético

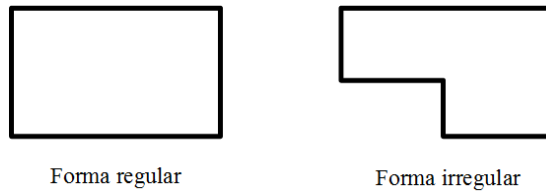
### 3.2 PROBLEMA DE DISTRIBUCIÓN EN PLANTA DE ÁREAS DESIGUALES

Dentro del *FLP* existen diferentes tipos de problemas. Estos vienen determinados por distintas características que los definen. En este sentido, varias clasificaciones de *FLP* pueden encontrarse en revisiones y estados del arte publicados sobre este tema [49], [163], [164].

Dentro de los problemas definidos, el problema de distribución en planta de departamentos de área desigual, conocido por sus siglas en inglés *UA-FLP*, formulado por Armour y Buffa [21], ha sido uno de los más estudiados en la literatura [41], [129], [165]-[171].

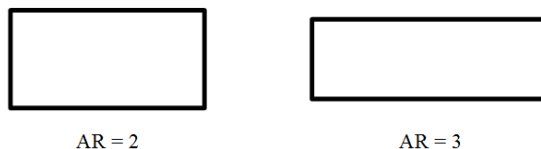
Respecto a la forma que adoptan las instalaciones, existen dos tipos diferentes: el primer tipo, el más común, se refiere a instalaciones con formas regulares [48]; mientras que, en el segundo

tipo, se admite que las instalaciones estén formadas por polígonos irregulares de ángulos rectos [172] (ver figura 3.5).



**Figura 3.5:** Formas de las instalaciones

Las instalaciones regulares más habituales son las rectangulares. Estas pueden ser definidas de dos maneras: *a)* por las dimensiones fijas del ancho y alto, o *b)* por un área fija, en cuyo caso el ancho y alto del rectángulo, al no estar definidos, pueden tomar diferentes valores, cumpliendo con el área fijada. En este punto entra en juego la relación de aspecto (*Aspect Ratio (AR)*). El *AR* es el cociente entre el lado mayor y el menor del rectángulo. Cuando se fija el área de una instalación, generalmente se añade alguna restricción como el límite superior y/o inferior del *AR*, de manera que se evite formas no deseables de la instalación. En la figura 3.6 se muestran dos ejemplos de una instalación con un mismo área y diferentes *AR*.



**Figura 3.6:** Instalación con la misma área y diferentes *aspect ratio*

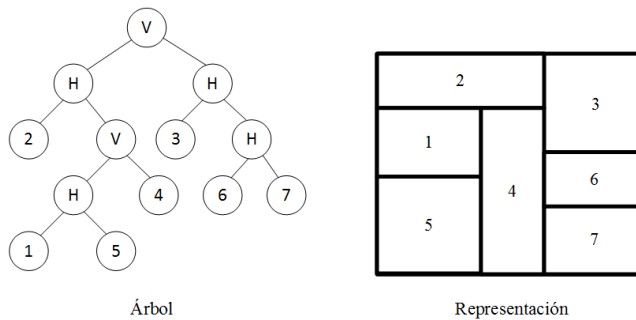
El problema que se analiza en la presente tesis está definido por una planta rectangular de dimensiones fijas donde se distribuye

un número de instalaciones rectangulares desiguales [21]. Además, debe tenerse en cuenta que todas las instalaciones tienen que distribuirse dentro de la planta, y que no está permitido el solape de las mismas. Este problema pretende distribuir las instalaciones que componen la planta industrial de la manera más efectiva, teniendo en cuenta una función objetivo. En este sentido, la más común trata de minimizar el flujo de material que existe entre los departamentos que componen la planta [173].

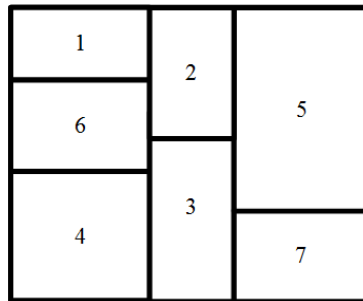
Según Arapoglu, Norman y Smith [174], para representar las plantas se han utilizado comúnmente dos estrategias: *a*) estructura de bahías flexibles (Flexible Bay Structure (*FBS*)), y *b*) estructura de árbol de cortes (Slicing Tree Structure (*STS*)). En el primer tipo, el espacio disponible se divide en bandas que, o bien son horizontales, o bien verticales, pero no ambos tipos simultáneamente [175], [176]. Mientras que en el segundo, el espacio se divide recursiva y alternativamente en cortes horizontales o verticales [11], [72], [165], [177]. En las figuras 3.8 y 3.7 se muestran, respectivamente, un ejemplo de cada estrategia de representación de planta.

Son dos tipos de representaciones alternativas, muy frecuentemente usadas en la bibliografía y que proporcionan resultados distintos y no comparables entre sí. El método *FBS* tiene la ventaja de su simplicidad y facilidad de formulación, por lo que el tiempo de cómputo suele ser menor que en *STS* [71], ya que este último amplía el espacio de búsqueda, con la contra de añadir complejidad al problema. Además, en los enfoques para la resolución del

*FLP*, la representación *FBS* ha sido más utilizada en la literatura, como por ejemplo en los siguientes trabajos: [166], [178]-[180]. Es por ello que en los enfoques desarrollados en la presente tesis se hace uso de la representación *FBS*.



**Figura 3.7:** Representación de una planta con *STS*



**Figura 3.8:** Representación de una planta con *FBS*

### 3.2.1 Formulación del problema

En esta sección se presenta la formulación del problema *UA-FLP*. Este problema consiste en posicionar un conjunto de  $n$  espacios

rectangulares que representan instalaciones, máquinas, procesos o cualquier actividad dentro de una planta industrial, oficinas, hospitales o cualquier otro espacio en el que sea necesario optimizar el flujo de materiales o de información, sujeto a una serie de restricciones.

La restricción más importante para este problema es la existencia de un espacio rectangular (planta) de dimensiones  $W \times H$  en el que es necesario disponer las  $n$  instalaciones, ocupando cada una de ellas un área específica ( $A_i$ ), de forma que no se supere el espacio total disponible. Es decir, la suma de las áreas de las instalaciones debe ser menor o igual que el área de la planta (ecuación 1).

Aunque existe un conjunto de criterios a optimizar dentro del diseño de la planta, en la función objetivo del presente trabajo solo se considera la minimización del flujo de materiales [173] para el cómputo del *fitness* de cada solución (ver ecuación 2).

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

$$\text{mín OF} = \sum_{i=1}^n \sum_{j=1, i \neq j}^n f_{ij} d_{ij} \quad (2)$$

Donde  $OF$  es la función objetivo;  $n$  es el número de instalaciones a disponer sobre el espacio disponible;  $f_{ij}$  es el flujo de materiales entre las instalaciones  $i$  y  $j$ ; y  $d_{ij}$  es la distancia entre los centros de las actividades  $i$  y  $j$ , que puede calcularse con diferentes métodos, los cuales serán descritos posteriormente. Tanto el método del cálculo de la distancia como el flujo  $f_{ij}$  es parte de la definición de la instancia del problema.

Por otro lado, se debe tener en cuenta que durante el proceso de búsqueda de soluciones pueden aparecer diseños de planta inviables, como por ejemplo tener instalaciones con un lado excesivamente estrecho, es decir, no cumplir con los límites de  $AR$ . Si se considerara exclusivamente el flujo de materiales sin ningún control sobre la forma de las instalaciones, estas soluciones inviables podrían ser consideradas como óptimas.

Sin embargo, desechar estas soluciones inviables durante el proceso de búsqueda puede no ser una buena decisión, ya que se disminuiría la diversidad y se podría caer con mayor facilidad en óptimos locales de manera prematura [10]. Por esta razón Tate y Smith [10] propusieron una formulación (ecuación 3) para penalizar las soluciones inviables, de manera que estas permanecieran en la población con un *fitness* penalizado en proporción al incum-



plimiento de la restricción del AR, al mismo tiempo que aportan diversidad.

$$P = (D_{inf})^k (V_{feas} - V_{all}) \quad (3)$$

Donde  $P$  es la penalización;  $D_{inf}$  es el número de instalaciones inviables según el criterio del AR;  $k$  es un factor de penalización que ajusta la severidad de la penalización (Tate y Smith [10] recomiendan  $k \geq 3$ );  $V_{feas}$  el valor más pequeño de la función objetivo encontrado hasta el momento entre todas las soluciones viables encontradas; y  $V_{all}$  es el valor más pequeño encontrado hasta el momento entre todas las soluciones evaluadas, sean viables o no.

Por lo tanto, la formulación de la función objetivo teniendo en cuenta la penalización se expresa de acuerdo con la ecuación 4.

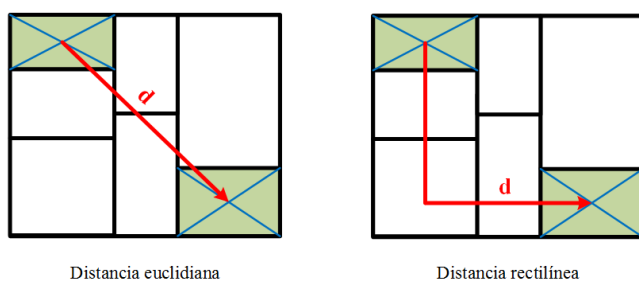
$$\text{mín OF} = \sum_{i=1}^n \sum_{j=1, i \neq j}^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \quad (4)$$

Como se ha indicado anteriormente, dependiendo de la formulación de la instancia del problema, el cálculo de la distancia se realiza mediante una de las siguientes formulaciones: *a*) distancia euclidiana (ecuación 5), y *b*) distancia *Manhattan* o rectilínea (ecuación 6). Este cálculo se realiza a partir de las coordenadas del

centro de cada departamento  $x_k, y_k$ . En la figura 3.9 se presentan ejemplos gráficos de ambos tipos de distancias en sendas plantas.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \quad (6)$$



**Figura 3.9:** Tipos de distancias

### 3.2.2 Codificación de las plantas

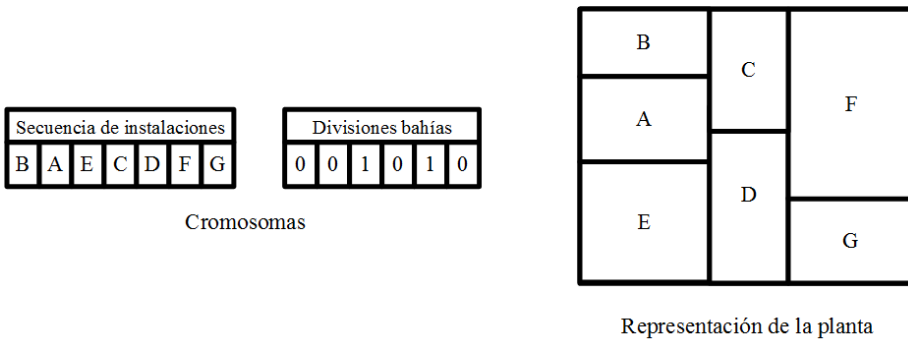
Cada una de las soluciones candidatas (los *individuos* en la población del algoritmo evolutivo) se codifica en una estructura denominada *genotipo* según lo propuesto por Gomez, Fernandez, Fuente Garcia y otros [181], consistente en dos  *cromosomas*:

1. Secuencia de las instalaciones: orden en el que las actividades se colocan en la planta (de arriba a abajo y de izquierda a

derecha). Este segmento del cromosoma es una permutación desde 1 hasta el número total de actividades en la planta ( $n$ ).

2. Divisiones en bahías: describe la estructura de la planta marcando las actividades que determinan el fin de una bahía. Para representar esta información, se utiliza un vector binario en el que la posición de cada valor 1 marca el final de una bahía.

La figura 3.10 muestra un ejemplo de cromosomas y su correspondiente planta asociada.



**Figura 3.10:** Ejemplo de codificación de una planta

### 3.3 ENFOQUES

En esta sección se presentan los enfoques novedosos en los que se basan ambas propuestas fruto del resultado de la presente tesis doctoral.

En primer lugar se presentan las técnicas de *niching*, las cuales serán aplicadas en la propuesta desarrollada para la mejora en la búsqueda de soluciones en base a aspectos cualitativos mediante el guiado del diseñador experto.

En segundo lugar, se describe el enfoque de técnicas de islas, cuya meta es preservar la diversidad de la población (soluciones) con el objeto de ampliar el espacio de búsqueda, reduciendo las probabilidades de caer en óptimos locales de manera prematura.

### 3.3.1 *Técnicas de niching*

La consideración de aspectos cualitativos en el diseño de distribución en planta enriquece enormemente las soluciones obtenidas. Con la propuesta de García-Hernández, Pierreval, Salas-Morera y otros [135] se permitió la inclusión de las preferencias del diseñador en el diseño de las plantas mediante una propuesta interactiva. En ella, el diseñador experto evaluaba y evolucionaba el proceso hacia un conjunto de soluciones satisfactorias para él. Sin embargo, esta propuesta tenía el inconveniente de que la misma solución podía presentarse al diseñador en varias ocasiones en la misma etapa del algoritmo y también entre diferentes iteraciones. Esto provocaba que el usuario tuviera que evaluar más soluciones de las realmente necesarias y acabara el proceso cansado y sobrecargado.

Para intentar corregir esta problemática, se han evaluado diferentes técnicas para preservar la población de soluciones. Como resultado de este estudio previo se han considerado los métodos de *niching* [182]-[184], con el objetivo de preservar la población de soluciones candidatas al problema y además considerar las preferencias cualitativas del diseñador experto en los diseños de distribución en planta.

Los métodos de *niching* tienen la capacidad de mantener ciertas propiedades dentro de la población de soluciones, preservando múltiples nichos estables cercanos de los óptimos locales, y reduciendo la probabilidad de quedar atrapado en ellos [185]. Estos métodos se han aplicado en numerosas propuestas de optimización *multimodal* [186]-[190], cuyo objetivo consiste en la localización de múltiples óptimos en una misma población de manera simultánea.

Según Yu y Suganthan [191], en la computación evolutiva, la técnica de *niching* implica la formación de subgrupos dentro de una población en la que cada subgrupo se dirige a una tarea específica. En comparación con un algoritmo genético simple, la diferencia radica en el hecho de que se preservan tanto los individuos altamente puntuados como aquellos peor puntuados. Esto le da a la población la oportunidad de transmitir la información genética de tales individuos a su descendencia y asegura que esta información no se extinga rápidamente. Al mantener una variedad razonablemente equilibrada de información genética, los procedimientos de *niching* permiten a la población enfocarse simultáneamente en más

de una región en el espacio de búsqueda, lo cual es esencial para descubrir varios óptimos en una sola ejecución.

En nuestra propuesta, se aplican dos técnicas de *niching* para la resolución del problema *UA-FLP* guiado por el diseñador experto: Deterministic Crowding (*DC*) y Restricted Tournament Selection (*RTS*).

El algoritmo *DC* fue presentado por Mahfoud [192], como una mejora del esquema *crowding*, cuyo objetivo es que solo una parte de la población pueda reproducirse y morir en cada generación [193]. En este algoritmo se preserva la diversidad de la descendencia basada en su *fitness* y la similitud entre sus padres y esta descendencia.

En la figura 3.11 se presenta el funcionamiento del algoritmo *DC*. En primer lugar, se seleccionan, de manera aleatoria, dos soluciones o individuos, denominados *padres* ( $p_1; p_2$ ). A continuación, se aplica el cruce, un proceso clásico del *GA*. Una vez realizado, se cuentan con dos nuevos individuos, en este caso *hijos* ( $h_1; h_2$ ), a los que se le aplicará el proceso de mutación, si procede. Seguidamente, se evalúa el *fitness* tanto de los *padres* ( $f p_1; f p_2$ ) como de los *hijos* ( $f h_1; f h_2$ ), así como la distancia entre ellos ( $d p_1, h_1; d p_1, h_2; d p_2, h_1; d p_2, h_2$ ). Finalmente, se realiza un proceso denominado *reemplazo*. En este, cada *hijo* compite contra uno de sus *padres* para decidir qué individuo sobrevive, que será determinado mediante selección por torneo (sobrevive el de mejor *fitness*).

En resumen, este método decide si los *hijos* van a reemplazar a sus *padres* en la población.

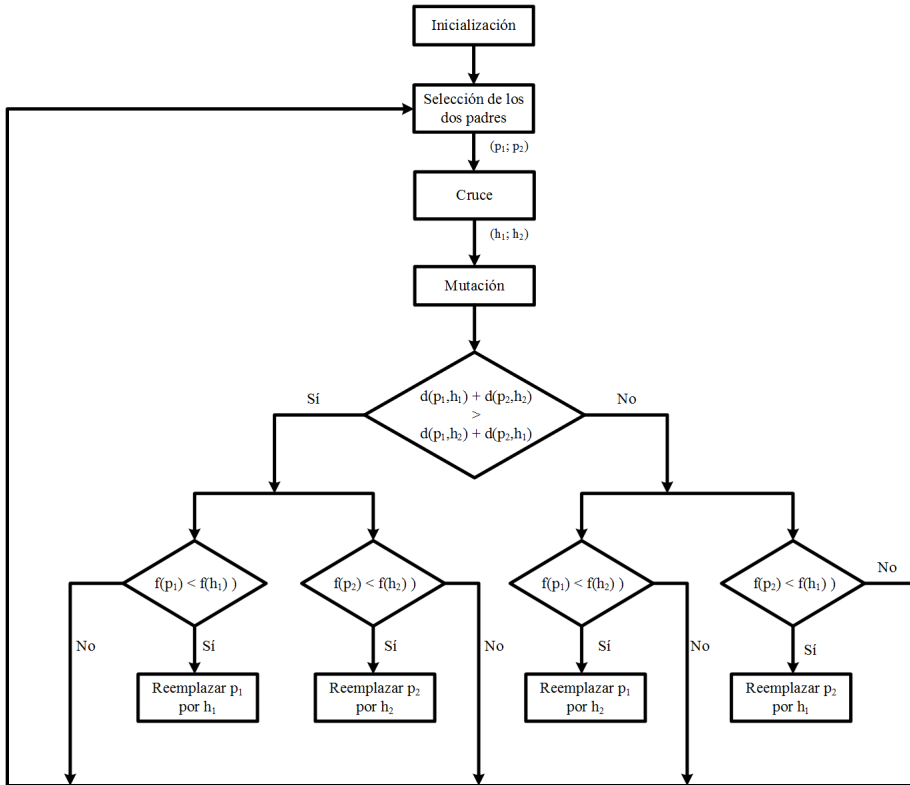
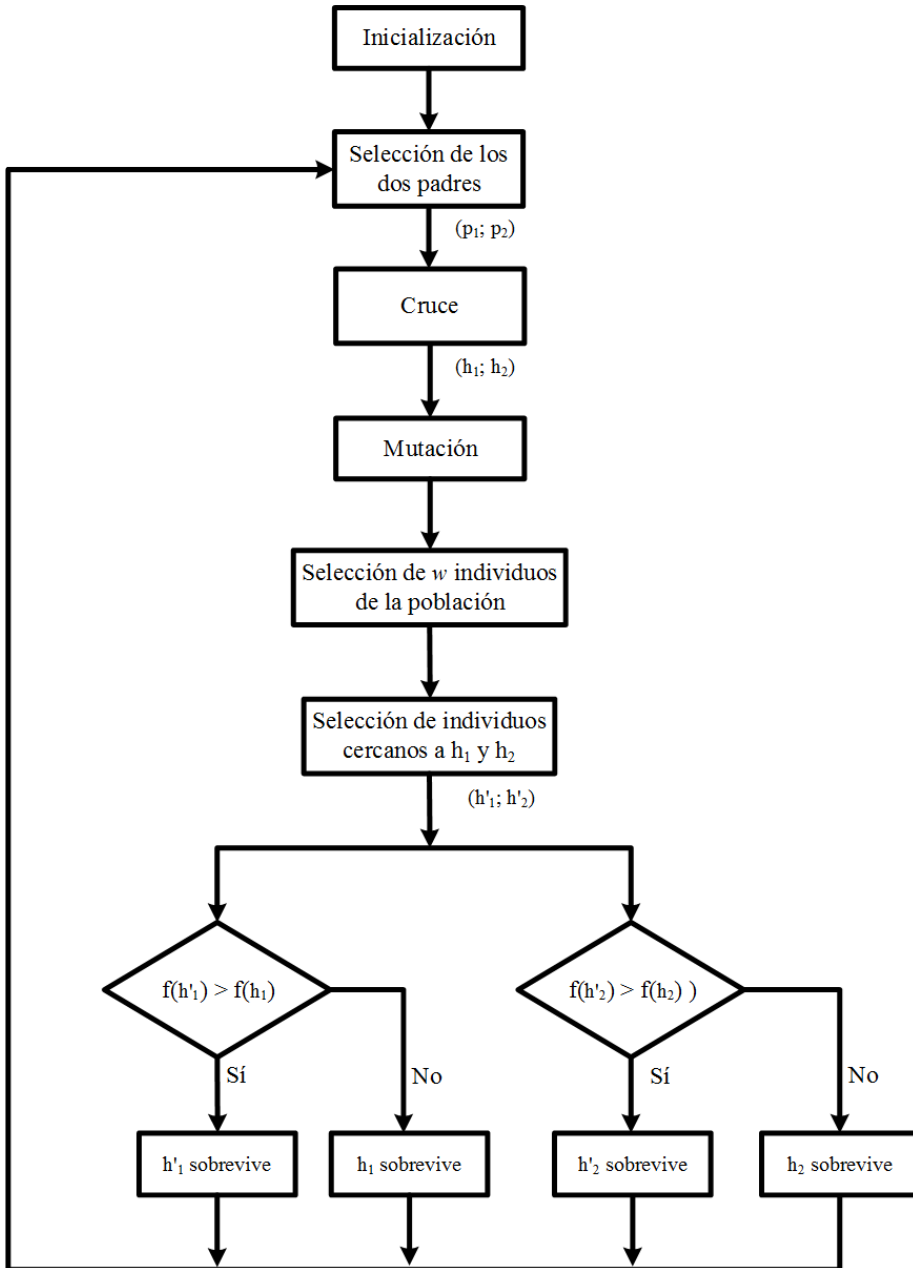


Figura 3.11: Diagrama de flujo del algoritmo *Deterministic Crowding*

Por otro lado, el algoritmo *RTS* es una modificación de la selección por torneo tradicional, que se basa en la competencia local, con objetivo de realizar una optimización *multimodal* [194]. El aumento del *fitness* de los individuos provocará competencias entre miembros del mismo nicho, mejorando cada nicho de manera independiente.



**Figura 3.12:** Diagrama de flujo del algoritmo *Restricted Tournament Selection*

En este enfoque, se seleccionan aleatoriamente de la población ( $p$ ), al igual que en el método anterior, dos *padres* ( $p_1; p_2$ ). A con-



tinuación, se aplican los procesos de cruce y mutación (si procede) obteniéndose dos hijos  $(h_1; h_2)$ . Seguidamente, se seleccionan  $w$  individuos de  $p$ , y de estos se obtienen los más cercanos a  $h_1$  y  $h_2$ , denominados  $h'_1$  y  $h'_2$ , respectivamente. Finalmente, por cada par  $(h_1, h'_1; h_2, h'_2)$ , se selecciona el de mejor *fitness*. Estos dos individuos seleccionados serán los que sobrevivan para formar parte de la siguiente generación o iteración del algoritmo.

### 3.3.2 Algoritmos genéticos paralelos

Otra línea de investigación prometedora para resolver los inconvenientes clásicos de los algoritmos genéticos (quedar atrapado en los óptimos locales, la convergencia prematura y el tiempo de ejecución), reside en los algoritmos genéticos paralelos (*PGA*).

En los últimos años, los *PGA* han sido ampliamente utilizados para resolver diferentes problemas reales tan diversos como la eficiencia energética [195]-[198], la gestión de redes viarias [199], [200], y el diseño de lanzamiento de satélites [201], entre muchos otros.

Estos algoritmos surgen debido al volumen de cómputo que requieren los problemas de alta complejidad, ya que para resolver estos problemas mediante *GA* tradicionales se requiere un tiempo de ejecución, a menudo, prohibitivo [202]. Cantú-Paz [203] propone clasificar los *PGA* en tres categorías:

- Maestro-esclavo de población única: la población se evalúa considerando varios procesos independientes, aunque la selección y cruce se aplica a toda la población.
- Grano fijo de población única: este modelo se basa en una población espacialmente estructurada en la que la selección y cruce se aplica entre individuos cercanos (por vecindario).
- Grano grueso de múltiples poblaciones: se evolucionan varias subpoblaciones independientemente, entre las que, ocasionalmente, se intercambian individuos mediante un proceso denominado *migración*.

En cuanto al problema que nos ocupa (*FLP*), se han propuesto algunos enfoques basados en *PGA*. Tam [204] desarrolló y comparó cuatro *PGA* de grano grueso. Al-Hakim [205] modificó el trabajo de Tam [204] al introducir una nueva operación para la preservación, denominada *trasplante*, que garantizaría la validez de la descendencia. Este modelo, basado en la evolución de una población dividida en subpoblaciones independientes, ayuda a presentar la diversidad genética y a explorar múltiples espacios de búsqueda [206].

Por lo tanto, podemos concluir que los *PGA* juegan un papel importante en el mantenimiento de la diversidad a lo largo de las diferentes generaciones del algoritmo, objetivo que perseguimos con su aplicación en esta tesis, el cual consideramos que es un requisito para la obtención de mejores soluciones para el problema *UA-FLP*.

En la presente tesis se desarrolla una propuesta denominada Island Model Genetic Algorithm (*IMGA*), basada en un *PGA* de grano grueso con múltiples poblaciones o *islas*. Debido al grado de independencia que suponen las *islas*, y al intercambio de individuos a través de la *migración*, es posible explorar diferentes regiones del espacio de búsqueda, mejorando así la calidad de las soluciones en un menor número de iteraciones. Este enfoque puede considerarse como un medio para mantener la diversidad de la población [207], además de ofrecer un mejor rendimiento de búsqueda que los modelos secuenciales basados en una sola población [203], [208].

### 3.4 CONJUNTO DE PROBLEMAS (*datasets*)

En esta sección se presentan los problemas sobre los cuales se realiza la experimentación de los algoritmos desarrollados en la presente tesis. Estos problemas han sido empleados en la literatura para probar las diferentes propuestas, con el objetivo de comparar los resultados entre ellas.

En primer lugar, se describen los dos problemas utilizados en la experimentación de la propuesta orientada a ayudar al diseñador experto en la búsqueda de buenos diseños de planta mediante técnicas de *niching* evolutivas. El primero de ellos definido por Aiello, Enea y Galante [73], se compone de 20 instalaciones que deben asignarse dentro de un área cuadrada de  $70\text{ m} \times 70\text{ m}$ , con un  $AR_{max}$  2,5 (*AR* máximo) para cada instalación. El segundo, extraído del trabajo de Salas-Morera, Cubero-Atienza y Ayuso-Muñoz

[8], consiste en un matadero de ovinos, cuyas instalaciones (con  $AR_{max}$  4) deben ser colocadas dentro de un área rectangular de  $55\text{ m} \times 30\text{ m}$ .

A continuación, se enumeran los problemas utilizados en la experimentación de la propuesta basada en *PGA* desarrollada en el presente trabajo: los problemas *Ba12* y *Ba14* de Bazaraa [61]; *Ba12TS* y *Ba14TS*, modificaciones realizadas por Tate y Smith [10] basadas en sendos problemas de Bazaraa [61]; el problema *MB12* propuesto por Bozer y Meller [209]; los problemas *O7*, *O8* y *O9* descritos por Meller, Narayanan y Vance [12]; el problema *VC10* definido por Camp, Carter y Vannelli [210], en los que se consideran diferentes medidas de distancia y *AR*; *AB20*, el problema descrito por Armour y Buffa [21], para el que se ha considerado hasta 12 *AR* diferentes; y por último, dos problemas de gran tamaño, *SC30* y *35*, presentados por Liu y Meller [51], y posteriormente modificados por Komarudin y Wong [72], quienes agregaron instalaciones ficticias.

En la tabla 3.1 se muestran las principales características del conjunto de problemas utilizado durante la experimentación de las propuestas desarrolladas en la presente tesis. En esta tabla se indica: *a)* el nombre del problema; *b)* el número de departamentos; *c)* dimensiones de la planta; *d)* restricción de forma; *e)* tipo de distancia, y *f)* referencia del problema. La restricción de forma puede ser de dos tipos:  $\alpha$  (*AR* máximo) y  $l_{min}$  (lado mínimo). Finalmente,

en las tablas 3.2-3.13 se presentan los flujos de materiales entre departamentos de los diferentes problemas.

**Tabla 3.1:** Características de los problemas para la experimentación

Nombre del problema	Núm. dptos.	Tamaño de planta (W x H)	Restricción de forma	Tipo de distancia	Referencia
O7	7	8.54 x 13.00	$\alpha$ 4	Rectilínea	Meller, Narayanan y Vance [12]
O8	8	11.31 x 13.00	$\alpha$ 4	Rectilínea	Meller, Narayanan y Vance [12]
O9	9	12.00 x 13.00	$\alpha$ 4	Rectilínea	Meller, Narayanan y Vance [12]
vC10E-a <sup>a</sup>	10	25.00 x 51.00	$\alpha$ 5	Euclidiana	Camp, Carter y Vannelli [210]
vC10E-s <sup>a</sup>	10	25.00 x 51.00	$l_{min}$ 5	Euclidiana	Camp, Carter y Vannelli [210]
vC10R-a <sup>a</sup>	10	25.00 x 51.00	$\alpha$ 5	Rectilínea	Camp, Carter y Vannelli [210]
vC10R-s <sup>a</sup>	10	25.00 x 51.00	$l_{min}$ 5	Rectilínea	Camp, Carter y Vannelli [210]
Ba12	12	6.00 x 10.00	$l_{min}$ 1	Rectilínea	Bazaraa [61]
Ba12TS <sup>b</sup>	12	6.00 x 10.00	$l_{min}$ 1	Rectilínea	Tate y Smith [10]
MB12	12	6.00 x 8.00	$\alpha$ 4	Rectilínea	Bozer y Meller [209]
Slaughterhouse	12	55.00 x 30.00	$\alpha$ 4	Euclidiana	Salas-Morera, Cubero-Atienza y Ayuso-Muñoz [8]
Ba14	14	7.00 x 9.00	$l_{min}$ 1	Rectilínea	Bazaraa [61]
Ba14TS <sup>c</sup>	14	7.00 x 9.00	$l_{min}$ 1	Rectilínea	Tate y Smith [10]
Aiello20	20	70.00 x 70.00	$\alpha$ 2,5	Rectilínea	Aiello, Enea y Galante [73]
AB20-ar170667 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 1,70667	Rectilínea	Armour y Buffa [21]
AB20-ar175 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 1,75	Rectilínea	Armour y Buffa [21]
AB20-ar2 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 2	Rectilínea	Armour y Buffa [21]
AB20-ar3 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 3	Rectilínea	Armour y Buffa [21]
AB20-ar4 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 4	Rectilínea	Armour y Buffa [21]
AB20-ar5 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 5	Rectilínea	Armour y Buffa [21]
AB20-ar7 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 7	Rectilínea	Armour y Buffa [21]
AB20-ar10 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 10	Rectilínea	Armour y Buffa [21]
AB20-ar15 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 15	Rectilínea	Armour y Buffa [21]
AB20-ar25 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 25	Rectilínea	Armour y Buffa [21]
AB20-ar50 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 50	Rectilínea	Armour y Buffa [21]
AB20-ar1000 <sup>d</sup>	20	2.00 x 3.00	$\alpha$ 1000	Rectilínea	Armour y Buffa [21]
SC30	30	12.00 x 15.00	$\alpha$ 5	Rectilínea	Liu y Meller [51] Komarudin y Wong [72]
SC35	35	15.00 x 16.00	$\alpha$ 4	Rectilínea	Liu y Meller [51] Komarudin y Wong [72]

- <sup>a</sup> Problema vC10 con distintos tipos de distancia y restricciones de forma.  
<sup>b</sup> Problema Ba12 con cambios en el área de algunas instalaciones, pero manteniendo el mismo flujo de materiales.  
<sup>c</sup> Problema Ba14 con cambios en el área de algunas instalaciones, pero manteniendo el mismo flujo de materiales.  
<sup>d</sup> Problema AB20 con distintas restricciones de forma.













Tabla 3.11: Problema Aiello20. Flujo de materiales entre departamentos.

Dpto.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-	150	200	280	240	55	0	50	95	55	890	0	0	320	655	395	0	845	80	95
2	25	-	0	235	935	0	0	515	0	0	0	75	0	0	95	840	50	915	135	0
3	935	0	-	910	675	0	0	265	60	800	180	0	70	0	0	0	90	0	665	995
4	620	75	160	-	80	60	180	165	0	385	175	890	955	0	60	0	430	345	555	90
5	0	75	0	50	-	80	0	70	520	95	70	435	0	0	65	95	50	0	175	0
6	0	0	925	50	55	-	510	0	760	0	0	95	0	0	0	0	195	900	110	0
7	55	60	305	235	385	0	-	305	0	0	0	0	95	760	60	60	75	80	0	555
8	380	65	50	80	525	90	975	-	85	120	400	980	0	95	0	830	0	55	90	0
9	50	70	65	0	90	95	60	530	-	0	0	90	80	265	0	60	0	85	95	85
10	170	50	90	90	0	665	50	975	85	-	0	75	90	55	640	0	255	385	0	80
11	50	0	90	765	0	0	0	980	180	425	-	75	65	725	515	930	180	85	515	65
12	90	195	0	110	0	585	185	385	95	60	55	-	0	545	0	0	0	0	220	670
13	0	0	135	550	55	690	310	410	60	0	0	50	-	165	85	0	195	545	80	55
14	65	90	220	0	80	0	65	605	90	425	0	0	70	-	65	750	95	0	0	785
15	80	0	0	55	70	0	75	0	70	0	545	90	95	0	-	0	80	775	70	755
16	65	140	0	0	730	80	55	0	80	50	90	80	740	65	0	-	95	0	455	70
17	0	70	880	95	0	0	50	685	70	0	870	725	0	450	0	70	-	0	450	50
18	70	75	0	60	80	65	0	50	65	0	430	80	0	130	190	80	90	-	765	0
19	80	725	310	0	0	0	85	0	885	50	0	85	85	90	75	0	485	0	-	65
20	80	60	365	0	80	0	0	0	50	0	50	0	0	110	65	0	90	0	0	-

**Tabla 3.12:** Problema SC30. Flujo de materiales entre departamentos.

Desde	Hasta	Flujo (%)	Desde	Hasta	Flujo (%)
1	24	2.95	10	12	38.03
1	25	6.32	11	12	8.84
1	26	1.26	12	13	12.97
1	27	2.11	12	14	4.67
1	28	1.26	12	15	53.42
1	30	13.91	12	16	1.83
1	6	20.38	13	15	12.97
1	19	4.5	14	15	4.67
1	20	3.63	15	4	63.95
1	21	2.93	15	18	190.74
1	22	1.29	16	15	4.58
1	23	1.43	17	15	0.76
4	3	394.11	18	4	190.74
5	11	4.09	19	29	9.18
5	12	33.09	20	29	7.4
6	5	8.92	21	29	5.97
6	7	2.07	22	29	2.63
6	8	4.84	23	29	2.92
6	10	4.56	24	29	5.9
6	12	1.83	25	29	12.65
6	17	0.61	26	29	2.53
7	8	12.97	27	29	4.22
8	9	43.23	28	29	2.53
9	11	4.76	29	4	190.13
9	12	38.47	30	29	59.64

**Tabla 3.13:** Problema SC35. Flujo de materiales entre departamentos.

Desde	Hasta	Flujo (%)	Desde	Hasta	Flujo (%)
1	24	2.95	11	12	8.84
1	25	6.32	12	32	72.89
1	26	1.26	13	15	12.97
1	27	2.11	14	15	4.67
1	28	1.26	15	4	63.95
1	30	13.91	15	18	190.74
1	6	20.38	16	15	4.58
1	19	4.5	17	15	0.76
1	20	3.63	18	4	190.74
1	21	2.93	19	29	9.18
1	22	1.29	20	29	7.4
1	23	1.43	21	29	5.97
4	3	225.65	22	29	2.63
5	11	4.09	23	29	2.92
5	12	33.09	24	29	5.9
6	5	8.92	25	29	12.65
6	7	2.07	26	29	2.53
6	8	4.84	27	29	4.22
6	10	4.56	28	29	2.53
6	12	1.83	29	33	190.13
6	17	0.61	30	29	59.64
6	31	3.93	31	32	22.92
7	8	12.97	32	13	12.97
8	9	43.23	32	14	4.67
9	11	4.76	32	15	53.42
9	12	38.47	32	16	1.83
10	12	38.03	33	34	168.45

# 4

---

## RESULTADOS

---

Como resultado de la investigación realizada se han publicado dos artículos en la revista *Expert Systems with Applications*, indexada en el primer decil del *Journal Citation Reports*. Esta revista internacional de referencia se centra en el intercambio de información relacionada con sistemas expertos e inteligentes aplicados en la industria, el gobierno y las universidades de todo el mundo. El objetivo de la revista es publicar documentos relacionados con el diseño, desarrollo, prueba, implementación y/o gestión de sistemas expertos e inteligentes, y también proporcionar pautas prácticas en el desarrollo y gestión de estos sistemas.

El primer artículo trata sobre la incorporación del conocimiento experto al proceso de diseño de distribución en planta, de forma que se puedan considerar características cualitativas que no serían posibles con un enfoque meramente cuantitativo

El diseñador experto guía el proceso evolutivo del algoritmo con la particularidad de que no necesita explicitar sus preferencias, ni formalizarlas de ningún modo. Es más, puede incluso modificar su criterio a lo largo del proceso, bien porque las soluciones propuestas por el algoritmo le hagan ver alguna mejora sobre su idea inicial, o bien porque simplemente advierte haber olvidado algún aspecto importante.

Sin embargo, no es posible mostrar al diseñador experto todas las posibles soluciones que están siendo consideradas en la población del algoritmo, debido al alto número de las mismas, por lo que una correcta selección de soluciones que mostrar al experto es crucial. En primer lugar, el conjunto de soluciones mostradas debe ser representativo de la totalidad de la población, a la vez que las soluciones mostradas deben ser suficientemente distintas entre sí para evitar que el usuario tenga que evaluar soluciones parecidas o incluso idénticas.

Para seleccionar las soluciones a mostrar se ha utilizado la técnica de *fuzzy c-means clustering* [211]. Asimismo, para hacer la selección de individuos que se reproducirán para la siguiente generación, preservando la diversidad de la población, se han utilizado dos técnicas de *niching*: *Deterministic Crowding* [212] y *Restricted Tournament Selection* [194]. Con este tipo de estrategia se consigue un doble objetivo: *a)* se incorporan las preferencias del diseñador a la estrategia de búsqueda de soluciones, y *b)* se consigue minimizar su fatiga, haciéndolo intervenir solo cada cierto número de

generaciones y mostrándole un número limitado de soluciones sin perder generalidad en el proceso. El artículo publicado se presenta en el *anexo A*, y cuya referencia es la siguiente:

L. García-Hernández, J. M. Palomo-Romero, L. Salas-Morera, A. Arauzo-Azofra y Henri Pierreval. *A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem*. Expert Systems with Applications, 42 (10), páginas 4697-4708, 2015.

Por otro lado, en la segunda propuesta desarrollada, el enfoque es meramente cuantitativo. En este caso se pretende mejorar la calidad de las soluciones estableciendo una estrategia que prevenga la convergencia prematura y que a la vez preserve la variabilidad de soluciones permitiendo también incrementar la capacidad exploratoria del algoritmo. Para ello, se ha implementado un enfoque basado en la evolución por islas, que se clasifica dentro de las técnicas de computación paralela o evolución paralela.

En términos generales, la estrategia consiste en desarrollar la evolución de la población simultáneamente repartida en un conjunto de subpoblaciones, de forma que en cada una de ellas se ejecuta una exploración intensiva sobre un subespacio de soluciones. Cada cierto número de generaciones del algoritmo, se realiza una *migración* de individuos de unas islas a otras, de forma que los mejores individuos sustancialmente distintos de una sub pobla-



ción con respecto a otra, son intercambiados, incrementando así la diversidad de la población con una exploración más extensiva.

Mediante este enfoque de evolución paralela, se ha conseguido mejorar las soluciones referidas en la bibliografía para la mayoría de los casos probados y con un tiempo de cómputo menor.

El segundo artículo publicado, en el que se desarrolla esta propuesta, puede consultarse en el *anexo B*:

J. M. Palomo-Romero, L. Salas-Morera, y L. García-Hernández. ***An island model genetic algorithm for unequal area facility layout problems***. Expert Systems with Applications, 68, páginas 151-162, 2017.

# 5

---

## CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

---

En la presente tesis doctoral se han desarrollado dos nuevas estrategias evolutivas para la resolución del problema *UA-FLP*. Cada una de ellas aborda el problema desde un punto de vista distinto: la primera permite la intervención de un usuario experto en el proceso de generación y selección de soluciones; y, la segunda, implementa una nueva estrategia para incrementar la diversidad de la población a lo largo de todo el proceso evolutivo, lo que permite evitar caer en óptimos locales, por un lado, y encontrar un conjunto más diverso de posibles soluciones para cada problema, por otro.

La introducción de las preferencias subjetivas de un usuario experto enriquece el conjunto de soluciones obtenidas, permitiendo tener en cuenta aspectos difíciles de considerar en una función objetivo tradicional. Dentro de este conjunto de preferencias pue-

den encontrarse consideraciones estéticas, de seguridad, o cualquier otra que afecte al proceso productivo más allá del mero flujo de materiales. Sin embargo, la necesidad de la intervención humana dentro de la evolución del algoritmo produce considerable fatiga física y visual. Para ello, se ha recurrido a agrupar las soluciones que componen la población en grupos más o menos homogéneos, a través de un procedimiento de *clustering*, para mostrar al usuario únicamente aquellas más representativas de la población, distribuyendo las evaluaciones al resto de la población por medio de un sistema de pesos ponderados.

La intervención de un usuario experto ha demostrado ser útil para incluir sus preferencias subjetivas sacrificando solo una pequeña parte de la optimización de los criterios objetivos. Por otro lado, la estrategia evolutiva de búsqueda de soluciones permite que el usuario cambie de opinión, o descubra nuevas opciones, a lo largo de las distintas generaciones, adaptándose dinámicamente a sus nuevos requerimientos. Por todo ello, la intervención del usuario enriquece las soluciones obtenidas, incorporando aspectos que serían imposible que aparecieran en un algoritmo cuantitativo tradicional, lo que incrementa sustancialmente su aplicabilidad práctica en casos reales.

Por otro lado, las dos carencias fundamentales de los métodos de computación evolutiva son: *a)* el peligro de caer rápidamente en óptimos locales, con una convergencia prematura hacia soluciones alejadas del óptimo global, y *b)* la falta de diversidad de

la población, aspectos ambos que están íntimamente relacionados. Para incrementar la variabilidad de la población y evitar la convergencia prematura, se han propuesto dos técnicas: en primer lugar el uso de técnicas de *niching* y, en segundo lugar, un nuevo enfoque consistente en la evolución de la población por islas independientes.

El primero de ellos (*niching*), consiste en identificar subgrupos de soluciones distintas con características similares, incluso aunque en el subgrupo existan soluciones con *fitness* mejores y peores. El objetivo es mantener estos grupos a lo largo del proceso evolutivo, de forma que la información genética de cada subgrupo se preserve y se transmite de una generación a otra, asegurando la diversidad.

Por su parte, el enfoque evolutivo por islas consiste en evolucionar, no de una manera global una única población, sino un conjunto de subpoblaciones simultáneamente, de forma que cada una avanza hacia la obtención de soluciones satisfactorias independientemente. En un momento dado del proceso, se producen migraciones de individuos de unas islas a otras, intercambiando las soluciones más aptas y diferentes entre sí, y aportando gran variabilidad al proceso evolutivo de cada isla. Este nuevo método ha sido probado sobre un amplio conjunto de problemas bien conocidos en la bibliografía, obteniendo soluciones aptas y mejores que las aportadas hasta el momento por otras propuestas, incluso con coste computacional inferior.

La evolución por islas representa una novedad que abre la puerta a analizar todo un conjunto de posibles estrategias evolutivas, como por ejemplo, mejorar la estrategia de migraciones entre las distintas poblaciones, analizar la aplicación de distintos operadores de cruce y mutación a cada una de las poblaciones, y otras que pudieran aportar variabilidad y profundidad en la búsqueda de nuevas soluciones.

La proyección natural de la investigación realizada en la presente tesis tiene dos vertientes fundamentales relacionadas con las dos principales líneas cubiertas. En primer lugar, respecto a la intervención de un usuario experto dentro del proceso evolutivo de búsqueda de soluciones, habría que explorar dos posibles líneas: *a)* la intervención multiusuario, y *b)* analizar la forma de mejorar la interacción del usuario con la máquina al objeto de disminuir su fatiga.

La primera de ellas, la intervención multiusuario, debería permitir la interacción de varios usuarios dentro del proceso evolutivo de búsqueda de soluciones, con la idea de permitir un diseño colaborativo en el que pudieran tenerse en cuenta las preferencias subjetivas de varias personas distintas al objeto de enriquecer el diseño. Esta intervención multiusuario debería poder ser asíncrona o síncrona. En el primer caso, un usuario ejecutaría el lanzamiento del problema y guiaría la búsqueda de soluciones hasta obtener una solución que satisfaga sus preferencias, inmediatamente intervendría el siguiente usuario y así sucesivamente. En el segundo

caso, todos los usuarios intervendrían evaluando simultáneamente el paquete de soluciones desde diferentes puestos de trabajo vía web. Es de suponer que el comportamiento de las dos estrategias no dé el mismo grado de prioridad a todos los usuarios, por lo que habría que analizar el conjunto de soluciones aportadas por el algoritmo para ver cuál se adapta en mayor medida a ese diseño en grupo.

Por otro lado, la disminución de la fatiga del usuario puede obtenerse mejorando los algoritmos de *clustering* utilizados para presentarle un subconjunto cada vez más pequeño de soluciones a evaluar, así como mejorando la estrategia de distribución de las evaluaciones. Otra prometedora línea a investigar es la de evitar la necesidad del uso del ratón por parte del usuario. En este caso, se combinaría la evolución del algoritmo con el uso de un dispositivo de seguimiento ocular para estimar las preferencias del usuario a partir de los parámetros de su comportamiento visual: el tiempo hasta la primera fijación, duración de la primera fijación, conteo de visitas, tiempo total de visitas, parpadeo, etc. La estimación de las preferencias del usuario tendría que ser sintonizada previamente mediante un paquete de ensayos sobre un conjunto de problemas bien conocidos y sobre un conjunto de usuarios expertos tan amplio como fuera posible, usando algún algoritmo de inteligencia artificial, como por ejemplo redes neuronales u otro tipo de algoritmo de aprendizaje.

Por su parte, la obtención de soluciones en menos tiempo manteniendo la variabilidad de la población, podría mejorarse incorporando nuevas estrategias evolutivas, como por ejemplo el algoritmo de arrecifes de coral. Este algoritmo evolutivo imita el crecimiento de los arrecifes de coral y mantiene un buen equilibrio entre la búsqueda extensiva y la intensiva, evitando caer en óptimos locales [213]. Igualmente, podría implementarse la versión mejorada de este algoritmo que permite, en una estrategia multicapa [214], poner en práctica distintas estrategias evolutivas (GA, búsqueda armónica, búsqueda tabú, etc.) y con distintos operadores simultáneamente sobre la misma población de soluciones candidatas, lo que también incrementaría sustancialmente la potencia del algoritmo.

Por último, una línea de investigación con buenas expectativas de desarrollo de propuestas para la obtención de soluciones mejores y más diversas entre sí, en un menor tiempo computacional, es la de los *PGA*. Aunque en este sentido ya se ha realizado una primera aproximación con la presente tesis, los resultados satisfactorios de la misma permiten augurar resultados prometedores en esta línea. Un interesante aporte al modelo de islas propuesto en esta tesis es la estrategia de localización descrita en el trabajo de Gozali y Fujimura [215]. Esta estrategia toma como referencia la variedad de hábitats del mundo real, y consiste, en términos generales, en configurar diferentes parámetros del GA a cada una de las islas, de manera que estas se comportarían de manera diferente, potenciando así distintas características de los individuos en cada

una de las islas. El resultado que se persigue con esta estrategia es, al fin y al cabo, aumentar la diversidad de las soluciones exploradas.





---

## BIBLIOGRAFÍA

---

- [1] P. M. Vázquez y J. M. Ruíz, «Elementos que Afectan el Nivel de Inventario en Proceso (WIP) y los Costos de una Línea de Producción», *Conciencia Tecnológica*, n.º 43, págs. 36-41, 2012.
  
- [2] S. Heragu, *Facilities Design*. CRC Press, 2018, ISBN: 9781498732901.
  
- [3] J. Tompkins, J. White, Y. Bozer y J. Tanchoco, *Facilities Planning*, 4rd ed. New York: Wiley, 2010.
  
- [4] P. Hicks, *Ingeniería industrial y administración: una nueva perspectiva*. CECSA, 2000, ISBN: 9789682612169.
  
- [5] P. Kouvelis, A. A. Kurawarwala y G. J. Gutierrez, «Algorithms for robust single and multiple period layout planning for manufacturing systems», *European Journal of Operational Research*, vol. 63, n.º 2, págs. 287-303, 1992.

- [6] R. Muther, *Practical plant layout*. McGraw-Hill, 1955.
- [7] T. Koopmans y M. J. Beckmann, «Assignment Problems and the Location of Economic Activities», Cowles Foundation for Research in Economics, Yale University, Cowles Foundation Discussion Papers 4, 1955.
- [8] L. Salas-Morera, A. Cubero-Atienza y R. Ayuso-Muñoz, «Computer-aided plant layout», *Información Tecnológica*, vol. 7, n.º 4, págs. 39-46, 1996.
- [9] J. Moore, *Plant Layout and Design*. Macmillan, 1962.
- [10] D. M. Tate y A. E. Smith, «Unequal Area Facility Layout Using Genetic Search», *IIE Transactions*, vol. 27, págs. 465-472, 1995.
- [11] E. Shayan y A. Chittilappilly, «Genetic algorithm for facilities layout problems based on slicing tree structure», *International Journal of Production Research*, vol. 42, n.º 19, págs. 4055-4067, 2004.
- [12] R. D. Meller, V. Narayanan y P. H. Vance, «Optimal facility layout design», *Operations Research Letters*, vol. 23, n.º 3-5, págs. 117 -127, 1998.

- 
- [13] F. Azadivar y J. Wang, «Facility layout optimization using simulation and genetic algorithms», *International Journal of Production Research*, vol. 38, n.º 17, págs. 4369-4383, 2000.
- [14] R. Meller y K. Gau, «The Facility Layout Problem: Recent and Emerging trends and perspectives», *Journal of Manufacturing Systems*, vol. 15, págs. 351-366, 1996.
- [15] M. J. Rosenblatt, «The facilities layout problem: a multi-goal approach», *International Journal of Production Research*, vol. 17, n.º 4, págs. 323-332, 1979.
- [16] J. C. Fortenberry y J. F. Cox, «Multiple criteria approach to the facilities layout problem», *International Journal of Production Research*, vol. 23, n.º 4, págs. 773-782, 1985.
- [17] T. L. Urban, «A multiple criteria model for the facilities layout problem», *International Journal of Production Research*, vol. 25, n.º 12, págs. 1805-1812, 1987.
- [18] C.-W. Chen y D. Sha, «Heuristic approach for solving the multi-objective facility layout problem», *International Journal of Production Research*, vol. 43, n.º 21, págs. 4493-4507, 2005.

- [19] A. Saraswat, U. Venkatadri e I. Castillo, «A framework for multi-objective facility layout design», *Computers & Industrial Engineering*, vol. 90, págs. 167 -176, 2015.
- [20] C. Guan, Z. Zhang, S. Liu y J. Gong, «Multi-objective particle swarm optimization for multi-workshop facility layout problem», *Journal of Manufacturing Systems*, vol. 53, págs. 32 -48, 2019.
- [21] G. Armour y E. Buffa, *A heuristic Algorithm and Simulation Approach to the Relative Locations of Facilities*, vol. 9, pág. 294, 1963.
- [22] C. C. Pegels, «Plant layout and discrete optimizing», *International Journal of Production Research*, vol. 5, n.º 1, págs. 81-92, 1966.
- [23] T. E. Vollmann y E. S. Buffa, «The facilities layout problem in perspective», *Management Science*, vol. 12, n.º 10, B-450, 1966.
- [24] C. E. Nugent, T. E. Vollmann y J. Ruml, «An experimental comparison of techniques for the assignment of facilities to locations», *Operations research*, vol. 16, n.º 1, págs. 150-173, 1968.

- 
- [25] A. V. Cabot, R. L. Francis y M. A. Stary, «A Network Flow Solution to a Rectilinear Distance Facility Location Problem», *A I I E Transactions*, vol. 2, n.º 2, págs. 132-141, 1970.
- [26] C. Friedrich, A. Klausnitzer y R. Lasch, «Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance», *European Journal of Operational Research*, vol. 270, n.º 3, págs. 837 -851, 2018.
- [27] J. Feng y A. Che, «Novel integer linear programming models for the facility layout problem with fixed-size rectangular departments», *Computers & Operations Research*, vol. 95, págs. 163 -171, 2018.
- [28] R. Wang, H. Zhao, Y. Wu, Y. Wang, X. Feng y M. Liu, «An industrial facility layout design method considering energy saving based on surplus rectangle fill algorithm», *Energy*, vol. 158, págs. 1038 -1051, 2018.
- [29] G. Cravo y A. Amaral, «A GRASP algorithm for solving large-scale single row facility layout problems», *Computers & Operations Research*, vol. 106, págs. 49 -61, 2019.

- [30] A. Klausnitzer y R. Lasch, «Optimal facility layout and material handling network design», *Computers & Operations Research*, vol. 103, págs. 237 -251, 2019.
- [31] L. García-Hernández, A. Arauzo-Azofra, L. Salas-Morera, H. Pierreval y E. Corchado, «Recycling Plants Layout Design by Means of an Interactive Genetic Algorithm», *Intelligent Automation & Soft Computing*, vol. 19, n.º 3, págs. 457-468, 2013.
- [32] B. Ulutas y A. A. Islier, «Dynamic facility layout problem in footwear industry», *Journal of Manufacturing Systems*, vol. 36, págs. 55 -61, 2015.
- [33] S. D. Jena, J.-F. Cordeau y B. Gendron, «Solving a dynamic facility location problem with partial closing and reopening», *Computers & Operations Research*, vol. 67, págs. 143 -154, 2016.
- [34] M. Marufuzzaman, R. Gedik y M. S. Roni, «A Benders based rolling horizon algorithm for a dynamic facility location problem», *Computers & Industrial Engineering*, vol. 98, págs. 462 -469, 2016.

- 
- [35] Y. Xiao, Y. Xie, S. Kulturel-Konak y A. Konak, «A problem evolution algorithm with linear programming for the dynamic facility layout problem—A general layout formulation», *Computers & Operations Research*, vol. 88, págs. 187 -207, 2017.
- [36] S. Vitayasak, P. Pongcharoen y C. Hicks, «A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a Genetic Algorithm or modified Backtracking Search Algorithm», *International Journal of Production Economics*, vol. 190, págs. 146 -157, 2017.
- [37] H. Pourvaziri y H. Pierreval, «Dynamic facility layout problem based on open queuing network theory», *European Journal of Operational Research*, vol. 259, n.º 2, págs. 538 -553, 2017.
- [38] J. Li, X. Tan y J. Li, «Research on Dynamic Facility Layout Problem of Manufacturing Unit Considering Human Factors», *Mathematical Problems in Engineering*, vol. 2018, 2018.
- [39] G. Emirhüseyinoğlu y A. Ekici, «Dynamic facility location with supplier selection under quantity discount», *Computers & Industrial Engineering*, vol. 134, págs. 64 -74, 2019.



- [40] X. Wei, S. Yuan e Y. Ye, «Optimizing facility layout planning for reconfigurable manufacturing system based on chaos genetic algorithm», *Production & Manufacturing Research*, vol. 7, n.º 1, págs. 109-124, 2019.
- [41] M.-J. Wang, M. H. Hu y M.-Y. Ku, «A solution to the unequal area facilities layout problem by genetic algorithm», *Computers in Industry*, vol. 56, n.º 2, págs. 207 -220, 2005.
- [42] A. Diponegoro y B. R. Sarker, «Machine assignment in a nonlinear multi-product flowline», *Journal of the Operational Research Society*, vol. 54, n.º 5, págs. 472-489, 2003.
- [43] G. Palubeckis, «A branch-and-bound algorithm for the single-row equidistant facility layout problem», *OR spectrum*, vol. 34, n.º 1, págs. 1-21, 2012.
- [44] M. Solimanpur, P. Vrat y R. Shankar, «Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing», *European Journal of Operational Research*, vol. 157, n.º 3, págs. 592 -606, 2004.
- [45] S. Wang y B. R. Sarker, «Locating cells with bottleneck machines in cellular manufacturing systems», *International*

- 
- Journal of Production Research*, vol. 40, n.º 2, págs. 403-424, 2002.
- [46] A. Baykasoğlu y N. N. Gindy, «A simulated annealing algorithm for dynamic layout problem», *Computers & Operations Research*, vol. 28, n.º 14, págs. 1403 -1426, 2001.
- [47] M. M. Azevedo, J. A. Crispim y J. P. de Sousa, «A dynamic multi-objective approach for the reconfigurable multi-facility layout problem», *Journal of Manufacturing Systems*, vol. 42, págs. 140 -152, 2017.
- [48] J.-G. Kim e Y.-D. Kim, «Layout planning for facilities with fixed shapes and input and output points», *International Journal of Production Research*, vol. 38, n.º 18, págs. 4635-4653, 2000.
- [49] A. Drira, H. Pierreval y S. Hajri-Gabouj, «Facility layout problems: A survey», *Annual Reviews in Control*, vol. 31, n.º 2, págs. 255-267, 2007.
- [50] B. Montreuil, «A Modelling Framework for Integrating Layout Design and flow Network Design», en *Material Handling '90*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, págs. 95-115, ISBN: 978-3-642-84356-3.

- [51] Q. Liu y R. D. Meller, «A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments», *IIE Transactions*, vol. 39, n.º 4, págs. 377-394, 2007.
- [52] Y. A. Bozer y C.-T. Wang, «A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem», *European Journal of Operational Research*, vol. 218, n.º 2, págs. 382 -391, 2012.
- [53] I. Castillo y T. Westerlund, «An E-accurate model for optimal unequal-area block layout design», *Computers & Operations Research*, vol. 32, n.º 3, págs. 429 -447, 2005.
- [54] H. D. Sherali, B. M. P. Fraticelli y R. D. Meller, «Enhanced Model Formulations for Optimal Facility Layout», *Operations Research*, vol. 51, págs. 629-644, 2003.
- [55] R. D. Meller, W. Chen y H. D. Sherali, «Applying the sequence-pair representation to optimal facility layout designs», *Operations Research Letters*, vol. 35, n.º 5, págs. 651 -659, 2007.
- [56] K. Park, J. Koo, D. Shin, C. J. Lee y E. S. Yoon, «Optimal multi-floor plant layout with consideration of safety dis-

- 
- tance based on mathematical programming and modified consequence analysis», *Korean Journal of Chemical Engineering*, vol. 28, n.º 4, págs. 1009-1018, 2011.
- [57] J. Hathhorn, E. Sisikoglu y M. Y. Sir, «A multi-objective mixed-integer programming model for a multi-floor facility layout», *International Journal of Production Research*, vol. 51, n.º 14, págs. 4223-4239, 2013.
- [58] N. Izadinia, K. Eshghi y M. H. Salmani, «A robust model for multi-floor layout problem», *Computers & Industrial Engineering*, vol. 78, págs. 127 -134, 2014.
- [59] H. Neghabi y F. Ghassemi Tari, «An optimal approach for maximizing the number of adjacencies in multi floor layout problem», *International Journal of Production Research*, vol. 53, n.º 11, págs. 3462-3474, 2015.
- [60] A. Che, Y. Zhang y J. Feng, «Bi-objective optimization for multi-floor facility layout problem with fixed inner configuration and room adjacency constraints», *Computers & Industrial Engineering*, vol. 105, págs. 265 -276, 2017.

- [61] M. S. Bazaraa, «Computerized Layout Design: A Branch and Bound Approach», *A I I E Transactions*, vol. 7, n.º 4, págs. 432-438, 1975.
- [62] J.-G. Kim e Y.-D. Kim, «A branch and bound algorithm for locating input and output points of departments on the block layout», *Journal of the Operational Research Society*, vol. 50, n.º 5, págs. 517-525, 1999.
- [63] D. Kohara, H. Yamamoto y A. Suzuki, «Efficient algorithms based on branch and bound methods for multi floor facility layout problems», en *Proceedings of the 9th Asia Pacific industrial engineering & management systems conference, Bali, Indonesia*, 2008, págs. 387-395.
- [64] M. Karatas y E. Yakıcı, «An iterative solution approach to a multi-objective facility location problem», *Applied Soft Computing*, vol. 62, págs. 272 -287, 2018.
- [65] M. J. Rosenblatt, «The Dynamics of Plant Layout», *Management Science*, vol. 32, n.º 1, págs. 76-86, 1986.
- [66] J. Li y A. E. Smith, «Block layout for attraction-based enterprises», *European Journal of Operational Research*, vol. 266, n.º 3, págs. 1100 -1112, 2018.

- 
- [67] M. Abdel-Basset, G. Manogaran, D. El-Shahat y S. Mirjalili, «Integrating the whale algorithm with Tabu search for quadratic assignment problem: A new approach for locating hospital departments», *Applied Soft Computing*, vol. 73, págs. 530 -546, 2018.
- [68] F. Glover, «Future paths for integer programming and links to artificial intelligence», *Computers & Operations Research*, vol. 13, n.º 5, págs. 533 -549, 1986.
- [69] R. Şahin, K. Ertoğral y O. Türkbey, «A simulated annealing heuristic for the dynamic layout problem with budget constraint», *Computers & Industrial Engineering*, vol. 59, n.º 2, págs. 308 -313, 2010.
- [70] S. Wang, X. Zuo, X. Liu, X. Zhao y J. Li, «Solving dynamic double row layout problem via combining simulated annealing and mathematical programming», *Applied Soft Computing*, vol. 37, págs. 303 -310, 2015.
- [71] K. Y. Wong y Komarudin, «Solving facility layout problems using Flexible Bay Structure representation and Ant System algorithm», *Expert Systems with Applications*, vol. 37, n.º 7, págs. 5523 -5527, 2010.

- [72] Komarudin y K. Y. Wong, «Applying Ant System for solving Unequal Area Facility Layout Problems», *European Journal of Operational Research*, vol. 202, n.º 3, págs. 730 -746, 2010.
- [73] G. Aiello, M. Enea y G. Galante, «A multi-objective approach to facility layout problem by genetic search algorithm and Electre Method», *Robotics and Computer-Integrated Manufacturing*, vol. 22, págs. 447-455, 2006.
- [74] W.-C. Chiang y P Kouvelis, «An improved tabu search heuristic for solving facility layout design problems», *International Journal of Production Research*, vol. 34, n.º 9, págs. 2565-2585, 1996.
- [75] H. Samarghandi y K. Eshghi, «An efficient tabu algorithm for the single row facility layout problem», *European Journal of Operational Research*, vol. 205, n.º 1, págs. 98 -105, 2010.
- [76] R. Kothari y D. Ghosh, «Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods», *European Journal of Operational Research*, vol. 224, n.º 1, págs. 93 -100, 2013.

- 
- [77] D. Romero y A. Sánchez-Flores, «Methods for the one-dimensional space allocation problem», *Computers & Operations Research*, vol. 17, n.º 5, págs. 465 -473, 1990.
- [78] B. Keller y U. Buscher, «Single row layout models», *European Journal of Operational Research*, vol. 245, n.º 3, págs. 629 -644, 2015.
- [79] D. M. Simmons, «One-dimensional space allocation: an ordering algorithm», *Operations Research*, vol. 17, n.º 5, págs. 812-826, 1969.
- [80] J.-C. Picard y M. Queyranne, «On the one-dimensional space allocation problem», *Operations Research*, vol. 29, n.º 2, págs. 371-391, 1981.
- [81] S. S. Heragu y A. Kusiak, «Machine layout problem in flexible manufacturing systems», *Operations research*, vol. 36, n.º 2, págs. 258-268, 1988.
- [82] M. Solimanpur, P. Vrat y R. Shankar, «An ant algorithm for the single row layout problem in flexible manufacturing systems», *Computers & Operations Research*, vol. 32, n.º 3, págs. 583 -598, 2005.



- [83] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller y E. Teller, «Equation of state calculations by fast computing machines», *The journal of chemical physics*, vol. 21, n.º 6, págs. 1087-1092, 1953.
- [84] S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi, «Optimization by Simulated Annealing», *Science*, vol. 220, n.º 4598, págs. 671-680, 1983.
- [85] S. Jajodia, I. Minis, G. Harhalakis y J.-M. Proth, «CLASS: computerized layout solutions using simulated annealing», *The International Journal of Production Research*, vol. 30, n.º 1, págs. 95-108, 1992.
- [86] R. D. Meller e Y. A. Bozer, «A new simulated annealing algorithm for the facility layout problem», *International Journal of Production Research*, vol. 34, n.º 6, págs. 1675-1692, 1996.
- [87] O. Al-Araidah, A. Krishnamurthy y C. J. Malmborg, «A two-stage simulated annealing procedure for block layout problems», *International Journal of Production Research*, vol. 44, n.º 20, págs. 4417-4429, 2006.

- 
- [88] R. Şahin, «A simulated annealing algorithm for solving the bi-objective facility layout problem», *Expert Systems with Applications*, vol. 38, n.º 4, págs. 4460 -4465, 2011.
- [89] G. Palubeckis, «Fast simulated annealing for single-row equidistant facility layout», *Applied Mathematics and Computation*, vol. 263, págs. 287 -301, 2015.
- [90] F. M. Defersha y A. Hodiya, «A mathematical model and a parallel multiple search path simulated annealing for an integrated distributed layout design and machine cell formation», *Journal of Manufacturing Systems*, vol. 43, págs. 195 -212, 2017.
- [91] L. Chwif, M. R. Barretto y L. A. Moscato, «A solution to the facility layout problem using simulated annealing», *Computers in Industry*, vol. 36, n.º 1, págs. 125 -132, 1998.
- [92] J. Grobelny y R. Michalski, «A novel version of simulated annealing based on linguistic patterns for solving facility layout problems», *Knowledge-Based Systems*, vol. 124, págs. 55 -69, 2017.

- [93] J. Grobelny, «The fuzzy approach to facilities layout problems», *Fuzzy Sets and Systems*, vol. 23, n.º 2, págs. 175-190, 1987.
- [94] J. Grobelny, «On one possible fuzzy approach to facilities layout problems», *International Journal of Production Research*, vol. 25, n.º 8, págs. 1123-1141, 1987.
- [95] J. Grobelny, «The linguistic pattern method for a workstation layout analysis», *The International Journal Of Production Research*, vol. 26, n.º 11, págs. 1779-1798, 1988.
- [96] G. Suresh y S. Sahu, «Multiobjective facility layout using simulated annealing», *International Journal of Production Economics*, vol. 32, n.º 2, págs. 239 -254, 1993.
- [97] S. Singh y V. Singh, «An improved heuristic approach for multi-objective facility layout problem», *International Journal of Production Research*, vol. 48, n.º 4, págs. 1171-1194, 2010.
- [98] R. Matai, «Solving multi objective facility layout problem by modified simulated annealing», *Applied Mathematics and Computation*, vol. 261, págs. 302 -311, 2015.

- 
- [99] A. R. McKendall y J. Shang, «Hybrid ant systems for the dynamic facility layout problem», *Computers & Operations Research*, vol. 33, n.º 3, págs. 790 -803, 2006.
- [100] M. Dorigo y L. M. Gambardella, «Ant colony system: a cooperative learning approach to the traveling salesman problem», *IEEE Transactions on Evolutionary Computation*, vol. 1, n.º 1, págs. 53-66, 1997.
- [101] M. Dorigo, V. Maniezzo, A. Colorni y otros, «Ant system: optimization by a colony of cooperating agents», *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, vol. 26, n.º 1, págs. 29-41, 1996.
- [102] N. Çetin Demirel y M. D. Toksarı, «Optimization of the quadratic assignment problem using an ant colony algorithm», *Applied Mathematics and Computation*, vol. 183, n.º 1, págs. 427 -435, 2006.
- [103] J. Yang e Y. Zhuang, «An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem», *Applied Soft Computing*, vol. 10, n.º 2, págs. 653 -660, 2010.

- [104] M. Salari, M. Reihaneh y M. S. Sabbagh, «Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem», *Computers & Industrial Engineering*, vol. 83, págs. 244 -251, 2015.
- [105] X. Xia e Y. Zhou, «Performance analysis of ACO on the quadratic assignment problem», *Chinese Journal of Electronics*, vol. 27, n.º 1, págs. 26-34, 2018.
- [106] A. Baykasoglu, T. Dereli e I. Sabuncu, «An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems», *Omega*, vol. 34, n.º 4, págs. 385 -396, 2006.
- [107] G. Y.-H. Chen, «A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems», *International Journal of Production Economics*, vol. 142, n.º 2, págs. 362-371, 2013.
- [108] C. C. Murray, A. E. Smith y Z. Zhang, «An efficient local search heuristic for the double row layout problem with asymmetric material flow», *International Journal of Production Research*, vol. 51, n.º 20, págs. 6129-6139, 2013.

- 
- [109] Z. Zhang y W. Cheng, «Decomposition strategies and heuristic for double row layout problem», *Computer Integrated Manufacturing Systems*, vol. 20, n.º 3, págs. 559-568, 2014.
- [110] J. Guan, G. Lin, H.-B. Feng y Z.-Q. Ruan, «A decomposition-based algorithm for the double row layout problem», *Applied Mathematical Modelling*, vol. 77, págs. 963 -979, 2020.
- [111] D. E. Goldberg, *Genetic algorithms in Search, Optimization, and Machine Learning*. Adisson-Wesley Professional, 1989.
- [112] T.-J. Chang, S.-C. Yang y K.-J. Chang, «Portfolio optimization problems in different risk measures using genetic algorithm», *Expert Systems with Applications*, vol. 36, n.º 7, págs. 10 529-10 537, 2009.
- [113] S. Safarzadeh y H. Koosha, «Solving an extended multi-row facility layout problem with fuzzy clearances using GA», *Applied Soft Computing*, vol. 61, págs. 819 -831, 2017.
- [114] A. Sanz-Garcia, A. Pernia-Espinoza, R. Fernandez-Martinez y F. Martinez-De-Pison-Ascacibar, «Combining genetic algorithms and the finite element method to improve steel industrial processes», *Journal of Applied Logic*, vol. 10, n.º 4, págs. 298-308, 2012.

- [115] Y. Wang, C. Chen, Y. Tao, Z. Wen, B. Chen y H. Zhang, «A many-objective optimization of industrial environmental management using NSGA-III: A case of China's iron and steel industry», *Applied Energy*, vol. 242, págs. 46 -56, 2019.
- [116] J. Domínguez-Navarro, R. Dufo-López, J. Yusta-Loyo, J. Artal-Sevil y J. Bernal-Agustín, «Design of an electric vehicle fast-charging station with integration of renewable energy and storage systems», *International Journal of Electrical Power & Energy Systems*, vol. 105, págs. 46 -58, 2019.
- [117] R. K. Rahman, S. Ibrahim y A. Raj, «Multi-objective optimization of sulfur recovery units using a detailed reaction mechanism to reduce energy consumption and destruct feed contaminants», *Computers & Chemical Engineering*, vol. 128, págs. 21 -34, 2019.
- [118] J. Arthur, R. Bahran, J. Hutchinson y S. A. Pozzi, «Genetic algorithm for nuclear data evaluation applied to subcritical neutron multiplication inference benchmark experiments», *Annals of Nuclear Energy*, vol. 133, págs. 853 -862, 2019.
- [119] P.-C. Tsai, X. Huang, Y.-H. Hung, C.-H. Huang y S. Smith, «The computer aided cutting planning of components using genetic algorithms for decommissioning of a nuclear reac-

- 
- tor», *Annals of Nuclear Energy*, vol. 130, págs. 200 -207, 2019.
- [120] P. Liu, M. D. E. Basha, Y. Li, Y. Xiao, P. C. Sanelli y R. Fang, «Deep Evolutionary Networks with Expedited Genetic Algorithms for Medical Image Denoising», *Medical Image Analysis*, vol. 54, págs. 306 -315, 2019.
- [121] L. Rundo, A. Tangherloni, M. S. Nobile, C. Militello, D. Besozzi, G. Mauri y P. Cazzaniga, «MedGA: A novel evolutionary method for image enhancement in medical imaging systems», *Expert Systems with Applications*, vol. 119, págs. 387 -399, 2019.
- [122] S. Cui, D. Wang, Y. Wang, P-W. Yu e Y. Jin, «An improved support vector machine-based diabetic readmission prediction», *Computer Methods and Programs in Biomedicine*, vol. 166, págs. 123 -135, 2018.
- [123] A. Badnjević, L. G. Pokvić, M. Hasičić, L. Bandić, Z. Mašetić, Živorad Kovačević, J. Kevrić y L. Pecchia, «Evidence-based clinical engineering: Machine learning algorithms for prediction of defibrillator performance», *Biomedical Signal Processing and Control*, vol. 54, págs. 101-129, 2019.



- [124] S. Torabi y M. Wahde, «A method for performance analysis of a genetic algorithm applied to the problem of fuel consumption minimization for heavy-duty vehicles», *Applied Soft Computing*, vol. 80, págs. 735 -741, 2019.
- [125] J. Liu, B. Ma y H. Zhao, «Combustion parameters optimization of a diesel/natural gas dual fuel engine using genetic algorithm», *Fuel*, vol. 260, pág. 116365, 2020.
- [126] H. Pourvaziri y B. Naderi, «A hybrid multi-population genetic algorithm for the dynamic facility layout problem», *Applied Soft Computing*, vol. 24, págs. 457 -469, 2014.
- [127] B. Turanoğlu y G. Akkaya, «A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem», *Expert Systems with Applications*, vol. 98, págs. 93 -104, 2018.
- [128] J. Guan y G. Lin, «Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem», *European Journal of Operational Research*, vol. 248, n.º 3, págs. 899 -909, 2016.
- [129] F. G. Paes, A. A. Pessoa y T. Vidal, «A hybrid genetic algorithm with decomposition phases for the Unequal Area

- 
- Facility Layout Problem», *European Journal of Operational Research*, vol. 256, n.º 3, págs. 742 -756, 2017.
- [130] J. F. Gonçalves y M. G. Resende, «A biased random-key genetic algorithm for the unequal area facility layout problem», *European Journal of Operational Research*, vol. 246, n.º 1, págs. 86 -107, 2015.
- [131] C. Ou-Yang y A. Utamima, «Hybrid Estimation of Distribution Algorithm for solving Single Row Facility Layout Problem», *Computers & Industrial Engineering*, vol. 66, n.º 1, págs. 95 -103, 2013.
- [132] B. Malakooti y A. Tsurushima, «An expert system using priorities for solving multiple-criteria facility layout problems», *International Journal of Production Research*, vol. 27, n.º 5, págs. 793-808, 1989.
- [133] A. M. Brintup, J. Ramsden y A. Tiwari, «An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization», *Computers in Industry*, vol. 58, págs. 279-291, 2007.
- [134] A. M. Brintup, H. Takagi, A. Tiwari y J. Ramsden, «Evaluation of sequential, multi-objective, and parallel interacti-

- ve genetic algorithms for multi-objective optimization problems», *Journal of Biological Physics and Chemistry*, vol. 6, págs. 137-146, 2006.
- [135] L. García-Hernández, H. Pierreval, L. Salas-Morera y A. Arauzo-Azofra, «Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm», *Appl. Soft Comput.*, vol. 13, n.º 4, págs. 1718-1727, 2013.
- [136] T. Sato y M. Hagiwara, «IDSET: Interactive Design System using Evolutionary Techniques», *Computer-Aided Design*, vol. 33, n.º 5, págs. 367-377, 2001.
- [137] I. Jeong y K. Kim, «An interactive desirability function method to multiresponse optimization», *European Journal of Operational Research*, vol. 195, n.º 2, págs. 412-426, 2009.
- [138] J. C. Quiroz, S. J. Louis, A. Banerjee y S. M. Dascalu, «Towards creative design using collaborative interactive genetic algorithms», en *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, Trondheim, Norway: IEEE Press, 2009, págs. 1849-1856, ISBN: 978-1-4244-2958-5.

- 
- [139] F. Zhao, G. Li, C. Yang, A. Abraham y H. Liu, «A human-computer cooperative particle swarm optimization based immune algorithm for layout design», *Neurocomputing*, vol. 132, págs. 68-78, 2014.
- [140] R. S. Kumar, P. Asokan, S. Kumanan y otros, «An artificial immune system-based algorithm to solve linear and loop layout problems in flexible manufacturing systems», *International Journal of Product Development*, vol. 10, n.º 1-3, págs. 165-179, 2010.
- [141] X.-d. Xu y C.-x. Li, «Research on immune genetic algorithm for solving the job-shop scheduling problem», *The international journal of advanced manufacturing technology*, vol. 34, n.º 7-8, págs. 783-789, 2007.
- [142] H. M. Pandey, A. Chaudhary y D. Mehrotra, «A comparative review of approaches to prevent premature convergence in GA», *Applied Soft Computing*, vol. 24, págs. 1047 -1077, 2014.
- [143] D. B. Fogel, «An introduction to simulated evolutionary optimization», *IEEE transactions on neural networks*, vol. 5, n.º 1, págs. 3-14, 1994.

- [144] D. Whitley, «An overview of evolutionary algorithms: practical issues and common pitfalls», *Information and Software Technology*, vol. 43, n.º 14, págs. 817 -831, 2001.
- [145] T. Friedrich, P. S. Oliveto, D. Sudholt y C. Witt, «Analysis of diversity-preserving mechanisms for global exploration», *Evolutionary Computation*, vol. 17, n.º 4, págs. 455-476, 2009.
- [146] M. Kurdi, «An effective new island model genetic algorithm for job shop scheduling problem», *Computers & Operations Research*, vol. 67, págs. 132 -142, 2016.
- [147] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992, ISBN: 0-262-58111-6.
- [148] C. Darwin, *On the Origin of the Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life*. John Murray, 1859.
- [149] L. García-Hernández, A. Araúzo-Azofra, H. Pierreval y L. Salas-Morera, «Encoding Structures and Operators Used in Facility Layout Problems with Genetic Algorithms», en *ISDA '09: Proceedings of the 2009 Ninth International Conference*

- 
- on Intelligent Systems Design and Applications*, Washington, DC, USA: IEEE Computer Society, 2009, págs. 43-48, ISBN: 978-0-7695-3872-3.
- [150] J. S. Kochhar, B. T. Foster y S. S. Heragu, «HOPE: A genetic algorithm for the unequal area facility layout problem», *Computers & Operations Research*, vol. 25, n.º 7-8, págs. 583-594, 1998.
- [151] G. Aiello, G. L. Scalia y M. Enea, «A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems», *Expert Systems with Applications*, vol. 40, n.º 12, págs. 4812-4819, 2013.
- [152] X. Sun, L.-F. Lai, P. Chou, L.-R. Chen y C.-C. Wu, «On GPU implementation of the island model genetic algorithm for solving the unequal area facility layout problem», *Applied Sciences (Switzerland)*, vol. 8, n.º 9, 2018.
- [153] E. Israeli y E. Gilad, «Novel genetic algorithm for loading pattern optimization based on core physics heuristics», *Annals of Nuclear Energy*, vol. 118, págs. 35-48, 2018.
- [154] S. Maity, A. Roy y M. Maiti, «A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling

- Salesman Problems», *Computers & Industrial Engineering*, vol. 83, págs. 273 -296, 2015.
- [155] N. Xie y N. Chen, «Flexible job shop scheduling problem with interval grey processing time», *Applied Soft Computing*, vol. 70, págs. 513 -524, 2018.
- [156] J. Lozano, P. Larrañaga, M. Graña y F. Albizuri, «Genetic algorithms: bridging the convergence gap», *Theoretical Computer Science*, vol. 229, n.º 1, págs. 11 -22, 1999.
- [157] T. Kellegöz, B. Toklu y J. Wilson, «Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem», *Applied Mathematics and Computation*, vol. 199, n.º 2, págs. 590 -598, 2008.
- [158] M. Gen y R. Cheng, *Genetic Algorithms and Engineering Design*, ép. Engineering Design and Automation. Wiley, 1997, ISBN: 9780471127413.
- [159] D. E. Goldberg, R. Lingle y otros, «Alleles, loci, and the traveling salesman problem», en *Proceedings of an international conference on genetic algorithms and their applications*, Lawrence Erlbaum, Hillsdale, NJ, vol. 154, 1985, págs. 154-159.

- 
- [160] I. Oliver, D. Smith y J. R. Holland, «Study of permutation crossover operators on the traveling salesman problem», en *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*, Hillsdale, NJ: L. Erlbaum Associates, 1987., 1987.
- [161] L. Davis, «Applying adaptive algorithms to epistatic domains.», en *IJCAI*, vol. 85, 1985, págs. 162-164.
- [162] J. Grefenstette, «Genetic Algorithms for Changing Environments», en *Parallel Problem Solving from Nature 2*, Elsevier, 1992, págs. 137-144.
- [163] A. Kusiak y S. S. Heragu, «The facility layout problem», *European Journal of Operational Research*, vol. 29, n.º 3, págs. 229-251, 1987.
- [164] A. Ahmadi, M. S. Pishvaei y M. R. A. Jokar, «A survey on multi-floor facility layout problems», *Computers & Industrial Engineering*, vol. 107, págs. 158 -170, 2017.
- [165] D. Scholz, A. Petrick y W. Domschke, «STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area



- facility layout problem», *European Journal of Operational Research*, vol. 197, n.º 1, págs. 166 -178, 2009.
- [166] B. H. Ulutas y S. Kulturel-Konak, «An artificial immune system based algorithm to solve unequal area facility layout problem», *Expert Systems with Applications*, vol. 39, n.º 5, págs. 5384 -5395, 2012.
- [167] J. Liu, D. Wang, K. He e Y. Xue, «Combining Wang–Landau sampling algorithm and heuristics for solving the unequal-area dynamic facility layout problem», *European Journal of Operational Research*, vol. 262, n.º 3, págs. 1052 -1063, 2017.
- [168] S. Kang y J. Chae, «Harmony search for the layout design of an unequal area facility», *Expert Systems with Applications*, vol. 79, págs. 269 -281, 2017.
- [169] J. Liu, H. Zhang, K. He y S. Jiang, «Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem», *Expert Systems with Applications*, vol. 102, págs. 179 -192, 2018.
- [170] M. Z. Allahyari y A. Azab, «Mathematical modeling and multi-start search simulated annealing for unequal-area fa-

- 
- ility layout problem», *Expert Systems with Applications*, vol. 91, págs. 46 -62, 2018.
- [171] J. Liu y J. Liu, «Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems», *Applied Soft Computing*, vol. 74, págs. 167 -189, 2019.
- [172] G. C. Lee e Y. D. Kim, «Algorithms for adjusting shapes of departments in block layouts on the grid-based plane», *Omega*, vol. 28, n.º 1, págs. 111-122, 2000.
- [173] G. Aiello, G. L. Scalia y M. Enea, «A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding», *Expert Systems with Applications*, n.º 0, 2012.
- [174] R. A. Arapoglu, B. A. Norman y A. E. Smith, «Locating input and output points in facilities design - a comparison of constructive, evolutionary, and exact methods», *IEEE Transactions on Evolutionary Computation*, vol. 5, n.º 3, págs. 192-203, 2001.

- [175] R. D. Meller, «The multi-bay manufacturing facility layout problem», *International Journal of Production Research*, vol. 35, n.º 5, págs. 1229-1237, 1997.
- [176] S. Kulturel-Konak y A. Konak, «A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem», *Engineering Optimization*, vol. 43, n.º 12, págs. 1263-1287, 2011.
- [177] S. Kang y J. Chae, «Harmony search for the layout design of an unequal area facility», *Expert Systems with Applications*, vol. 79, págs. 269 -281, 2017.
- [178] S. Kulturel-Konak, «A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays», *European Journal of Operational Research*, vol. 223, n.º 3, págs. 614 -625, 2012.
- [179] A. Asef-Vaziri, H. Jahandideh y M. Modarres, «Loop-based facility layout design under flexible bay structures», *International Journal of Production Economics*, vol. 193, págs. 713 -725, 2017.
- [180] J. Liu y J. Liu, «Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout

- 
- problems», *Applied Soft Computing*, vol. 74, págs. 167 -189, 2019.
- [181] A. Gomez, Q. Fernandez, D. D. la Fuente Garcia y P. Garcia, «Using genetic algorithms to resolve layout problems in facilities where there are aisles», *International Journal of Production Economics*, vol. 84, n.º 3, págs. 271 -282, 2003.
- [182] S. W. Mahfoud, «Niching Methods for Genetic Algorithms», Tesis doct., Champaign, IL, USA, 1995.
- [183] A. Pétrowski, «A Clearing Procedure as a Niching Method for Genetic Algorithms», en *International Conference on Evolutionary Computation (1996)*, 3 de ene. de 2002, págs. 798-803.
- [184] B. Sareni y L. Krahenbuhl, «Fitness sharing and niching methods revisited», *IEEE transactions on Evolutionary Computation*, vol. 2, n.º 3, págs. 97-106, 1998.
- [185] B. Qu, J. Liang y P. Suganthan, «Niching particle swarm optimization with local search for multi-modal optimization», *Information Sciences*, vol. 197, págs. 131 -143, 2012.

- [186] S. Das, S. Maity, B.-Y. Qu y P. Suganthan, «Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art», *Swarm and Evolutionary Computation*, vol. 1, n.º 2, págs. 71 -88, 2011.
- [187] S. Gören y H. Pierreval, «Taking advantage of a diverse set of efficient production schedules: A two-step approach for scheduling with side concerns», *Computers & Operations Research*, vol. 40, n.º 8, págs. 1979 -1990, 2013.
- [188] X. Yan, J. Zhao, C. Hu y D. Zeng, «Multimodal optimization problem in contamination source determination of water supply networks», *Swarm and Evolutionary Computation*, vol. 47, págs. 66 -71, 2019.
- [189] W. Luo, X. Lin, T. Zhu y P. Xu, «A clonal selection algorithm for dynamic multimodal function optimization», *Swarm and Evolutionary Computation*, vol. 50, pág. 100 459, 2019.
- [190] X. Wang, M. Sheng, K. Ye, J. Lin, J. Mao, S. Chen y W. Sheng, «A multilevel sampling strategy based memetic differential evolution for multimodal optimization», *Neurocomputing*, vol. 334, págs. 79 -88, 2019.

- 
- [191] E. Yu y P. Suganthan, «Ensemble of niching algorithms», *Information Sciences*, vol. 180, n.º 15, págs. 2815 -2833, 2010.
- [192] S. W. Mahfoud, «Crowding and preselection revisited», *Parallel Problem Solving from Nature*, págs. 27-36, 1992.
- [193] K. A. De Jong, «An analysis of the behavior of a class of genetic adaptive systems», Tesis doct., University of Michigan, Ann Arbor, 1975.
- [194] G. R. Harik, «Finding multimodal solutions using restricted tournament selection», en *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995, págs. 24-31, ISBN: 1558603700.
- [195] P. Claudio y C. Lapa, «Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem», *Annals of Nuclear Energy*, vol. 30, págs. 555-565, 2003.
- [196] M. Mezmoz, N. Melab, Y. Kessaci, Y. Lee, E.-G. Talbi, A. Zomaya y D. Tuyttens, «A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud compu-

- ting systems», *Journal of Parallel and Distributed Computing*, vol. 71, n.º 11, págs. 1497 -1508, 2011.
- [197] Z. kai Feng, W. jing Niu y C. tian Cheng, «Optimization of hydropower reservoirs operation balancing generation benefit and ecological requirement with parallel multi-objective genetic algorithm», *Energy*, vol. 153, págs. 706 -718, 2018.
- [198] J. Luo, S. Fujimura, D. E. Baz y B. Plazolles, «GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem», *Journal of Parallel and Distributed Computing*, vol. 133, págs. 244 -257, 2019.
- [199] Y. Zhang e Y. Zhou, «Distributed coordination control of traffic network flow using adaptive genetic algorithm based on cloud computing», *Journal of Network and Computer Applications*, vol. 119, págs. 110 -120, 2018.
- [200] K. Nitisiri, M. Gen y H. Ohwada, «A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling», *Computers & Industrial Engineering*, vol. 130, págs. 381 -394, 2019.

- 
- [201] X. Hu, X. Chen, Y. Zhao y W. Yao, «Optimization design of satellite separation systems based on Multi-Island Genetic Algorithm», *Advances in Space Research*, vol. 53, n.º 5, págs. 870 -876, 2014.
- [202] N. Sitkoff, M. Wazlowski, A. Smith y H. Silverman, «Implementing a genetic algorithm on a parallel custom computing machine», en *Proceedings IEEE Symposium on FPGAs for Custom Computing Machines*, 1995, págs. 180-187.
- [203] E. Cantú-Paz, «A Survey of Parallel Genetic Algorithms», *Calculateurs Paralleles*, vol. 10, 1998.
- [204] K. Y. Tam, «Solving facility layout problems with geometric constraints using parallel genetic algorithms: Experimentation and findings», *International Journal of Production Research*, vol. 36, n.º 12, págs. 3253-3272, 1998.
- [205] L. Al-Hakim, «On solving facility layout problems using genetic algorithms», *International Journal of Production Research*, vol. 38, n.º 11, págs. 2573-2582, 2000.
- [206] D. Whitley, S. Rana y R. Heckendorn, «The Island Model Genetic Algorithm: On Separability, Population Size and



Convergence», *Journal of Computing and Information Technology*, vol. 7, 1998.

- [207] T. Starkweather, D. Whitley y K. Mathias, «Optimization using distributed genetic algorithms», en *Parallel Problem Solving from Nature*, H.-P. Schwefel y R. Manner, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, págs. 176-185, ISBN: 978-3-540-70652-6.
- [208] H. Mühlenbein, «Evolution in Time and Space – The Parallel Genetic Algorithm», en, ép. *Foundations of Genetic Algorithms*, G. J. Rawling, ed., vol. 1, Elsevier, 1991, págs. 316-337.
- [209] Y. A. Bozer y R. D. Meller, «A reexamination of the distance-based facility layout problem», *IIE Transactions*, vol. 29, n.º 7, págs. 549-560, 1997.
- [210] D. J. van Camp, M. W. Carter y A. Vannelli, «A nonlinear optimization approach for solving facility layout problems», *European Journal of Operational Research*, vol. 57, n.º 2, págs. 174 -189, 1992.

- 
- [211] J. C. Bezdek, R. Ehrlich y W. Full, «FCM: The fuzzy c-means clustering algorithm», *Computers & Geosciences*, vol. 10, n.º 2-3, págs. 191-203, 1984.
- [212] S. W. Mahfound, «Crowding and Preselection Revisited», en *Parallel Problem Solving from Nature, 2: Proceedings of the Second Conference on Parallel Problem Solving from Nature*, vol. 2, 1992, págs. 27-36.
- [213] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López y J. A. Portilla-Figueras, «The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems», *Scientific World Journal*, vol. 2014, 2014.
- [214] S. Salcedo-Sanz, C. Camacho-Gómez, R. Mallol-Poyato, S. Jiménez-Fernández y J. DelSer, «A novel Coral Reefs Optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids», *Soft Computing*, vol. 20, n.º 11, págs. 4287-4300, 2016.
- [215] A. A. Gozali y S. Fujimura, «Localization Strategy for Island Model Genetic Algorithm to Preserve Population Diversity», en *Computer and Information Science*, R. Lee, ed. Cham: Springer International Publishing, 2018, págs. 149-161, ISBN: 978-3-319-60170-0.



# ANEXOS

---



A

---

A NOVEL HYBRID EVOLUTIONARY APPROACH FOR  
CAPTURING DECISION MAKER KNOWLEDGE INTO  
THE UNEQUAL AREA FACILITY LAYOUT PROBLEM

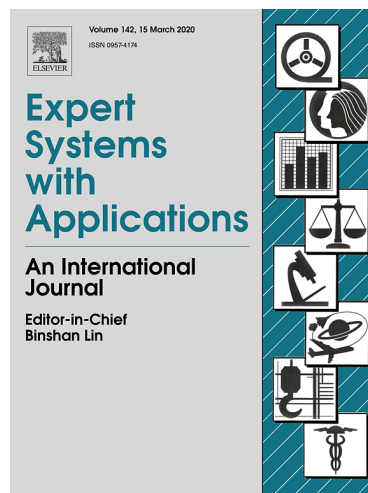
---



TÍTULO: A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem.

AUTORES: Laura García Hernández, Juan María Palomo Romero, Lorenzo Salas Morera, Antonio Arauzo Azofra y Henri Pierreval.

PUBLICACIÓN: Expert Systems with Applications, 42 (10), páginas 4697-4708, 2015. DOI: 10.1016/j.eswa.2015.01.037



### RANKING JCR 2015

FACTOR DE IMPACTO: 2.981

Categoría	Cuartil	Ranking
Operations Research & Management Science	Q1 <sup>a</sup>	6/82
Engineering, Electrical & Electronic	Q1	27/257
Computer Science, Artificial Intelligence	Q1	19/130

<sup>a</sup> La revista se encuentra situada en el **decil 1** de la categoría.



MÉTRICAS SCOPUS

FECHA DE CONSULTA: noviembre de 2019

CITAS: 26

FIELD-WEIGHTED CITATION IMPACT: 4.29 \*

<b>Categoría</b>	<b>CiteScore</b>
<b>Categoría</b>	<b>Percentile <sup>a</sup></b>
Computer Science	93 %
Engineering	93 %
Computer Science Applications	92 %
Artificial Intelligence	91 %

<sup>a</sup> Indica la posición relativa de un artículo en su campo de estudio. Cada campo se divide en 100 percentiles de igual tamaño, siendo 0% el menor, y 99% el mayor.

\* Índice que calcula cuánto se cita este artículo en comparación con artículos similares. Un valor mayor que 1.00 significa que el artículo se cita más de lo esperado.



## A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem



Laura García-Hernández<sup>a,\*</sup>, Juan M. Palomo-Romero<sup>a</sup>, Lorenzo Salas-Morera<sup>a</sup>, Antonio Arauzo-Azofra<sup>a</sup>, Henri Pierreval<sup>b</sup>

<sup>a</sup> Area of Project Engineering, University of Córdoba, Spain

<sup>b</sup> LIMOS UMR CNRS 6158 IFMA, Clermont-Ferrand, France

### ARTICLE INFO

#### Article history:

Available online 9 February 2015

#### Keywords:

Unequal area facility layout problem  
Interactive evolutionary computation  
Expert knowledge  
Nicheing method

### ABSTRACT

Introducing expert knowledge into evolutionary algorithms for the facility layout design problem can provide better solutions than the mathematically optimal solutions by considering qualitative aspects in the design. However, this approach requires the direct intervention of a designer (normally called the decision maker) in the evolutionary algorithm that guides the search process to adjust it to his/her preferences. To do this, the designer scores each of the most representative designs of the population to avoid fatigue. The selection of the solutions to be presented for human assessment is crucial, so a small number of solutions that represents the characteristics of the population must be selected without losing the variability of the solutions. The novel hybrid system proposed in this study consists of an interactive genetic algorithm that is combined with two different niching methods to allow interactions between the algorithm and the expert designer. The inclusion of niching techniques into the approach allows for the preservation of diversity, which avoids presenting similar solutions to the designer in the same iteration of the algorithm. The proposed approach was tested using two case studies of facility layout designs. The results of the experiments, which successfully validate the approach, are presented, compared and discussed.

© 2015 Elsevier Ltd. All rights reserved.

### 1. Introduction

The placement of facilities in a plant, which is often referred to as the *facility layout problem*, is known to have a significant impact on manufacturing costs, work in process, lead times and productivity (Drira, Pierreval, & Hajri-Gabouj, 2007). Where to locate facilities and the efficient design of those facilities are important and fundamental strategic issues in any manufacturing industry (Singh & Sharma, 2006). Well laid out facilities contribute to the overall efficiency of operations and can reduce total operating costs by between 20% and 50% (Tompkins, White, Bozer, & Tanchoco, 2010).

Depending on the features that are considered, several problems are included in the field of facility layout design (FLD) (Kusiak & Heragu, 1987). In particular, the novel proposal

presented in this article focuses on the unequal area facility layout problem (UA-FLP) as described by Armour and Buffa (Armour & Buffa, 1963), who formulated the UA-FLP as a rectangular plant layout that is composed of unequal rectangular facilities that should be arranged in an effective way.

FLD problems can be modeled using linear integer programming, mixed integer programming (Izadinia, Eshghi, & Salmani, 2014) and Graph Theoretic methods (Heragu & Kusiak, 1991). However, because the number of facilities limits the application of optimal methods, suboptimal methods have been used to solve more complex problems. Several techniques have been applied, such as the branch and bound method (Solimanpur & Jafari, 2008) and graph theory (Kim & Kim, 1995). Recently, researchers have focused on meta-heuristic methods, such as tabu search (McKendall et al., 2006; Scholz, Petrick, & Domschke, 2009), simulated annealing (Şahin, Ertoğral, & Türkbey, 2010), the ant system (Kumarudin & Wong, 2010, 2012a) and genetic algorithms (GAs) (Aiello, Scalia, & Enea, 2013; Goldberg, 1989). The latter methods have been commonly used in UA-FLP (García-Hernández, Arauzo-Azofra, Pierreval, & Salas-Morera, 2009; Michalewicz, Dasgupta, Riche, & Schoenauer, 1996).

\* Corresponding author at: Area of Project Engineering, University of Córdoba, Rabanales Campus, Leonardo Da Vinci Building, 14071 Córdoba, Spain. Fax: +34 957 218 550.

E-mail addresses: [irlgahel@uco.es](mailto:irlgahel@uco.es) (L. García-Hernández), [i52paroj@uco.es](mailto:i52paroj@uco.es) (J.M. Palomo-Romero), [lsalas@uco.es](mailto:lsalas@uco.es) (L. Salas-Morera), [arauzo@uco.es](mailto:arauzo@uco.es) (A. Arauzo-Azofra), [henri.pierreval@ifma.fr](mailto:henri.pierreval@ifma.fr) (H. Pierreval).

In general, the problem of designing a physical layout involves the optimization and consideration of certain objectives (mainly, the material flow) (AIELLO, Scalia, & Enea, 2012) and criteria. However, Babbar-Sebens and Minsker (2012) stated that these optimization approaches may not appropriately describe all of the important qualitative information that is considered to be essential by a human expert who is involved in the design phase (e.g., engineers). However, it is difficult to take into account these qualitative features using a classical heuristic or meta-heuristic optimization system (Brintup, Ramsden, & Tiwari, 2007). Supposedly, the most reasonable way to consider all of these features in the selection process is to personally include the decision maker (DM) in the process (Brintup et al., 2007), which provides additional advantages such as eliminating the requirement of specifying all of the desirable information about the facility design in advance, offering the DM the ability to learn about his/her own preferences, and stimulating the users creativity (García-Hernández, Pierreval, Salas-Morera, & Arauzo-Azofra, 2013).

Brintup, Takagi, Tiwari, and Ramsden (2006) highlighted the fact that interactive evolutionary computation (IEC) can greatly contribute to improving the optimized design by involving users in searching for a satisfactory solution. However, in real-world optimization problems, we are sometimes not satisfied with only one optimal solution. In this sense, GAs often lack the ability to find multiple optima. Many authors have used niching methods to solve this problem, which can maintain a diverse population and are not as prone to converging prematurely as simple GAs (Yu & Suganthan, 2010).

Few studies have incorporated human expert knowledge into the FLD process. Quiroz, Louis, Banerjee, and Dascalu (2009) suggested an approach to handle collaborative design issues in constructing floor plans. Zhao, Li, Yang, Abraham, and Liu (2014) proposed an approach for packing problems that consisted of a human-computer cooperative particle swarm optimization-based immune algorithm. Although they considered the designer knowledge in their approach by asking him/her for possible preliminary designs, the algorithm does not allow the designer to evaluate the solutions that are obtained by the algorithm. Additionally, this approach does not allow the designer to change his/her initial preferences to new ones that could appear during the process. Moreover, in this approach, the designer must know his/her preferences perfectly before the beginning of the computing process, which could be difficult because these preferences may be unclear at this point. García-Hernández et al. (2013) presented a proposal to consider qualitative features in UA-FLP. Although it improved upon previous methods, the algorithm was mainly focused on introducing designer knowledge by means of his/her evaluation of representative solutions that were selected from the entire population using a clustering method. Unfortunately, the selection method resulted in a lack of diversity in the population, so similar solutions were frequently presented to the DM in the same algorithm iteration and during the entire process. This caused fatigue for the DM, which led him/her to evaluate inappropriate solutions (which had been evaluated previously) and unnecessarily extended the duration of the design process. This was pointed out in the conclusions of this study, where the authors introduced the necessity of investigating new ways to avoid tiring the DM. To provide a better way to preserve the population diversity. It could be interesting to explore the possibilities of introducing new selection techniques to improve the set of solutions that will be shown to the DM while not allowing the same solution to appear more than once in the same evaluation set. Moreover, to the best of our knowledge, no studies have addressed the UA-FLP and are able to provide solutions that satisfy the DMs qualitative preferences while preserving the population diversity.

This article presents a novel hybrid evolutionary algorithm (HEA) for incorporating human expert knowledge into the UA-FLP. This approach consists of an interactive genetic algorithm that is combined with two niching methods to allow interactions between the algorithm process and the expert human designer while preserving the diversity of the population of solutions.

The remainder of this article is organized as follows. The problem is formulated in Section 2. Section 3 provides an explanation of the computational models that are used in our proposed method. Section 4 details how the suggested new HEA works. In Section 5, the methodology is tested, and the obtained results are analyzed. Finally, concluding remarks and future work are presented in Section 6.

## 2. Problem formulation

The UA-FLP was proposed by Armour and Buffa (1963). It considers a rectangular plant with fixed dimensions,  $W$  (width)  $\times$   $H$  (height), and a set of facilities that each has a required area ( $A_i$ ), where the sum of the facility areas must be less than or equal to the total plant area; see Eq. (1). The aim is to allocate the facilities in the plant based on a given optimization criterion subject to the non-overlapping restriction of the facilities.

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

Many authors have taken into account quantitative criteria or constraints for the UA-FLP, including material handling costs, adjacency requests, distance requirements or a desired aspect ratio. However, as was discussed in the previous section, other qualitative features could be considered by the DM in the facility layout design. These features are difficult to consider with an optimization method because they are difficult to quantify, change depend on the design case and are frequently unknown at the beginning of the process. This study specifically emphasizes these qualitative features. Thus, the aim of this proposal is to find solutions that satisfy the DM's preferences, which could differ greatly between DMs. For example, the DM's preferences could be those proposed by García-Hernández et al. (2013), which are:

- The distribution of the remaining space in the plant layout. The DM may prefer designs in which the remaining space is dispersed all over the plant (for example, for use as storage areas), or in which all of the remaining space is aggregated in a particular location.
- The facility placement. This aspect implies the DM's preference about the location of a certain facility in the layout, which could be at the top, bottom, center or corner of the plant.
- The orientation that the DM prefers for a facility to match the sequence of the production process. For example, the DM may want the longest side of a facility to be parallel to the right side of the plant.
- Locations that are not desired for a certain facility to avoid certain factors (e.g., noise, bad smells, humidity) that exist in the plant.
- Any other subjective preference that the DM considers to be necessary in the final facility layout design.

## 3. Bio-inspired systems

This section describes the bio-inspired computational models (Helmy, Fatai, & Faisal, 2010) that have been applied in our novel approach to solve the UA-FLP based on qualitative features.

### 3.1. Genetic algorithms

GAs (Holland, 1992) are computational methods that are based on the principles of natural selection that were proposed by Darwin, 1859. According to Darwin, in nature, individuals of a population compete against themselves to obtain resources (such as food and water) and to find a partner to procreate. The more successful individuals of the population will have a greater probability to reproduce and create new offspring than the other individuals.

According to Holland (1992), GAs have a direct correspondence with natural behavior. These algorithms manipulate a population of individuals that represent candidate solutions to a problem. Depending on the solution's goodness, a fitness (or score) is assigned to each individual of the population. High individual fitness values lead to greater chances for the individuals to be selected to combine their genetic material with another individual. The crossover operation will produce new individuals who will share the characteristics of their parents. Thus, a new population of generally improved individuals is generated and will replace the previous population. A mutation process is then applied with a low probability of randomly altering genes within the chromosome.

GAs have been commonly used to solve optimization and search problems (Chang, Yang, & Chang, 2009). Moreover, these techniques have been successfully applied to other fields, such as steel industrial processes (Sanz-García, Pernía-Espinoza, Fernández-Martínez, & de Pisón Ascacibar, 2012), content-based image retrieval (Arevalillo-Herraez, Ferri, & Moreno-Picot, 2013), and chopped plastic recycling plants (García-Hernández, Arauzo-Azofra, Salas-Morera, Pierreval, & Corchado, 2013).

### 3.2. Clustering method

A clustering method (MacQueen, 1967) is a technique of grouping sets of objects into clusters whose members have similar characteristics. Clustering is sometimes referred to as a method of unsupervised learning (Pimentel & de Souza, 2013). In this study, a clustering method is used to group a population into clusters (with similar characteristics) and select a representative element from each (called the centroid). This allows a representative subset of the solutions of the population to be shown to avoid tiring the DM with too many evaluations during the process.

Due to the number and complexity of the features that define the UA-FLP individuals, it is preferable to allow each population element to belong to more than one cluster simultaneously. Therefore, it can receive inherited evaluations from more than one centroid, which allows the algorithm to accurately adjust it to the scores of the elements that are near the bounds of the clusters. To do this, the fuzzy c-means clustering method (described by Bezdek, Ehrlich, & Full (1984)) has been selected.

The c-means clustering method is based on minimizing the following objective equation:

$$F_m = \sum_{i=1}^c \sum_{j=1}^n m_{ij}^f \|x_j - c_i\|^2 \tag{2}$$

The membership grade of each individual with respect to each representative element of the cluster is calculated as follows:

$$m_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_j - c_i\|}{\|x_j - c_k\|} \right)^{\frac{2}{f}}} \tag{3}$$

The centroid of each cluster is computed using Eq. 4:

$$c_i = \frac{\sum_{j=1}^n m_{ij}^f x_j}{\sum_{j=1}^n m_{ij}^f} \tag{4}$$

where  $F_m$  is the function used to minimize,  $m_{ij}$  is the membership grade of individual  $j$  in cluster  $i$ ,  $c$  is the number of clusters,  $n$  is the number of individuals,  $f$  is the fuzziness parameter, which is any real number greater than 1,  $x_j$  is individual  $j$  of the population,  $c_i$  is the centroid of cluster  $i$ , and  $\| \cdot \|$  is any norm that defines the similarity between the element and the centroid of the cluster. This study uses the Euclidean squared distance as the metric.

The procedure for applying the clustering method is described below:

1.  $M_0 = [m_{ij}]_0$ . The membership matrix is initialized with uniform random numbers in the range of [0, 1] such that the members of an element sum to 1.
2. During iteration  $k$ , the centroids of the clusters are calculated using Eq. 4.
3. The matrix  $M_k$  is updated with the centroid values that were obtained in step 2, using Eq. 3.
4. If  $\|M_{k+1} - M_k\| = \max_{ij} |m_{ij}|_{k+1} - [m_{ij}]_k| < \varepsilon$  or the algorithm reaches the iteration  $k_{max}$ , the process ends. Else, the process continues and goes to step 2.  $\varepsilon$  is the criterion that defines the maximum error that is allowed by the algorithm. In our case,  $\varepsilon = 0.001$  and  $k_{max} = 25$  as suggested by Bezdek et al., 1984.

### 3.3. Nicheing techniques

Nicheing is a generic term that refers to a technique of finding and preserving multiple stable niches or favorable parts of the solution landscape, possibly around local optima, to prevent convergence to a single solution (Gren & Pierreval, 2013).

A nicheing method must be able to form and maintain multiple diverse final solutions regardless of whether these solutions are of identical fitness or of varying fitness. A nicheing method must maintain these solutions for an exponential to infinite time period with respect to the population size (Mahfoud, 1995).

According to Yu and Suganthan (2010), in evolutionary computation, nicheing involves the formation of subgroups within a population in which each subgroup targets a specific task, such as discovering one peak in a multimodal function. Compared with the simple GA, the uniqueness of nicheing GAs lies in the fact that they preserve both highly fit individuals and weaker individuals as long as they belong to groups without the highly fit ones. This gives the population the opportunity to pass the genetic information of such individuals to their offspring and ensures that this information does not become extinct quickly. By maintaining a reasonably balanced variety of genetic information, nicheing procedures allow the population to simultaneously focus on more than one region in the search space, which is essential to discovering several optima in a single run.

The first nicheing method was proposed by Cavicchio (1970), whose approach defines the preselection scheme to preserve the diversity of genetic searches. A design that is created through gene variation will replace the parent that is most similar to the new design (Lin & Wu, 2002). Since then, several nicheing methods have emerged, such as sequential nicheing (Beasley, Bull, & Martin, 1993; Zhang, Huang, Lok, & Lyu, 2006), crowding (De Jong, 1975; Martínez-Alvarez, Calvo-Zaragoza, Cuenca-Asensi, Ortiz, & Jimeno-Morenila, 2014; Yannibelli & Amandi, 2012), sharing (Dong et al., 2011; Goldberg & Richardson, 1987), clearing (Fayek, Darwish, & Ali, 2010; Pétrowski, 1996) and immune systems (fu Li, Wang, Xiao, & Yin, 2014; Ulutas & Kulturel-Konak, 2012b).

In this article, two different algorithms that are based on the crowding method are used: deterministic crowding (DC) and restricted tournament selection (RTS). Crowding methods are inspired by a corresponding ecological phenomenon. In a natural population, similar individuals compete against each other for

limited resources, and dissimilar individuals normally do not compete (they take up different niches). The end result is that in a fixed-size population at equilibrium, new members of a particular species replace older members of that species; the total number of members of a particular species does not change (Mahfoud, 1992).

3.3.1. Deterministic crowding

De Jong (1975) proposed the crowding scheme based on Cavicchios dissertation. The algorithm allows only a fraction of the population to reproduce and die in each generation. A percentage of the population, which is specified by the generation gap, is chosen to reproduce via fitness proportionate roulette wheel selection. For every newly created individual, CF (Crowding Factor) individuals are randomly taken from the population, and the most similar to the new individual is replaced. Fitness values are not considered in the replacement procedure (Yu & Suganthan, 2010).

Mahfoud (1995) suggested an improvement for crowding called DC. In this algorithm, the diversity of the offspring is preserved based on their fitness and the similarity between their parents and themselves. No new parameters are added in this approach. This algorithm is implemented by the following steps:

1. From the current population ( $p$ ), randomly select two different parents ( $p_1, p_2$ ).
2. Through crossover and mutation (if applicable), obtain two offspring ( $c_1, c_2$ ) from the selected parents ( $p_1, p_2$ ).
3. Compute the fitness of the parents and offspring ( $f(p_1), f(p_2), f(c_1), f(c_2)$ ).
4. Compute the distances between the parents and their offspring. Four different distances are calculated:  $d(p_1, c_1), d(p_1, c_2), d(p_2, c_1)$  and  $d(p_2, c_2)$ .
5. The offspring are generated using the following algorithm:

```

if  $d(p_1, c_1) + d(p_2, c_2) > d(p_1, c_2) + d(p_2, c_1)$ 
then
  if  $f(p_1) < f(c_1)$  then
    Exchange  $p_1$  for  $c_1$ 
  end if
  if  $f(p_2) < f(c_2)$  then
    Exchange  $p_2$  for  $c_2$ 
  end if
else
  if  $f(p_1) < f(c_2)$  then
    Exchange  $p_1$  for  $c_2$ 
  end if
  if  $f(p_2) < f(c_1)$  then
    Exchange  $p_2$  for  $c_1$ 
  end if
end if
    
```

3.3.2. Restricted tournament selection

Restricted tournament selection (RTS) is a modification of regular binary tournament selection, which is based on local competition, for multimodal optimization (Harik, 1995). The increasing fitness of individuals will cause increased competition between members of the same niche, which results in the improvement of each independent niche. The algorithm's steps are described below:

1. From the current population ( $p$ ), randomly select two different parents ( $p_1, p_2$ ).
2. Through crossover and mutation (if applicable), obtain two offspring ( $c_1, c_2$ ) from the selected parents ( $p_1, p_2$ ).

3. Randomly select  $w$  individuals from  $p$ , and from these find the closest ones to  $c_1$  ( $c'_1$ ) and  $c_2$  ( $c'_2$ ). The distance measures are Euclidean (for real variables) or Hamming (for binary variables).
4. For each pair of unique elements ( $\{c_1, c'_1\}; \{c_2, c'_2\}$ ), select the one with the best fitness. These two individuals will go to the next generation.

4. Proposed approach

4.1. Layout representation

The flexible bay structure (FBS) proposed by Tong (1991) is used to represent the plant layout. The rectangular plant is defined by its height and width dimensions. This area is divided in one direction into bays of different widths, which are subdivided to allocate the facilities that make up the plant layout.

4.2. Encoding structure

The encoding structure that is used in our approach to represent each individual of the population is similar to that proposed by García-Hernández et al. (2013). This structure contains a chromosome that is composed of 2 segments. The first segment encloses the facility sequence organized bay by bay, from left to right and from top to bottom. It contains  $n$  labels that represent each of the facilities in the plan layout. The second segment includes the bay divisions in the plant layout. It contains  $n1$  binary elements. A value of 0 indicates that the corresponding facility is placed in the same bay as the previous one, and a value of 1 indicates that this element is the last facility in the present bay.

An example of an individuals encoding structure is shown in Fig. 1., and its representation is shown in Fig. 2.

4.3. The suggested hybrid evolutionary algorithm

This section explains the suggested hybrid evolutionary algorithm in detail. The algorithm is illustrated in Fig. 3.

In the first stage of the algorithm, an initial population of  $n$  facility layouts is randomly generated. Next, the population is classified into clusters using the fuzzy c-means algorithm that was described in Section 3.2. A subset of individuals that is composed of the representative elements of the clusters is displayed to the DM. Nine solutions are displayed to the DM in each generation as was proposed by García-Hernández et al. (2013). The DM evaluates the solutions according to his/her preferences for each design. During every iteration, the DM gives each of the displayed facility plant

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

Fig. 1. An example of the encoding structure of a chromosome.

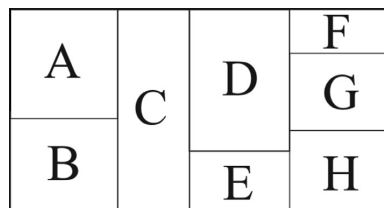


Fig. 2. The representation of the facility associated with the previous chromosome.

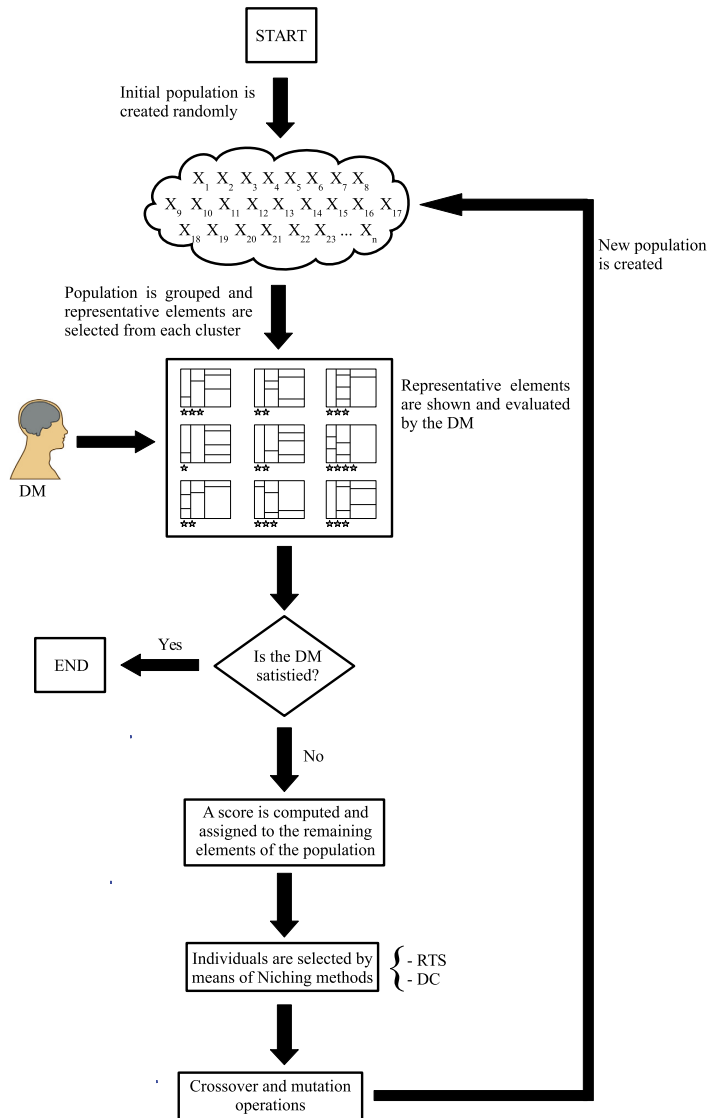


Fig. 3. Schematic of the suggested hybrid evolutionary algorithm.

layouts a score, which can range from 1 (very unsatisfactory solution) to 5 (very satisfactory solution). The fitness of the remaining facility layout designs (eval<sub>j</sub>) is computed using Eq. (5).

$$fit_j = \sum_{i=1}^c m_{ij} e_i \tag{5}$$

where *c* is the number of clusters, *m<sub>ij</sub>* is the membership grade of individual *j* in cluster *i*, and *e<sub>i</sub>* is the score given by the DM to the representative individual of cluster *i*.

In this stage, a selection method is applied to the population. In this study, two niching techniques are used: DC (explained in Section 3.3.1) and RTS (described in Section 3.3.2). Finally, crossover

and mutation operators (which will be described in Sections 4.3.1 and 4.3.2, respectively) are applied, and the new population is created, which will most likely be more adapted to the DM's preferences than the previous population. The process is repeated until the DM finds a facility layout design that satisfies his/her preferences.

#### 4.3.1. Crossover operators

In our proposed method, two different crossover operators are applied to the two segments of the chromosome (see Section 4.2). The facility sequence segment is recombined using the partially mapped crossover (PMX) (Sadrzadeh, 2012), which guarantees that

no facilities are repeated in a facility layout design. The steps of this method are as follows:

1. Randomly select a substring.
2. Exchange the selected substring between the parents.
3. Regulate the offspring with the mapping relationship.

The second segment (bays division segment) is crossed by means of the n-point crossover (Holland, 1992; Starkweather et al., 1991). In this case, a single crossover point (which is selected randomly) is used to exchange the substrings between the parents to create the offspring. Examples of both crossover operators are shown in Fig. 4.

4.3.2. Mutation operator

In our approach, the mutation operator is applied with a certain probability (normally low) for each chromosome segment. In the facility sequence segment, two facilities are randomly chosen and switched. In the bays division segment, an index is randomly picked, and its value is reversed.

5. Experiments

This section describes the experimental methodology, which is performed based on the preferences that are considered to be relevant by the DM. Two UA-FLP cases were selected to test our hybrid approach, and the results are validated by comparing them with those obtained by García-Hernández et al. (2013).

5.1. Problem cases

The following subsections describe the selected UA-FLP cases and the groups of preferences that are considered to be relevant for facility layout designs by the DM.

5.1.1. UA-FLP case 1

This case was defined by Aiello, Enea, and Galante (2006). In this problem, the plant layout is composed of 20 facilities that must be allocated within a square area of 70 m × 70 m. Table 1 shows the areas associated with each of these facilities.

The groups of preferences that are important to the DM for UA-FLP case 1 are listed below:

1. A plant layout where facility 'G' touches any side of the plant, facility 'J' is located in a corner, facility 'T' is adjacent to facility 'G' and facility 'T' is an interior facility.
2. A plant layout where facility 'G' is an exterior facility, facility 'T' is not adjacent to facility 'G', facility 'J' is located in a corner, and facility 'G' is adjacent to facility 'P' (which is an interior facility).
3. A plant layout is divided into five to seven bays, where: facility 'C' is an exterior facility, and facility 'G' is an interior facility that is not adjacent to facility 'T'.

5.1.2. UA-FLP case 2

This UA-FLP case is an ovine slaughterhouse and was proposed by Salas-Morera, Cubero-Atienza, and Ayuso-Munoz (1996). Table 2 shows the facilities and the associated areas that must be placed within a rectangular area of 30 m × 55m.

The groups of preferences that were defined by the DM for UA-FLP case 2 are:

1. A plant layout that is divided into three bays, where facility 'B' is an exterior facility, facility 'A' is located in a corner, and facility 'A' is adjacent to facility 'F'.
2. A plant layout where a block of empty space and facility 'B' are placed in corners of the plant, facility 'A' is adjacent to a block of empty space, and facility 'C' is an exterior facility.
3. A plant layout where facility 'F' is an interior facility, facility 'C' touches any side of the plant layout, a block of empty space is located in any corner of the plant layout, and the longest side of facility 'A' is parallel to the left side of the plant.

5.2. Methodology

The main goal of the proposed methodology is to define a plant facility layout that satisfies the DMs preferences, which is a facility layout with a score of 5 (the maximum possible value). Additionally, to avoid DM fatigue, the number of generations that is required to reach the goal should be as low as possible.

The proposed hybrid evolutionary algorithm contains parameters that have been tuned empirically. Because the DM

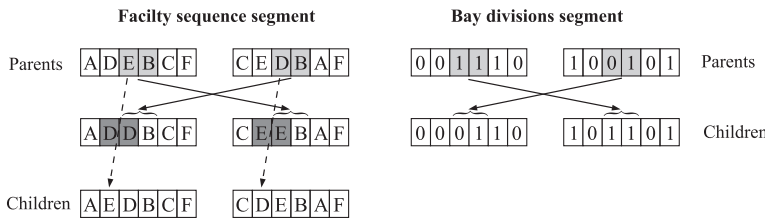


Fig. 4. Examples of the crossover operators that are applied to each chromosome segment.

Table 1 Facilities and associated areas (m<sup>2</sup>) for UA-FLP case 1.

A	B	C	D	E	F	G	H	I	J	K	L	M
100	120	230	150	130	240	300	340	210	180	180	110	180
N	O	P	Q	R	S	T						
110	110	230	220	160	300	200						

Table 2 Facilities and associated areas (m<sup>2</sup>) for UA-FLP case 2.

A	B	C	D	E	F	G	H	I	J	K	L
570	206	150	55	114	102	36	26	46	109	80	40

**Table 3**  
Tuning of the HEA parameters.

Parameter	Value
Population size	100
Crossover probability	0.5
Mutation probability	0.01
Clustering fuzziness	1.2
w (RTS)	9

evaluates nine designs during each algorithm iteration, it is not viable to perform a massive test. Thus, tests were performed while the algorithm was being developed to determine parameter values that fit well. The parameter values that provide good results are shown in Table 3. To avoid random effects, the test was performed three times with the proposed algorithm for the two cases with each parameter.

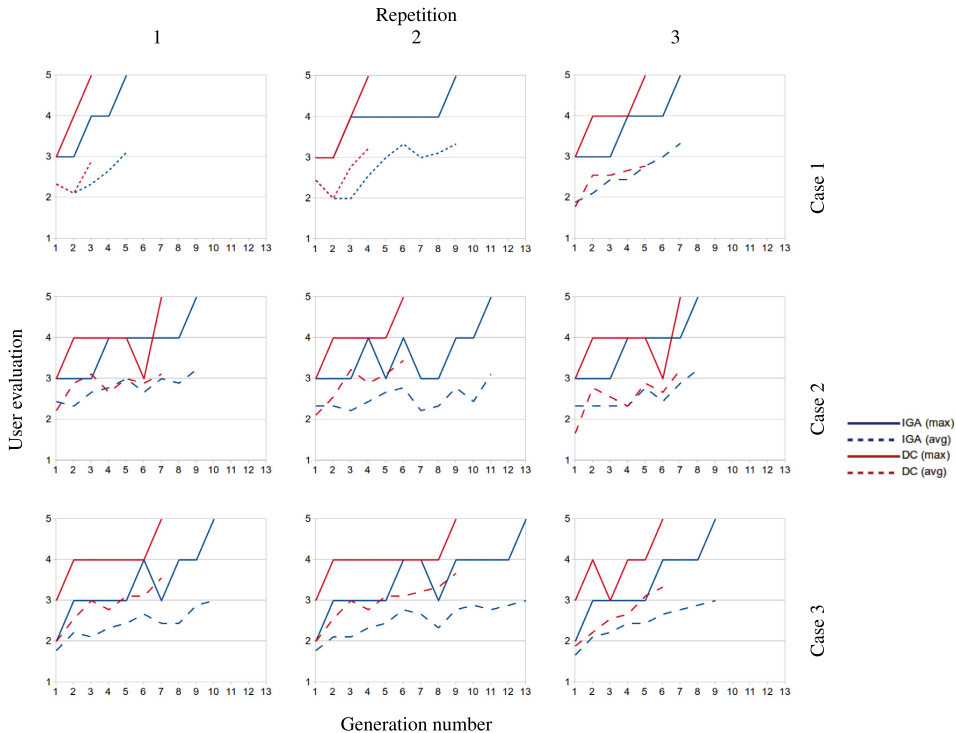
To test how our approach adapts to the DM's preferences, the proposed hybrid evolutionary algorithm was executed while con-

**Table 4**  
Comparison of the numbers of generations that are needed to satisfy the DM using the proposed method versus IGA for case 1.

Problem case 1															
G.I.	IGA					Niching (DC)					Niching (RTS)				
	Generations			Average	Std. Dev	Generations			Average	Std. Dev	Generations			Average	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

**Table 5**  
Comparison of the numbers of generations that are needed to satisfy the DM using the proposed method versus IGA for case 2.

Problem case 2															
G.I.	IGA					Niching (DC)					Niching (RTS)				
	Generations			Average	Std. Dev	Generations			Average	Std. Dev	Generations			Average	Std. Dev
1	17	14	15	15.33	1.52	11	10	7	9.33	2.08	5	8	9	7.33	2.08
2	22	23	19	21.33	2.08	6	11	12	9.66	3.21	11	8	13	10.66	2.51
3	9	6	10	8.33	2.08	7	12	6	8.33	3.21	5	8	9	7.33	2.08



**Fig. 5.** Evolution of the DM evaluations for UA-FLP case 1 – DC niching vs IGA.



sidering a set of preferences that the DM would like to satisfy in the facility layout designs. These preferences were defined for each problem in Sections 5.1.1 and 5.1.2. Additionally, to ensure that the results were not obtained by chance, our proposed approach was run three times for each group of preferences that was defined for each problem case. The results of these experiments and comparisons with those found using the approach proposed by García-Hernández et al. (2013) are described in the next section.

5.3. Results

The following information was considered to analyze the evolution of the proposed hybrid evolutionary algorithm through several generations: the number of generations that are needed to satisfy the DM and the average and standard deviation of the number of generations that are necessary to find a satisfactory solution for each group of preferences.

Tables 4 and 5 show the results that were obtained using our approach and those obtained using the methodology of García-Hernández et al. (2013) (sometimes called IGA), which is sometimes called IGA. The tables show the numbers of iterations of the algorithms that are needed to reach a satisfactory solution, which is defined as the solution that has the maximum score. The best values that were reached in each set of preferences are highlighted in bold. Additionally, Figs. 5–8 show the highest scores and the averages of the scores that were assigned by the DM to the representative individuals in each generation.

As shown in Tables 4 and 5, the number of generations that is needed to reach a satisfactory solution by the DM varies between repetitions with the same group of preferences. The minimum

number of generations that is required to reach a solution that satisfies all of the DMs preferences varies between 3 and 6 for case 1 and between 5 and 6 for case 2.

For case 1 (Table 4), HEA reached a satisfactory solution in fewer generations than IGA in all three repetitions. The DC niching method obtains the lowest number of generations in a repetition for two groups of preferences, and the RTS niching method obtains the lowest number of generations for the remaining group. However, the DC niching method has the lowest average number of generations for all groups of preferences (it is tied with RTS for group 2).

For case 2 (Table 5), HEA reaches a satisfactory solution with a lower number of generations than IGA in all three repetitions. Moreover, for groups 1 and 2, IGA may cause the DM to become fatigued and overloaded because it requires too many generations to reach a satisfactory solution. In this case, the RTS niching method uses the lowest average number of generations for two groups of preferences, and the DC niching method uses the lowest average number of generations for the other group.

Tables 6 and 7 show the diversity of facility plant layout solutions that were generated by the proposed HEA and compares them with the IGA results. The tables show the numbers of representative individuals that are shown to the DM that are identical in that generation for all of the executions as well as the average and standard deviation for each group of preferences (the best values are displayed in bold).

As shown in Table 6, for case 1, HEA generated a greater diversity of representative individuals than IGA in all of the repetitions. IGA generated a maximum of 5 repeated elements per generation during all of the executions, and the HEA generated a maximum

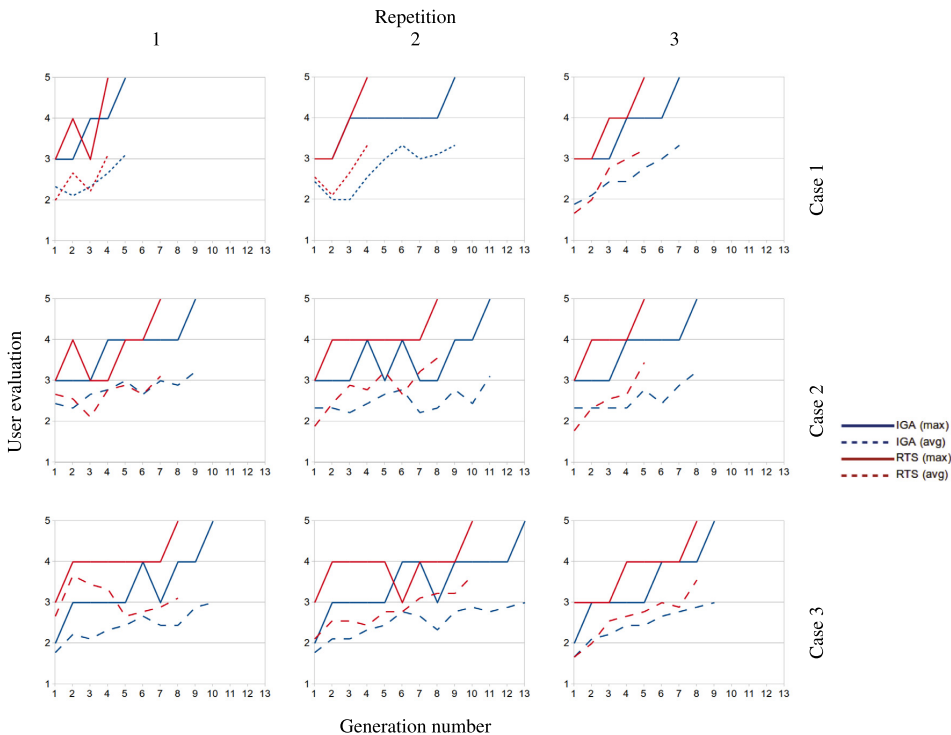


Fig. 6. Evolution of the DM evaluations for UA-FLP case 1 – RTS niching vs IGA.

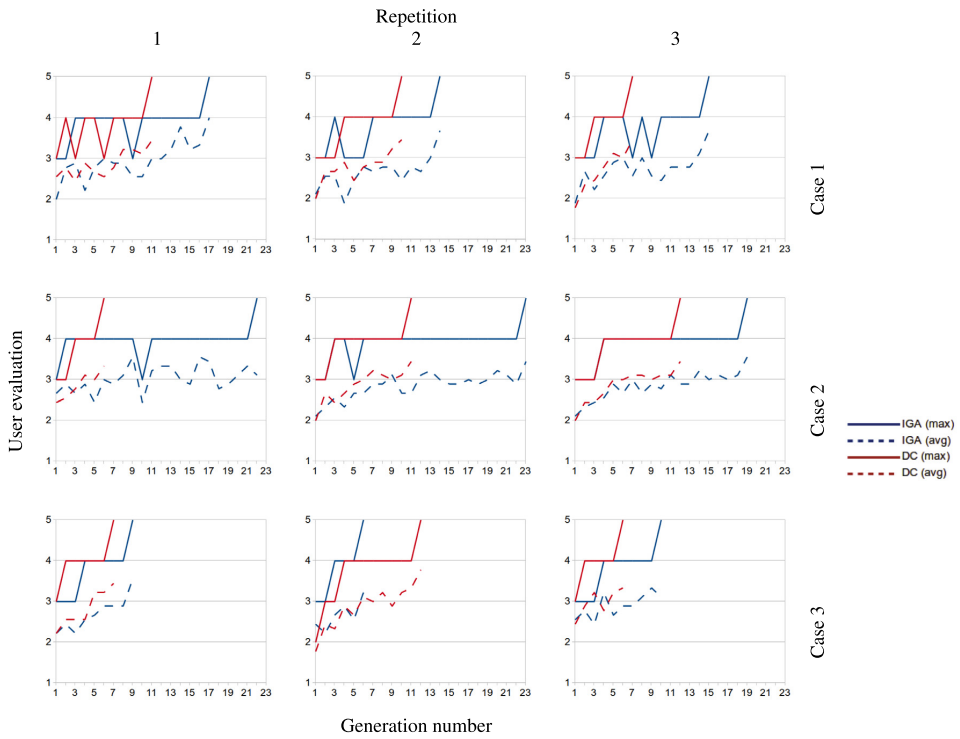


Fig. 7. Evolution of the DM evaluations for UA-FLP case 2 - DC niching vs IGA.

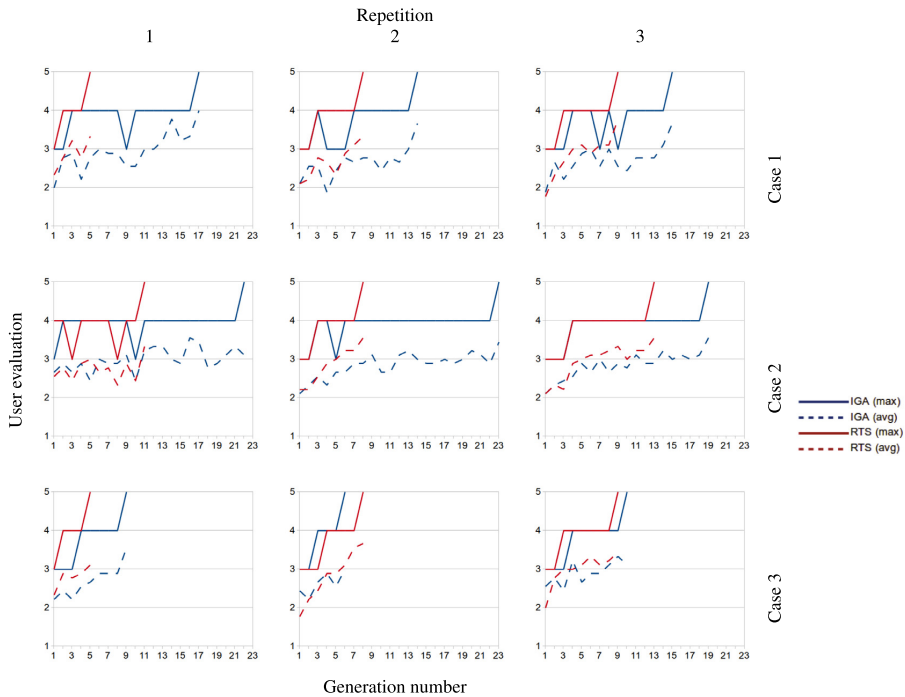


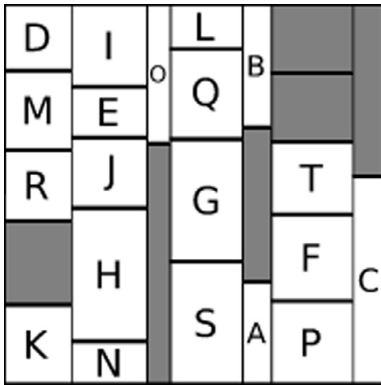
Fig. 8. Evolution of the DM evaluations for UA-FLP case 2 - RTS niching vs IGA.

**Table 6**  
Comparison of the diversity of the displayed layout solutions provided by the proposed method versus IGA for case 1.

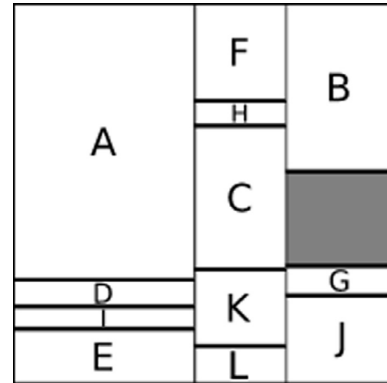
Problem case 1															
G.I.	IGA			Average	Std. Dev	Niching (DC)			Average	Std. Dev	Niching (RTS)				
	R. indiv.	repeated				Repeated					Repeated				
1	3	4	4	3.66	0.57	1	1	0	<b>0.66</b>	0.57	2	1	0	1	1
2	4	5	4	4.33	0.57	0	1	0	<b>0.33</b>	0.57	2	0	0	0.66	1.15
3	5	3	4	4	1	0	0	0	<b>0</b>	0	0	0	1	0.33	0.57

**Table 7**  
Comparison of the diversity of the displayed layout solutions provided by the proposed method versus IGA for case 2.

Problem case 2															
G.I.	IGA			Average	Std. Dev	Niching (DC)			Average	Std. Dev	Niching (RTS)				
	R. indiv.	repeated				Repeated					Repeated				
1	9	8	6	7.66	1.52	3	1	1	1.66	1.15	0	1	0	<b>0.33</b>	0.57
2	11	9	12	10.66	1.52	0	3	2	<b>1.66</b>	1.52	2	3	2	2.33	0.57
3	4	2	4	3.33	1.15	0	0	1	<b>0.33</b>	0.57	2	3	1	2	1



**Fig. 9.** A solution considered to be satisfactory by the DM for UA-FLP case 1.



**Fig. 10.** A solution considered to be satisfactory by the DM for UA-FLP case 2.

of 2 repeated elements. Of the two niching methods that were applied, DC provides more diversity than RTS for all groups of preferences.

For the case 2 (Table 7), HEA produced better results than IGA in all of the repetitions. In this case, IGA generated a maximum of 12 repeated elements per generation during all of the executions, and the HEA generated 3 repeated elements. In this case, the DC niching method provides more diversity for two groups of preferences, and the RTS niching method provides more diversity for the remaining group.

Examples of satisfactory solutions that were obtained with the HEA are shown in Figs. 9 and 10. Fig. 9 shows a solution that was considered to be satisfactory by the DM for UA-FLP case 1. This solution, which was generated by the RTS niching method, satisfies the third group of preferences, so the plant is divided into seven bays, facility 'C' is an exterior facility, facility 'G' is an interior facility, and facility 'G' is not adjacent to facility 'T'.

Fig. 10 shows a satisfactory solution for the UA-FLP case 2. This solution, which was generated by the DC niching method, satisfies the first group of preferences, so the plant is divided into three bays, facility 'B' is an exterior facility, and facility 'A' is located in a corner and is adjacent to facility 'F'.

In summary, the results show that the proposed approach (HEA) provides solutions that satisfy the DM faster than the method that

was proposed in García-Hernández et al. (garcia-herandez2013iga) (IGA). It should be noted that HEA presents substantially more diverse representative individuals to the DM than IGA. This significantly reduces the number of repeated representative elements that must be evaluated by the DM in the same generation.

## 6. Conclusions

This study presents a novel hybrid evolutionary algorithm for the unequal area facility layout problem. Using the suggested approach, it is possible to include a DMs preferences into the design process by means of an interactive genetic algorithm. The DMs knowledge guides the search process towards their preferences without needing to specify them at the beginning of the process.

Because the DMs intervention is essential for the proper evolution of the algorithm, the selection of solutions that will be shown to him/her during each iteration is crucial to the algorithms success. The set of solutions must be representative of the population, and its components must be sufficiently different from one to another to prevent the DM from having to evaluate similar or identical solutions in the same step of the algorithm. A hybrid approach was developed that includes a combination of an interactive

genetic algorithm with two different niching methods to preserve the diversity of solutions.

The empirical analysis shows that the suggested approach is able to provide facility designs that satisfy the DMs preferences and preserve the diversity of solutions during the entire design process. The experimental results for the two analyzed cases and all of the groups of preferences that were tested show that the method provides satisfactory solutions that do not repeat and is faster than the comparable approach. Additionally, the new approach causes less fatigue and overload of the DM. Thus, the new hybrid approach improves the results that are obtained by the previous method. Moreover, the standard deviation of the solutions is smaller in all cases than those obtained using the previous approach, which indicates that the new method is more consistent.

Incorporating the expert knowledge of the DM into the UA-FLP increases the quality and applicability of the designs over other solutions that are obtained by only mathematical optimization because it allows the qualitative preferences of the DM to be considered, which is impossible in other approaches. Nevertheless, the method suffers from requiring the continued participation of the DM in manually assessing solutions during the process, so finding a way to shorten the process would be beneficial. In addition, shortening the computational process would allow the DM to incorporate more complex qualitative requirements, which would increase the possible applications of the method.

In the future, it could be interesting to utilize more than one DM in the approach. This could enrich the facility layout designs by incorporating the knowledge and experience from several human experts in the facility layout design. Furthermore, to assist the DM in the evaluation process, an approach that includes system learning in the algorithm could also be investigated.

## References

- Aiello, G., Enea, M., & Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and electromechanical 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 electromechanical method for unequal area facility layout problems. *Expert Systems with Applications*, 40(12), 4812–4819.
- Arellano-Herrera, M., Ferri, F. J., & Moreno-Picot, S. (2013). A hybrid multi-objective optimization algorithm for content based image retrieval. *Applied Soft Computing*, 13(11), 4358–4369.
- 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.
- Beasley, D., Bull, D. R., & Martin, R. R. (1993). A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2), 101–125.
- 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.
- Cavicchio, D. (1970). *Adaptive Search Using Simulated Evolution* (Ph.D. thesis), Ann Arbor, MI, USA: University of Michigan.
- Chang, T.-J., Yang, S.-C., & Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, 36(7), 10529–10537.
- 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.
- 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.
- Dong, N., Wu, C.-H., Ip, W.-H., Chen, Z.-Q., Chan, C.-Y., & Yung, K.-L. (2011). An improved species based genetic algorithm and its application in multiple template matching for embroidered pattern inspection. *Expert Systems with Applications*, 38(12), 15172–15182.
- Drira, A., Pierrel, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2), 255–267.
- Fayek, M. B., Darwish, N. M., & Ali, M. M. (2010). Context based clearing procedure: A niching method for genetic algorithms. *Journal of Advanced Research*, 1(4), 301–307.
- Fu, L. S., Wang, X.-L., Xiao, J.-Z., & Yin, Z.-J. (2014). Self-adaptive obtaining water-supply reservoir operation rules: Co-evolution artificial immune system. *Expert Systems with Applications*, 41(Part 14), 1262–1270.
- García-Hernández, L., Arauzo-Azofra, A., Salas-Morera, L., Pierrel, 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., Pierrel, 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., Pierrel, 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.
- Goldberg, D. E. & Richardson, J. J. (1987). Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the 2nd International Conference on Genetic Algorithms*, (pp. 41–49).
- Gren, S., & Pierrel, H. (2013). Taking advantage of a diverse set of efficient production schedules: A two-step approach for scheduling with side concerns. *Computers & Operations Research*, 40(8), 1979–1990.
- Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. In L. J. Eshelman (Ed.), *ICGA* (pp. 24–31). Morgan Kaufmann.
- Helmy, T., Fatai, A., & Faisal, K. (2010). Hybrid computational models for the characterization of oil and gas reservoirs. *Expert Systems with Applications*, 37(7), 5353–5363.
- Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1), 1–13.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press.
- 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.
- Lin, C.-Y., & Wu, W.-H. (2002). Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software*, 33(11), 779–791.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceeding of fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). Univ. of Calif. Press.
- 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.
- Martinez-Alvarez, A., Calvo-Zaragoza, J., Cuenca-Asensi, S., Ortiz, A., & Jimeno-Morenila, A. (2014). Multi-objective adaptive evolutionary strategy for tuning compilations. *Neurocomputing*, 123(0), 381–389.
- 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.
- Pérowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In: *International Conference on Evolutionary Computation*, (pp. 798–803).
- Pimentel, B. A., & de Souza, R. M. (2013). A multivariate fuzzy c-means method. *Applied Soft Computing*, 13(4), 1592–1607.
- Quiroz, J. C., Louis, S. J., Banerjee, A., & Dascalu, S. M. (2009). Towards creative design using collaborative interactive genetic algorithms. In *CEC'09: Proceedings of the eleventh conference on congress on evolutionary computation* (pp. 1849–1856). Piscataway, NJ, USA: IEEE Press.
- 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.
- Şahin, R., Ertoğral, K., & Türkbey, O. (2010). A simulated annealing heuristic for the dynamic layout problem with budget constraint. *Computers and Industrial Engineering*, 59(2), 308–313.
- Salas-Morera, L., Cubero-Atienza, A. J., & Ayuso-Munoz, R. (1996). Computer-aided plant layout. *Información Tecnológica*, 7(4), 39–46.
- Sanz-García, A., Pernía-Espinoza, A. V., Fernandez-Martinez, R., & de Pisón Ascáibar, F. J. M. (2012). Combining genetic algorithms and the finite element method to improve steel industrial processes. *Journal of Applied Logic*, 10(4), 298–308.
- 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.
- Ulutas, B. H., & Kulturel-Konak, S. (2012a). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), 5384–5395.
- Ulutas, B. H., & Kulturel-Konak, S. (2012b). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), 5384–5395.
- Yannibelli, V., & Amandi, A. (2012). A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications*, 39(10), 8584–8592.
- Yu, E., & Suganthan, P. (2010). Ensemble of niching algorithms. *Information Sciences*, 180(15), 2815–2833.
- Zhang, J., Huang, D.-S., Lok, T.-M., & Lyu, M. R. (2006). A novel adaptive sequential niche technique for multimodal function optimization. *Neurocomputing*, 69(1618), 2396–2401.
- Zhao, F., Li, G., Yang, C., Abraham, A., & Liu, H. (2014). A human computer cooperative particle swarm optimization based immune algorithm for layout design. *Neurocomputing*, 132(0), 68–78.

# B

---

AN ISLAND MODEL GENETIC ALGORITHM FOR  
UNEQUAL AREA FACILITY LAYOUT PROBLEMS

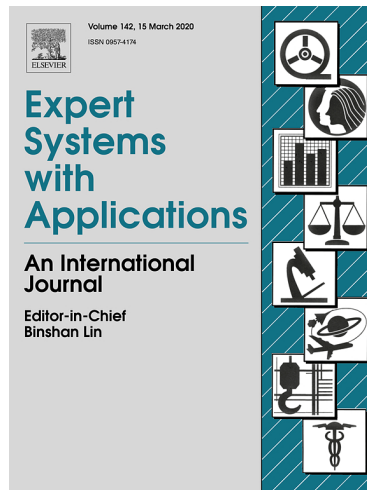
---



TÍTULO: An island model genetic algorithm for unequal area facility layout problems.

AUTORES: Juan María Palomo Romero, Lorenzo Salas Morera y Laura García Hernández.

PUBLICACIÓN: Expert Systems with Applications, 68, páginas 151-162, 2017. DOI: 10.1016/j.eswa.2016.10.004



RANKING JCR 2017

FACTOR DE IMPACTO: 3.768

Categoría	Cuartil	Ranking
Operations Research & Management Science	Q1 <sup>a</sup>	8/83
Computer Science, Artificial Intelligence	Q1	20/132
Engineering, Electrical & Electronic	Q1	42/260

<sup>a</sup> La revista se encuentra situada en el **decil 1** de la categoría.



MÉTRICAS SCOPUS

FECHA DE CONSULTA: noviembre de 2019

CITAS: 25

FIELD-WEIGHTED CITATION IMPACT: 8.44 \*

<b>Categoría</b>	<b>CiteScore</b>
<b>Categoría</b>	<b>Percentile <sup>a</sup></b>
Computer Science	92 %
Artificial Intelligence	90 %
Computer Science Applications	90 %
Engineering	88 %

<sup>a</sup> Indica la posición relativa de un artículo en su campo de estudio. Cada campo se divide en 100 percentiles de igual tamaño, siendo 0% el menor, y 99% el mayor.

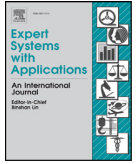
---

\* Índice que calcula cuánto se cita este artículo en comparación con artículos similares. Un valor mayor que 1.00 significa que el artículo se cita más de lo esperado.



Contents lists available at ScienceDirect

## Expert Systems With Applications

journal homepage: [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

# An island model genetic algorithm for unequal area facility layout problems



Juan M. Palomo-Romero\*, Lorenzo Salas-Morera, Laura García-Hernández

Area of Project Engineering, University of Córdoba, Rabanales Campus, Leonardo Da Vinci Building, Córdoba 14071, Spain

## ARTICLE INFO

### Article history:

Received 22 April 2016  
 Revised 30 September 2016  
 Accepted 1 October 2016  
 Available online 14 October 2016

### Keywords:

Unequal area facility layout problem  
 Parallel genetic algorithms  
 Island model  
 Flexible bay structure

## ABSTRACT

The Unequal Area Facility Layout Problem (UA-FLP) has been addressed by various methods, including mathematical modelling, heuristic and metaheuristic approaches. Nevertheless, each type of approach presents problems such as premature convergence, lack of diversity, or high computational cost. In this paper, for the first time, an Island Model Genetic Algorithm (IMGA) is proposed to solve these subjects in the UA-FLP. The parallel evolution of several populations is used to maintain the population diversity and to obtain a wider sampling of the search space to obtain better quality solutions in fewer generations. Our novel approach was tested with a well-known set of problems taken from the literature and the results were compared with those of previous reports. In most cases, the results obtained by our novel approach improved on the previous results. Additionally, the proposed approach is able to reach good solutions with a wide range of problem sizes and in a reasonable computational time.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

The problem of facilities placement within a plant, often called the facility layout problem (FLP), is known to have a significant impact on work processes, manufacturing costs, productivity and lead times (Drira, Pierrelval, & Hajri-Gabouj, 2007). A facility layout consists of an arrangement of everything needed for production of goods or delivery of services. A facility is an entity that facilitates the performance of a job; for example, a facility may be a machine tool, a work center, a manufacturing cell, a machine shop, a department, a warehouse, and so on (Heragu, 2008). In most cases, the main objective in facility layout problems is to minimize material handling costs.

In any manufacturing industry, an efficiently designed plant layout and facility locations are both important (Singh & Sharma, 2006). A superior facilities layout contributes to the overall operational efficiency and can reduce total operating costs by 20% to 50% (Tompkins, White, Bozer, & Tanchoco, 2010).

The particular goal depends on the features of the facilities. For example, facilities have different shapes (e.g., circular, rectangular) and many have the same or different dimensions. Since the 1990s, research in this field has focused primarily on the unequal area FLP (UA-FLP) (Scholz, Petrick, & Domschke, 2009), which has been

one of the most important facility layout problems (Meller & Gau, 1996).

Different methods exist to represent facility layout designs. Among the different ways to represent a layout are: (a) continuous representations and (b) discrete representations. The most commonly used representations are two types of continuous representation structures: (a) the slicing tree representation, a concept presented by Tam (1992), that involves dividing an initial rectangle (in the horizontal or vertical direction) from one side to the other recursively, repeating the division with the newly generated rectangles; and (b) the flexible bay structure (FBS), a continuous layout representation defined by Tong (1991), that is common in many manufacturing and retail facilities (Kulturel-Konak, 2012). This method of representation consists of dividing the rectangular area in one direction into parallel bays with varying widths. Lately, this method has been one of the most common ways to represent unequal area facility layout designs. The method is described in more detail in Section 3.2.

Several classical techniques have been applied to the FLP, including the branch and bound method (Solimanpur & Jafari, 2008), mixed integer programming (Bozer & Wang, 2012) and graph theoretic methods (Kim & Kim, 1995). The results obtained by the classical methods reveal that they are not particularly computationally efficient (Achterberg, 2007), particularly on large-sized problems. Later, some proposals for hybrid classic methods were presented; for example, Kulturel-Konak (2012), proposed a probabilistic tabu search approach that used linear programming to determine department shapes and their locations within the bays. The

\* Corresponding author. Tel.: +34957212246. Fax: +34957218550.

E-mail addresses: [juanmaria@uco.es](mailto:juanmaria@uco.es) (J.M. Palomo-Romero), [lsalas@uco.es](mailto:lsalas@uco.es) (L. Salas-Morera), [ir1gahel@uco.es](mailto:ir1gahel@uco.es) (L. García-Hernández).

proposed linear programming approach was used to solve 13 FLP instances of different sizes from the literature and its results equaled the best results to that date for many problems and improved on the results for some problems. However, the linear programming approach has the same drawback as other classical methods for large-sized problems: lengthy execution time.

To solve the UA–FLP problem, recent research has focused on metaheuristic methods such as local search. A disadvantage of applying local search methods is the possibility of the methods becoming trapped in a local optimum. To escape from such optima, an interchange and insertion neighborhood exploration combined with local search was proposed by Palubeckis (2015a). However, the major disadvantage of iterative search methods is that these algorithms may require large amounts of computation time to produce sufficiently good solutions. Other studies have presented proposals based on simulated annealing (SA) (Matai, 2015; Palubeckis, 2015b; Sahin, 2011), although in problems with few local minima the SA algorithm may not be efficient, and it can be too slow (Ingber, 1993).

The FLP can be viewed as a generalization of the quadratic assignment problem. It is most unlikely that any exact solution to the general layout problem can be obtained in an amount of time bounded by a polynomial with the size of the problem, resulting in prohibitive computation time for large problems (Ertay, Ruan, & Tuzkaya, 2006). Therefore, in recent years, various evolutionary strategies have been used to address classical problems such as the FLP (Abotaleb, Nassar, & Hosny, 2016; Li & Parrott, 2016).

Genetic algorithms (GAs) (Holland, 1992) are computational methods that are based on the principles of natural selection (Darwin, 1859). In accordance with Darwin, individuals in a population compete to obtain natural resources and find mates. The more successful individuals in a population have a greater probability of reproducing; therefore, they will create more offspring than less successful individuals. GAs are global search algorithms that have typically been used to solve optimization and search problems (Goldberg, 1989). In many different fields, including control systems engineering (Wang & Kwok, 1994), robotics (Kker, 2013), pattern recognition (Ijjina & Chalavadi, 2016), speech recognition (Kwong, Ng, & Man, 1995), the water industry (Shin, Joo, & Koo, 2016), job-shop scheduling (Kundakc & Kulak, 2016), logistics parks (Chen, Jiang, Wahab, & Long, 2015), the footwear industry (Ulutas & Islier, 2015), chemical plants (Alves, de Medeiros, & de Queiroz F. Arajo, 2016) and, naturally, facility layout (Ahmadi & Jocar, 2016; Pierreval, Caux, Paris, & Viguer, 2003). Moreover, GAs have recently proven their effectiveness at finding “good enough” solutions to many NP-hard problems, such as UA–FLP (Aiello, Scalia, & Enea, 2013). Various published approaches include Aiello, Scalia, and Enea (2012); Datta, Amaral, and Figueira (2011); Dunker, Radons, and Westk mpfer (2005); Islier (1998); Mak, Wong, and Chan (1998); Tam (1992); and Pourvaziri and Naderi (2014). As these studies show, genetic algorithms have proven to be an appropriate strategy for addressing many different problems.

Several studies have specifically addressed the UA–FLP using GAs. First, Tate and Smith (1995) implemented a simple genetic search for UA–FLP. Their approach used an adaptive penalty function to find good feasible solutions. Furthermore, they provided the results from experiments with several test problems that demonstrated the robustness of their algorithm. Kochhar, Foster, and Heragu (1998) proposed a GA-based algorithm to solve the single-floor FLP while considering departments of both equal and unequal sizes. They evaluated their algorithm using several test problems and concluded that GAs can offer a better alternative when the objective is to find a number of “reasonably good” layouts. A shape-based block layout approach using a hybrid genetic algorithm was presented by Lee and Lee (2002) in which, the objec-

tive function minimizes the total material handling cost and maximizes the space utilization. Wang, Hu, and Ku (2005) presented a study that combines a GA with analysis of variance (ANOVA). The experimental results showed that this approach is more feasible in addressing facilities layout problems in the real world. Approaches based on traditional GAs usually exhibit a typical problem: premature convergence. Moreover, normally, all the individuals in a population should be selected and evaluated in each generation and they can be crossed and mutated, all of which requires additional CPU time when searching for good solutions.

Several GA variants have also been proposed. Aiello et al. (2013) used a Multi Objective Genetic Algorithm (MOGA) to evolve the population. This approach involves ranking according to a set of criteria given by the decision maker using a multi-criteria decision making procedure. Garc a-Hern andez, Arauzo-Azofra, Salas-Morera, Pierreval, and Corchado (2013) proposed approaches to solve the UA–FLP using interactive genetic algorithms with subjective criteria. In these proposals, the user guides the evolution of the population by implementing subjective scores that evaluate a set of representative facilities (Garc a-Hern andez, Arauzo-Azofra, Salas-Morera, Pierreval, & Corchado, 2015a; Garc a-Hern andez, P erez-Ortiz, Arauzo-Azofra, Salas-Morera, & Herv as-Mart nez, 2014). Interactive algorithms may also execute slowly because they require the intervention of a decision maker (generally, a human expert); however, the decision maker can be at risk of fatigue due to the amount of information to be evaluated.

One technique that seeks to overcome the main disadvantage of sequential GA is Ant Colony Optimization (ACO). This is a fairly novel technique for solving computational problems; it mimics the natural behavior of a colony of ants as they find and select paths to a food source (Dorigo & St utzle, 2004). Some hybrid approaches based on ant colony optimization have also been developed. Komarudin and Wong (2010) presented an ant colony optimization algorithm using the slicing tree to easily represent the problems without excessively restricting the solution space. Later, Wong and Komarudin (2010) published an ant colony approach to solving the UA–FLP. The proposed algorithm uses the *best-so-far* and *iteration-best* ants for pheromone updating. The algorithm stops when it reaches one of the following criteria: (a) a number of maximum iterations in a single implementation or (b) a maximum number of iterations without improving the *best-so-far solution*. Along the same direction, an ant colony optimization algorithm was proposed by Kulturel-Konak and Konak (2011b) who compared the results with the previously best known solutions. This algorithm finds the previously known best solutions in a very short amount of CPU time. Finally, a hybrid algorithm based on variable neighborhood search and ant colony optimization was recently proposed by Guan and Lin (2016) to solve the single row facility layout problem. However ACO algorithms exhibit two problems: (a) their probability distribution changes by iteration and (b) their convergence time is uncertain (Umarani & Selvi, 2010), which explain the excessive CPU time required to reach good solutions in medium-sized and large-sized problems.

In recent years, other approaches using hybrid algorithms based on GAs have been published. For example, Vitayasak, Pongcharoen, and Hicks (2016) focused on solving dynamic facility layout problems using GAs and the Backtracking Search Algorithm (BSA). This algorithm is a population-based iterative evolutionary algorithm designed to achieve global optimization. Nevertheless, the backtracking algorithms have these disadvantages: (a) a drawback of backtracking is having to perform redundant work, and (b) the basic backtracking algorithm may detect a conflict too late because it is unable to detect conflict before it occurs. In contrast, a biased random-key genetic algorithm (BRKGA) for the UA–FLP was proposed by Goncalves and Resende (2015). This work combines a BRKGA to determine the order of placement and the dimensions of

each facility, a novel placement strategy to position each facility, and a linear programming model to fine-tune the solutions. This combination improves the results in most of the cases tested. A heuristic algorithm for FLP and a software package with collision detection and collision response methods presented by Sikaroudi and Shahanaghi (2016). This approach was compared with the BRKGA method presented by Gonçalves and Resende (2015), and concluded that BRKGA required fewer generations and less execution time. In another approach, Paes, Pessoa, and Vidal (2016) introduced a strategy of decomposition phases oriented to a facility layout with many departments: a GA combined with a decomposition strategy via partial solution deconstructions and reconstructions. This algorithm has been tested on classical benchmark instances from the literature, and it returns high-quality solutions. However, these proposals are based on unique representation structures that have not been explored in the literature. Moreover, these proposals did not consider maintaining the diversity of the population to explore more search space in less execution time.

Another line of research to solve the problems of becoming trapped in local optima, premature convergence, and high CPU time lies in the parallel genetic algorithms (PGA). Adopting a parallel approach can be considered as a way to improve search diversification and delay premature convergence (Kurdi, 2016), which are key features for exploring a larger search space and obtaining better solutions.

To date, only a few parallel genetic algorithms (PGA) approaches based on the island model have been applied to solve FLP. Tam (1998) presented a PGA approach for the FLP in which four coarse-grained parallel genetic algorithms (PGAs) were developed and compared. Al-Hakim (2000) modified Tam's work by introducing a new preserving operation (called transplanting) that provides a guarantee of the validity of the offspring. This model, based on the evolution of a population that is divided into independent multiple subpopulations, helps preserve genetic diversity and explore multiple search trajectories (Whitley, Rana, & Heckendorn, 1998). Furthermore, this model appears to offer better search performance than the sequential single-population models (Cantú-Paz, 1998; Mhlenbein, 1991). As shown by the papers described above, we can conclude that PGAs play an important role in maintaining diversity throughout different generations, helping to generate better and more diverse solutions.

To the best of our knowledge, no previous work has considered solving the UA-FLP using a PGA based on the island model. However, because of the main characteristics of PGAs, such as maintaining the population diversity, performing parallel searches from multiple points in the space, and higher efficiency and efficacy compared with sequential GAs (Alba & Troya, 1999), it appears that this line of research could be useful in obtaining promising results for the UA-FLP.

In this paper, a novel island model genetic algorithm (IMGA) is presented. To the best of our knowledge, this is the first time that a PGA based on the island model has been proposed to solve the UA-FLP using the FBS representation. This new approach will be tested with several well-known problems in the literature. Comparisons with previous approaches to solve these problems will be shown later.

The remainder of this article is organized as follows. A brief description of the PGAs is presented in Section 2. Section 3 explains the newly proposed algorithm. In Section 4, we use well-known problems to test our algorithm and we analyze the best solutions obtained in the literature. A comparison of the results found using our proposal and the results found by the best approaches from the literature based on FBS representation are presented in Section 5. Finally, Section 6 presents concluding remarks and future work.

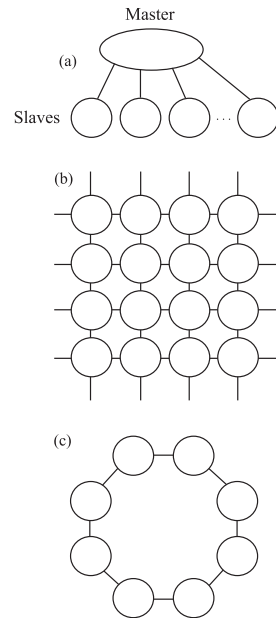


Fig. 1. Categories of parallel genetic algorithms.

## 2. Parallel genetic algorithms

PGAs arise from the volume of computation required by problems of extreme complexity. The execution time required to solve such problems using traditional sequential GAs is often prohibitive (Sitkoff, 1995). A PGA algorithm is usually composed of various GAs. Each GA processes either a part of the full population or processes independent populations. Communication may or may not occur among these GAs (Solar, Parada, & Urrutia, 2002).

Cantú-Paz (1998) proposed classifying PGAs into three categories (presented in Fig. 1):

- Global single-population master-slave GAs. As in a GA, there is a single population, but in this case, this population is evaluated by considering several independent processes. However, selection and crossover are applied to the entire population.
- Single-population fine-grained GAs. This model of PGA is based on a spatially structured population. The ideal case is to have only one individual for every node. Selection and crossover are restricted to a small neighborhood. For each individual in each generation, crossover occurs with a neighbor, selected from all possible neighboring candidates.
- Multiple-population coarse-grained GAs. This model consists of several distributed subpopulations that occasionally exchange individuals (*a migration process*). Consequently, this type of PGA is somewhat more difficult to understand than the others because adding the migration approach changes the behavior of the simple GAs.

In our work, we use a multiple-population coarse-grained model. In this model, because of the degree of independence imposed by the islands and the exchange of individuals through migration, it is possible to explore different regions of the search space, thereby improving the search quality, offering better-quality solutions and requiring fewer evaluations to find them. This approach can also be considered as a means of maintaining population diversity (Starkweather, Whitley, & Mathias, 1991b).

**3. Problem formulation**

Our work is focused on solving the UA-FLP, problem that is one of the most studied facility layout problems in the related literature (Aiello et al., 2013; Bozer & Wang, 2012; Cantú-Paz, 1998; García-Hernández, Palomo-Romero, Salas-Morera, Arauzo-Azofra, & Pierreval, 2015b; García-Hernández, Pierreval, Salas-Morera, & Arauzo-Azofra, 2013; Gonçalves & Resende, 2015; Kim & Kim, 1995; Kochhar et al., 1998; Kulturel-Konak, 2012; Ripon, Glette, Khan, Hovin, & Torresen, 2013; Scholz et al., 2009; Ulutas & Kulturel-Konak, 2012; Wang et al., 2005; Wong & Komarudin, 2010).

The UA-FLP was described by Armour and Buffa (1963) as a rectangular region that has fixed dimensions  $H(\text{height}) \times W(\text{width})$  and a collection of  $n$  required facilities, each of which occupies a specified area ( $A_i$ ). The sum of the facility areas must be less than or equal to the plant area ( $HW$ ) (see Eq. (1)).

$$\sum_i^n A_i \leq H \times W \tag{1}$$

To solve the UA-FLP, the goal is to allocate the facilities in the plant based on some given optimization criteria while taking into consideration that the facilities cannot overlap. The optimization criteria can be to maximize the adjacency requirement (Seppanen & Moore, 1970) and/or minimize the cost of transporting materials between the facilities (Koopmans & Beckmann, 1955), among others.

**3.1. Objective function**

Each individual of the population is composed of a chromosome that represents a genetically encoded solution to the problem, that is to say, a facility layout design. Each individual has a fitness score that represents the quality of the solution. This fitness score is calculated by a predetermined objective function. As mentioned above, various objectives and criteria have been considered in UA-FLP: adjacency requirements, distance requests, material handling costs or the satisfaction of a desired aspect ratio. The optimization can be performed using one criterion or a combination of criteria (Aiello & Enea, 2001).

Most studies consider only the material flow; therefore, they directly minimize the total material handling cost between facilities (sometimes called departments) (Wang et al., 2005).

The fitness score used to evaluate the test problems in our proposal is based on the material handling cost and incorporates an adaptive penalty function (Tate & Smith, 1995) to direct the search process to the feasible solution regions. The objective function that minimizes the total material handling cost is presented in Eq. (2).

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

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

where:

$n$  = the number of departments;

$f_{ij}$  = the cost to move materials from department  $i$  to department  $j$ ;

$d_{ij}$  = the distance between departments  $i$  and  $j$ . The distance can be one of the following: rectilinear (Eq. (3)) or Euclidean (Eq. (4)), where the point defined by  $x$  and  $y$  is the center of the department;

$$d_{i,j} = |x_i - x_j| + |y_i - y_j| \tag{3}$$

$$d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{4}$$

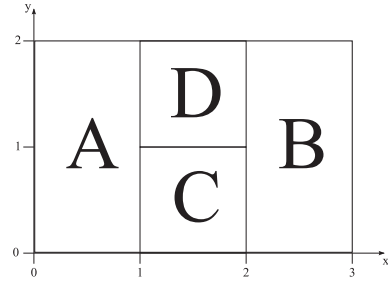


Fig. 2. An example of facility layout plan.

$D_{inf}$  = the number of infeasible departments (determined by a shape constraint);

$V_{feas}$  = the best feasible objective function value yet found;

$V_{all}$  = the best overall objective function value yet found;

$k$  = the parameter that adjusts the value of the penalty function (which was set to '3' (Tate & Smith, 1995)).

**3.1.1. Example graph**

To make the calculation process easier to understand, we use the simple facility layout shown in Fig. 2. The shape constraint is:  $\alpha = 4$ ; and the distance metric is: rectilinear. The cost to move materials between departments is detailed below:

$$f_{AB} = 3;$$

$$f_{AC} = 2;$$

$$f_{AD} = 3;$$

$$f_{BC} = 4;$$

$$f_{BD} = 1;$$

$$f_{CD} = 2;$$

The steps to calculate the fitness score are explained as follows:

1. Determine the center of each department.

$$A = (0.5, 1)$$

$$B = (2.5, 1)$$

$$C = (1.5, 0.5)$$

$$D = (1.5, 1.5)$$

2. Calculate the (rectilinear) distance between departments.

$$d_{AB} = |x_A - x_B| + |y_A - y_B| = |0.5 - 2.5| + |1 - 1| = 2$$

$$d_{AC} = |x_A - x_C| + |y_A - y_C| = 1.5$$

$$d_{AD} = |x_A - x_D| + |y_A - y_D| = 1.5$$

$$d_{BC} = |x_B - x_C| + |y_B - y_C| = 1.5$$

$$d_{BD} = |x_B - x_D| + |y_B - y_D| = 1.5$$

$$d_{CD} = |x_C - x_D| + |y_C - y_D| = 1$$

3. Compute the number of infeasible departments ( $D_{inf}$ ).

In our example, a department is infeasible if  $\alpha > 4$ ;

where:  $\alpha = \frac{\max(\text{Width}, \text{Height})}{\min(\text{Width}, \text{Height})}$

$$\alpha_A = \frac{\max(W_A, H_A)}{\min(W_A, H_A)} = \frac{\max(1, 2)}{\min(1, 2)} = \frac{2}{1} = 2 \text{ (feasible)}$$

$$\alpha_B = \frac{\max(W_B, H_B)}{\min(W_B, H_B)} = 2 \text{ (feasible)}$$

$$\alpha_C = \frac{\max(W_C, H_C)}{\min(W_C, H_C)} = 1 \text{ (feasible)}$$

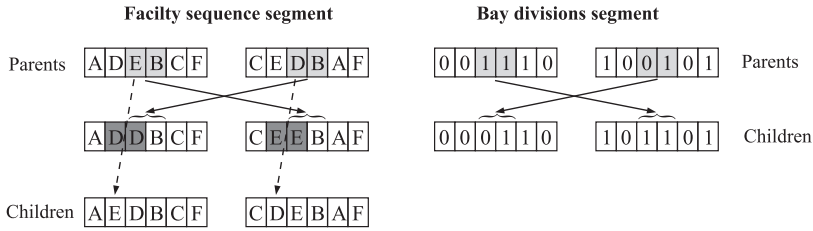


Fig. 3. Examples of crossover operators.

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

Fig. 4. A chromosome example.

$$\alpha_D = \frac{\max(W_D, H_D)}{\min(W_D, H_D)} = 1 \text{ (feasible)}$$

$$D_{inf} = 0$$

4. Calculate the objective function value.

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

$$OF = (3 \cdot 2 + 2 \cdot 1.5 + 3 \cdot 1.5 + 4 \cdot 1.5 + 1 \cdot 1.5 + 2 \cdot 1) + 0$$

$$OF = 23$$

### 3.2. Layout representation

We use the flexible bay structure (FBS), a representation for illustrating plant layouts that was presented by Tong (1991). Among the different mechanisms available from the literature to represent plant designs, the FBS is widely used. A sample of the large number of research studies that use this representation scheme demonstrates its suitability: Alagoz, Norman, and Smith (2008); Konak, Kulturel-Konak, Norman, and Smith (2006); Kulturel-Konak (2012); Kulturel-Konak and Konak (2011a); Tate and Smith (1995); Ulutas and Kulturel-Konak (2012); Wong and Komarudin (2010); and García-Hernández et al. (2013). This representation has the following features: (a) the plant is a rectangular area defined by its height and width; (b) the plant area is divided in one direction into parallel bays with varying widths; and (c) each bay is subdivided to allocate the facilities that integrate the plant design.

### 3.3. Encoding

To represent each individual of the population, we used an encoding structure that contains a chromosome divided into 2 segments. The first segment encodes the facility sequence organized bay by bay, from left to right and from top to bottom. The sequence is composed of  $n$  labels that represent each of the facilities in the plan layout. The second segment encodes the bay divisions in the plant layout ( $n - 1$  binary elements). A value of 0 indicates that the facility is placed in the same bay as the previous facility, while value of 1 indicates that the facility is the last facility in the present bay. Fig. 4 shows an example chromosome. Finally, the facility layout defined by the chromosome in Fig. 4 is presented in Fig. 5.

### 3.4. The island model genetic algorithm approach

This section explains the suggested island model genetic algorithm in detail. First, we describe our approach using pseudocode.

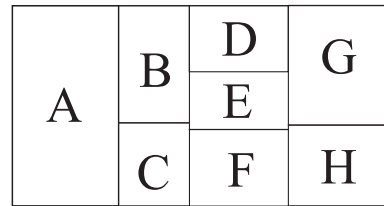


Fig. 5. The facility layout representation related to the example chromosome in Fig. 4.

The following subsections provide the details of the most important steps in the algorithm. First, an initial population of  $p$  facility designs is randomly generated. Then, all the individuals are evaluated by a fitness function (see Section 3.1).

The entire population is randomly distributed into  $i$  subpopulations (islands) of  $p/i$  individuals, where each subpopulation implements an independent sequential GA. Specifically, and each subpopulation implements an independent sequential GA. Specifically, for each of the subpopulations, the algorithm performs the following processes: first it applies a selection method to the subpopulation, as described in Section 3.4.1. Then, crossover and mutation operators (see Sections 3.4.2 and 3.4.3, respectively) are applied. Finally, the subpopulation is evaluated. The new subpopulation that is produced is likely to be better than the previous subpopulation.

The sequential GA is repeated for  $g$  iterations for each island. Then, a migration process is conducted (see Section 3.4.4), which consists of exchanging the  $m$  best individuals between islands. Thereafter, the independent evolutionary process for the islands is executed again.

The algorithm ends when  $c$  cycles of generations have been conducted.

The pseudocode of our algorithm is presented in Algorithm 1.

#### 3.4.1. Selection operator

Many selection operators exist for GAs, and each has different characteristics. An ideal selection operator would be simple to code and efficient for both nonparallel and parallel architectures (Miller & Goldberg, 1996).

We use tournament selection to select the individuals for whom crossover will be conducted. Tournament selection (Goldberg & Deb, 1991) is a variant of rank-based selection methods. Some number of individuals (2 in our approach) are randomly chosen from the population, and the individual with the best fitness from this pair is selected as a parent. This process is repeated as often as the individuals must be chosen.

#### 3.4.2. Crossover operators

Two different crossover methods are applied in our proposal one for each segment. The partially mapped crossover (PMX) (Sadrzadeh, 2012) is used to recombine the facility sequence

---

**Algorithm 1:** Pseudocode of the island model genetic algorithm approach.

---

```

Generate an initial population of  $p$  individuals;
Evaluate the fitness of all of the individuals;
Distribute the population across  $i$  islands;
repeatc times
  foreach island do
    repeatg times
      Select the best individuals for reproduction;
      Use the crossover operator;
      Use the mutation operator;
      Update the fitness of the modified individuals;
      Generate the population for the next generation;
    until;
    Migrate the  $m$  best individuals to a neighboring island;
  end
until;

```

---

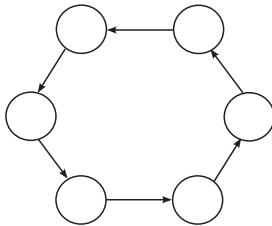


Fig. 6. Ring topology.

segment. This method ensures that no facilities are repeated in a facility layout design. The  $n$ -point crossover method (Starkweather et al., 1991a) is applied to the bay's division segment. Fig. 3 shows examples of both crossover operations.

### 3.4.3. Mutation operators

The purpose of mutation in GAs is to allow the algorithm to avoid local solutions and to force the population contain unprecedented examples (Albayrak & Allahverdi, 2011). The mutation operator is applied with a determinate probability to each chromosome segment. In the facility sequence segment, two facilities are randomly chosen and switched. In the bays division segment, a division index is randomly selected and the selected division's value is reversed.

### 3.4.4. Migration process

Our island model uses a ring topology. This approach is easy to implement and its use has often been extended (Alba & Troya, 1999). In this approach, individuals from each island migrate to only one other island, called a *neighboring island* (see Fig. 6).

Migration is the operator that guides the exchange of individuals among islands. As migration occurs, information about different regions of the search space is exchanged between islands, providing greater population diversity (Pereira & Sacco, 2008).

This operator is composed of two parameters (Tanese, 1989): (a) migration rate, which is related to the number of individuals ( $k$  in our algorithm) exchanged between islands; and (b) migration frequency, which indicates how often migration occurs (the number of generations between each exchange, namely,  $g$  in our proposal).

## 4. Experiments

We tested our proposal using a set of well-known problems that have been described in the literature. These are Ba12 and

Ba14 from Bazaraa (1975); Ba12TS and Ba14TS, which are modifications of the Ba12 and Ba14 problems presented by Tate and Smith (1995); MB12 from Bozer and Meller (1997); O7, O8 and O9, described by Meller, Narayanan, and Vance (1998); VC10 (which considers different measures of distance and different shape constraints) from van Camp, Carter, and Vannelli (1992); AB20 (which considers different departmental shape constraints) from Armour and Buffa (1963); and SC30 and SC35, two large problems presented by Liu and Meller (2007) that were later modified by Komarudin and Wong (2010), who added dummy departments.

The algorithm is coded in Python 2.7.3. All experiments were performed using an Intel Xeon E5420 (2.50 GHz), 8 GB RAM and a Linux operating system.

Table 1 presents the properties of the test problems used, such as the facility size, number of departments, shape constraint ( $\alpha$  is the maximum aspect ratio constraint, and  $l_{\min}$  is the minimum side length constraint), and the distance measure used in the objective function (Euclidean or rectilinear), and the references for the problem data sources.

### 4.1. IMGA configuration

The proposed IMGA algorithm contains parameters that have been tuned empirically. Using this method, tests to determine the parameters that fit best were performed while the algorithm was being developed.

The values considered for the IMGA parameters are presented in Table 2. From these values, a full-factorial design was developed (shown in Table 3). This design included a total of 144 different tuning configurations of the IMGA parameters. Each configuration was tested with a representative subset of well-known problems. Finally, we compared the results obtained by each option to select the best tuning.

In our tuning tests, the best IMGA configuration was option 51. The tuning of the IMGA parameters used for our numerical experiments is shown in Table 4.

## 5. Comparisons with other approaches

We compared the solutions found by our IMGA approach on the well-known problems described previously with the best solutions reported by other approaches based on the FBS representation that are found in the literature. Specifically, these other approaches were a mixed-integer programming formulation from Konak et al. (2006), an ant system approach presented by Wong and Komarudin (2010), a colony optimization algorithm proposed by Kulturel-Konak and Konak (2011b), and a probabilistic tabu search approach from Kulturel-Konak (2012). These works use two different types of termination conditions: (a) maximum CPU time and (b) maximum number of iterations. All the conditions of these previous works have been satisfied in our approach. The termination criterion used in our approach is 1050 iterations (70 cycles with 15 generations between each migration). Moreover, the CPU time required for each execution of our approach was far less than 24 h.

Table 5 shows a comparison Table 5 of the results from the different approaches. The best values achieved for each problem are highlighted in bold.

At first glance, by looking at Table 5, we can conclude that in most cases, the proposed algorithm found better solutions than did the methods from the literature. However, our algorithm is compared with the other algorithms in more detail.

As shown, compared with the algorithm proposed by Konak et al. (2006), our approach produced the best solution (or even better) for all of the problems except Ba12. In total, our algorithm improved 5 of the 8 problem sets compared with the algorithm developed by Konak et al. (2006).

**Table 1**  
Properties of the test problems.

Problem name	Number of departments	Facility size (W × H)	Common shape constraint	Distance measure	Reference
O7	7	8.54 × 13.00	$\alpha = 4$	Rectilinear	Meller et al. (1998)
O8	8	11.31 × 13.00	$\alpha = 4$	Rectilinear	Meller et al. (1998)
O9	9	12.00 × 13.00	$\alpha = 4$	Rectilinear	Meller et al. (1998)
VC10E-a	10	25.00 × 51.00	$\alpha = 5$	Euclidean	van Camp et al. (1992)
VC10E-s	10	25.00 × 51.00	$l_{min} = 5$	Euclidean	van Camp et al. (1992)
VC10R-a	10	25.00 × 51.00	$\alpha = 5$	Rectilinear	van Camp et al. (1992)
VC10R-s	10	25.00 × 51.00	$l_{min} = 5$	Rectilinear	van Camp et al. (1992)
Ba12	12	6.00 × 10.00	$l_{min} = 1$	Rectilinear	Bazaraa (1975)
Ba12TS	12	6.00 × 10.00	$l_{min} = 1$	Rectilinear	Tate and Smith (1995)
MB12	12	6.00 × 8.00	$\alpha = 4$	Rectilinear	Bozer and Meller (1997)
Ba14	14	7.00 × 9.00	$l_{min} = 1$	Rectilinear	Bazaraa (1975)
Ba14TS	14	7.00 × 9.00	$l_{min} = 1$	Rectilinear	Tate and Smith (1995)
AB20-ar170667	20	2.00 × 3.00	$\alpha = 1.70667$	Rectilinear	Armour and Buffa (1963)
AB20-ar175	20	2.00 × 3.00	$\alpha = 1.75$	Rectilinear	Armour and Buffa (1963)
AB20-ar2	20	2.00 × 3.00	$\alpha = 2$	Rectilinear	Armour and Buffa (1963)
AB20-ar3	20	2.00 × 3.00	$\alpha = 3$	Rectilinear	Armour and Buffa (1963)
AB20-ar4	20	2.00 × 3.00	$\alpha = 4$	Rectilinear	Armour and Buffa (1963)
AB20-ar5	20	2.00 × 3.00	$\alpha = 5$	Rectilinear	Armour and Buffa (1963)
AB20-ar7	20	2.00 × 3.00	$\alpha = 7$	Rectilinear	Armour and Buffa (1963)
AB20-ar10	20	2.00 × 3.00	$\alpha = 10$	Rectilinear	Armour and Buffa (1963)
AB20-ar15	20	2.00 × 3.00	$\alpha = 15$	Rectilinear	Armour and Buffa (1963)
AB20-ar25	20	2.00 × 3.00	$\alpha = 25$	Rectilinear	Armour and Buffa (1963)
AB20-ar50	20	2.00 × 3.00	$\alpha = 50$	Rectilinear	Armour and Buffa (1963)
AB20-ar1000	20	2.00 × 3.00	$\alpha = 1000$	Rectilinear	Armour and Buffa (1963)
SC30	30	12.00 × 15.00	$\alpha = 5$	Rectilinear	Komarudin and Wong (2010); Liu and Meller (2007)
SC35	35	15.00 × 16.00	$\alpha = 4$	Rectilinear	Komarudin and Wong (2010); Liu and Meller (2007)

**Table 2**  
Values considered for the parameters.

Parameter	Value
Total population size ( <i>p</i> )	500
Number of islands ( <i>i</i> )	10, 15
Cycle generations ( <i>c</i> )	50, 70
Number of generations between migration ( <i>g</i> )	15, 30
Number of individuals who migrate ( <i>m</i> )	5, 10
Crossover probability ( <i>p<sub>c</sub></i> )	0.6, 0.7, 0.8
Mutation probability ( <i>p<sub>m</sub></i> )	0.01, 0.03, 0.05

Our algorithm found the same or improved solutions for all the problems except one configuration of AB20 ( $\alpha = 1.75$ ), compared to the method developed by Wong and Komarudin (2010). In particular, our proposed algorithm improved the best solution for 5 of the 8 problems.

With respect to the colony optimization from Kulturel-Konak and Konak (2011b), the proposed algorithm obtained better solutions for 12 of the 20 problems tested and found the same solution for VC10R-s. This work is the only compared approach based on the FBS representation that experimented with the large problems (SC30 and SC35). Our proposal is able to reach similar results as those obtained by Kulturel-Konak and Konak (2011b) but in a remarkably shorter CPU time.

In contrast, our proposal found better solutions for 12 problems than did the probabilistic tabu search approach presented by Kulturel-Konak (2012). Furthermore, our algorithm equaled the best solution found by Kulturel-Konak (2012) for the problem VC10R-s.

In brief, the proposed algorithm was able to match or beat previous results presented in the literature by achieving the best solution for 2 problems and improving on the best solution in the literature for 14 problems (out of a total of 24 test problems).

Furthermore, we chose two problems (O7 and O8) that have not previously been tested by approaches using FBS representations in the literature to give future researchers the opportunity to provide appropriate comparisons.

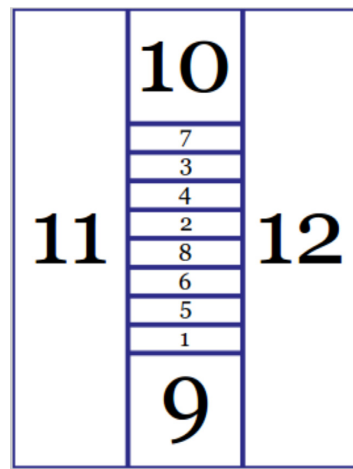


Fig. 7. Best facility layout found for MB12.

A summary comparison between the proposed algorithm and the best known solution found in the literature is presented in Table 6. In this table, we present the following: the best solution produced by IMGA, the percent difference of between the best solution found by our algorithm and the best known solution found in the literature, the CPU time (given in seconds), and the facilities design of the IMGA best solutions (read bay by bay, from left to right and from top to bottom). Moreover, several of the facility layouts generated by the proposed algorithm for problems MB12, VC10Es and VC10Ra, which are substantially better solutions than those found by previous approaches from the literature, are illustrated in Figs. 7, 8, and 9, respectively.

In conclusion, 26 problems that involve a number of departments between 7 to 35 were analyzed.



**Table 3**  
Full factorial design.

#	i	c	g	m	$p_c$	$p_m$	#	i	c	g	m	$p_c$	$p_m$	#	i	c	g	m	$p_c$	$p_m$
1	10	50	15	5	0.6	0.01	49	10	50	15	5	0.6	0.03	97	10	50	15	5	0.6	0.05
2	15	50	15	5	0.6	0.01	50	15	50	15	5	0.6	0.03	98	15	50	15	5	0.6	0.05
3	10	70	15	5	0.6	0.01	51	10	70	15	5	0.6	0.03	99	10	70	15	5	0.6	0.05
4	15	70	15	5	0.6	0.01	52	15	70	15	5	0.6	0.03	100	15	70	15	5	0.6	0.05
5	10	50	30	5	0.6	0.01	53	10	50	30	5	0.6	0.03	101	10	50	30	5	0.6	0.05
6	15	50	30	5	0.6	0.01	54	15	50	30	5	0.6	0.03	102	15	50	30	5	0.6	0.05
7	10	70	30	5	0.6	0.01	55	10	70	30	5	0.6	0.03	103	10	70	30	5	0.6	0.05
8	15	70	30	5	0.6	0.01	56	15	70	30	5	0.6	0.03	104	15	70	30	5	0.6	0.05
9	10	50	15	10	0.6	0.01	57	10	50	15	10	0.6	0.03	105	10	50	15	10	0.6	0.05
10	15	50	15	10	0.6	0.01	58	15	50	15	10	0.6	0.03	106	15	50	15	10	0.6	0.05
11	10	70	15	10	0.6	0.01	59	10	70	15	10	0.6	0.03	107	10	70	15	10	0.6	0.05
12	15	70	15	10	0.6	0.01	60	15	70	15	10	0.6	0.03	108	15	70	15	10	0.6	0.05
13	10	50	30	10	0.6	0.01	61	10	50	30	10	0.6	0.03	109	10	50	30	10	0.6	0.05
14	15	50	30	10	0.6	0.01	62	15	50	30	10	0.6	0.03	110	15	50	30	10	0.6	0.05
15	10	70	30	10	0.6	0.01	63	10	70	30	10	0.6	0.03	111	10	70	30	10	0.6	0.05
16	15	70	30	10	0.6	0.01	64	15	70	30	10	0.6	0.03	112	15	70	30	10	0.6	0.05
17	10	50	15	5	0.7	0.01	65	10	50	15	5	0.7	0.03	113	10	50	15	5	0.7	0.05
18	15	50	15	5	0.7	0.01	66	15	50	15	5	0.7	0.03	114	15	50	15	5	0.7	0.05
19	10	70	15	5	0.7	0.01	67	10	70	15	5	0.7	0.03	115	10	70	15	5	0.7	0.05
20	15	70	15	5	0.7	0.01	68	15	70	15	5	0.7	0.03	116	15	70	15	5	0.7	0.05
21	10	50	30	5	0.7	0.01	69	10	50	30	5	0.7	0.03	117	10	50	30	5	0.7	0.05
22	15	50	30	5	0.7	0.01	70	15	50	30	5	0.7	0.03	118	15	50	30	5	0.7	0.05
23	10	70	30	5	0.7	0.01	71	10	70	30	5	0.7	0.03	119	10	70	30	5	0.7	0.05
24	15	70	30	5	0.7	0.01	72	15	70	30	5	0.7	0.03	120	15	70	30	5	0.7	0.05
25	10	50	15	10	0.7	0.01	73	10	50	15	10	0.7	0.03	121	10	50	15	10	0.7	0.05
26	15	50	15	10	0.7	0.01	74	15	50	15	10	0.7	0.03	122	15	50	15	10	0.7	0.05
27	10	70	15	10	0.7	0.01	75	10	70	15	10	0.7	0.03	123	10	70	15	10	0.7	0.05
28	15	70	15	10	0.7	0.01	76	15	70	15	10	0.7	0.03	124	15	70	15	10	0.7	0.05
29	10	50	30	10	0.7	0.01	77	10	50	30	10	0.7	0.03	125	10	50	30	10	0.7	0.05
30	15	50	30	10	0.7	0.01	78	15	50	30	10	0.7	0.03	126	15	50	30	10	0.7	0.05
31	10	70	30	10	0.7	0.01	79	10	70	30	10	0.7	0.03	127	10	70	30	10	0.7	0.05
32	15	70	30	10	0.7	0.01	80	15	70	30	10	0.7	0.03	128	15	70	30	10	0.7	0.05
33	10	50	15	5	0.8	0.01	81	10	50	15	5	0.8	0.03	129	10	50	15	5	0.8	0.05
34	15	50	15	5	0.8	0.01	82	15	50	15	5	0.8	0.03	130	15	50	15	5	0.8	0.05
35	10	70	15	5	0.8	0.01	83	10	70	15	5	0.8	0.03	131	10	70	15	5	0.8	0.05
36	15	70	15	5	0.8	0.01	84	15	70	15	5	0.8	0.03	132	15	70	15	5	0.8	0.05
37	10	50	30	5	0.8	0.01	85	10	50	30	5	0.8	0.03	133	10	50	30	5	0.8	0.05
38	15	50	30	5	0.8	0.01	86	15	50	30	5	0.8	0.03	134	15	50	30	5	0.8	0.05
39	10	70	30	5	0.8	0.01	87	10	70	30	5	0.8	0.03	135	10	70	30	5	0.8	0.05
40	15	70	30	5	0.8	0.01	88	15	70	30	5	0.8	0.03	136	15	70	30	5	0.8	0.05
41	10	50	15	10	0.8	0.01	89	10	50	15	10	0.8	0.03	137	10	50	15	10	0.8	0.05
42	15	50	15	10	0.8	0.01	90	15	50	15	10	0.8	0.03	138	15	50	15	10	0.8	0.05
43	10	70	15	10	0.8	0.01	91	10	70	15	10	0.8	0.03	139	10	70	15	10	0.8	0.05
44	15	70	15	10	0.8	0.01	92	15	70	15	10	0.8	0.03	140	15	70	15	10	0.8	0.05
45	10	50	30	10	0.8	0.01	93	10	50	30	10	0.8	0.03	141	10	50	30	10	0.8	0.05
46	15	50	30	10	0.8	0.01	94	15	50	30	10	0.8	0.03	142	15	50	30	10	0.8	0.05
47	10	70	30	10	0.8	0.01	95	10	70	30	10	0.8	0.03	143	10	70	30	10	0.8	0.05
48	15	70	30	10	0.8	0.01	96	15	70	30	10	0.8	0.03	144	15	70	30	10	0.8	0.05

**Table 4**  
IMGA parameter values.

Parameter	Value
Total population size ( $p$ )	500
Number of islands ( $i$ )	10
Subpopulation size per island ( $p/i$ )	50
Cycle generations ( $c$ )	70
Number of generations between each migration process ( $g$ )	15
Number of individuals who migrate ( $m$ )	5
Crossover probability ( $p_c$ )	0.6
Mutation probability ( $p_m$ )	0.03

The solutions were compared with those obtained by the authors of previous studies for the same well-known problems. Our novel proposal obtained better solutions on 14 problems, matched the best previous solution on 2 problems, and found satisfactory solutions for two new problems that have not previously been reported in the literature. These best solutions belong to a wide range of problem sizes, from small to large. Our approach found

better solutions in all these ranges than the solutions of previous methods from the literature.

Compared with the previous solutions, the novel algorithm is considerably more versatile. The previous solutions tend to find better solutions for only a certain type of problem; they do not behave equally well for all problem sizes.

Finally, the execution times required by our approach range between 67 and 4015 s. These times achieved by our proposal are comparable to, or better than, the times of previous approaches in all cases. This difference is particularly clear on largest problems: our method reduced the execution time nearly fourfold compared to the previous proposals.

### 6. Conclusions

This paper presented a novel island model genetic algorithm with a flexible bay representation for the unequal area facility layout problem. To the best of our knowledge, this approach is the first study to investigate the use of parallel genetic algorithms to solve UA-FLPs. The parallel approach helps to avoid premature convergence and excessive execution time. In addition, this method

**Table 5**  
Test results comparisons between the IMGA algorithm and other approaches.

Problem name	The novel IMGA proposed	Best known previous FBS solutions			
		Kulturel-Konak (2012)	Kulturel-Konak and Konak (2011b)	Wong and Komarudin (2010)	Konak et al. (2006)
O7	<b>134.19</b>	–	–	–	–
O8	<b>245.51</b>	–	–	–	–
O9	<b>241.06</b>	–	–	<b>241.06</b>	<b>241.06</b>
vC10E-a	<b>18554.59</b>	18798.7	–	–	–
vC10E-s	<b>18823.74</b>	20320.52	–	–	–
vC10R-a	<b>20142.13</b>	21463.07	21463.07	21463.1	21463.07
vC10R-s	<b>22899.65</b>	<b>22899.65</b>	<b>22899.65</b>	<b>22899.65</b>	<b>22899.65</b>
Ba12	8435.83	<b>8021</b>	8083	8786	8801.33
Ba12TS	8627.05	<b>8587.05</b>	8600.38	<b>8587.05</b>	8600.38
MB12	<b>125</b>	–	–	–	145.28
Ba14	<b>4665.93</b>	4739.74	4913.22	5004.55	–
Ba14TS	<b>4834.65</b>	4927.69	4927.69	4927.69	4927.69
AB20-ar170667	5885.68	5845.33	<b>5845.3</b>	–	–
AB20-ar175	6291.87	5845.33	5845.3	<b>5677.83</b>	6890.82
AB20-ar2	5858.41	5845.33	<b>5845.3</b>	–	–
AB20-ar3	<b>5419.49</b>	5594.29	5594.3	–	–
AB20-ar4	<b>5286.65</b>	5360.8	5360.8	–	–
AB20-ar5	<b>5256.1</b>	5297.6	5297.6	–	–
AB20-ar7	4844.49	<b>4793.47</b>	4793.5	–	–
AB20-ar10	<b>4367.57</b>	4440.68	4440.7	–	–
AB20-ar15	<b>4100.17</b>	4119.8	4116.8	–	–
AB20-ar25	<b>3324.49</b>	3399.16	3526.5	–	–
AB20-ar50	<b>2382.74</b>	2457.79	2706.5	–	–
AB20-ar1000	<b>1588.49</b>	1638.53	1638.5	–	–
SC30	3613.11	–	<b>3563.95</b>	–	–
SC35	3885.29	–	<b>3814.98</b>	–	–

**Table 6**  
Summary of results.

Problem name	Best solution found by IMGA	Percent difference (%) <sup>a</sup>	CPU time	Facility layout by IMGA
O7	<b>134.19</b>	–	67	7-5-3   2-6-4-1
O8	<b>245.51</b>	–	75	7-4-1-2   3-6-8-5
O9	<b>241.06</b>	0	84	7-8   4-1-2   3-6-9-5
vC10E-a	<b>18554.59</b>	1.32	94	3-4-7   1-6-2-9-5-8-10
vC10E-s	<b>18823.74</b>	7.95	96	9-10   2-4-8   6-7-5   1-3
vC10R-a	<b>20142.13</b>	6.56	91	5-8-10-9-2-6-1   4-7-3
vC10R-s	<b>22899.65</b>	0	91	1   7-6   4-2   8-10-9   5-3
Ba12	8435.83	–4.92	190	10   4   5   11-8-12   3   2   1   9-6-7
Ba12TS	8627.05	–0.46	154	4   5   9-8-10   3   2   1   11-6-12   13-7
MB12	<b>125</b>	16.22	109	11   10-7-3-4-2-8-6-5-1-9   12
Ba14	<b>4665.93</b>	1.58	181	11-5-10   1   3   13   4   2-12   14-6-8-9-7
Ba14TS	<b>4834.65</b>	1.92	131	10-5-11-13   1   3   4   14-2   7-9-8-6-12
AB20-ar170667	5885.68	–0.69	208	16-17   11-15-12   13-14-9-10   1-3-19   5-6-8-7-4   18-20-2
AB20-ar175	6291.87	–9.76	206	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	207	20-6-18   5-8-7-2-4   13-9-3-19   15-14-10   17-1-12   16-11
AB20-ar3	<b>5419.49</b>	3.23	207	16-11   17-13-15   12-9-10-14   1-19-3   6-4-2-7-8-5   18-20
AB20-ar4	<b>5286.65</b>	1.40	205	20-18   5-8-7-2-4-6   3-19   15-14-10-12   1-13-9-17   11-16
AB20-ar5	<b>5256.1</b>	0.79	209	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	207	11   16   17-15   12-13   1-10-9-14   5-19-3   20-8-7-2-4-6-18
AB20-ar10	<b>4367.57</b>	1.67	204	11   16   17   12   15-13-14-10-9-19-3-1   5-20-8-7-2-4-6-18
AB20-ar15	<b>4100.17</b>	0.41	210	1   18   20-8-7-2-4-19-10-9-14-3-6   12   15   17   5-13   16   11
AB20-ar25	<b>3324.49</b>	2.25	210	11   16   13   17   12   15   10   14   3   19   4   2   20-8-7-9-5-1   6   18
AB20-ar50	<b>2382.74</b>	3.15	212	1   18   5   20   8   7   6   2   4   19   3   10   14   9   15   12   17   13   16   11
AB20-ar1000	<b>1588.49</b>	3.42	210	1   18   5   20   8   7   6   2   4   19   3   10   14   9   15   12   17   13   16   11
SC30	3613.11	–1.36	419	7-8-9-10-14-16   12   6-11-5-15-13   2-23-18   21-4   19-28-3-17   22-20-24-29   27-1-25-30-26
SC35	3885.29	–1.80	804	27-21-7-8-9-11-31-16   24-2-6-5-12-32-14   23-1-25-30-10-13   19-29-17-15   28-33-18   35-20-34-22-4-26   3

<sup>a</sup> 100 \* (the best known solution found in the literature - the novel IMGA best solution) / (the novel IMGA best solution)

improves search diversity, allowing it to explore a larger search space and obtain better solutions.

Our novel approach was tested on a total of 26 well-known UAF-FLPs, and the results were compared with the best results found in the literature. Our algorithm was able to improve the best solution for 14 problems, match the best solution for two problems and present two new solutions for problems that were previously unreported in the literature. Moreover, the execution times of the

novel proposal were similar to or lower than the execution times of the previous approaches in all cases.

The success of this novel approach is due to the degree of independence imposed by the islands. The islands make possible to explore different regions of the search space, helping to (a) improve search quality, (b) obtain better quality solutions, (c) provide greater population diversity, and (d) reduce the number of evaluations required to find good solutions.

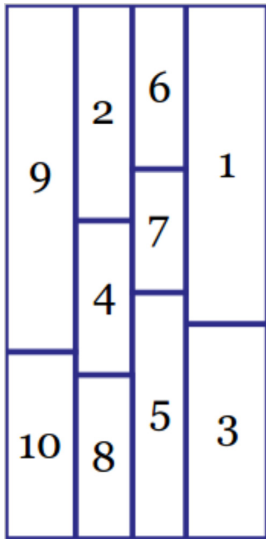


Fig. 8. Best facility layout found for VC10Es.

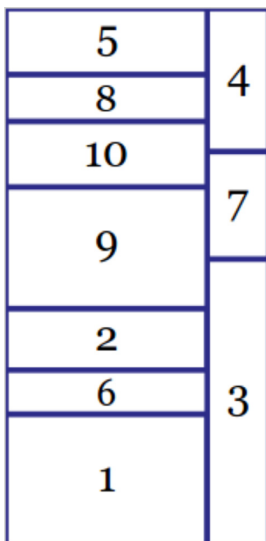


Fig. 9. Best facility layout found for VC10Ra.

On one hand, it could be interesting to continue working in the future along the promising line of using PGAs to reduce running times by parallelizing the operations involved. For example, optimizing the number of islands (subpopulations) might be an issue to be considered for a future study. On the other hand, it might be even more promising to develop a novel strategy by redesigning the migration method so that the best subpopulations migrate to help improve the worst ones. Finally, it could also be interesting to apply the new approach to the interactive genetic algorithms, to try to reduce the fatigue of the decision maker by having them evaluate a more diverse population, which could help find a desirable facility layout in a reduced number of generations.

## References

- Abotaleb, I., Nassar, K., & Hosny, O. (2016). Layout optimization of construction site facilities with dynamic freeform geometric representations. *Automation in Construction*, 66, 15–28. <http://dx.doi.org/10.1016/j.autcon.2016.02.007>.
- Achterberg, T. (2007). Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1), 4–20. <http://dx.doi.org/10.1016/j.disopt.2006.10.006>.
- Ahmadi, A., & Jokar, M. R. A. (2016). An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems. *Applied Mathematical Modelling*, 40(9–10), 5605–5620. <http://dx.doi.org/10.1016/j.apm.2016.01.014>.
- Aiello, G., & Enea, M. (2001). Fuzzy approach to the robust facility layout in uncertain production environments. *International Journal of Production Research*, 39(18), 4089–4101.
- 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 electromechanical method for unequal area facility layout problems. *Expert Systems with Applications*, 40(12), 4812–4819. <http://dx.doi.org/10.1016/j.eswa.2013.02.026>.
- Al-Hakim, L. (2000). On solving facility layout problems using genetic algorithms. *International Journal of Production Research*, 38(11), 2573–2582. doi:10.1080/00207540050031922.
- Alagoz, O., Norman, B. A., & Smith, A. E. (2008). Determining aisle structures for facility designs using a hierarchy of algorithms. *IIE Transactions*, 40(11), 1019–1031. doi:10.1080/07408170802167621.
- Alba, E., & Troya, J. M. (1999). A survey of parallel distributed genetic algorithms. *Complex*, 4(4), 31–52. doi:10.1002/(SICI)1099-0526(199903/04)4:4<31::AID-CPLX5>3.3.CO;2-W.
- Albayrak, M., & Allahverdi, N. (2011). Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. *Expert Systems with Applications*, 38(3), 1313–1320. <http://dx.doi.org/10.1016/j.eswa.2010.07.006>.
- Alves, D. T. S., de Medeiros, J. L., & de Queiroz F. Arajo, O. (2016). Optimal determination of chemical plant layout via minimization of risk to general public using monte carlo and simulated annealing techniques. *Journal of Loss Prevention in the Process Industries*, 41, 202–214. <http://dx.doi.org/10.1016/j.jlp.2016.03.026>.
- Armour, G. C., & Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9, 294–309.
- Bazaraa, M. S. (1975). Computerized layout design: A branch and bound approach. *AIIE Transactions*, 7(4), 432–438. doi:10.1080/0569557508975028.
- Bozer, Y. A., & Meller, R. D. (1997). A reexamination of the distance-based facility layout problem. *IIE Transactions*, 29(7), 549–560. doi:10.1080/07408179708966365.
- 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. <http://dx.doi.org/10.1016/j.ejor.2011.10.052>.
- van Camp, D. J., Carter, M. W., & Vannelli, A. (1992). A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research*, 57(2), 174–189. [http://dx.doi.org/10.1016/0377-2217\(92\)90041-7](http://dx.doi.org/10.1016/0377-2217(92)90041-7).
- Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateur Paralleles*, 10.
- Chen, Y., Jiang, Y., Wahab, M., & Long, X. (2015). The facility layout problem in non-rectangular logistics parks with split lines. *Expert Systems with Applications*, 42(21), 7768–7780. <http://dx.doi.org/10.1016/j.eswa.2015.06.009>.
- 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.
- Datta, D., Amaral, A. R., & Figueira, J. R. (2011). Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2), 388–394. <http://dx.doi.org/10.1016/j.ejor.2011.03.034>.
- Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Scituate, MA, USA: Bradford Company.
- Drira, A., Pierrel, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2), 255–267.
- Dunker, T., Radons, G., & Westkämper, E. (2005). Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research*, 165(1), 55–69. <http://dx.doi.org/10.1016/j.ejor.2003.01.002>.
- Ertay, T., Ruan, D., & Tuzkaya, U. R. (2006). Integrating data envelopment analysis and analytic hierarchy for the facility layout design in manufacturing systems. *Information Sciences*, 176(3), 237–262. <http://dx.doi.org/10.1016/j.ins.2004.12.001>.
- García-Hernández, L., Arauzo-Azofra, A., Salas-Morera, L., Pierrel, 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., Salas-Morera, L., Pierrel, H., & Corchado, E. (2015a). Facility layout design using a multi-objective interactive genetic algorithm to support the dm. *Expert Systems*, 32(1), 94–107. doi:10.1111/esity.12064.
- García-Hernández, L., Palomo-Romero, J. M., Salas-Morera, L., Arauzo-Azofra, A., & Pierrel, H. (2015b). A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem. *Expert Systems with Applications*, 42(10), 4697–4708. <http://dx.doi.org/10.1016/j.eswa.2015.01.037>.

- García-Hernández, L., Pérez-Ortiz, M., Araúz-Azofra, A., Salas-Morera, L., & Hervás-Martínez, C. (2014). An evolutionary neural system for incorporating expert knowledge into the ua-flp. *Neurocomputing*, 135, 69–78. <http://dx.doi.org/10.1016/j.neucom.2013.01.068>.
- García-Hernández, L., Pierrelval, H., Salas-Morera, L., & Araúz-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.
- Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms* (pp. 69–93). Morgan Kaufmann.
- Gonçalves, J. F., & Resende, M. G. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*, 246(1), 86–107. <http://dx.doi.org/10.1016/j.ejor.2015.04.029>.
- Guan, J., & Lin, G. (2016). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. *European Journal of Operational Research*, 248(3), 899–909. <http://dx.doi.org/10.1016/j.ejor.2015.08.014>.
- Heragu, S. (2008). *Facilities design, third edition. 500 Tips*. Taylor & Francis.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press.
- Ijjina, E. P., & Chalavadi, K. M. (2016). Human action recognition using genetic algorithms and convolutional neural networks. *Pattern Recognition*, 59, 199–212. <http://dx.doi.org/10.1016/j.patcog.2016.01.012>.
- Ingeber, L. (1993). Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11), 29–57. [http://dx.doi.org/10.1016/0895-7177\(93\)90204-C](http://dx.doi.org/10.1016/0895-7177(93)90204-C).
- Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research*, 36(6), 1549–1569. doi:10.1080/002075498193165.
- 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(7–8), 583–594. [http://dx.doi.org/10.1016/S0305-0548\(97\)00100-7](http://dx.doi.org/10.1016/S0305-0548(97)00100-7).
- Komarudin, & Wong, K. 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.
- Koopmans, T., & Beckmann, M. J. (1955). *Assignment Problems and the Location of Economic Activities*. Cowles Foundation Discussion Papers 4. Cowles Foundation for Research in Economics, Yale University.
- 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. <http://dx.doi.org/10.1016/j.ejor.2012.07.019>.
- Kulturel-Konak, S., & Konak, A. (2011a). A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Engineering Optimization*, 43(12), 1263–1287. doi:10.1080/0305215X.2010.548864.
- Kulturel-Konak, S., & Konak, A. (2011b). Unequal area flexible bay facility layout using ant colony optimisation. *International Journal of Production Research*, 49(7), 1877–1902. doi:10.1080/00207541003614371.
- Kundakc, N., & Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *Computers & Industrial Engineering*, 96, 31–51.
- Kurdi, M. (2016). An effective new island model genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, 67, 132–142. <http://dx.doi.org/10.1016/j.cor.2015.10.005>.
- Kwong, S., Ng, A., & Man, K. (1995). Improving local search in genetic algorithms for numerical global optimization using modified grid-point search technique. In *Genetic algorithms in engineering systems: Innovations and applications, 1995. galesia, first international conference on (conf. publ. no. 414)* (pp. 419–423). doi:10.1049/cp:19951085.
- Kker, R. (2013). A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. *Information Sciences*, 222(0), 528–543. <http://dx.doi.org/10.1016/j.ins.2012.07.051>.
- Lee, Y. H., & Lee, M. H. (2002). A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. *Computers & Industrial Engineering*, 42(2–4), 237–248. [http://dx.doi.org/10.1016/S0360-8352\(02\)00018-9](http://dx.doi.org/10.1016/S0360-8352(02)00018-9).
- Li, X., & Parrott, L. (2016). An improved genetic algorithm for spatial optimization of multi-objective and multi-site land use allocation. *Computers, Environment and Urban Systems*, 59, 184–194. <http://dx.doi.org/10.1016/j.compenvurbysys.2016.07.002>.
- Liu, Q., & Meller, R. D. (2007). A sequence-pair representation and mip-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions*, 39(4), 377–394.
- Mak, K., Wong, Y., & Chan, F. (1998). A genetic algorithm for facility layout problems. *Computer Integrated Manufacturing Systems*, 11(1–2), 113–127. [http://dx.doi.org/10.1016/S0951-5240\(98\)00018-4](http://dx.doi.org/10.1016/S0951-5240(98)00018-4).
- Matali, R. (2015). Solving multi-objective facility layout problem by modified simulated annealing. *Applied Mathematics and Computation*, 261, 302–311. <http://dx.doi.org/10.1016/j.amc.2015.03.107>.
- Meller, R., & Gau, K. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15, 351–366.
- Meller, R. D., Narayanan, V., & Vance, P. H. (1998). Optimal facility layout design. *Operations Research Letters*, 23(3–5), 117–127. [http://dx.doi.org/10.1016/S0167-6377\(98\)00024-8](http://dx.doi.org/10.1016/S0167-6377(98)00024-8).
- Miller, B. L., & Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2), 113–131. doi:10.1162/evco.1996.4.2.113.
- Mhlenbein, H. (1991). Evolution in time and space – the parallel genetic algorithm. In *Foundations of genetic algorithms* (pp. 316–337). Morgan Kaufmann.
- Paes, F. G., Pessoa, A. A., & Vidal, T. (2016). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research*. <http://dx.doi.org/10.1016/j.ejor.2016.07.022>.
- Palubeckis, G. (2015a). Fast local search for single row facility layout. *European Journal of Operational Research*, 246(3), 800–814. <http://dx.doi.org/10.1016/j.ejor.2015.05.055>.
- Palubeckis, G. (2015b). Fast simulated annealing for single-row equidistant facility layout. *Applied Mathematics and Computation*, 263, 287–301. <http://dx.doi.org/10.1016/j.amc.2015.04.073>.
- 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. International Nuclear Atlantic Conference 2005. <http://dx.doi.org/10.1016/j.pnucene.2007.12.007>.
- Pierrelval, H., Caux, C., Paris, J., & Viguier, F. (2003). Evolutionary approaches to the design and organization of manufacturing systems. *Computers & Industrial Engineering*, 44(3), 339–364. [http://dx.doi.org/10.1016/S0360-8352\(02\)00195-X](http://dx.doi.org/10.1016/S0360-8352(02)00195-X).
- Pourvaziri, H., & Naderi, B. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing*, 24(0), 457–469. <http://dx.doi.org/10.1016/j.asoc.2014.06.051>.
- Ripon, K. S. N., Glette, K., Khan, K. N., Hovim, M., & Torresen, J. (2013). Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities. *Swarm and Evolutionary Computation*, 8(0), 1–12. <http://dx.doi.org/10.1016/j.swevo.2012.07.003>.
- 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. doi:10.1016/j.cie.2011.12.033.
- Sahin, R. (2011). A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Systems with Applications*, 38(4), 4460–4465. <http://dx.doi.org/10.1016/j.eswa.2010.09.117>.
- 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.
- Seppanen, J., & Moore, J. M. (1970). Facilities planning with graph theory. *Management Science*, 17(4), B242–B253.
- Shin, H., Joo, C., & Koo, J. (2016). Optimal rehabilitation model for water pipeline systems with genetic algorithm. *Procedia Engineering*, 154, 384–390. 12th International Conference on Hydroinformatics (HIC 2016) – Smart Water for the Future. <http://dx.doi.org/10.1016/j.proeng.2016.07.497>.
- Sikaroudi, A. M. E., & Shahanaghi, K. (2016). Facility layout by collision detection and force exertion heuristics. *Journal of Manufacturing Systems*, 41, 21–30. <http://dx.doi.org/10.1016/j.jmsy.2016.07.001>.
- 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.
- Sitkoff, N. (1995). Implementing a genetic algorithm on a parallel custom computing machine. In *Proceedings of the IEEE symposium on FPGA's for custom computing machines*. In FCCM '95. Washington, DC, USA: IEEE Computer Society. 180–.
- Solar, M., Parada, V., & Urrutia, R. (2002). A parallel genetic algorithm to solve the set-covering problem. *Computers & Operations Research*, 29(9), 1221–1235. [http://dx.doi.org/10.1016/S0305-0548\(01\)00260-0](http://dx.doi.org/10.1016/S0305-0548(01)00260-0).
- 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. doi:10.1016/j.cie.2008.01.018.
- Starkweather, T., McDaniel, S., Whitley, D., Mathias, K., Whitley, D., & Dept, M. E. (1991a). A comparison of genetic sequencing operators. In *Proceedings of the fourth international conference on genetic algorithms* (pp. 69–76). Morgan Kaufmann.
- Starkweather, T., Whitley, D., & Mathias, K. (1991b). Optimization using distributed genetic algorithms. In H.-P. Schwefel, & R. Männer (Eds.), *Parallel problem solving from nature*. In *Lecture Notes in Computer Science*: 496 (pp. 176–185). Springer Berlin Heidelberg. doi:10.1007/BFb0029750.
- Tam, K. Y. (1992). Genetic algorithms, function optimization, and facility layout design. *European Journal of Operational Research*, 63(2), 322–346. Strategic Planning of Facilities. [http://dx.doi.org/10.1016/0377-2217\(92\)90034-7](http://dx.doi.org/10.1016/0377-2217(92)90034-7).
- Tam, K. Y. (1998). Solving facility layout problems with geometric constraints using parallel genetic algorithms: Experimentation and findings. *International Journal of Production Research*, 36(12), 3253–3272. doi:10.1080/002075498192058.
- Tanese, R. (1989). Distributed genetic algorithms. In *Proceedings of the 3rd international conference on genetic algorithms* (pp. 434–439). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- 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* (4th). New York: Wiley.
- Tong, X. (1991). *SECOT: A sequential construction technique for facility design*. University of Pittsburgh: Doctoral Dissertation.

- Ulutas, B., & Islier, A. A. (2015). Dynamic facility layout problem in footwear industry. *Journal of Manufacturing Systems*, 36, 55–61. <http://dx.doi.org/10.1016/j.jmsy.2015.03.004>.
- Ulutas, B. H., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), 5384–5395. <http://dx.doi.org/10.1016/j.eswa.2011.11.046>.
- Umarani, R., & Selvi, V. (2010). Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications*, 5(4), 1–6. Published By Foundation of Computer Science
- Vitayasak, S., Pongcharoen, P., & Hicks, C. (2016). A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a genetic algorithm or modified backtracking search algorithm. *International Journal of Production Economics*. <http://dx.doi.org/10.1016/j.ijpe.2016.03.019>.
- Wang, M.-J., Hu, M. H., & Ku, M.-Y. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *Computers and Industrial*, 56(2), 207–220. doi:10.1016/j.compind.2004.06.003.
- Wang, P., & Kwok, D. (1994). Optimal design of {PID} process controllers based on genetic algorithms. *Control Engineering Practice*, 2(4), 641–648. [http://dx.doi.org/10.1016/0967-0661\(94\)90008-6](http://dx.doi.org/10.1016/0967-0661(94)90008-6).
- 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, K.-Y., & Komarudin (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Expert Systems with Applications*, 37(7), 5523–5527. <http://dx.doi.org/10.1016/j.eswa.2009.12.080>.